# TASK-CUSTOMIZED MASKED AUTOENCODER VIA MIXTURE OF CLUSTER-CONDITIONAL EXPERTS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Masked Autoencoder (MAE) is a prevailing self-supervised learning method that achieves promising results in model pre-training. However, when the various downstream tasks have data distributions different from the pre-training data, the semantically irrelevant pre-training information might result in negative transfer, impeding MAE's scalability. To address this issue, we propose a novel MAE-based pre-training paradigm, named Mixture of Cluster-conditional Experts (MoCE), which can be trained once but provide customized pre-training models for diverse downstream tasks. Different from the mixture of experts (MoE), our MoCE trains each expert only with semantically relevant images by using cluster-conditional gates, which can automatically format the hierarchical cluster architecture of the pre-training data. Thus, each downstream task can be allocated to its customized model pre-trained with data most similar with the downstream data. Experimental results show that our MoCE achieves 2.45% average performance improvement compared to the vanilla MAE on a collection of 11 different downstream tasks.

## 1 INTRODUCTION

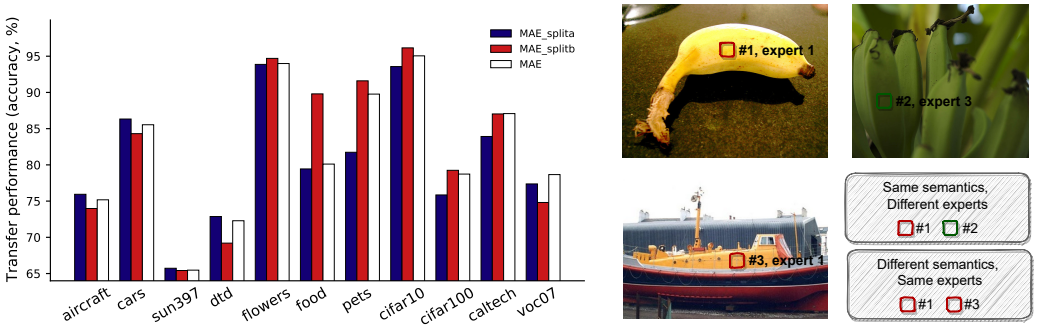
Self-supervised learning (SSL) has become a prevailing model pre-training paradigm, which learns effective transferable representations without human annotations (He et al., 2020; Chen et al., 2021a; Bao et al., 2021a). Among different SSL methods, Masked Autoencoder (MAE) (He et al., 2022) might be the most prevalent one currently, which proposes to construct supervision signals with raw image data by masking random patches of the input image and reconstructing missing pixels. This simple but effective self-supervisory strategy has been proved efficient to train large-scale models, such as ViT (Dosovitskiy et al., 2020) with impressive performance on widely adopted benchmarks, such as ImageNet-1K [1] (Deng et al., 2009). However, can MAE, the so-called "scalable vision learner", really be efficient for various downstream tasks?

Unfortunately, we observe that MAE still suffers from the *negative transfer* phenomenon (Liu et al., 2022) when transferring towards downstream tasks with distinct semantics. Specifically, for a given downstream task, involving more data in pre-training might instead degrade transfer performance, due to the negative effects of semantic irrelevant pre-training data. A preliminary is conducted in Sec. 3.1, where MAE pre-trained on the full ImageNet is outperformed by MAE pre-trained on the semantically relevant subsets on 9 of 11 downstream tasks, as shown in Fig. 1(a).

To achieve task-customized pre-training, Mixture of Expert (MoE) (Riquelme et al., 2021) seems promising, which proposes a multi-expert architecture to provide customized models for different input tokens. In the scenarios of self-supervised learning, however, it is difficult to train experts with diversity due to the missing semantic labels used in supervised pre-training, as demonstrated in Fig. 1(b). Our experiments in Sec. 4.2 show that naively adopting MoE to MAE, referring as TokenMoE, is not feasible as the experts tend to degenerate to one or a few experts without the data labels during pre-training.

In this paper, we propose the Mixture of Cluster-conditional Expert (MoCE), a novel paradigm to achieve task-customized self-supervised pre-training with data clustering and explicitly training each expert with images of similar semantics. Specifically, our MoCE consists of three stages. We first

[1] We refer to ImageNet-1K as ImageNet if not specified in this paper.




(a) Negative transfer phenomenon on MAE.

(b) Disadvantages of TokenMoE.

Figure 1: (a) Transfer performance of MAE separately pre-trained on the Split-A (blue), Split-B (red) and the full ImageNet (white), as discussed in Sec. 3.1. Only two of the eleven downstream tasks benefit from the usage of full ImageNet for pre-training. (b) Two kind of mistakes made by TokenMoE (#1 & #2) and (#1 & #3). Due to the usage of pixel RGB values as the reconstruction targets, tokens with similar low-level information tend to be routed to the same expert, while our MoCE can make routing decisions more correlated with the semantics, with the introduction of the cluster-conditional gates.

cluster the pre-trained dataset based on a pre-trained dense MAE model. Our MoCE is then constructed under the multi-expert structure, and we train each expert with different clusters through routing tokens in the image based on the *cluster embedding* instead of *token embedding*. Additional regularization loss is proposed to enhance the confidence of gate results, stabilizing the training procedure. Finally, when the downstream tasks come, we propose a searching method reusing the cluster module by feeding downstream images to it and the closet cluster is selected by the majority vote. We show that our MoCE achieves remarkable strong performance on a collection of 11 downstream tasks over MAE and achieves state-of-the-art averaged accuracy. Besides, the MoCE also enjoys the benefit of sub-model deployment that saves inference time and model capacity.

To summarize, our main contributions are:

1. We systematically analyze the negative transfer phenomenon with MAE, and show that naively adopting MoE to MAE cannot improve the transfer performance in downstream tasks.
2. We propose MoCE that train each expert explicitly with semantic-aware clusters by automatically routing similar clusters with the same expert.
3. We demonstrate the effectiveness of our model on a collection of 11 downstream tasks and achieve up to 2.45% performance improvement in Top-1 accuracy. To the best of our knowledge, this is the first work that successfully trains Vision MoE models under SSL settings with ImageNet only and achieves state-of-the-art transfer performances.

## 2 Related Work

**Self-supervised Learning** Previous works mainly focus on the design of pretext tasks with image transformations (Doersch et al., 2015; Gidaris et al., 2018), inpainting (Pathak et al., 2016), colorization (Zhang et al., 2016), contrastive learning (Chen et al., 2020; He et al., 2020; Grill et al., 2020b; Caron et al., 2020), and for specific downstream tasks (Wang et al., 2020; Xie et al., 2020; 2021a; Chen et al., 2021a). Motivated by BERT Devlin et al. (2018), masked image modeling (MIM) is recently proposed to learn by reconstructing masked images. BEiT (Bao et al., 2021a) is the pioneering work to predict visual tokens generated by a pre-trained tokenizor (Radford et al., 2021). SimMIM (Xie et al., 2021b) further simplifies the framework by directly utilizing pixel RGB values as the reconstruction target, and MAE (He et al., 2022) proposes an asymmetric encoder-decoder architecture for better training efficiency. Due to its effectiveness and efficiency, MAE is selected as the primary baseline.

Although achieving significant transfer performance, most previous methods can only provide a unified representation for various downstream tasks, suffering from the risk of negative transfer, as demonstrated in Sec. 3.1. The most relevant work to ours is SDR (Liu et al., 2022), which can train 256 subnets with 256 disjoint ImageNet subsets simultaneously. Instead, our work differs from three perspectives, 1) the mapping between subnets and subsets in SDR is randomly selected and fixed during pre-training, while MoCE achieves self-adaptive mapping with the cluster-conditional gates; 2) progressive training is required in SDR, while MoCE shares the same dataloader with MAE; 3) SDR is built upon SimSiam (Chen & He, 2020), a representative contrastive method, while MoCE focuses on the recently proposed MAE.

**Mixture of Experts** Mixture of experts(MoE) has a long history through decades (Jacobs et al., 1991; Jordan & Jacobs, 1994; Shazeer et al., 2017). Recently, MoE is regarded as an effective way to scale up the model in natural language processing (Lepikhin et al., 2020; Fedus et al., 2021; Yang et al., 2021; Lewis et al., 2021). With the growth of Vision Transformer, MoE for vision (Riquelme et al., 2021; Wu et al., 2022) is also explored recently. However, we still lack a self-supervised learning MoE model that can be trained on the moderate size of the dataset, like ImageNet-1k.

We notice (Kudugunta et al., 2021; Ma et al., 2018) most resemble our work. They regard MoE as a multi-task learning method and focus on the multi-language translation and recommendation systems respectively. We show that under the self-supervised learning setting for image modality, an additional clustering component is crucial for learning a meaningful MoE model using ImageNet-1K only. Moreover, benefiting from the unsupervised pre-training, MoCE is able to be deployed to any image downstream task, which differs from these two works in that downstream tasks should follow the pre-training task. (Puigcerver et al., 2020) shares a similar setting with us except their model is pre-trained supervisedly. However, their mapping between the expert and the data is pre-defined and fixed during training, while MoCE is learned dynamically to achieve a better result, referring to Table 2. Their idea is implemented as MoCE(fix) in the table.

**Multi-Task Learning** Multi-task Learning targets to learn a model that is appropriate for several tasks. Some works propose to adopt a shared backbone together with multi-heads for different specific tasks, namely hard-parameter sharing, showing the effectiveness on time-series data, language and graph data separately (Liu et al., 2019; Hu et al., 2019; McDermott et al., 2021). Some works (Gao et al., 2021) claim that the network design may benefit from the relationship of tasks, and trains masks for different tasks. These works assume that the information of tasks are available during model training. In SSL pre-training, on the other hand, information of the downstream tasks are unknown.

# 3 METHOD

In this section, we first demonstrate the negative transfer phenomenon in MAE with a preliminary experiment in Sec. 3.1. After that, we discuss the limitations of adopting TokenMoE (Riquelme et al., 2021) with MAE in Sec. 3.2, and propose our Mixture of Cluster-conditional Experts (MoCE), a novel paradigm for customized pre-training for various downstream tasks in Sec. 3.3.

## 3.1 NEGATIVE TRANSFER IN MASKED AUTOENCODER

To demonstrate the negative transfer phenomenon on MAE, in this section, we evaluate the transfer performance of MAE models pre-trained with data of distinct semantics on various downstream tasks. First, we separate ImageNet into two disjoint splits, called Split-A and Split-B, based on their label semantic dissimilarity in WordNet tree (Miller, 1998) following (Huh et al., 2016; Liu et al., 2022). Split-A mainly contains inanimate objects, such as cars and airplanes, while Split-B primarily involves organisms, such as plants and animals. Then, we pre-train MAE on Split-A, Split-B and the full ImageNet respectively without data annotation, and evaluate the resulting three models on 11 downstream tasks. Check Sec. 4.1 for more implementation details.

As demonstrated in Fig. 1, MAE pre-trained with Split-A performs the best on Aircraft, Cars, SUN397 and DTD, while achieving the highest accuracy on Flowers, Food, Pets, Cifar-10 and Cifar-100 when pre-trained with Split-B. Only two of the eleven tasks benefit from the usage of the full ImageNet, suggesting that for datasets whose semantics are close to inanimate objects, like Aircraft,

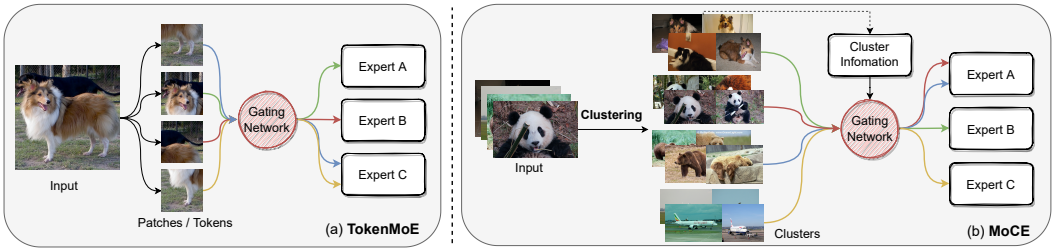



Figure 2: Model design comparison between (a) TokenMoE (Riquelme et al., 2021) and (b) MoCE. Both methods utilize the multi-expert architecture with the main difference about the input of the gating network. MoCE adopts the corresponding cluster embedding of the current token as in Eqn. 5, instead of the token embedding in Eqn. 1. Therefore, each expert can be trained by semantically similar images to alleviate the negative transfer phenomenon.

Cars, SUN397 and DTD, adding pre-training data from Split-B would in contrast degrade the downstream model performance, and vice versa for datasets whose semantics are close to organisms, such as Flowers, Food, and Pets. To conclude, the introduction of semantically irrelevant pre-training data might impede the transfer performance for specific downstream tasks. This negative transfer phenomenon motivates us to develop an efficient and automatic paradigm for task-customized pre-training.

### 3.2 Exploring TokenMoE to Masked AutoEncoder

**Overview of TokenMoE** Here, we introduce TokenMoE (Riquelme et al., 2021), a successful customized supervised pre-training paradigm built upon ViT (Dosovitskiy et al., 2020), which mainly consists of Transformer blocks with alternating multi-head self-attention (MSA) and multi-layer perceptron (MLP). Specifically, TokenMoE converts several Transformer blocks to Mixture of Expert (MoE) ones by expanding the MLP layer $N$ times, with each considered as an *expert*, denoted as $E_i(\cdot),\ i = 0, 1, 2, ..., N$. Conditional computation is performed between $N$ experts controlled by *gate*, denoted as $G(\cdot)$, which is a linear layer whose input is the token embedding $\boldsymbol{x}$, and the output is the top-$K$ probability among experts as,

$$G(x) = TopK(\sigma(\boldsymbol{W}_g \boldsymbol{x})), \tag{1}$$

where $\boldsymbol{W}_g$ is the parameters of the gate and $\sigma$ is the softmax function. $TopK(\cdot)$ will remain the top-$K$ largest values unchanged, but set others to be 0, and $K$ is the number of activated experts. Thus, each token will be routed to its corresponding experts. The final output can be represented as,

$$y = \sum_{i=1}^{N} G(\boldsymbol{x})_i E_i(\boldsymbol{x}). \tag{2}$$

Importance loss and load loss are also defined to enforce the balance usage of experts as in Riquelme et al. (2021). If not specified, we set $K$ to be 1 and $N$ to be 8 by default in all our experiments.

**Limitation of TokenMoE.** As shown in Table 2, naively adopting TokenMoE to MAE cannot achieve explicit improvement even if with intense hyper-parameter tuning and complicated data augmentations (*e.g.*, Repeat Augment and RandAugment with larger magnitude). We visualize the routing heatmaps of the pre-trained TokenMoE model in Fig. 3(a), and observe that the routing process demonstrates little correlation with ImageNet labels. Moreover, Expert-3 nearly dominates all the semantic classes, selected by 91.8% classes as the most frequently chosen expert, which degenerates the multi-expert design by utilizing single expert frequently. Check detailed analysis in Sec. 4.2.

As demonstrated in Fig. 1(b), we hypothesize that the degeneration is due to the usage of low-level pixel RGB values as the reconstruction target in MAE instead of the semantic class labels used in the original TokenMoE, which is also observed in Li et al. (2022).

### 3.3 MIXTURE OF CLUSTER-CONDITIONAL EXPERTS

To address the degeneration of TokenMoE, we propose the Mixture of Cluster-conditional Experts (MoCE), which trains each expert in an explicit semantic-aware manner. The general procedure mainly consists of data clustering, architecture and gate design, and deployment.

**Data Clustering.** To train each expert semantically, a clustering procedure is first performed to simulate the label partitioning in Sec. 3.1. With a pre-trained MAE model, we collect all image features normalized to be unit vector, so the feature matrix can be represented as $\boldsymbol{F} = [\boldsymbol{f_1}, \boldsymbol{f_2}, ..., \boldsymbol{f_n}]$, $\forall \|\boldsymbol{f_i}\|_2 = 1$. We denote $n$ as the number of images and $m$ as the desired number of clusters. The learnable centroids of the clusters are represented as $\boldsymbol{C} = [\boldsymbol{c_1}, \boldsymbol{c_2}, ..., \boldsymbol{c_m}]$, $\forall \|\boldsymbol{c_i}\|_2 = 1$. The assignment of each feature to the clusters can be calculated as $\boldsymbol{A} = \boldsymbol{F}^T\boldsymbol{C}$. Following Asano et al. (2019), we set $\boldsymbol{Q} \in \mathbb{R}^{m*n}$ as the posterior distribution of clustering, and the objective can be described as,

$$\max_{\boldsymbol{Q}} \; Tr(\boldsymbol{Q}^T\boldsymbol{A}) + \epsilon H(\boldsymbol{Q}) \quad s.t. \quad \boldsymbol{Q}\mathbf{1}_n = \frac{1}{m}\mathbf{1}_m, \; \boldsymbol{Q}^T\mathbf{1}_m = \frac{1}{n}\mathbf{1}_n, \tag{3}$$

where $\mathbf{1}_m$ is the vector of all ones in dimension $m$, and $H$ is the entropy function. The constraints force the clustering results to be balanced. During optimization, $\boldsymbol{Q}$ is solved by Sinkhorn-Knopp algorithm (Cuturi, 2013), while $\boldsymbol{C}$ is trained by SGD to minimize the cross entropy between $\boldsymbol{Q}$ and $\boldsymbol{A}$. We optimize $\boldsymbol{Q}$ and $\boldsymbol{C}$ iteratively, and take the final $\boldsymbol{C}$ and $\boldsymbol{Q}$ as the centroid of each cluster and the clustering assignments respectively.

**Architecture.** The whole network is trained on the full ImageNet simultaneously, with each expert trained by images of certain clusters depending on the routing results of the MoCE gates. Considering the size of each cluster could be relatively small and the effective training time of each expert is $K$ times shorter by expectation than other parameters with dense modeling (*e.g.*, MSA parameters (Riquelme et al., 2021)), we further adopt a distillation loss $\mathcal{L}_{distill}$ defined as the L2 distance between the feature generated by the whole network and each expert. In general, the loss function can be formulated as,

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{m} \mathcal{L}_{MAE}(D_i; \boldsymbol{\theta_i}) + \mathcal{L}_{distill}, \tag{4}$$

where $D_i$ means the $i$-th cluster and $\boldsymbol{\theta_i}$ denotes the parameters trained by $D_i$, which consists of several experts in different layers, as explained in the following.

**Gate Design.** Following the design of TokenMoE, we also replace several MLP layers in ViT with our MoCE layers equipped with the MoCE gates. Different from TokenMoE where each token is routed to different experts separately, we route the tokens from the images in the same cluster to the same expert. Therefore, the output of our MoCE gates can be represented as,

$$G(x) = TopK(\sigma(\boldsymbol{W}_g \cdot Cluster_Emb(x))), \tag{5}$$

where $Cluster_Emb(\cdot)$ returns the embedding of the cluster that $x$ belongs to. In practice, we fix it to be centroid matrix $\boldsymbol{C}$. Furthermore, we empirically find that the confidence of $G(x)$ is low, and consequently the mapping between clusters and experts varies a lot during pre-training. Inspired by the design of importance and load losses, we adopt another regularization loss $\mathcal{L}_{imbalance}$ to enhance the confidence of the gate result as,

$$\mathcal{L}_{imbalance} = -\sum_{i=1}^{n} \left( \frac{std(G(x)_i)}{mean(G(x)_i)} \right)^2. \tag{6}$$

n is the number of samples here. For practical implementation, the loss is calculated over the samples in a batch. The imbalance loss penalizes on the negative variance of the gate confidence. We find MoCE benefits from such design with more stable training.

**Deployment.** During the deployment stage, we try to customize the model for fine-tuning with best transfer results. As shown in the Sec. 3.1, the model sharing the closest semantics to the downstream task is preferred. We achieve this by re-using the data clustering. Specifically, we

Table 1: Transfer performance(%) of different self-supervised learning models on 11 downstream tasks, including both ResNet-50 and Vision Transformer based ones. The results of the supervised pre-trained model (DeiT (Touvron et al., 2021)) is also provided for reference. We utilize the officially released checkpoints for all baselines. Our MoCE outperforms the vanilla MAE model significantly by 2.45%, achieving state-of-the-art averaged accuracy. *: we continue to train a 1600-epoch pre-trained MAE for 200 epochs with total ImageNet to have a fair comparison with MoCE.

| | Aircraft | Caltech | Cars | C10 | C100 | DTD | Flowers | Food | Pets | SUN | VOC | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *ResNet-50* | | | | | | | | | | | | |
| BYOL | 82.39 | 90.12 | 87.33 | 96.28 | 82.15 | 74.57 | 95.96 | 82.13 | 88.52 | 64.41 | 83.97 | 84.35 |
| DeepClusterV2 | 78.75 | 90.51 | 86.33 | 96.48 | 82.28 | 75.43 | 96.16 | 83.68 | 90.33 | 66.68 | 81.37 | 84.36 |
| *Vision Transformer* | | | | | | | | | | | | |
| Supervised | 76.55 | 89.98 | 86.19 | 96.79 | 83.96 | 75.09 | 93.94 | 85.17 | 92.54 | 64.54 | 87.22 | 84.72 |
| DINO | 66.50 | 91.65 | 76.37 | 98.12 | 86.69 | 75.73 | 96.40 | 93.77 | 93.97 | 59.33 | 86.62 | 84.10 |
| MoCoV3 | 76.29 | 91.64 | 85.18 | 97.99 | 86.98 | 72.64 | 95.33 | 83.94 | 92.35 | 65.54 | 84.21 | 84.74 |
| BEiT | 53.16 | 79.02 | 68.11 | 94.34 | 73.54 | 68.04 | 91.33 | 79.59 | 84.02 | 56.13 | 65.65 | 73.90 |
| MAE | 72.38 | 90.47 | 83.51 | 95.69 | 68.40 | 75.48 | 96.10 | 79.98 | 92.35 | 62.43 | 84.79 | 81.96 |
| MAE* | 72.71 | 91.24 | 84.47 | 96.15 | 77.33 | 75.05 | 96.25 | 80.49 | 92.78 | 62.46 | 85.02 | 83.09 |
| MoCE (Ours) | 78.73 | 90.61 | 88.56 | 97.79 | 84.68 | 74.04 | 96.94 | 86.24 | 93.07 | 65.05 | 85.26 | **85.54**$^{+2.45}$ |

feed the downstream images through the pre-trained MAE model and collect all features $\boldsymbol{F}_{task}$. Then, the assignment of downstream images to the clusters can be computed by $\boldsymbol{A}_{task} = \boldsymbol{F}_{task}^T \boldsymbol{C}$. We select the cluster with the most downstream images, and extract out the corresponding experts for deployment, which we call a **sub-model** of the total MoCE model. Therefore, only a regular ViT model is used for downstream fine-tuning, which is much more efficient than TokenMoE, as demonstrated in Table 3.

## 4 Experiment

In this section, we first introduce the implementation details of pre-training and fine-tuning stage of MoCE in Sec. 4.1. Then we demonstrate the effectiveness of MoCE by evaluating the pre-trained models on a collection of 11 downstream tasks with detailed analysis of our MoCE superior to vanilla MAE and TokenMoE in Sec. 4.2. Finally we take ablation studies on the key components of MoCE in Sec. 4.3.

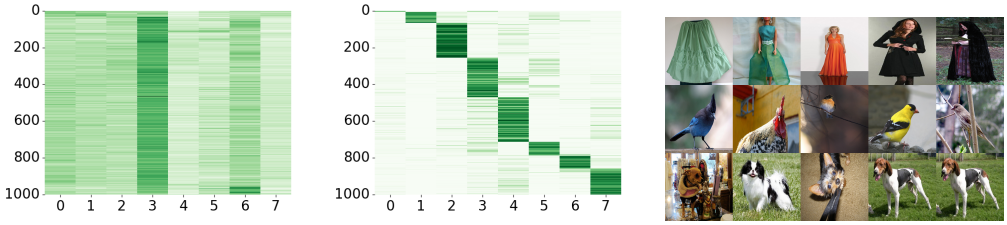
### 4.1 Setup

**Implementation details.** For all experiments, we replace two MLP layers with MoCE layers in the original ViT-B. Following (Wu et al., 2022), layers with the greatest gradient magnitude are selected (*i.e.* the last two MLP layers in ViT-B according to our experiments). The number of experts and the clusters are set to be 8 and 256 by default if not specified. Our model utilizes the officially released 1600-epoch pre-trained MAE model and performs continuous training with extra 200 epochs. Each expert is initialized by the corresponding dense model in addition with a small magnitude of noise. The training procedure mainly follows MAE except we multiply the base learning rate by 0.1. All regularization loss weight is set to 0.01 by default. For distillation loss, we store the features of the whole network at the end of each epoch to avoid heavy computation for inferring the whole network at each iteration.

**Downstream fine-tuning.** For downstream classification tasks, we try to set up a fair benchmark for Vision Transformers. We mainly follow the hyper-parameter settings of existing works (Dosovitskiy et al., 2020; Riquelme et al., 2021) and benchmark settings of (Ericsson et al., 2021). Specifically, all models are trained by SGD with a momentum of 0.9. Weight decay is set to be 0 and the learning rate is searched among [1e-4, 3e-4, 1e-3, 3e-3, 1e-2, 3e-2, 1e-1, 3e-1]. Each model is fine-tuned for 2500 steps with cosine learning rate decay, a batch size of 64, and 224×224 resolu-

Table 2: Transfer performance of MAE, TokenMoE, SDR and MoCE. SDR(ViT) is our re-implementation of SDR under ViT. We observe that TokenMoE cannot outperform vanilla MAE, while SDR(ViT) achieves better performance, which is further outperformed by MoCE.

| | Aircraft | Caltech | Cars | C10 | C100 | DTD | Flowers | Food | Pets | SUN | VOC | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MAE* | 72.71 | 91.24 | 84.47 | 96.15 | 77.33 | 75.05 | 96.25 | 80.49 | 92.78 | 62.46 | 85.02 | 83.09 |
| TokenMoE | 70.51 | 89.70 | 81.40 | 95.18 | 76.44 | 73.67 | 95.09 | 77.45 | 90.71 | 61.12 | 80.15 | 81.04 |
| SDR | 75.77 | 89.73 | 86.65 | 95.31 | 83.60 | 73.62 | 95.53 | 84.77 | 91.25 | 64.64 | 83.51 | 84.03 |
| SDR(ViT) | 76.57 | 90.04 | 86.95 | 96.92 | 81.42 | 73.09 | 96.14 | 82.90 | 92.65 | 64.40 | 85.37 | 84.22 |
| MoCE | 78.73 | 90.61 | 88.56 | 97.79 | 84.68 | 74.04 | 96.94 | 86.24 | 93.07 | 65.05 | 85.26 | **85.54** |




(a) Routing heatmap for each expert in TokenMoE.

(b) Routing heatmap for each expert in MoCE.

(c) Visualization of samples selected by different experts.

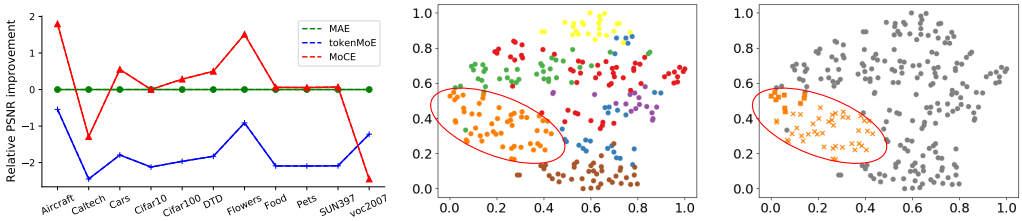
Figure 3: (a),(b): The routing heatmap for each expert in TokenMoE and MoCE. The x-axis is the Expert ID, and the y-axis is the ImageNet semantic label ID. Darker green means a higher proportional of tokens belonging to the corresponding class are allocated by the expert. (c): Samples randomly chosen from the pre-training dataset of 3 MoCE experts in (b). Samples in the same row corresponds to the same expert. We can see that each expert select images with similar semantics.

tion. We fine-tune each model 3 times and report the average performance. We find such a setting generates a stable result. We compare our method with current state-of-the-art ViT self-supervised models, DINO (Caron et al., 2021), MoCoV3 (Chen et al., 2021b), BEiT (Bao et al., 2021b), in addition to the several strong-performing ResNet-50 models, e.g. BYOL (Grill et al., 2020a) and DeepClusterV2 (Caron et al., 2018).

### 4.2 Results and Analysis

**Transfer Results.** The transfer performance of different self-supervised models is summarized in Table 1. It can be seen that MoCE achieves 2.45% improvement over the MAE baseline and reaches the state-of-the-art averaged accuracy, demonstrating the effectiveness of the task-customized pre-training paradigm. In addition, MoCE exhibits distinct superiority on different downstream tasks. For the fine-grained datasets like Aircraft, Cars and Food, MoCE outperforms the baseline model by a large margin, while for tasks like Caltech, Cifar-10 and VOC, MoCE shows limited improvement, which is reasonable, since for downstream fine-grained tasks that might share similar distributions with only a subset of the pre-training dataset, MoCE is able to find the model that is trained by the cluster the most similar with the downstream task, and thus alleviate negative transfer. However, when the downstream tasks contain images covering the whole semantic of the pre-train dataset, the negative transfer does not exist. Therefore, MoCE only gets limited improvement.

**Comparison between MoCE, TokenMoE and MAE.** The comparison is provided in Table 2, with all models trained with the same epochs for a fair comparison. We also provide the results of SDR (Liu et al., 2022), a task-customized model aiming at alleviating negative transfer, and further re-implement their idea under ViT architecture notated as SDR(ViT). It can be seen that TokenMoE performs even worse than MAE, suggesting that naively adopting MoE to MAE is not feasible. Both MoCE and SDR models exceed vanilla MAE, demonstrating the effectiveness of task-customized method for alleviating negative transfer. MoCE further outperforms SDR(ViT), indicating the importance of self-adaptive routing. Besides transfer accuracy, we also provide the mean of peak signal-to-noise ratio (PSNR) (Sara et al., 2019) comparison to show the generation quality in Fig. 4(a) and MoCE still exhibits consistent improvement over TokenMoE and MAE on most datasets.




(a) Relative PSNR improvement. (b) Mapping between the cluster and the gate for two MoCE layers.

Figure 4: (a): PSNR comparison of MAE, TokenMoE and MoCE for each downstream task. Here shows the relative improvement over MAE. MoCE performs consistently better than other two methods on most datasets. (b): The assignment of each cluster to the expert for the former MoCE layer (left figure) and the latter one (right figure). Left: each point represents a cluster, and the color denotes the expert it is routed to. Right: only the clusters belongs to the expert *Orange* is painted for clear understanding. Different shapes represents the expert it is routed to. It can be seen that clusters routed to expert *Orange* will be split again in the next MoCE layer. Therefore we obtain a meaningful hierarchy structure for clusters of pre-training data.

**Analysis on experts.** We contribute the improvement of MoCE to the effective learning of experts. As shown in the routing heatmap in Fig. 3(a), each expert of TokenMoE shows little correlation with the semantics, while each expert of MoCE in Fig. 3(b) is trained by several classes, showing a more balanced assignment of images to experts. We also visualize the samples for 3 random experts in Fig. 3(c). It is interesting to find that expert 1 is mostly trained by images containing clothes, and experts 2 and 3 are trained mostly by birds and dogs respectively.

In addition to the visualization, we quantitatively show that each expert is trained by close semantics. Following (Mikolov et al., 2013), we select the corresponding label set that is trained by each expert and compute the L2 distance between the labels inside and between the sets with label embedding produced by CLIP Radford et al. (2021). The L2 distance inside the set averaged on 8 experts is 0.84, smaller than that between the sets 0.92, indicating that MoCE gates automatically cluster similar semantics to each expert, and consequently benefit the downstream transfer.

We further show that by stacking multiple MoCE layers, a hierarchy cluster structure is automatically formatted. As shown in Fig. 4(b), the former MoCE layer dispatches the clusters into several experts, and the latter MoCE layer further splits the clusters into its experts. The latter MoCE layer handles the clusters in a finer way.

**Training and testing efficiency.** Besides the transfer performance of MoCE, we provide the training and testing efficiency comparison with TokenMoE. As shown in Table 3, all three models share similar FLOPs. However, TokenMoE needs to be deployed the full model during both the training and testing phase, while MoCE only needs to deploy a single expert and therefore halves the required number of parameters when testing. In addition, training and testing speed are improved by 10% and 30% respectively, which is attributed to the reduction of token shuffle operation as tokens in the same image do not split and will be dispatched to the same expert, significantly reducing the communication overhead.

### 4.3 Ablation

**Searching method.** When the downstream task comes, it is expensive to fine-tune each expert to choose the best one, requiring an efficient searching method to provide the task-customized expert. Here we ablate several methods including early stop, KNN, LogME You et al. (2021), and our method based on the clustering results, as discussed in Sec. 3.3. We also provide the best and worst sub-model for reference. We choose Aircraft, Flowers, Food as representatives since the size of these datasets covers the smallest and largest ones. It can be seen in Table 4 that MoCE performs stably among different sizes of the dataset, and the searching time is negligible as we only need to infer the downstream task once. We attribute this to the nature design of MoCE that combines clustering and pre-training into a single paradigm.

Table 3: Training (top 3 rows) and testing (last 3 rows) efficiency comparison.

| | MAE | TokenMoE | MoCE |
|---|---|---|---|
| Params (M) | 111.91 | 178.03 | 178.03 |
| FLOPs (G) | 9.80 | 9.81 | 9.81 |
| Speed↑ | 1.41x | 1x | **1.18x** |
| # Params (M) | 85.88 | 152.00 | 85.88 |
| FLOPs (G) | 16.88 | 16.88 | 16.88 |
| Speed↑ | 1.37x | 1x | **1.37x** |

Table 4: Comparison between expert searching algorithms.

| | Aircraft | Flowers | Food | Time(GPUh) |
|---|---|---|---|---|
| Best | 79.92 | 97.96 | 86.24 | 288 |
| Worst | 69.84 | 94.97 | 81.51 | 288 |
| Early stop | 77.00 | 96.83 | 85.33 | 144 |
| KNN | 71.40 | 95.10 | 83.32 | 5 |
| LogME | 73.84 | 96.54 | 85.11 | 5 |
| MoCE | 78.73 | 96.94 | 86.24 | 1 |

Table 5: Ablations of the expert number in a MoCE layer.

| # experts | Acc. |
|---|---|
| 1 | 83.09 |
| 2 | 83.01 |
| 4 | 84.22 |
| **8** | **85.54** |

Table 6: Ablations of the number and location of MoCE layers.

| # MoCE layers | Acc. |
|---|---|
| 1 | 83.09 |
| 2 (10th & 12th) | 83.08 |
| **2 (11th & 12th)** | **85.54** |
| 4 | 85.59 |

Table 7: Ablations of the number of clusters.

| # clusters | Acc. |
|---|---|
| 16 | 82.00 |
| 64 | 84.02 |
| **256** | **85.54** |
| 512 | 85.33 |

**Scalability.** We ablate on the scalability of MoCE in three aspects. All default settings are marked in **bold**. Firstly, we vary the number of experts in each MoCE layer. As shown in Table 5, with more experts comes explicitly consistent improvement to the averaged accuracy. We further ablate the number and location of MoCE layers. As mentioned in Sec. 4.1, we select MoCE layers based on the gradient magnitude, which is the 11th and 12th MLP layers (3rd row in Table 6) according to our experiments, while the last 2 *even* MLP layers (*i.e.*, 10th and 12th layers, 2nd row in Table 6) are chosen in TokenMoE by default. As shown in Table 6, we notice that it is essential to replace the right layer to be MoCE layer. Adding more MoCE layers shows little improvement. We also train MoCE with different number of clusters. As shown in Table 7, the accuracy continuously increases when the number of clusters goes up to 256, but start to drop later. We hypothesize that with moderate number of clusters, MoCE can produce a variety of task-customized models. However, with even more clusters, the number of experts become the bottleneck and performance starts to saturate.

## 5 Conclusion

In this work, we first show the negative transfer phenomenon that exists in the prevailing self-supervised learning method MAE through extensive experiments. It will impede the scalability of MAE as more pre-training data may instead degenerate the downstream performance. In order to tackle the problem, we introduce Mixture of Expert to MAE as the multi-experts design can equip MAE with different ability that aids transfer. However, different from supervised pre-training, TokenMoE suffers from the fact that the gate shows no correlation to the semantic and the transfer ability is not improved. Based on this, we propose MoCE to explicitly train each expert with different clusters through the MoCE gate design and several losses to stablize the training process. A search algorithm for selecting the best model for transfer is also proposed based on the clustering priors. Extensive experiment shows that MoCE trains each expert with meaningful semantics and achieves state-of-art transfer performance on a collection of 11 downstream tasks.

In addition to the transfer accuracy, MoCE can automatically format the hierarchy structure of pre-train data and is also deployed more lightly than TokenMoE. It is the first work that successfully trains a self-supervised learning MoE model on ImageNet only. We hope such a design will motivate more research on the self-supervised MoE models.

## REFERENCES


Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019.

Yutong Bai, Zeyu Wang, Junfei Xiao, Chen Wei, Huiyu Wang, Alan Yuille, Yuyin Zhou, and Cihang Xie. Masked autoencoders enable efficient knowledge distillers. *arXiv preprint arXiv:2208.12256*, 2022.

Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021a.

Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021b.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, pp. 132–149, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv:2006.09882*, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Kai Chen, Lanqing Hong, Hang Xu, Zhenguo Li, and Dit-Yan Yeung. Multisiam: Self-supervised multi-instance siamese representation learning for autonomous driving. In *International Conference on Computer Vision*, pp. 7546–7554, 2021a.

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

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9640–9649, 2021b.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pp. 1422–1430, 2015.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Linus Ericsson, Henry Gouk, and Timothy M Hospedales. How well do self-supervised models transfer? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5414–5423, 2021.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2021.

Dehong Gao, Wenjing Yang, Huiling Zhou, Yi Wei, Yi Hu, and Hao Wang. Network clustering for multi-task learning. *arXiv preprint arXiv:2101.09018*, 2021.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020a.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020b.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. Beyond distillation: Task-level mixture-of-experts for efficient inference. *arXiv preprint arXiv:2110.03742*, 2021.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. Base layers: Simplifying training of large, sparse models. In *International Conference on Machine Learning*, pp. 6265–6274. PMLR, 2021.

Xiaotong Li, Yixiao Ge, Kun Yi, Zixuan Hu, Ying Shan, and Ling-Yu Duan. mc-beit: Multi-choice discretization for image bert pre-training. *arXiv preprint arXiv:2203.15371*, 2022.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019.

Zhili Liu, Jianhua Han, Kai Chen, Lanqing Hong, Hang Xu, Chunjing Xu, and Zhenguo Li. Task-customized self-supervised pre-training with scalable dynamic routing. *Transfer*, 55:65, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1930–1939, 2018.

Matthew McDermott, Bret Nestor, Evan Kim, Wancong Zhang, Anna Goldenberg, Peter Szolovits, and Marzyeh Ghassemi. A comprehensive ehr timeseries pre-training benchmark. In *Proceedings of the Conference on Health, Inference, and Learning*, pp. 257–278, 2021.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

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

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, Cedric Renggli, André Susano Pinto, Sylvain Gelly, Daniel Keysers, and Neil Houlsby. Scalable transfer learning with expert models. *arXiv preprint arXiv:2009.13239*, 2020.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, and J. Clark. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021.

Umme Sara, Morium Akter, and Mohammad Shorif Uddin. Image quality assessment through fsim, ssim, mse and psnr—a comparative study. *Journal of Computer and Communications*, 7(3):8–18, 2019.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers amp; distillation through attention. In *International Conference on Machine Learning*, volume 139, pp. 10347–10357, July 2021.

Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. *arXiv preprint arXiv:2011.09157*, 2020.

Lemeng Wu, Mengchen Liu, Yinpeng Chen, Dongdong Chen, Xiyang Dai, and Lu Yuan. Residual mixture of experts. *arXiv preprint arXiv:2204.09636*, 2022.

Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. *arXiv preprint arXiv:2102.04803*, 2021a.

Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. *arXiv preprint arXiv:2011.10043*, 2020.

Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. *arXiv preprint arXiv:2111.09886*, 2021b.

An Yang, Junyang Lin, Rui Men, Chang Zhou, Le Jiang, Xianyan Jia, Ang Wang, Jie Zhang, Jiamang Wang, Yong Li, et al. M6-t: Exploring sparse expert models and beyond. *arXiv preprint arXiv:2105.15082*, 2021.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. Logme: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning*, pp. 12133–12143. PMLR, 2021.

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

Table 8: Comparison of MAE and MoCE under equal parameter counts. We train MAE with a larger model that shares the same parameter count as the whole model of MoCE.

| | # Params | Aircraft | Caltech | Cars | C10 | C100 | DTD | Flowers | Food | Pets | SUN | VOC | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MAE | 178.03 | 74.43 | 90.3 | 85.5 | 96.9 | 83.8 | 74.84 | 96.3 | 81.86 | 92.97 | 62.98 | 85.51 | 84.13 |
| MoCE | 178.03 | 78.73 | 90.61 | 88.56 | 97.79 | 84.68 | 74.04 | 96.94 | 86.24 | 93.07 | 65.05 | 85.26 | 85.54 |

Table 9: Comparison of MAE and MoCE on ViT-L. We also train MAE with 2 subsets of ImageNet, namely Split-A and Split-B, following the same setting mentioned in Sec. 3.1. This table shows that negative transfer still exists on larger architectures, while MoCE can alleviate this problem and achieve better transfer results.

| | Aircraft | Caltech | Cars | C10 | C100 | DTD | Flowers | Food | Pets | SUN | VOC | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MAE(L) | 74.30 | 93.97 | 88.60 | 97.85 | 82.47 | 77.61 | 96.67 | 81.22 | 93.97 | 67.99 | 88.30 | 85.72 |
| Split-A(MAE-L) | 79.70 | 91.59 | 89.33 | 96.97 | 80.38 | 78.67 | 95.44 | 82.97 | 92.49 | 68.73 | 82.41 | 85.33 |
| Split-B(MAE-L) | 73.42 | 90.80 | 86.00 | 96.18 | 78.73 | 77.34 | 96.75 | 83.63 | 94.92 | 66.06 | 85.85 | 84.52 |
| MoCE(L) | 87.04 | 94.86 | 90.72 | 98.29 | 87.49 | 76.65 | 97.38 | 88.21 | 95.89 | 69.49 | 89.13 | **88.65**$^{+2.93}$ |

# A Appendix

## A.1 Details for clustering.

For data clustering, the features are computed by inferring the pre-train MAE, and the matrix $Q$ and $C$ are solved by the Sinkhorn-Knopp algorithm and SGD optimizer iteratively. For the Sinkhorn-Knopp algorithm, we set the iteration number as 3. The learning rate of SGD is set to 0.1, the momentum is 0.9 and weight decay is set to 0.9 for the sparsity assignment of cluster results. We train 10 epochs in total and it costs 3 minutes and 20 seconds on average for a single GPU.

## A.2 Comparison under fair parameter counts.

The setting used in our work focuses on a fair comparison of **FLOPs**, referring to Table 4 in the main paper. Since TokenMoE and MoCE always activate only one expert throughout the whole pre-training and fine-tuning procedure, the FLOPs value is maintained close to MAE. Apart from this criterion, as suggested, we further provide the comparison on equal **parameter** counts. As shown in the Table 8, we train MAE under same parameter count as the whole model of MoCE, and MoCE still outperforms MAE consistently.

## A.3 MoCE for larger architecture.

Here we provide the analysis on the larger architecture(ViT-L, 2.57× larger than ViT-B) to explore the scalability of MoCE. We first demonstrate that negative transfer still exists for larger architecture by training MAE with total ImageNet, Split-A and Split-B following the same setting in Sec. 3.1. As shown in the first three rows of Table. 9, similar phenomenon is oberserved that MAE trained by Split-A performs better in Aircraft, Cars, DTD and SUN while Split-B in Flowers, Food, and Pets. On the other hand, MoCE(L) can still alleviate the problem and therefore transfers better. We believe that the negative transfer phenomenon mainly exists when a common pre-trained model is used for various downstream tasks, due to the inevitable semantic gaps between the pre-training and downstream datasets, rather than the architecture.

## A.4 Performance of MoCE without pre-training.

We provide results of MoCE trained from scratch for 200 epochs in Table 10. In this experiment, for clustering, we first pretrain MAE for 50 epochs and perform clustering. We then train MoCE from scratch for 200 epochs based on the clustering results. Although it is a common practice to utilize pre-trained dense models as initialization to accelerate pre-training (Wu et al., 2022; Bai et al., 2022), MoCE still outperforms MAE consistently in various downstream tasks when trained from scratch.

Table 10: Comparison of MAE and MoCE both training from scratch for 200 epochs.

| | Aircraft | Caltech | Cars | C10 | C100 | DTD | Flowers | Food | Pets | SUN | VOC | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MAE | 64.73 | 85.91 | 77.10 | 92.92 | 72.50 | 73.30 | 93.11 | 73.14 | 88.70 | 57.84 | 73.27 | 77.50 |
| MoCE | 71.16 | 90.55 | 82.46 | 96.06 | 76.56 | 74.57 | 95.70 | 79.67 | 92.58 | 62.20 | 84.25 | 82.34 |

### A.5 Results on semantic segmentation.

We also evaluate MoCE on the semantic segmentation task that aims to predict the class for each pixel in the image. We report the metric of mean Intersection of Union (mIoU) averaged over all semantic categories in ADE20K. We use Adam (Loshchilov & Hutter, 2017) as the optimizer. The learning rate is set to 1e-3 with layer-wise decay to be 0.65. We conduct fine-tuning for 160K steps. The batch size is 16. The detailed hyperparameters can refer to Bao et al. (2021a). It can be seen that MoCE still outperform MAE and achieves the state-of-the-art results.

Table 11: Semantic segmentation on ADE20K.

| | mIoU |
|---|---|
| Deit | 46.9 |
| MoCov3 | 46.8 |
| DINO | 46.9 |
| MAE | 48.1 |
| MoCE | **48.3**$^{+0.2}$ |