MoDE: CLIP Data Experts via Clustering

Jiawei Ma[2,*] Po-Yao Huang[1] Saining Xie[3] Shang-Wen Li[1]
Luke Zettlemoyer[1,4] Shih-Fu Chang[2] Wen-Tau Yih[1] Hu Xu[1,+]
[1]FAIR, Meta [2]Colubmia University [3]New York University [4]University of Washington

Abstract

The success of contrastive language-image pretraining (CLIP) relies on the supervision from the pairing between images and captions, which tends to be noisy in web-crawled data. We present Mixture of Data Experts (MoDE) and learn a system of CLIP data experts via clustering. Each data expert is trained on one data cluster, being less sensitive to false negative noises in other clusters. At inference time, we ensemble their outputs by applying weights determined through the correlation between task metadata and cluster conditions. To estimate the correlation precisely, the samples in one cluster should be semantically similar, but the number of data experts should still be reasonable for training and inference. As such, we consider the ontology in human language and propose to use fine-grained cluster centers to represent each data expert at a coarse-grained level. Experimental studies show that four CLIP data experts on ViT-B/16 outperform the ViT-L/14 by OpenAI CLIP and OpenCLIP on zero-shot image classification but with less (<35%) training cost. Meanwhile, MoDE can train all data expert asynchronously and can flexibly include new data experts. The code is available here.

1. Introduction

Contrastive Language-Image Pretraining (CLIP) learns versatile vision-language representations which are transferable across diverse downstream tasks. Existing models, such as OpenAI CLIP [35], OpenCLIP [40] and MetaCLIP [46], are trained with a large collection of web-crawled image-caption pairs. Specifically, for each image, its paired caption is viewed as a *positive* example, and the captions of all the other images are viewed as *negatives*. The model then learns to project both images and captions into a shared space, where the embedding of the positive caption is drawn closer to the image embedding, compared to the embeddings of all the other negative captions.

The key to the success of contrastive vision-language representation learning lies in the creation of quality *negative* examples for training [7, 13]. A single image can be depicted by texts with different meanings (*i.e.*, semantics), covering multiple details and interpretations, as illustrated in Fig. 1. Because the paired caption usually describes limited visual content, it is common to see that two similar images have drastically different textual descriptions, especially in noisy web-crawled data. When those image-caption pairs are sampled in the same batch, captions of other images become *false negatives* — acceptable captions yet being treated as negative descriptions of the target image. Conversely, if only dissimilar image-caption pairs are sampled, the contrastive learning problem becomes trivial. Incorporating *hard negatives* [7, 33, 45] (e.g., incorrect yet similar captions that share many words of a correct textual description) in training batches has often been shown to improve the model performance.

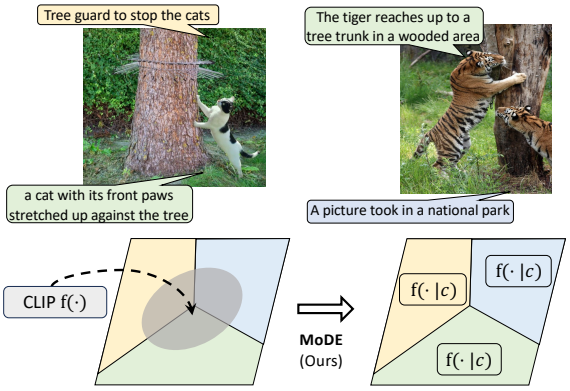

Figure 1. For an image-caption pair, the caption may describe limited visual content or even be unrelated, and such noises unavoidably hurt the quality of negative examples to learning a single model. We propose to uncover the clusters from training data, where 1) the pairs with similar images but different captions are assigned to different clusters and 2) the samples in each cluster are of related meanings, and learn a *Data Expert* for each cluster. These experts are then selectively ensembled for inference.

In this work, we introduce the Mixture of Data Experts (MoDE) framework (shown in Fig. 1-bottom) via clustering. MoDE separates false negative samples into different

[*] Research done while Jiawei Ma was an intern at FAIR.
[+] Project Lead.

clusters and groups the pairs with similar semantics, which mitigates noise from false-negative captions while incorporating a more challenging set of hard-negative examples, thereby enhancing vision-language pre-training. MoDE consists of two main steps: (1) the training data (*i.e.*, image-caption pairs) is first clustered into several disjoint subsets by the captions; each cluster is then used to train a model following the standard contrastive learning method. In this way, each model is specialized by the training data in one cluster and thus termed as a *Data Expert*. (2) When applied to downstream tasks, such as image classification, the task metadata (*i.e.*, class names), are first compared to the centroid of each data cluster to determine which data expert needs to be activated. Selected data experts are then used to create the embeddings of the test image and classes. The class with the highest ensembled similarity is then output as the classification result.

Empirically, MoDE outperforms several state-of-the-art vision-language models when applied to multiple standard benchmarks, including +3.7% on image classification in CLIP benchmark [31, 35], +3.3% on image-to-text retrieval and +2.7% on text-to-image retrieval on COCO [26]. The superiority of MoDE can be attributed to better trained individual data expert models, due to the fact that examples in the same cluster, when used for contrastive learning, provide more quality negatives. Because captions in the same cluster are different but semantically similar (*e.g.*, "a cat climbs a tree", "a tiger reaches up to a tree"), they become challenging negative examples when compared with images that are not the originally paired ones. On the other hand, it is also less likely to encounter a false negative case where a very different caption validly describes the same image (*e.g.*, "tree guards to stop the cats" in Fig. 1). MoDE is also uniquely positioned for large-scale training when billions of image-caption pairs are available. As each data expert uses only a fraction of the whole dataset, it can be more easily trained with fewer compute resources asynchronously. From experiments across different ViT [5] model scales, we show that four ViT-B/16 data experts can outperform the single ViT-L/14 model by OpenAI CLIP [35] and OpenCLIP [39] on image classification but requires much less ($<$35%) training cost. In summary, our contributions are:

- We investigate the quality *negative* samples in contrastive language-image pretraining, and in particular, the noise of *false negatives* in web-crawled image-caption pairs.
- We propose the MoDE framework to learn a system of CLIP data experts via clustering, and adaptively ensemble data experts for downstream tasks at inference time.
- Extensive experimental study has demonstrated the effects in zero-shot transfer benchmarks with low training cost. MoDE can include new data experts flexibly and is thus beneficial for continual pre-training.

2. Related Work

Contrastive Language Image Pretraining (CLIP) aims to learns robust & transferable visual representations from large-scale data. Scaling up [17, 34] existing approaches and improving the effectiveness is critical. Recent progress in the field involves the exploration of regularization techniques [49] and hyperbolic embedding methods [4] but they require significant effort for data annotation. Data curation is then proposed to remove noisy web-crawled image-caption pairs. Additionally, methods like image masking [25] and concise captions [24] efficiently decrease memory demands, enabling the use of larger batch sizes and model sizes. However, a trade-off between training cost and effectiveness still exists. Following the studies [20, 37] in contrastive learning [2, 15], recent work investigated negative samples in CLIP training but still focuses on image side [27, 44]. The noise exhibited in captions [47] is then overlooked. In this study, we tackle the data noise and the discovery of negative samples via clustering. Rather than training a single model, we asynchronously train multiple data experts and then directly ensemble them for inference adaptively, which also shows benefits for model scaling.

Mixture-of-Expert (MoE) trains a set of sub-models and a routing module. Originally, each expert is defined as an entire network [16, 18], and a single model is selected for each data adaptively. As restricting to hard model selection may limit the practicality, deep mixture of expert [6], is then proposed where the MoE layer is set to softly ensemble layer outputs via weighted sum, which is then investigated with different architectures [8, 22] in various tasks [36, 41]. However, all expert models are still trained on the same data simultaneously, resulting in much heavier training costs. Recently, BTM [12, 23] proposes to learn expert models on different document types (*e.g.*, papers, posts) separately but is only validated on language models. Meanwhile, both MoE and BTM can only determine the model routing for each input separately. Instead, MoDE generalizes to task-level adaptation and ensembles the models by task metadata (*e.g.*, class names in classification task [3]).

Inference-Time Adaptation adapts a pre-trained model quickly and effectively to new tasks. Initially, transductive learning [9] is studied and leverages all unlabeled test data for model update. To mitigate the dependence on the presumed distribution of test data, test-time training [10, 38, 43] is developed to generate individual models for each input. Subsequent explorations into meta-learning [14, 28, 42] introduced a separate module (*i.e.*, meta-learner) that can adapt the pre-trained model for each task with a few annotated examples. MoDE has inference-time task adaptation but without annotation or parameter update.

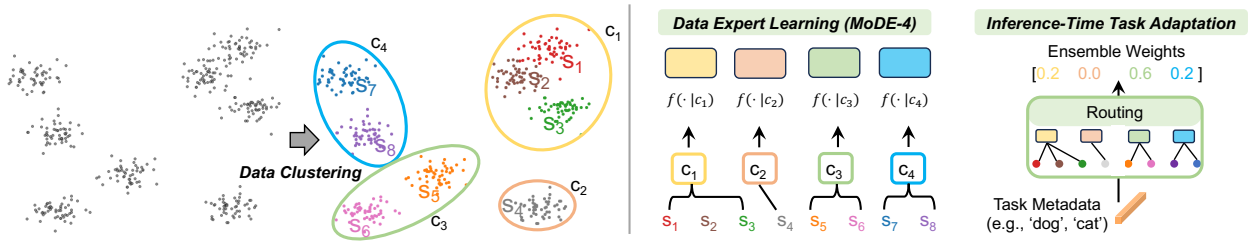

Figure 2. Framework of MoDE via clustering. (Left) We perform a two-step clustering on captions to decide clusters / conditions for data experts. The colored scatter plots are fine-grained clusters and the circles are clusters at coarse-grained level. (Right) Each coarse-grained cluster (c) conditions the learning of one data expert $f(\cdot|c)$ and all data experts (colored boxes) are learned asynchronously. For inference, the similarity between task metadata and fine-grained cluster centers ($\{s\}$) is used to decide the routing of data experts. To keep reasonable training cost, all data experts can be initialized with a model partially trained on all data without clustering (omitted for simplicity).

3. CLIP Data Experts

For contrastive image-language pre-training, the model is trained to accurately align each image with the captions describing the visual content. In a manner of divide-and-conquer [1], for each CLIP data expert training on one cluster, we reduce the amount of false negatives and increase the hard negatives within each mini-batch. In this way, we mitigate noise exhibited in web-crawled image-caption pairs and make the model training more effective.

As shown in Fig. 2, on top of the established CLIP training that learns a single dense CLIP model $f(\cdot)$ (Sec. 3.1), we propose to learn a set of CLIP data experts $\{f(\cdot|c)\}$ via unsupervised clustering (Sec. 3.2) and each CLIP data expert $f(\cdot|c)$ is trained on the cluster c (Sec. 3.3). In this way, the conditioned data expert $f(\cdot|c)$ is less sensitive to the noise from other clusters and can be effectively trained among the data of coherent semantics. For each evaluation task, by measuring the correlation between the task metadata (*e.g.*, class names) and the conditions, the outputs can be jointly decided by multiple data experts (Sec. 3.4).

3.1. Background: Vanilla CLIP Training

CLIP [35] learns separate vision and language encoders in a joint embedding space. By contrasting positive pairs from negative samples within the same batch, CLIP can accurately model the similarity of the image and caption in each pair. We denote CLIP as $f\big((\mathbf{x}_v, \mathbf{x}_l)\big)$ for an image-caption input $(\mathbf{x}_v, \mathbf{x}_l)$, and simplify CLIP model as $f(\cdot)$. As a reminder, instead of learning a single dense CLIP model $f(\cdot)$, we propose to learn a set of CLIP data experts independently given a set of conditions C, *i.e.*, $\{f(\cdot|c)|c \in C\}$.

3.2. Clustering

This subsection discusses how to formulate conditions C, and how to use clustering to automatically discover conditions for data experts from the pre-train set. In a nutshell, the desiderata for the conditions are twofold: 1) as each task at test time requires detailed description (*e.g.*, recognize the "cat" species instead of just "animal"), the conditions should be *representative* such that the correlation with tasks can be precisely modeled for reliable data experts selection; 2) the number of conditions should be *reasonable* since each condition is used to learn one data expert. As each condition is represented by a cluster, the ideals of *representative* likely ask for more fine-grained clustering whereas the latter may require for fewer data experts.

Instead, motivated by the ontology in human language, we propose to capture such a hierarchical structure via clustering, *i.e.*, determine the condition of a data expert at the coarse-grained level and represent it via the set of fine-grained clusters. For simplicity and efficiency of scaling, we design a *two-step K-means clustering*. We employ fine-grained clustering to locate each cluster whose samples are of similar semantics, such that the cluster centers are representative (Step 1), and then group fine-grained clusters to determine coarse-grained clustering among data for data experts' specialization (Step 2). In this way, instead of using a single coarse-grained center, the condition is symbolized by the fine-grained cluster centers. The features for clustering are extracted from captions (details studied in Sec. 5).

Step 1: Fine-grained Clustering. As the amount of pre-train data $\mathcal{D}$ is huge (hundreds of millions to billions level for CLIP [35]), it could be inefficient to train K-means over all pre-training data. Instead, we first uniformly sample a subset from the pre-training set: $\mathcal{D}' \sim \mathcal{D}$ and $|\mathcal{D}'| \ll |\mathcal{D}|$. Then, we perform K-means training [30] over $\mathcal{D}'$:

$$S \leftarrow \text{K-means}(\mathcal{D}'), \tag{1}$$

where S is a set of learned cluster centers. Note that the number of fine-grained clusters $m = |S|$ can be substantially large such that the cluster center of each cluster well represents coherent semantic information for each cluster.

Step 2: Coarse-grained Clustering. To efficiently allocate the training/inference of a data expert, we perform a second round, *i.e.*, coarse-grained, K-means clustering on top of fine-grained cluster centers S:

$$C \leftarrow \text{K-means}(S), \tag{2}$$

where each coarse-grained cluster center $c \in C$ is the condition for a data expert. We denote $n = |C|$ as the number of data experts where $n \ll m$, and S_c as set of fine-grained clusters assigned to data expert $f(\cdot|c)$ where $S = \cup_{c \in C} S_c$.

3.3. Data Experts Training

Next, we formulate training data for each data expert. We first collect the data assigned for each fine-grained cluster s: $\mathcal{D}_s = \{d | s = \arg\min_{s \in S}(\|\mathbf{e}_d - \mathbf{e}_s\|_2^2) \text{ and } d \in \mathcal{D}\}$, where $\mathbf{e}_d$ and $\mathbf{e}_s$ are the embeddings for training example d and fine-grained cluster center s respectively. To train a data expert $f(\cdot|c)$, its corresponding CLIP training data is:

$$\mathcal{D}_c = \bigcup\nolimits_{s \in S_c} \mathcal{D}_s. \tag{3}$$

For convenience, we use MoDE-n to indicate the system with n CLIP data experts. For training efficiency, all data experts are specialized from the same seed CLIP model that is partially trained over the entire set $\mathcal{D}$. Then, each data expert $f(\cdot|c)$ is trained only on $\mathcal{D}_c$.

3.4. Inference Time Task-Adaptation

As our framework conditions the model expertise on clusters to train data experts, it also gives multiple models to choose from during inference (instead of the only choice on a single CLIP model). This gives the room to adapt different data experts to various downstream tasks.

We propose a simple approach to adapt data experts (no parameter updates) to downstream tasks using the task metadata. Intuitively, this approach routes each downstream task adaptively and efficiently to data experts during inference. For simplicity, we formulate the data experts routing as a weighted sum of data experts' outputs. Formally, given an evaluation task $\mathbf{T}$, the output of CLIP data experts is

$$\sum\nolimits_{c \in C} f(\cdot|c) p(c|\mathbf{T}), \tag{4}$$

where $p(c|\mathbf{T})$ is the normalized weight for the data expert $f(\cdot|c)$ and the data expert will not be used for inference if the weight is close to zero. The weight is proportional to the correlation, *i.e.*, similarity, between metadata of task $\mathbf{T}$ and condition c. Below we provide simple implementations for zero-shot classification and retrieval, respectively.

Zero-Shot Classification. To have accurate routing, we leverage fine-grained cluster centers S in Step 1 to route a task to data experts. We treat the set of class names L as metadata, and define the similarity matrix between classes and data experts as $\mathbf{A} \in \mathbb{R}^{|L| \times m}$. To compute $\mathbf{A}$, we first compute $\mathbf{e}_l$ as the embedding for class $l \in L$ via the same encoder for the embedding of fine-grained cluster center $\mathbf{e}_s$. Then each entry is defined as

$$\mathbf{A}_{l,s} = \exp(-\|\mathbf{e}_l - \mathbf{e}_s\|_2^2 / \lambda), \tag{5}$$

where $\lambda \in \mathbb{R}^+$ is a temperature to sharpen the similarities. Further, the weight routing to a data expert $f(\cdot|c)$ is proportional to

$$p(c|\mathbf{T}) \propto \exp(\sum\nolimits_{l \in L} \sum\nolimits_{s \in S_c} \mathbf{A}_{l,s}). \tag{6}$$

In practice, we found that using the nearest neighboring fine-grained cluster center ($\arg\max_{s \in S} \mathbf{A}_{l,s}$) for each class $l \in L$ is good enough to reduce noises in routing.

Zero-Shot Retrieval. The retrieval tasks consist of text retrieval and image retrieval. For text retrieval where each image is used to retrieve a text from a large corpus Q, we leverage Q as metadata to build similarity matrix $\mathbf{A} \in \mathbb{R}^{|Q| \times m}$. Similar to the classification task, the weights for ensembling can be naturally adopted for MoDE:

$$p(c|\mathbf{T}) \propto \exp(\sum\nolimits_{q \in Q} \sum\nolimits_{s \in S_c} \mathbf{A}_{q,s}), \tag{7}$$

where each entry $\mathbf{A}_{q,s}$ is computed as $\exp(-\|\mathbf{e}_q - \mathbf{e}_s\|_2^2 / \lambda)$, where $\mathbf{e}_q$ is the embedding for text q. For image retrieval where each text q retrieves an image separately, we treat the retrieval by text q as an independent task $\mathbf{T}_q$ such that the ensembling weights are then $p(c|\mathbf{T}_q) \propto \exp(\sum_{s \in S_c} \mathbf{A}_{q,s})$.

4. Experiment

4.1. Data

We use the datasets collected in MetaCLIP [46] for evaluation and conduct experiments on image-caption pairs at two scales: 400M (similar to the scale in OpenAI CLIP), and 2.5B to scale MoDE. All images are pre-processed with face-blurring and de-duplication against benchmarks.

4.2. Training Setup

Clustering Setup. We use the pre-trained language model SimCSE [11] to extract the embeddings for all captions where the advantages of language encoders over CLIP encoders are studied in Sec. 5.3. We use balanced K-means [29] for both of the two unsupervised clustering steps. We set the number of fine-grained clusters $m = 1024$, and report performance for both MoDE-2 and MoDE-4 below to directly show the improvement by increase the number of data expert models on all evaluation tasks.

Data Experts Training Setup. We follow OpenAI CLIP's hyper-parameters [35] for fair comparison and train on the same budget of 12.8B image-caption pairs (32 epochs of 400M), with a global batch size of 32,768. We train MoDE under 3 scales: for ViT-B/32 and ViT-B/16, we use 64 Nvidia V100 GPUs with a per GPU batch size of 512, and for ViT-L/14, we use 128 GPUs with a 256 per GPU batch

	Average	ImageNet	Food-101	CIFAR10	CIFAR100	CUB	SUN397	Cars	Aircraft	DTD	Pets	Caltech-101	Flowers	MNIST	FER-2013	STL-10	EuroSAT	RESISC45	GTSRB	KITTI	Country211	PCAM	UCF101	Kinetics700	CLEVR	HatefulMemes	SST2
ViT-B/32																											
OpenAI CLIP	56.6	63.4	83.7	89.8	65.1	53.7	62.0	59.7	19.6	44.0	87.2	87.4	66.9	48.2	46.6	97.1	44.9	61.0	32.6	28.7	17.2	62.5	63.9	48.0	23.6	56.4	58.6
OpenCLIP	57.6	62.9	80.7	90.7	70.6	61.2	66.4	79.2	16.7	54.5	86.5	90.7	66.1	37.4	48.2	95.6	52.2	58.0	42.0	38.0	14.8	50.1	63.0	42.8	22.5	53.3	52.3
MetaCLIP	58.2	65.5	80.6	91.3	70.2	63.4	63.0	70.7	26.8	52.8	88.7	91.9	68.5	41.5	35.9	95.4	52.6	64.2	35.8	30.7	17.2	55.5	66.1	45.4	30.6	56.4	53.4
MoDE-2	58.6	66.1	81.2	90.9	70.5	65.2	63.0	72.0	28.3	53.5	89.4	92.3	68.2	45.2	33.5	95.4	51.9	63.7	34.9	34.2	17.3	54.3	65.9	45.5	29.3	56.6	54.6
MoDE-4	59.0	66.4	82.3	91.3	70.9	67.0	63.7	73.8	30.1	52.6	89.9	92.1	69.2	37.9	33.2	95.7	53.5	64.1	35.2	33.9	17.1	58.4	66.6	45.9	30.0	58.0	54.5
ViT-B/16																											
OpenAI CLIP	59.6	68.3	88.8	90.8	68.2	55.6	64.0	64.6	24.0	45.1	88.9	89.1	69.4	51.8	53.0	98.2	54.8	65.5	43.3	21.7	22.8	56.3	68.5	52.3	25.5	58.7	60.5
OpenCLIP	60.4	67.1	85.8	91.7	71.4	65.3	69.2	83.6	17.4	51.0	89.2	90.8	66.5	66.3	46.1	97.0	52.2	65.7	43.5	23.7	18.1	51.7	67.0	46.2	33.9	54.5	54.4
MetaCLIP	61.1	70.8	86.8	90.1	66.5	70.8	66.6	74.1	27.9	55.9	90.4	93.8	72.3	47.8	44.6	97.2	55.4	68.8	43.8	33.4	22.6	52.9	68.0	49.5	22.8	54.8	60.6
MoDE-2	61.8	71.2	87.2	91.3	67.4	71.7	66.8	75.5	29.9	57.0	90.5	94.1	73.0	51.0	44.9	97.2	55.4	68.7	44.5	32.9	22.7	52.9	67.2	49.4	28.1	56.0	60.1
MoDE-4	62.1	71.6	87.8	91.4	68.9	74.7	67.2	77.3	32.6	56.2	91.3	93.9	74.9	43.7	46.6	97.2	54.4	70.0	44.0	29.8	22.9	55.7	68.6	50.0	29.7	55.2	58.0
ViT-L/14																											
OpenAI CLIP	65.7	75.5	93.0	95.6	78.3	63.3	66.8	77.8	31.3	55.3	93.6	93.3	79.3	76.4	56.9	99.4	61.9	70.9	50.6	19.2	31.9	50.1	75.7	60.2	22.3	59.7	68.9
OpenCLIP	64.5	72.7	90.0	94.7	78.0	73.9	72.4	89.5	24.7	60.2	91.6	93.6	73.0	76.1	54.3	98.1	63.9	69.6	49.9	16.0	23.0	51.7	71.5	51.6	25.4	55.3	56.0
MetaCLIP	67.1	76.2	90.7	95.5	77.4	75.9	70.5	84.7	40.4	62.0	93.7	94.4	76.4	61.7	46.5	99.3	59.7	71.9	47.5	29.9	30.9	70.1	75.5	57.1	35.1	56.6	65.6
MoDE-2	67.1	76.5	91.1	95.9	77.8	76.7	70.6	85.1	40.9	62.4	93.9	94.8	76.8	63.0	46.2	99.4	57.8	71.7	47.4	26.7	31.1	69.9	75.6	57.3	33.1	56.6	65.5
MoDE-4	67.2	76.3	91.2	95.7	77.9	78.3	70.7	85.6	41.8	62.4	94.0	94.5	77.1	62.6	46.6	99.2	57.7	72.0	47.3	26.8	31.3	71.5	76.0	57.3	30.6	56.6	65.5

Table 1. Performance on CLIP benchmark [31, 35] by models trained on 400M image-caption pairs. MoDE-2 and MoDE-4 consistently outperform the MetaCLIP Baseline and MoDE-4 achieves the best score on average.

	Average	ImageNet	Food-101	CIFAR10	CIFAR100	CUB	SUN397	Cars	Aircraft	DTD	Pets	Caltech-101	Flowers	MNIST	FER-2013	STL-10	EuroSAT	RESISC45	GTSRB	KITTI	Country211	PCAM	UCF101	Kinetics700	CLEVR	HatefulMemes	SST2
ViT-B/32																											
OpenCLIP	61.5	66.6	82.0	93.6	75.8	66.0	68.3	86.0	23.9	56.1	90.5	91.9	70.5	70.0	50.4	96.6	49.3	65.7	49.3	32.7	16.7	51.7	64.9	45.6	24.2	52.4	57.2
MetaCLIP	59.8	67.6	82.6	95.2	77.7	67.8	66.8	77.2	26.9	58.9	90.9	92.5	69.7	42.7	48.3	96.3	49.9	66.5	39.2	29.3	17.7	50.0	68.0	47.6	19.4	53.5	53.1
MoDE-2	61.2	68.7	84.1	95.3	78.6	69.5	67.0	80.8	30.9	60.6	91.0	92.9	71.9	40.8	50.4	96.3	51.3	67.9	44.2	31.4	18.3	51.3	69.0	47.4	23.2	52.6	54.4
MoDE-4	61.7	68.8	85.8	95.2	79.0	74.4	67.5	83.3	29.5	60.3	91.9	92.9	72.1	49.7	46.9	96.4	50.3	66.8	51.6	28.5	19.6	50.1	68.4	48.3	21.6	52.6	52.2
ViT-B/16																											
OpenCLIP	62.4	70.2	86.2	94.9	76.9	70.5	70.6	88.2	26.6	56.3	90.4	93.1	71.0	65.8	53.3	97.9	55.2	68.3	48.3	11.9	20.3	51.2	68.1	48.9	24.8	53.0	59.5
MetaCLIP	63.5	72.1	88.3	95.7	79.0	71.4	68.5	82.9	30.3	62.1	91.7	93.3	73.9	66.1	47.0	98.4	51.1	71.1	46.6	16.6	22.7	50.5	73.0	52.5	30.8	57.4	59.0
MoDE-2	65.0	73.6	89.5	96.0	81.4	76.5	69.0	85.7	35.9	63.5	93.4	93.4	75.5	59.2	46.4	98.3	50.0	72.0	50.1	34.9	23.9	50.8	71.2	52.1	31.2	59.1	58.4
MoDE-4	67.2	74.2	91.6	96.5	82.0	80.9	71.2	88.9	42.2	63.0	93.6	93.6	78.9	66.8	49.0	98.5	53.8	71.5	57.5	32.4	26.7	61.7	73.8	53.9	27.4	57.0	59.4
ViT-L/14																											
OpenCLIP	65.7	74.0	88.6	95.8	78.3	73.5	73.5	91.4	34.6	61.2	92.7	93.3	74.4	64.4	53.9	98.5	58.6	71.9	51.6	26.1	24.4	58.0	73.3	52.0	27.4	55.1	60.4
MetaCLIP	69.8	79.2	93.4	97.6	84.2	80.1	73.8	88.7	44.6	68.1	94.7	95.4	81.8	64.4	55.1	99.3	59.2	74.6	56.3	29.7	34.0	67.3	81.6	62.0	25.9	58.0	66.7
MoDE-2	70.4	79.5	93.5	97.6	85.0	82.9	74.0	90.9	49.0	69.5	95.0	95.3	81.8	69.7	53.7	99.2	63.3	75.2	59.0	29.8	33.9	62.3	81.7	62.4	24.0	56.6	65.4
MoDE-4	71.2	79.4	94.0	97.8	85.6	83.5	74.2	91.2	48.7	69.1	95.6	95.6	81.4	71.4	54.3	99.3	61.0	76.5	63.3	34.7	34.0	70.9	81.6	62.2	24.6	55.7	66.7

Table 2. Performance on CLIP benchmark [31, 35] by models trained on billion-scale dataset (OpenCLIP: 2.3B, MetaCLIP/MoDE: 2.5B). MoDE-2 and MoDE-4 consistently outperform the MetaCLIP Baseline and MoDE-4 achieves the best score on average.

size. To maintain a reasonable training cost, we start from a partially trained (27th out of 32 epoches) MetaCLIP as the seed model and all data experts share the same seed model.

4.3. Evaluation

Zero-Shot Image Classification. We follow the evaluation protocol in CLIP benchmark [31, 35, 46] and use the same class names & prompts by OpenAI CLIP. For fair comparison, MetaCLIP [46] naturally serves as the single dense baseline. The checkpoints of OpenAI CLIP (WIT400M data) [35] and OpenCLIP (LAION-400M data, LAION-2B data) [40] are also re-evaluated for fair comparison.

The framework MoDE has shown *consistent performance gain across model scales and data scales*. Firstly, we compare the models learned from the 400M-scale dataset in Table 1, and summarize the results by different model scales. MoDE achieves consistent performance gain where increasing the number of data experts results in better performance. Next, we study the scaling property of MoDE on 2.5B image-caption pairs in Table 2. Comparing against MetaCLIP [46], the advantage of MoDE to learn four data expert models is better revealed on scaling pretraining data: +1.9% on B/32, +3.7% on B/16, and +1.4% on L/14. Lastly, we increase the number of data experts. As shown in Fig. 3, the performance can be kept improving when we increase the number of data experts where MoDE16 ViT-B/32 can outperform the MetaCLIP Baseline ViT-B/16 baseline.

Notably, MoDE provides *an efficient and scalable approach to consume large-scale data without a large batch size that requires more GPUs* (384 Nvidia A100 GPUs) as in OpenCLIP. As shown in Table 2, based on ViT-B/16 with a batch size of 32K, the MoDE-2 with two data expert models is on par with the ViT-L/14 model by OpenCLIP [39], while 4 data expert models can outperform the ViT-L/14 by

Approach	ViT	Avg.	IN-Sk	IN-V2	IN-A	IN-O	IN-R	Avg.	IN-Sk	IN-V2	IN-A	IN-O	IN-R
OpenAI CLIP	B/32	49.4	42.3	56.0	31.5	47.8	69.4	-	-	-	-	-	-
OpenCLIP	B/32	50.6	49.4	55.1	21.7	53.5	73.4	52.9	53.7	58.1	26.3	50.0	76.4
MetaCLIP	B/32	52.2	53.3	57.6	28.6	46.8	74.8	54.4	56.0	59.6	29.9	48.3	78.1
MoDE-2	B/32	53.0	53.9	57.9	29.4	48.0	75.7	55.2	57.1	60.5	31.2	48.4	79.0
MoDE-4	B/32	53.4	54.4	58.5	30.8	47.6	76.0	56.5	57.6	61.6	34.2	49.2	80.0
OpenAI CLIP	B/16	56.0	48.3	61.9	50.0	42.3	77.7	-	-	-	-	-	-
OpenCLIP	B/16	54.8	52.4	59.7	33.2	50.7	77.9	56.7	56.1	62.3	38.2	46.3	80.6
MetaCLIP	B/16	57.7	57.9	62.6	47.0	39.2	81.8	60.1	60.2	65.0	49.5	41.6	84.2
MoDE-2	B/16	58.4	58.5	63.2	47.9	39.9	82.3	62.3	62.4	66.5	52.0	45.2	85.5
MoDE-4	B/16	59.0	58.8	63.7	49.2	40.4	82.9	63.3	62.8	67.1	55.7	44.5	86.6
Pre-Train Data		400M Image-Caption Pairs						OpenCLIP:2.3B; MetaCLIP/MoDE:2.5B					

Table 3. **Zero-Shot Robustness Evaluation**. The results are separated by the scale of pre-train set. Entries in blue are the best ones. Results by ViT-L/14 can be found in the Suppl.

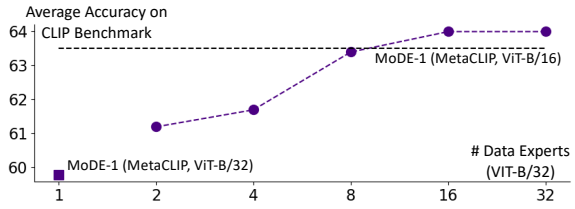

Figure 3. Average accuracy CLIP benchmark with increased number of data expert models in MoDE (Pretrain set: 2.5B pairs).

1.5% on CLIP benchmark dataset. Nevertheless, MoDE requires much less pretraining cost. As summarized in Fig. 4, MoDE-4 ViT-B/16 only requires less-than-35% of GPU-Hours used for OpenAI CLIP ViT-L/14. Compared with OpenCLIP trained on LAION-2B data, MoDE-8 ViT-B/32 data experts can even outperform a single ViT-B/16 model by OpenCLIP by but only use 31% of its GPU-Hours. In this way, our approach demonstrates great potential for efficient CLIP pretraining with limited GPUs in future.

Zero-Shot Robustness. In addition, to show a consistent gain on different tasks in the CLIP benchmark, we further validate the benefits towards robustness of MoDE in variants of ImageNet zero-shot classification. As summarized in Table 3, though there are systematic gaps across variants of ImageNet, learning a set of data experts can improve the zero-shot accuracy on all five variants over the MetaCLIP baseline for all model scales, and increasing the number of data experts can still introduce consistent gain. For the accuracies on IN-A and IN-O, the gap between baseline and other approaches is mitigated clearly by MoDE. Finally, MoDE-4 achieves the highest average accuracy of all dataset variants among all compared methods.

Zero-Shot Retrieval. We follow OpenCLIP [39] and reports the image/text retrieval results on COCO [26] and Flickr30k [48]. The compared models are trained on billion-scale datasets. As shown in Table 4, learning data experts can improve the scores consistently across all model sizes, on COCO, in particular, +3.3% and +2.7% in R@1 for image-to-text and text-to-image retrieval respectively by ViT-B/16 models, and we achieve the best performance.

Approach	ViT	Text Retrieval						Image Retrieval					
		COCO			Flickr30k			COCO			Flickr30k		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
OpenCLIP	B/32	56.3	79.8	87.1	84.1	96.2	98.3	39.3	65.4	75.6	66.7	88.4	93.1
MetaCLIP	B/32	55.2	78.9	86.5	80.7	95.2	97.3	38.1	64.1	74.3	65.1	87.7	92.7
MoDE-2	B/32	56.7	80.2	87.5	82.8	95.1	98.2	39.5	65.3	75.3	66.4	89.0	93.6
MoDE-4	B/32	57.4	80.1	87.3	82.9	95.6	97.7	39.9	66.1	75.7	66.7	88.4	93.3
OpenCLIP	B/16	59.5	81.8	88.6	86.2	98.0	99.5	42.3	67.7	77.1	69.8	90.4	94.6
MetaCLIP	B/16	59.4	80.6	87.8	85.5	97.4	98.9	41.4	67.2	76.9	70.7	90.8	94.5
MoDE-2	B/16	60.7	82.6	89.0	87.3	97.6	99.2	43.1	68.6	77.8	72.1	91.8	95.3
MoDE-4	B/16	62.7	82.9	89.8	89.4	98.0	99.6	44.1	69.5	78.7	72.6	91.8	95.4
Pretrain Data		OpenCLIP:2.3B; MetaCLIP/MoDE:2.5B											

Table 4. **Zero-shot Retrieval**. Entries in blue are the best ones. Results by model trained on 400M pairs and by ViT-L/14 can be found in the Suppl.

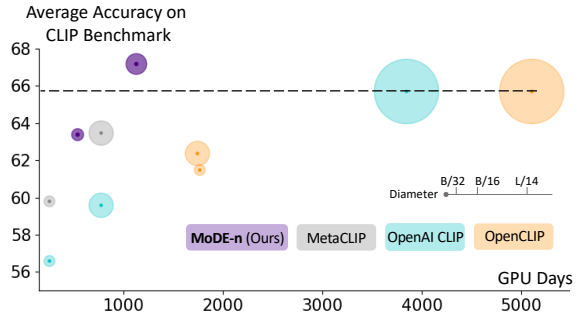

Figure 4. Summary of average accuracy on CLIP benchmark and pretraining cost (GPU-Hours). The diameter is proportional to the model size, different approaches are color-coded.

For the performance gap between MetaCLIP Baseline and OpenCLIP, *e.g.*, text retrieval on Flickr30k by ViT-B/32 models, the gap can also be mitigated clearly.

5. Discussion

We first analyze the importance of clustering (Sec. 5.1) and then study the MoDE design (Secs. 5.2 and 5.3). Finally, we investigate the potential of our approach in other important research directions (Sec. 5.4 and Sec. 5.5).

5.1. Effectiveness of Clustering

As MoDE ensembles the data experts learned from different clusters, we are first interested in the effects of clustering and consider two variants for ablation.

Though model ensembling [19] can provide gains over a single model, we are interested in how a naive ensembling of models trained on similar distribution performs compared to MoDE with data specialization. In Table 5, we train two ViT-B/32 CLIP models on the same training data without clustering, and then average the model outputs for prediction (Full-2). This achieves a similar performance as the baseline. Thus, the clustering is essential for MoDE.

Furthermore, we randomly split the training data into two subsets, and specialize a data expert for each subset (Random-2). For a fair comparison, we mimic the size of

	Average	ImageNet	Food-101	CIFAR10	CIFAR100	CUB	SUN397	Cars	Aircraft	DTD	Pets	Caltech-101	Flowers	MNIST	FER-2013	STL-10	EuroSAT	RESISC45	GTSRB	KITTI	Country211	PCAM	UCF101	Kinetics700	CLEVR	HatefulMemes	SST2
400M Image-Caption Pairs																											
MetaCLIP	58.2	65.5	80.6	91.3	70.2	63.4	63.0	70.7	26.8	52.8	88.7	91.9	68.5	41.5	35.9	95.4	52.6	64.2	35.8	30.7	17.2	55.5	66.1	45.4	30.6	56.4	53.4
Random-2	57.7	64.9	80.7	91.4	69.6	59.8	63.0	72.3	28.3	52.3	88.7	91.9	69.4	38.1	30.8	95.4	52.9	62.9	33.2	36.1	17.3	54.4	65.7	44.7	27.1	56.2	53.0
Full-2	58.3	65.9	81.0	91.2	69.9	63.8	63.3	71.0	27.3	52.3	88.9	91.8	69.2	42.9	33.3	95.4	52.5	64.6	35.8	31.2	17.0	56.1	67.0	45.5	28.7	57.5	53.5
MoDE-2	58.6	66.1	81.2	90.9	70.5	65.2	63.0	72.0	28.3	53.5	89.4	92.3	68.2	45.2	33.5	95.4	51.9	63.7	34.9	34.2	17.3	54.3	65.9	45.5	29.3	56.6	54.6
2.5B Image-Caption Pairs																											
MetaCLIP	59.8	67.6	82.6	95.2	77.7	67.8	66.8	77.2	26.9	58.9	90.9	92.5	69.7	42.7	48.3	96.3	49.9	66.5	39.2	29.3	17.7	50.0	68.0	47.6	19.4	53.5	53.1
Random-2	60.0	67.4	82.4	95.0	77.8	68.1	66.6	77.0	26.5	58.3	91.0	92.3	69.0	45.4	47.8	96.2	50.4	66.2	43.8	30.0	17.7	50.0	67.8	47.4	20.2	53.8	52.1
Full-2	60.0	67.8	82.6	95.2	77.7	68.4	66.7	77.7	27.7	58.6	90.9	92.5	69.9	43.6	48.7	96.4	50.1	66.0	41.7	28.2	17.9	50.0	68.4	47.7	19.3	53.9	52.8
MoDE-2	61.2	68.7	84.1	95.3	78.6	69.5	67.0	80.8	30.9	60.6	91.0	92.9	71.9	40.8	50.4	96.3	51.3	67.9	44.2	31.4	18.3	51.3	69.0	47.4	23.2	52.6	54.4

Table 5. Ablation Study for performance gain via Clustering by VIT-B/32.

Approach	CLIP Avg.	ImageNet	CLIP Avg.	ImageNet
MetaCLIP	58.2	65.6	59.8	67.7
OneStep-2	58.0	65.0	59.8	67.6
CoarseCluster-2	58.5	66.1	60.6	68.6
MoDE-2	58.6	66.1	61.2	68.7
CoarseCluster-4	58.7	66.2	61.3	68.5
MoDE-4	59.0	66.4	61.7	68.8
Pre-Train Dataset	400M Image-Caption Pairs		2.5B Image-Caption Pairs	

Table 6. Ablation study for Clustering Strategy by ViT-B/32.

subsets by MoDE-2 in the random splitting, and all data experts use the same seed model. As the data split is not obtained through clustering, we still only use the average of model outputs for evaluation. However, though Random-2 can provide small improvement when trained on 2.5B image-caption pairs (60.0 *vs*. 59.8), there is a noticeable drop when training on the 400M pairs (57.7 *vs*. 58.2).

5.2. Clustering Strategy

Instead of obtaining the data clusters in a single step, MoDE employs a two-step clustering strategy to discover the centers of fine-grained cluster S, which are used to properly model the correlation between task metadata and the conditions (Sec. 3.2). We provide ablation studies below to demonstrate this necessity for model ensembling.

Firstly, we evaluate the one-step clustering alternative, *i.e*., $m = n$, and for simplicity, we only learn two data experts (OneStep-2) based on ViT-B/32. As shown in Table 6, we summarize the average score on the CLIP benchmark and stand out the accuracy of ImageNet as it has the most number of classes. As the cluster centers are not representative enough to model the correlation with task metadata, model ensembling in OneStep-2 can even result in a slight drop. We do observe that each data expert alone can outperform MetaCLIP Baseline baseline on different tasks in the CLIP benchmark but it is difficult to pick correctly.

Then, we follow the two-step clustering but alter the number of fine-grained clusters m in the first step. As plotted in Fig. 5, we summarize the results of MoDE-2 trained on 400M image-caption pairs. With increasing m, we observed that the average accuracy on the CLIP evaluation benchmark improves consistently. Though the performance can be improved slightly when m is increased from 1024 to 2048, the computational cost during data clustering is also higher. We set $m = 1024$ in the main experiments.

Lastly, as another piece of evidence, we keep m as 1024 but use the coarse-grained cluster centers in Step 2, to determine the ensembling weights (CoarseCluster). As shown in Table 6 , as the meta clusters are not representative enough to obtain good ensembling weight, the resulting accuracy improvement is trivial. When we increase the number of data experts from 2 to 4, the gap between CoarseCluster-4 and MoDE-4 is even enlarged, which further demonstrates the importance of using fine-grained clusters to determine the ensembling weight for data experts in our MoDE.

5.3. Embeddings for Clustering

We further validate the importance of using language embeddings. In addition to SimCSE [11] language embedding, we investigate the following embeddings for clustering: (1) image embedding from the open-sourced DINOv2 [32]; (2) image and/or text embeddings from the seed model (*i.e*., the partially trained CLIP checkpoints on the 27th epoch). When the image embeddings are used for clustering, for each test image, we use its similarity with all fine-grained cluster centers to determine the logits ensemble weights. When both image and text embeddings are used, we use their concatenation as the feature for clustering. Without loss of generality, we compare with MoDE-2 trained on 400M pairs and set $m = 1024$ for fair comparison. We summarize the scores in Table 7 and report the zero-shot accuracy CLIP benchmark and ImageNet.

Firstly, by using image embeddings for clustering, the resulting models underperform MetaCLIP, in particular on ImageNet, and we believe the main reason is that the image embedding contains low-level details. As such, the cluster centers are not representative of model ensembling.

Furthermore, utilizing the language embeddings from the seed model yields only marginal performance improvement. This suggests that the CLIP embedding may still fall short of discerning high-level semantic correlations. This

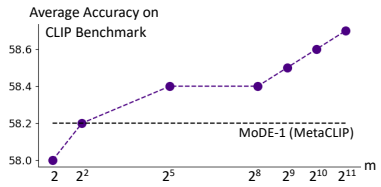

Figure 5. Ablation on # of clusters in Step 1.

Modality	Model	CLIP Eval.	ImageNet
Image	DINOv2	58.1	65.2
Image	CLIP Seed	58.3	64.7
Image & Lang.	CLIP Seed	58.4	65.5
Lang.	CLIP Seed	58.3	65.4
Lang.	SimCSE [11]	58.6	66.1

Table 7. Ablation Study on Embedding Types.

Model	B/32	B/16	L/14
MetaCLIP	67.5	73.8	82.3
MoDE-2	71.3	76.9	83.9
MoDE-4	74.1	79.6	84.7

Table 8. Performance comparison on ImageNet via linear probing.

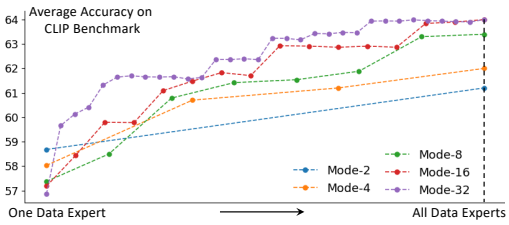

Figure 6. CLIP benchmark accuracy by MoDE-n when the data experts based on ViT-B/32 are developed in order and added to the system progressively. The pre-train set contains 2.5B pairs.

occurs as the language embeddings are influenced by image embeddings, potentially overlooking high-level semantics not depicted in corresponding images. For example, abstract concepts such as "travel", "product", and "politics" may lack corresponding visual elements. In contrast, the SimCSE text embeddings pretrained on large text corpora can understand abstract concepts for clustering,

5.4. Training Priority of Data Experts

As the data experts can be trained asynchronously, MoDE introduces flexibility in the data expert training priority. Below we demonstrate the robustness and effectiveness of MoDE when the data experts are trained in order.

Firstly, we rank the conditions to determine the training priority of data experts. This is useful when the computational resource is not sufficient to learn a giant dense model or all data experts together. We use the diversity of fine-grained clusters as a reference, and first train the model on the condition with the largest range, *i.e.*, the average distance between fine-grained clusters and the coarse-grained center. In Fig. 6, we vary the total number of ViT-B/32 data experts, *i.e.*, n, from 2 to 32 and summarize the average accuracy on the CLIP benchmark. When the data experts are gradually included, the performance keeps increasing.

In this way, instead of learning from all data simultaneously, MoDE enables progressive integration of new data experts, enabling dynamic updates. MoDE holds promise for applications such as *online* and continual learning. With each new set of data, it has the flexibility to update a pre-trained data expert, or to learn a new data expert. This is valuable when the incoming data are unprecedented to the existing system. We leave the trade-off between catastrophic forgetting [21] and adaption as the future work.

5.5. Application of Vision Encoders

The set of vision encoders can also be directly ensembled with equal weight in downstream application, which is free from any cluster center and can be generalizable to the case where the language metadata is not available.

Specifically, for each image, we concatenate the outputs from all (n) vision encoders as the representation and feed it into a linear layer for classification. To maintain reasonable training cost, only linear probing is considered where we exclusively train the linear classifier from scratch and fix all vision encoders. As shown in Table 8, we use ImageNet classification for evaluation and MoDE achieves consistent and clear performance gain over MetaCLIP. Besides, the parameters can also be averaged and used as initialization of a single network for finetuning (details studied in the Supp).

In summary, MoDE trains separate models to capture different fine-grained visual information, and can be applied to different types of downstream tasks, which could be a new pipeline to efficiently capture full visual semantics. Tentatively, the coarse-level clustering assumes the fine-grained clusters should be split into disjoint groups. We believe the fine-grained clusters can be grouped flexibly to improve ensemble strategy and leave it for future work.

6. Conclusion

The success of CLIP depends on the quality *negative* samples. As the *false negative* noise in web-crawled pairs hurts training effectiveness, scaling CLIP on large-scale data presents unique challenges in terms of training efficiency and computational bottlenecks. To this end, we have presented Mixture of Data Experts (MoDE) to asynchronously train a group of *data experts*. Each expert model is trained on a set of fine-grained clusters where the data in each cluster is of coherent semantics and all data experts are trained individually. During inference, the outputs are selectively ensembled based on the requirements for each task and modeled by the correlation between task metadata and fine-grained cluster centers. Empirically, MoDE significantly outperforms OpenCLIP and OpenAI CLIP on standard benchmarks with less than 30% training cost. We plan to adapt MoDE for generative models in the future.

Acknowledgements The authors would like to thank Xinlei Chen and Margaret Li for constructive discussion.

References

[1] Richard E Blahut. *Fast algorithms for signal processing*. Cambridge University Press, 2010. 3

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[4] Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, and Shanmukha Ramakrishna Vedantam. Hyperbolic image-text representations. In *International Conference on Machine Learning*, pages 7694–7731. PMLR, 2023. 2

[5] 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. 2

[6] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013. 2

[7] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. VSE++: improving visual-semantic embeddings with hard negatives. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 12. BMVA Press, 2018. 1

[8] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1):5232–5270, 2022. 2

[9] Alex Gammerman, Volodya Vovk, and Vladimir Vapnik. Learning by transduction. *arXiv preprint arXiv:1301.7375*, 2013. 2

[10] Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei Efros. Test-time training with masked autoencoders. *Advances in Neural Information Processing Systems*, 35:29374–29385, 2022. 2

[11] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 6894–6910. Association for Computational Linguistics (ACL), 2021. 4, 7, 8

[12] Suchin Gururangan, Margaret Li, Mike Lewis, Weijia Shi, Tim Althoff, Noah A Smith, and Luke Zettlemoyer. Scaling expert language models with unsupervised domain discovery. *arXiv preprint arXiv:2303.14177*, 2023. 2

[13] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. 1

[14] Guangxing Han, Jiawei Ma, Shiyuan Huang, Long Chen, and Shih-Fu Chang. Few-shot object detection with fully cross-transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5321–5330, 2022. 2

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[16] 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. 2

[17] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 2

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

[19] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015. 6

[20] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33:21798–21809, 2020. 2

[21] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016. 8

[22] 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. 2

[23] Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A Smith, and Luke Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language models. *arXiv preprint arXiv:2208.03306*, 2022. 2

[24] Xianhang Li, Zeyu Wang, and Cihang Xie. An inverse scaling law for clip training. *arXiv preprint arXiv:2305.07017*, 2023. 2

[25] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23390–23400, 2023. 2

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference,*

Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pages 740–755. Springer, 2014. 2, 6

[27] Haotian Liu, Kilho Son, Jianwei Yang, Ce Liu, Jianfeng Gao, Yong Jae Lee, and Chunyuan Li. Learning customized visual models with retrieval-augmented knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15148–15158, 2023. 2

[28] Jiawei Ma, Hanchen Xie, Guangxing Han, Shih-Fu Chang, Aram Galstyan, and Wael Abd-Almageed. Partner-assisted learning for few-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10573–10582, 2021. 2

[29] Mikko I Malinen and Pasi Fränti. Balanced k-means for clustering. In *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, S+ SSPR 2014, Joensuu, Finland, August 20-22, 2014. Proceedings*, pages 32–41. Springer, 2014. 4

[30] Tom M Mitchell. Machine learning, 1997. 3

[31] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *European Conference on Computer Vision*, pages 529–544. Springer, 2022. 2, 5

[32] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 7

[33] Mandela Patrick, Po-Yao Huang, Yuki Markus Asano, Florian Metze, Alexander G. Hauptmann, João F. Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 1

[34] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, et al. Combined scaling for zero-shot transfer learning. *Neurocomputing*, 555:126658, 2023. 2

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 4, 5

[36] 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. 2

[37] Vin Sachidananda, Ziyi Yang, and Chenguang Zhu. Global selection of contrastive batches via optimization on sample permutations. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. 2

[38] Stephan R Sain. The nature of statistical learning theory, 1996. 2

[39] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 2, 5, 6

[40] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 1, 5

[41] 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. In *International Conference on Learning Representations*, 2016. 2

[42] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 2

[43] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR, 2020. 2

[44] Chen-Wei Xie, Siyang Sun, Xiong Xiong, Yun Zheng, Deli Zhao, and Jingren Zhou. Ra-clip: Retrieval augmented contrastive language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19265–19274, 2023. 2

[45] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP, 7-11 November, 2021*, pages 6787–6800. Association for Computational Linguistics, 2021. 1

[46] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. *arXiv preprint arXiv:2309.16671*, 2023. 1, 4, 5

[47] Yuncong Yang, Jiawei Ma, Shiyuan Huang, Long Chen, Xudong Lin, Guangxing Han, and Shih-Fu Chang. TempCLR: Temporal alignment representation with contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[48] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 6

[49] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 2