MUGS: A MULTI-GRANULAR SELF-SUPERVISED LEARNING FRAMEWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

In self-supervised learning, multi-granular features are heavily desired though rarely investigated, as different downstream tasks (*e.g.*, general and fine-grained classification) often require different or multi-granular features, *e.g.* fine- or coarse-grained one or their mixture. In this work, for the first time, we propose an effective MUlti-Granular Self-supervised learning (Mugs) framework to explicitly learn multi-granular visual features. Mugs has three complementary granular supervisions: 1)an instance discrimination supervision (IDS), 2)a novel local-group discrimination supervision (LGDS), and 3) a group discrimination supervision (GDS). IDS distinguishes different instances to learn instance-level fine-grained features. LGDS aggregates features of an image and its neighbors into a local-group feature, and pulls local-group features from different crops of the same image together and push them away from others. It provides complementary instance supervision to IDS via an extra alignment on local neighbors, and scatters different local-groups separately to increase discriminability. Accordingly, it helps learn high-level fine-grained features at a local-group level. Finally, to prevent similar local-groups from being scattered randomly or far away, GDS brings similar samples close and thus pulls similar local-groups together, capturing coarse-grained features at a (semantic) group level. Consequently, Mugs captures three granular features that often enjoy higher generality on diverse downstream tasks over single-granular features, *e.g.* instance-level fine-grained features in contrastive learning. By only pretraining on ImageNet-1K, Mugs sets new SoTA linear probing accuracy 82.1% on ImageNet-1K and improves previous SoTA by 1.1%. It also surpasses SoTAs on other tasks, *e.g.* detection and segmentation.

1 INTRODUCTION

The family of self-supervised learning (SSL) approaches (He et al., 2020; Chen et al., 2020c) aims to learn highly transferable unsupervised representation for various downstream tasks by training deep models on a large-scale unlabeled dataset. To this end, a pretext task, *e.g.* jigsaw puzzle (Noroozi & Favaro, 2016) or orientation (Komodakis & Gidaris, 2018), is elaborately designed to generate pseudo labels of unlabeled visual data which are then utilized to train a model without using manual annotations. Since unlabeled visual data are of huger amount and also much cheaper than the manually annotated data, SSL has been very popularly adopted for visual representation learning recently (Caron et al., 2020a; Grill et al., 2020a), and is showing greater potential than supervised learning approaches for learning highly-qualified and well-transferable representation.

Motivation. In practice, various downstream tasks in SSL field often require different granular features, such as coarse- or fine-grained features. For instance, general classification downstream tasks distinguish a category from other categories and typically desire coarse-grained features, while fine-grained classification often discriminates subordinate categories and needs more fine-grained features. Actually, many downstream tasks highly desire multi-granular features. Take the classification task on ImageNet-1K (Deng et al., 2009) as an example. One needs coarse-grained features to distinguish a big category, *e.g.* dog, from other categories, *e.g.* bird and car, and also requires fine-grained features to discriminate different subordinate categories, such as Labrador and poodle in the dog category. However, this important multi-granularity requirement is ignored in the current state-of-the-art SSL approaches, including the representative contrastive learning family (He et al., 2020; Hjelm et al., 2018) and clustering learning family (Caron et al., 2018a; 2021). For contrastive learning, its instance discrimination task only aims to distinguish individual instances for learning

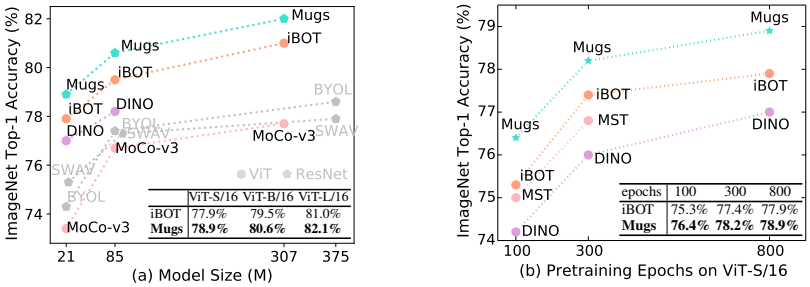


Figure 2: Comparison of **linear probing** accuracy on ImageNet-1K.

more instance-level fine-grained features, and does not consider the coarse-grained cluster structure in the data. As a result, it cannot well push semantically similar instances to be close either empirically (Zhou et al., 2021b) or theoretically (Wang & Isola, 2020), impairing performance. Clustering learning cluster similar instances into the same cluster and thus learns cluster-level coarse-grained features. But it cannot well handle the downstream tasks that require some fine-grained features. So in absence of prior feature preference of down stream tasks, one should build an SSL framework to learn multi-granular representation to well handle as many downstream tasks as possible.

Contributions. In this work, we propose an effective MUlti-Granular Self-supervised learning (Mugs) framework to explicitly learn multi-granular visual features. It adopts three complementary granular supervisions: 1) instance discrimination supervision (IDS), 2) local-group discrimination supervision (LGDS), and 3) group discrimination supervision (GDS). Inspired by contrastive learning, IDS distinguishes instances via scattering different instance features separately, and thus supervises instance-level fine-grained feature learning. To capture the higher-level fine-grained feature which is also called the "local-group feature" in this work, Mugs proposes a novel and effective LGDS. LGDS aggregates the features of an instance and its few highly similar neighbors into a local-group feature through a small transformer. Then it brings local-group features of different crops from the same image together and pushes them far away for others. This supervision enhances Mugs from two aspects: 1) it provides complementary instance supervision to the above IDS, since it enforces different crops of the same image to have highly similar neighbors, which is an extra challenging alignment, and boosts local-group semantic alignment; 2) it encourages highly similar instances to constitute small local-groups and scatters these groups separately, boosting more discriminative semantic learning. Finally, GDS is designed to avoid the cases that similar local-groups are scattered randomly or far away. GDS brings similar samples together and thus pulls similar local-groups close, capturing coarse-grained features at a (semantic) group level. With these complementary supervisions, Mugs can well learn multi-granular features which can well capture the data semantics, *e.g.* the shapes of "mugs" of ImageNet-1K in Fig. 1, and also often enjoy better generality and transferability on diverse downstream tasks than single-granular features.

Figure 1: Attention visualization on "mugs" of ViT-B/16 trained by our Mugs.

As shown in Fig. 2 (a), by only pretraining on ImageNet-1K, our Mugs sets a new state-of-the-art (SoTA) 82.1% linear probing accuracy on ImageNet-1K and surpasses the previous SoTA, i.e. iBOT (Zhou et al., 2021a), by a large margin 1.1%. Moreover, under different model sizes (see Fig. 2 (a)) and pretraining epochs (see Fig. 2 (b)), Mugs consistently improves previous SoTA pretrained on ImageNet 1K by a non-trivial 0.8% linear probing accuracy. Besides, on several downstream tasks, *e.g.* detection and segmentation, Mugs also beats previous SoTA with the same setting.

2 Related Works

As an effective family of SSL, contrastive learning, *e.g.*, MoCo (He et al., 2020), aims to train a network so that the positive pair, i.e. crops of the same image, are close but far from the negatives, i.e. other image crops. Though successful, they only distinguish individual instances to learn fine-grained feature, and often cannot well push similar instances close, impairing their performance.

Another line of SSL is clustering learning, *e.g.* (Caron et al., 2018a; Lin et al., 2021), which assigns pseudo cluster labels for each sample and then trains a network to learn unsupervised representation.

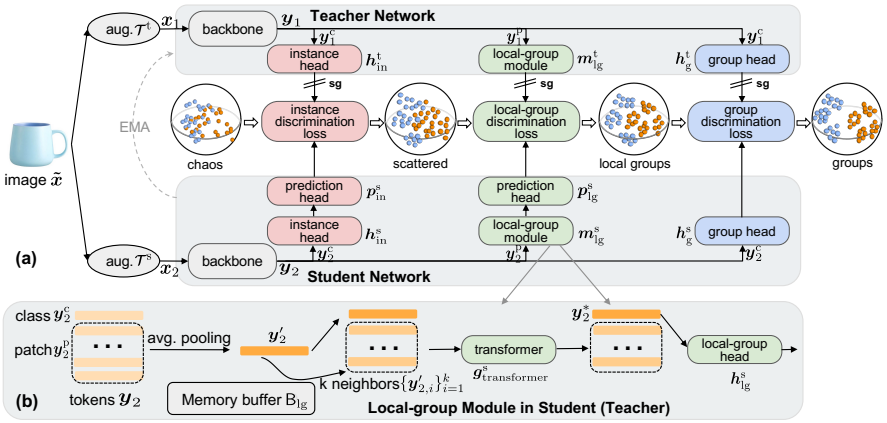


Figure 3: **Overall framework of Mugs**. **(a)** shows the overall framework. For each image, Mugs respectively feeds its two crops into backbones of student and teacher. Next, it uses three granular supervisions from instance, local-group and group levels. "sg" denotes stop-gradient. **(b)** shows the pipeline of local-group modules in both student and teacher. It averages all patch tokens, and then finds top-k neighbors from memory buffer. Next, it uses a transformer to aggregate the average and its k neighbors to obtain a local-group feature (class token) and feeds it into a local-group head.

For instance, deepcluster (Caron et al., 2018b) uses k-means to cluster all data features and generates pseudo clustering labels which are then used to train a network. DINO (Caron et al., 2021) and MST (Li et al., 2021) propose a much simple online labeling framework that generates pseudo-labels via a momentum teacher. Unfortunately, they often learns cluster-level coarse-grained (semantic) features, and cannot well handle the downstream tasks which desire fine-grained features.

Finally, the recently proposed masked auto-encoder (MAE), *e.g.* (He et al., 2021b; Xie et al., 2021b) is a new SSL family. It randomly masks image patches and then reconstructs the missing pixels or semantic features. But it emphasizes local region reconstruction, and lacks semantic discrimination ability. As a result, for adapting to downstream tasks via only fine-tuning a task head at the top of the pretrained backbone, it performs much worse than contrastive and clustering learning (He et al., 2021a). Indeed, to achieve good performance, these methods need to fine-tune the whole pretrained network to learn global semantics which are necessary for downstream tasks, *e.g.* classification. But this requires much higher extra training cost, and also results in very different models for different downstream tasks, which destroys model compatibility and increases deployment cost.

3 Multi-granular self-supervised learning

Here we first introduce the overall framework of our MUlti-Granular Self-supervised learning (Mugs), and then elaborate on its three granular supervisions. We evaluate the effectiveness of Mugs through ViT (Dosovitskiy et al., 2020) and thus will take ViT as an example to introduce Mugs, since ViT shows better performance than CNN (Touvron et al., 2021; He et al., 2021a) and great potential for unifying the vision and language models (He et al., 2021a; Baevski et al., 2022).

3.1 Overall Framework

As discussed in Sec. 1, different downstream tasks, *e.g.* general classification and its fine-grained variant, often require different granular features, *e.g.* coarse- or fine-grained one. More importantly, many real downstream tasks actually highly desire multi-granular feature. Unfortunately, this important multi-granular requirement is seldom brought to front and ignored in existing SSL methods.

To alleviate this issue, we propose a simple but effective Mugs framework to learn multi-granular features which can better satisfy different granular feature requirements of various downstream tasks and thus enjoy higher transferability and generality than single-granular features. As shown in Fig. 3 (a), given an image $\widetilde{\boldsymbol{x}}$, Mugs uses augmentations $\mathcal{T}^t$ and $\mathcal{T}^s$ to obtain its two crops $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Next, it respectively feeds $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ into the teacher and student backbones, and obtains their corresponding features $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ which contain class and patch tokens. Finally, Mugs builds three granular supervisions: 1) instance discrimination supervision for instance-level fine-grained features, 2) local-group discrimination supervision for high-level fine-grained features at a local-group level, 3)

group discrimination supervision for coarse-grained semantic features at a (semantic) group level. Accordingly, Mugs can learn multi-granular features and better handles as many downstream tasks as possible, in contrast with SSL methods that only consider single-granular features, *e.g.* MoCo for instance discriminative fine-grained features and deepclustering/DINO for group-discriminative coarse-grained features. Next, we will introduce our three complementary granular supervisions.

3.2 MULTI-GRANULAR SUPERVISIONS

Instance discrimination supervision (IDS). With this supervision, Mugs regards each instance as a unique class which is our finest level of granularity. Accordingly, it pulls the random crops of the same instance close and pushes other crops away. In this way, as shown in first and second spheres in Fig. 3 (a), it approximately scatters the instance features separately from the chaos distribution on the spherical surface, according with the empirical and theoretical observations in (Tian et al., 2020) and our results in Fig. 4 of Sec. 4. To implement IDS on the instance $\widetilde{\boldsymbol{x}}$, Mugs respectively feeds two class tokens $\boldsymbol{y}_1^{\text{c}}$ and $\boldsymbol{y}_2^{\text{c}}$ in the two features $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ into their corresponding instance heads h_{in}^{t} and h_{in}^{s} in Fig. 3 (a). Next, Mugs additionally passes $h_{\text{in}}^{\text{s}}(\boldsymbol{y}_2^{\text{c}})$ into an extra prediction head p_{in} which alleviates the side effects of feature alignment upon the generality of the feature learnt by student or teacher backbone. Finally, following MoCo, Mugs employs InfoNCE loss (Oord et al., 2018)

$$\mathcal{L}_{\text{instance}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = -\log \frac{\exp(\cos(\boldsymbol{z}_1, \boldsymbol{z}_2)/\tau_{\text{in}})}{\exp(\cos(\boldsymbol{z}_1, \boldsymbol{z}_2)/\tau_{\text{in}}) + \sum_{\boldsymbol{z}\in \mathsf{B}_{\text{in}}} \exp(\cos(\boldsymbol{z}_2, \boldsymbol{z})/\tau_{\text{in}})}, \tag{1}$$

where $\boldsymbol{z}_1 = h_{\text{in}}^{\text{t}}(\boldsymbol{y}_1^{\text{c}})$, $\boldsymbol{z}_2 = p_{\text{in}}(h_{\text{in}}^{\text{s}}(\boldsymbol{y}_2^{\text{c}}))$, and τ_{in} is a temperature. Buffer B_{in} stores the negative instances of $\boldsymbol{z}_2$, and is updated by the minibatch features $\{\boldsymbol{z}_1\}$ of teacher in a first-in and first-out (FIFO) order. Accordingly, Mugs pushes the crop $\boldsymbol{x}_2$ away from other instances and pulls its positive $\boldsymbol{x}_1$ close. So it helps learn fine-grained features and boosts instance-level feature diversity.

Local-group discrimination supervision (LGDS). As explained in Sec. 3.1, fine-grained features are often insufficient for diverse downstream tasks, *e.g.* classification, due to lack of sufficient high-level data semantics. To learn higher-level fine-grained features, also called "local-group features" here, Mugs proposes a novel and effective local-group supervision. Intuitively, as shown in the third sphere of Fig. 3 (a), LGDS encourages instance features to have small but separately scattered local-group structures, i.e. small/large distance among highly similar/dissimilar samples. Accordingly, it helps capture higher-level data semantics compared with instance discrimination supervision.

As shown in Fig. 3 (a), given the crop $\boldsymbol{x}_1$ of image $\widetilde{\boldsymbol{x}}$, the teacher backbone outputs $\boldsymbol{y}_1$ which contains class token $\boldsymbol{y}_1^{\text{c}}$ and patch tokens $\boldsymbol{y}_1^{\text{p}}$. Similarly, Mugs feeds another crop $\boldsymbol{x}_2$ of $\widetilde{\boldsymbol{x}}$ into student backbone to obtain $\boldsymbol{y}_2$ consisting of class token $\boldsymbol{y}_2^{\text{c}}$ and patch tokens $\boldsymbol{y}_2^{\text{p}}$. Next, Mugs respectively averages the patch tokens $\boldsymbol{y}_1^{\text{p}}$ and $\boldsymbol{y}_2^{\text{p}}$ to obtain two average tokens $\boldsymbol{y}_1'$ and $\boldsymbol{y}_2'$ shown in Fig. 3 (b). Here we use average tokens instead of class token, as we find that average token often contains more information and finds more accurate neighbors in the next step. Note, for a CNN backbone, we can average the last feature map to obtain $\boldsymbol{y}_1'$ and $\boldsymbol{y}_2'$. Then Mugs uses a buffer B_{lg} to store the historical minibatch average tokens $\{\boldsymbol{y}_1'\}$ and $\{\boldsymbol{y}_2'\}$ in a FIFO order. Next, as shown by Fig. 3 (b), for both $\boldsymbol{y}_1'$ and $\boldsymbol{y}_2'$, Mugs respectively finds their own top-k neighbors $\{\boldsymbol{y}_{1,i}'\}_{i=1}^k$ and $\{\boldsymbol{y}_{2,i}'\}_{i=1}^k$ from the buffer B_{lg}. Finally, it respectively uses a transformer to aggregate the average token and its k neighbors as

$$\boldsymbol{y}_1^* = g_{\text{transformer}}^{\text{t}}(\boldsymbol{y}_1'; \{\boldsymbol{y}_{1,i}'\}_{i=1}^k) \qquad \text{and} \qquad \boldsymbol{y}_2^* = g_{\text{transformer}}^{\text{s}}(\boldsymbol{y}_2'; \{\boldsymbol{y}_{2,i}'\}_{i=1}^k). \tag{2}$$

Here $g_{\text{transformer}}^{\text{t}}(\boldsymbol{y}_1'; \{\boldsymbol{y}_{1,i}'\}_{i=1}^k)$ denotes a 2-layered vanilla ViT without any patch embedding layers, and has input class token $\boldsymbol{y}_1'$, input patch tokens $\{\boldsymbol{y}_{1,i}'\}_{i=1}^k$ and output class token $\boldsymbol{y}_1^*$. Since the new feature $\boldsymbol{y}_1^*$ comes from $\boldsymbol{y}_1'$ and its top-k neighbors $\{\boldsymbol{y}_{1,i}'\}_{i=1}^k$ which together constitute a local group of $\boldsymbol{y}_1'$, $\boldsymbol{y}_1^*$ is also called a "local group feature". $g_{\text{transformer}}^{\text{s}}(\boldsymbol{y}_2'; \{\boldsymbol{y}_{2,i}'\}_{i=1}^k)$ has the same function. Finally, Mugs pulls these local-group features $\boldsymbol{y}_1^*$ and $\boldsymbol{y}_2^*$ from the same instance $\widetilde{\boldsymbol{x}}$ close and pushes away the local-group features of other instances by using following InfoNCE loss

$$\mathcal{L}_{\text{local-group}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = -\log \frac{\exp(\cos(\boldsymbol{z}_1, \boldsymbol{z}_2)/\tau_{\text{lg}})}{\exp(\cos(\boldsymbol{z}_1, \boldsymbol{z}_2)/\tau_{\text{lg}}) + \sum_{\boldsymbol{z}\in \mathsf{B}_{\text{lg}}'} \exp(\cos(\boldsymbol{z}_2, \boldsymbol{z})/\tau_{\text{lg}})}, \tag{3}$$

where $\boldsymbol{z}_1 = h_{\text{lg}}^{\text{t}}(\boldsymbol{y}_1^*)$ and $\boldsymbol{z}_2 = p_{\text{lg}}(h_{\text{lg}}^{\text{s}}(\boldsymbol{y}_2^*))$. h_{lg}^{t} and h_{lg}^{s} are two projection heads and p_{lg} is a prediction head. Buffer B_{lg}' stores the historical local-group features $\{\boldsymbol{y}_1^*\}$ produced by teacher in a FIFO order.

This LGDS supervision benefits Mugs from two aspects. **1)** It provides complementary instance supervision to the above instance discrimination supervision (IDS). It brings two local-group features

$\boldsymbol{y}_1^*$ and $\boldsymbol{y}_2^*$ from the same image $\widetilde{\boldsymbol{x}}$ close, where $\boldsymbol{y}_1^*/\boldsymbol{y}_2^*$ are the aggregation of the crop $\boldsymbol{x}_1/\boldsymbol{x}_2$ and its top-k neighbors. So to achieve small loss $\mathcal{L}_{\text{local-group}}(\boldsymbol{x}_1, \boldsymbol{x}_2)$, the two crops $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ of $\widetilde{\boldsymbol{x}}$ should have very similar top-k neighbors. This means besides the crops themselves, their corresponding neighbors should also be well aligned, which is an extra challenging alignment problem compared with IDS and enhances local-group semantic alignment. **2)** It encourages highly-similar instances to form local-groups and scatters these local-groups separately, increasing the semantic discrimination ability of the learnt feature. This is because a) LGDS uses a small k (around 10) for neighbors such that samples in the same local-group are highly similar and have small distance, helping form local-groups; 2) LGDS further pushes away local-group features of different instances, and thus scatters different local-groups separately. With these two aspects, LGDS boosts higher-level fine-grained feature learning by considering the local-group structures in data.

Group discrimination supervision (GDS). This supervision is the most coarse level supervision in Mugs. Intuitively, as shown in the last sphere in Fig. 3 (a), it targets at clustering semantically similar instances and local-groups into the same big group/cluster which could reveal more *global* semantics in data compared with the instance and local-group supervisions.

For the instance $\widetilde{\boldsymbol{x}}$, Mugs respectively feeds the class token $\boldsymbol{y}_1^{\text{c}}$ in the feature $\boldsymbol{y}_1$ from teacher backbone and the class token $\boldsymbol{y}_2^{\text{c}}$ in $\boldsymbol{y}_2$ from student backbone into two group heads h_{g}^{t} and h_{g}^{s}. Then, it builds a set of learnable cluster prototypes $\{\boldsymbol{c}_i\}_{i=1}^m$ and computes soft pseudo clustering labels:

$$\boldsymbol{p}_i^{\text{t}} = \frac{\exp(\sigma(h_{\text{g}}^{\text{t}}(\boldsymbol{y}_1^{\text{c}})) \cdot \boldsymbol{c}_i/\tau_{\text{g}})}{\sum_{i=1}^m \exp(\sigma(h_{\text{g}}^{\text{t}}(\boldsymbol{y}_1^{\text{c}})) \cdot \boldsymbol{c}_i/\tau_{\text{g}})} \text{ and } \boldsymbol{p}_i^{\text{s}} = \frac{\exp(h_{\text{g}}^{\text{s}}(\boldsymbol{y}_2^{\text{c}}) \cdot \boldsymbol{c}_i/\tau'_{\text{g}})}{\sum_{i=1}^m \exp(h_{\text{g}}^{\text{s}}(\boldsymbol{y}_2^{\text{c}}) \cdot \boldsymbol{c}_i/\tau'_{\text{g}})}. \tag{4}$$

Here the function $\sigma(h_{\text{g}}^{\text{t}}(\boldsymbol{y}_1^{\text{c}})) = h_{\text{g}}^{\text{t}}(\boldsymbol{y}_1^{\text{c}}) - \boldsymbol{p}_{\text{ema}}$ is to increase the diversity of the feature $h_{\text{g}}^{\text{t}}(\boldsymbol{y}_1^{\text{c}})$ and thus sharpens the soft pseudo label $\boldsymbol{p}^{\text{t}}$ in Eqn. (4), where $\boldsymbol{p}_{\text{ema}}$ denotes the estimated average statistics of all past $h_{\text{g}}^{\text{t}}(\boldsymbol{y}_1^{\text{c}})$ via an exponential moving average of the mini-batch $\boldsymbol{\mathcal{B}}$, namely, $\boldsymbol{p}_{\text{ema}} \leftarrow \rho \cdot \boldsymbol{p}_{\text{ema}} + (1-\rho) \cdot \frac{1}{|\boldsymbol{\mathcal{B}}|} \sum_{\boldsymbol{p}^{\text{t}} \in \boldsymbol{\mathcal{B}}} \boldsymbol{p}^{\text{t}}$ with a constant $\rho \in [0, 1]$. Such a technique is shown to be useful in (Caron et al., 2021). Next, similar to a supervised classification task, Mugs employs the cross-entropy loss but with soft labels as its training loss:

$$\mathcal{L}_{\text{group}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = -\sum\nolimits_{i=1}^m \boldsymbol{p}_i^{\text{t}} \log(\boldsymbol{p}_i^{\text{s}}). \tag{5}$$

Overall training objective. Now we introduce the overall training loss:

$$\mathcal{L}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \lambda_{\text{in}} \mathcal{L}_{\text{instance}}(\boldsymbol{x}_1, \boldsymbol{x}_2) + \lambda_{\text{lg}} \mathcal{L}_{\text{local-group}}(\boldsymbol{x}_1, \boldsymbol{x}_2) + \lambda_{\text{g}} \mathcal{L}_{\text{group}}(\boldsymbol{x}_1, \boldsymbol{x}_2), \tag{6}$$

where λ_{in}, λ_{lg} and λ_{g} are three constants. For simplicity, we set $\lambda_{\text{in}} = \lambda_{\text{lg}} = \lambda_{\text{g}} = \frac{1}{3}$ in all experiments. We then can minimize the objective $\mathcal{L}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ to optimize student network. Teacher network is updated via the exponential moving average of corresponding parameters in student.

Now we put our three granular supervisions (IDS, LGDS and GDS) together and discuss their co-effects on representation learning which also distinguishes it from existing methods, *e.g.* MoCo and DINO. As aforementioned, IDS is to pull the crops of the same image together and to approximately scatter the instance features separately on the spherical surface as shown in first and second spheres in Fig. 3 (a). It helps Mugs learn instance-level fine-grained features. Next, LGDS first provides complementary supervision for instance discrimination supervision by encouraging crops of the same instance to have highly similar neighbors. Then, as shown in the third sphere in Fig. 3 (a), LGDS scatters different local-groups formed by crops and its neighbors separately to boost the semantic discrimination ability of these local-groups. This supervision mainly learns higher-level fine-grained features at a local-group level. Finally, to avoid similar local-groups to be scattered randomly or far away, GDS brings similar samples together and thus pulls similar local-groups close, as intuitively illustrated by the last sphere in Fig. 3 (a). It is responsible to capture the coarse-grained features at a (semantic) group level. With these three granular supervisions, Mugs can well learn three different but complementary granular features, which are characterized by better generality and transferability on the various kinds of downstream tasks compared with single-granular features. Compared with existing methods, *e.g.* MoCo and DINO, the main novelties of Mugs lie in two folds: 1) Mugs learns multi-granular representation via three complementary supervisions and can often better handle diverse downstream tasks than the existing methods that often learn single-granular feature; 2) Mugs proposes a novel and effective local-group supervision which complements both instance and group supervisions and benefits Mugs from two aspects as discussed above.

Discussion. A few works also design (locally) clustering-involved SSL approaches. Given a positive pair $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ of an image $\widetilde{\boldsymbol{x}}$, NNCLR (Dwibedi et al., 2021) pushes $\boldsymbol{x}_2$ and the nearest neighbors $\{\boldsymbol{x}'_1\}_{i=1}^k$ of $\boldsymbol{x}_1$ closer via InfoNCE loss by regarding $\{\boldsymbol{x}'_1\}_{i=1}^k$ as the positives of $\boldsymbol{x}_2$. SwAV (Caron et al., 2020a) learns clustering prototypes via optimal transport while enforcing consistency between the similarity of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ on the prototypes. CLD (Wang et al., 2021) clusters current minibatch samples into local-groups via kmeans, and pulls samples in the same group closer but away from samples in other groups. But Mugs differs from them in several key aspects as shown in Table 1.

Table 1: Comparison of clustering-involved SSL.

	global clustering	scattered clusters	hierarchical clusters	end-to-end training
NNCLR	✗	✗	✗	✓
SwAV	✓	✗	✗	✗
CLD	✗	✓	✗	✗
Mugs	✓	✓	✓	✓

architecture	ViT-S				ViT-B	
method	Mugs	SwAV	Mugs	CLD	Mugs	NNCLR
pre. epoch	800	800	100	100	800	1000
top-1 acc. (%)	78.9	73.5	76.4	71.6	80.6	76.5

Specifically, **1)** on global clustering whether using whole dataset, NNCLR and CLD do not, while Mugs and SwAV do. In NNCLR, its local group considers only two samples, and it has severe degeneration if using more neighbors (*e.g.* 4, see its Tab. 7), indicating inferior clustering effect. CLD clusters minibatch via kmeans which however is sensitive and often cannot handle small-sized minibatch data. In contrast, Mugs and SwAV use whole data to iteratively improve their aggregation transformer or prototypes. **2)** On scattered clusters which avoid collapse and also increase discriminability, NNCLR and SwAV do not scatter their clusters separately, while LGDS in Mugs scatters different local groups. Optimal transport in SwAV only ensures uniform distribution of samples in each cluster. **3)** On hierarchical cluster structure (a group having several local groups), all three methods have no due to their single-granular cluster loss, while Mugs has because of its multi-granular loss. This structure can better depict real data and helps many downstream tasks. **4)** On end-to-end training, SwAV and CLD do not due to their optimal transport or kmeans, while Mugs and NNCLR do. **5)** On technique, our LGDS aggregates an image and its neighbors into a local-group feature via a transformer, and pulls positive local-groups closer while pushing negatives far away. This aggregation clustering differs from NNCLR, SwAV via optimal transport and CLD via local kmeans. Indeed, **our aggregation transformer enjoys several advantages**: **a)**as aforementioned, clustering given by aggregation transformer is global, **b)** it gives an end-to-end trainable framework; **c)** pushing or pulling local-group features explicitly affects each samples in the group and is much more efficient than NNCLR, SwAV and CLD which performs pushing and pulling actions on a single instance. **6)** On performance, Table 1 (also Table 2) shows large linear probing accuracy improvement of Mugs on ImageNet-1K over SwAV, CLD and NNCLR whose results are officially reported.

4 Experiments

Here we present the performance evaluation of our Mugs on benchmark tasks, *e.g.* classification and delectation and segmentation, with comparison against several representative SoTA SSL approaches.

Architectures. We test Mugs on ViT (Dosovitskiy et al., 2020). For IDS and LGDS, their projection heads are all 3-layered MLPs with hidden/output dimension 2,048/256, and their prediction heads p_{in} and p_{lg} are all 2-layered MLPs with hidden/output dimension 4,096/256. For group discrimination, its projection heads are all 3-layered MLP with hidden/output dimension of 2,048/256. Transformers $g^{\text{t}}_{\text{transformer}}$ and $g^{\text{s}}_{\text{transformer}}$ have 2 layers and have a total input token number of 9 as we set $k = 8$ for the neighbors. For three buffers (B_{in}, B_{lg} and B'_{lg}) and prototypes $\{\boldsymbol{c}_i\}_{i=1}^m$, their sizes are all 65,536.

Pretraining setup. We pretrain Mugs on ImageNet-1K (Deng et al., 2009). Following DINO and iBOT, we use symmetric training loss, i.e. $\frac{1}{2}(\mathcal{L}(\boldsymbol{x}_1, \boldsymbol{x}_2) + \mathcal{L}(\boldsymbol{x}_2, \boldsymbol{x}_1))$. For augmentation, we adopt weak augmentation in DINO to implement $\mathcal{T}^{\text{t}}$ in teacher, and use strong augmentation (mainly including AutoAugment (Cubuk et al., 2018)) in DeiT (Touvron et al., 2021) as the augmentation $\mathcal{T}^{\text{s}}$ in student. Following the multi-crop setting in SwAV and DINO, we crop each image into 2 large crops of size 224 and 10 extra small crops of size 96. For both large crops, we feed each of them into teacher, and use its output to supervise the student's output from the other 11 crops. For two-crop setting, Table 9 in Appendix A reports the results and shows superiority of Mugs over SoTAs.

For pretraining, Mugs has almost the same training cost with DINO, *e.g.* about 27 hours with 8 A100 GPUs for 100 pretraining epochs on ViT-S/16, as our projection/prediction heads and transformers $g_{\text{transformer}}$ are much smaller than the backbone. See more details of hyper-parameters (*e.g.* τ_{in}), the augmentation, multi-crop loss, pretraining cost, and optimizer settings in Appendix B.

Table 2: **Linear probing and k-NN** accuracy (%) on ImageNet-1K. "D" denotes which dataset is used to pretrain. "Epo." is the effective pretraining epochs in (Zhou et al., 2021a).

	Method	D	Epo.	Lin.	k-NN
ResNet-50	MoCo-v3 (Chen et al., 2021)	1K	1600	74.6	—
	SimCLR (Chen et al., 2020a)	1K	1600	69.3	—
	InfoMin Aug (Tian et al., 2020)	1K	1600	73.0	—
	SimSiam (Chen & He, 2021)	1K	1600	71.3	—
	BYOL (Grill et al., 2020b)	1K	2000	74.3	—
	SwAV (Caron et al., 2020b)	1K	2400	75.3	65.7
	DeepCluster (Caron et al., 2018b)	1K	2400	75.2	—
	DINO (Caron et al., 2021)	1K	3200	75.3	67.5
ViT-S	MoCo-v3 (Chen et al., 2021)	1K	3200	73.4	—
	SwAV (Caron et al., 2020b)	1K	3200	73.5	66.3
	DINO (Caron et al., 2021)	1K	3200	77.0	74.5
	iBOT (Zhou et al., 2021a)	1K	3200	77.9	75.2
	Mugs (ours)	1K	3200	**78.9**	**75.6**
ViT-B	MoCo-v3 (Chen et al., 2021)	1K	1200	76.7	—
	DINO (Caron et al., 2021)	1K	1600	78.2	76.1
	iBOT (Zhou et al., 2021a)	1K	1600	79.5	77.1
	Mugs (ours)	1K	1600	**80.6**	**78.0**
ViT-L	MoCo-v3 (Chen et al., 2021)	1K	1200	77.6	—
	iBOT (Zhou et al., 2021a)	1K	1000	81.0	78.0
	Mugs (ours)	1K	1000	**82.1**	**80.3**
	iBOT (Zhou et al., 2021a)	22K	200	82.3	72.9

Table 3: **Fine-tuning** accuracy (%) on ImageNet-1K. All are pretrained on ImageNet-1K. "Recons.", "con.", "clus." are respectively short for "Reconstruction", "contrastive", "clustering".

	Method	ViT-S/16 Epo.	ViT-S/16 Acc. (%)	ViT-B/16 Epo.	ViT-B/16 Acc. (%)
	Supervised (Touvron et al., 2021)	—	79.9	—	81.8
Recons.	BEiT (Bao et al., 2021)	800	81.4	800	83.4
	MAE (He et al., 2021a)	—	—	1600	83.6
	SimMIM (Xie et al., 2021b)	—	—	1600	83.8
	MaskFeat (Wei et al., 2021)	—	—	1600	84.0
	data2vec (Baevski et al., 2022)	—	—	1600	84.2
con./clus.	MoCo-v3 (Chen et al., 2021)	600	81.4	600	83.2
	DINO (Caron et al., 2021)	3200	82.0	1600	83.6
	iBOT (Zhou et al., 2021a)	3200	82.3	1600	83.8
	Mugs (ours)	3200	**82.6**	1600	**84.3**

Table 4: **Semi-supervised** classification accuracy (%) on ImageNet-1K.

Method	Arch.	logistic reg. 1%	logistic reg. 10%	fine-tuning 1%	fine-tuning 10%
SimCLRv2 (Chen et al., 2020b)	RN50	—	—	57.9	68.1
BYOL (Grill et al., 2020b)	RN50	—	—	53.2	68.8
SwAV (Caron et al., 2020b)	RN50	—	—	53.9	70.2
DINO (Caron et al., 2021)	ViT-S/16	64.5	72.2	60.3	74.3
iBOT (Zhou et al., 2021a)	ViT-S/16	65.9	73.4	61.9	75.1
Mugs (ours)	ViT-S/16	**66.9**	**74.0**	**66.8**	**76.8**

4.1 Results on ImageNet-1K

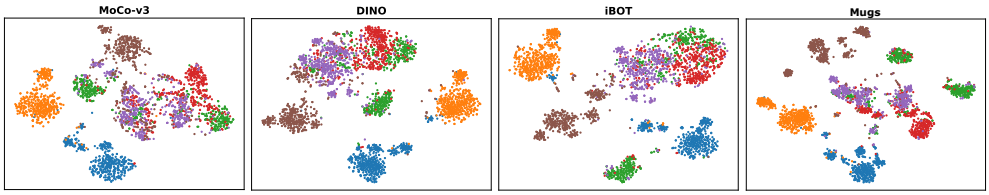
Linear Probing. It trains a linear classifier on top of frozen features generated by the backbone, *e.g.* ViT, for 100 epochs on ImageNet-1K. We follow iBOT, and use SGD with different learning rates for different models. Table 2 shows that by pretraining on ImageNet-1K, Mugs consistently outperforms other methods on different backbones of various sizes. Specifically, Mugs respectively achieves 78.9% and 80.6% top-1 accuracy on ViT-S and ViT-B, and improves corresponding SoTAs by at least 1.0%. Notably, on ViT-L, by only pretraining on ImageNet-1K, Mugs sets a new SoTA accuracy of 82.1%, even comparable to the accuracy 82.3% of iBOT pretrained on ImageNet-22K.

KNN. Table 2 shows that for all backbones, Mugs achieves the highest top-1 accuracy on ImageNet-1K. It respectively makes 0.4%, 0.9%, and 2.3% improvement on ViT-S, ViT-B and ViT-L over the runner-up, showing the advantages of multi-granular representation in Mugs.

Fine-tuning. It fine tunes the pretrained backbone with a linear classifier. Following iBOT, we use AdamW with layer-wise learning rate decay to train ViT-S/ViT-B/ViT-L for 200/100/50 epochs on ImageNet-1K. Table 3 reports the classification results, in which "Supervised" means randomly initializing model parameters and training scratch. On ViT-S and ViT-B, Mugs respectively achieves new SoTA of 82.5% and 84.3%, improving the runner-up, i.e., iBOT and data2vec, by 0.2% and 0.1% respectively. Note, the reconstruction frameworks, *e.g.* MAE, have unsatisfactory linear probing performance and thus are included in Table 2. Moreover, as explained in Sec. 2, this fine-tuning setting needs much higher extra training cost, and also destroys model compatibility for deployment.

Semi-supervised learning. We use 1% or 10% training data of ImageNet-1K to fine tune the pretrained backbones. Following iBOT, we consider two settings: 1) training a logistic regression classifier on frozen features; and 2) fine-tuning the whole pretrained backbone. Table 4 shows that for both 1% and 10% training data, Mugs surpasses previous SoTAs. Notably, under fine-tuning setting with 1% labeled data, Mugs improves iBOT by a significant 4.9% accuracy.

Result Analysis. Fig. 4 uses T-SNE (Van der Maaten & Hinton, 2008) to reveal the feature differences among MoCo-v3, DINO, iBOT, and Mugs, in which each color denotes a unique class. The last subfigure in Fig. 4 (a) shows that for one class, Mugs often divides it into several clusters in the feature space, *e.g.* 4 clusters for brown, 4 for purple, 6 for red, and 2 for orange, and scatters these small clusters in a big class. We further visualize two clusters of Mugs in Fig. 4 (b) and (c): the four clusters in (b) of electric ray (i.e. "brown" in (a)) respectively cluster the same small species together; hammerhead ("orange") has two clusters in (c) corresponding to its two poses. This partially reveals multi-granular structures in the feature: classes are separately scattered, which corresponds to a group-level coarse granularity; several small scattered clusters in a class show a local-group-level fine granularity; and some separate instances in a cluster reveal an instance-level fine granularity. In



(a) T-SNE visualization of the learned feature by ViT-B/16

(b) 4 clusters (2 images per cluster) in electric ray ("brown" in (a))

(c) 2 clusters in hammerhead ("orange")

Figure 4: T-SNE visualization of the learned feature by ViT-B/16. We show the fish classes in ImageNet-1K, i.e., the first six classes , *e.g.* hammerhead ("brown") and electric ray ("orange"). (b) and (c) respectively visualizes brown and orange clusters in Mugs. See more examples in Appendix.

Table 5: Classification accuracy (%) for **transfer learning** on six datasets.

Method	ViT-S/16						ViT-B/16					
	Cif_{10}	Cif_{100}	INat_{18}	INat_{19}	Flwrs	Car	Cif_{10}	Cif_{100}	INat_{18}	INat_{19}	Flwrs	Car
Sup. (Caron et al., 2021)	99.0	89.5	70.7	76.6	98.2	92.1	99.0	90.8	73.2	77.7	98.4	92.1
BEiT (Bao et al., 2021)	98.6	87.4	68.5	76.5	96.4	92.1	99.0	90.1	72.3	79.2	98.0	94.2
MAE (He et al., 2021a)	—	—	—	—	—	—	—	—	75.4	80.5	—	—
MoCo-v3 (Chen et al., 2021)	—	—	—	—	—	—	98.9	90.5	—	—	97.7	—
DINO (Caron et al., 2021)	99.0	90.5	72.0	78.2	98.5	93.0	99.1	91.7	72.6	78.6	98.8	93.0
iBOT (Zhou et al., 2021a)	99.1	90.7	73.7	78.5	98.6	**94.0**	99.2	92.2	74.6	79.6	**98.9**	**94.3**
Mugs (ours)	**99.2**	**91.8**	**74.4**	**79.8**	**98.8**	93.9	**99.3**	**92.8**	**76.4**	**80.8**	**98.9**	94.0

contrast, MoCo-v3, DINO and iBOT often do not show this multi-granular feature structure in Fig. 4 (a). Hence, for some challenging classes, *e.g.* electric ray, Mugs can well distinguish them, while MoCo-v3, DINO and iBOT cannot. This is because instead of regarding the class as a whole, Mugs utilizes its multi-granular supervisions to consider the multi-granular (hierarchical) data semantic structures and divide the whole class into several easily-distinguishable clusters in the pretraining phase. Differently, MoCo-v3, DINO and iBOT ignore the multi-granular semantic structures and only uses one granular supervision which often could not well handle the challenging classes. Fig. 5 (a) further visualizes the self-attention of ViT-B/16. One can observe Mugs can well capture object shapes and thus their semantics. See more details and examples in Appendix A.4.

4.2 Results on Downstream Tasks

Transfer learning. We fine-tune the pretrained backbone on various kinds of other datasets with same protocols and optimization settings in iBOT. Table 5 summarizes the classification accuracy, in which "Sup." denotes the setting where we pretrain the backbone on ImageNet-1K in a supervised manner and then fine tune backbone on the corresponding dataset. Table 5 shows our Mugs surpasses SoTAs on the first five datasets and achieves comparable accuracy on the Car dataset.

Object detection & Instance segmentation. Now we evaluate Mugs on object detection and instance segmentation on COCO (Lin et al., 2014). For fairness, we use the same protocol in iBOT. See optimization settings in Appendix B. Besides SSL approaches, *e.g.* MoBY (Xie et al., 2021a), we also compare supervised baselines, Swin-T/7 (Liu et al., 2021) with similar model size as ViT-S/16. Table 6 shows that on detection, Mugs makes 0.4 AP^{b} improvement over the runner-up, i.e. iBOT. Fig. 5 (b) shows that Mugs can accurately locate and classify objects in COCO. For instance segmentation, Mugs also improves 0.4 AP^{m} over the best baseline.

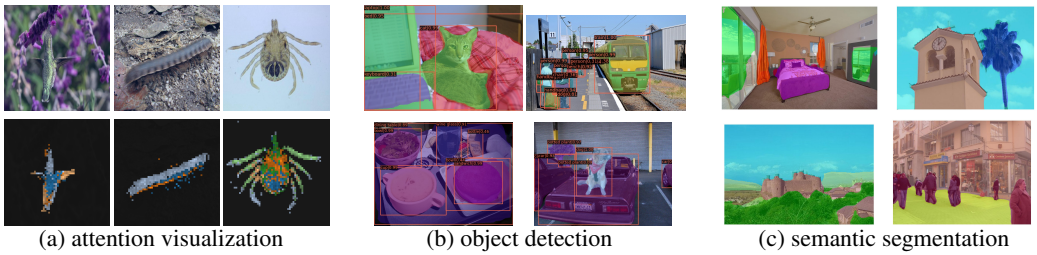
Semantic segmentation. We transfer the pretrained model to semantic segmentation task on the ADE20K dataset (Zhou et al., 2017). Following iBOT, we stack the task layer in UPerNet (Xiao

Table 6: **Object detection** (Det.) & **instance segmentation** (ISeg.) on COCO & **semantic seg.** (SSeg.) on ADE20K.

	Arch.	Param.	Det.	ISeg.	SSeg.
			AP^{b}	AP^{m}	mIoU
Sup. (Zhou et al., 2021a)	Swin-T	29	48.1	41.7	44.5
MoBY (Xie et al., 2021a)	Swin-T	29	48.1	41.5	44.1
Sup. (Zhou et al., 2021a)	ViT-S/16	21	46.2	40.1	44.5
iBOT (Zhou et al., 2021a)	ViT-S/16	21	49.4	42.6	45.4
Mugs (ours)	ViT-S/16	21	**49.8**	**43.0**	**47.4**

Table 7: **Video object segmentation** with ViT-B/16 on the DAVIS-2017 video dataset.

	$(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
DINO (Caron et al., 2021)	62.3	60.7	63.9
iBOT (Zhou et al., 2021a)	62.4	60.8	64.0
Mugs	**63.1**	**61.4**	**64.9**



(a) attention visualization (b) object detection (c) semantic segmentation

Figure 5: Visualization of pretrained ViT-B/16 (a) and ViT-S/16 (b) & (c) by Mugs.

et al., 2018) and fine-tune the whole backbone. Table 6 reports the mean intersection over union (mIoU) on all semantic categories. Mugs consistently outperforms the compared SoTAs by significant 2.0 mIoU. Fig. 5 (c) shows that Mugs can capture the object shape accurately.

Video object segmentation. We follow DINO and find nearest neighbors to segment objects in the video, since one can propagate segmentation masks via retrieving nearest neighbor between consecutive video frames. Table 7 reports the mean region similarity $\mathcal{J}_m$ and mean contour-based accuracy $\mathcal{F}_m$ on the DAVIS-2017 video segmentation dataset (Pont-Tuset et al., 2017) by using ViT-B/16. Mugs enjoys better feature transferability than DINO and iBOT even for video segmentation.

4.3 Ablation Study

Here we investigate the effects of each granular supervision in Mugs. Specifically, we train Mugs for 1,00 epochs on ImageNet-1K and report the linear probing accuracy in Table 8. One can observe that by independently removing each granular supervision, namely, the instance, local-group and group supervision, the performance of Mugs degenerates, which shows the benefit of each granular supervision, especially for the local-group supervision.

Table 8: Effects of the three granular supervisions in Mugs to the linear probing accuracy (%).

Mugs	Mugs w/o $\mathcal{L}_{\text{instance}}$	Mugs w/o $\mathcal{L}_{\text{local-group}}$	Mugs w/o $\mathcal{L}_{\text{group}}$
76.4	75.8	75.3	75.7

Next, we compare Mugs with DINO and iBOT under different augmentations and also show the effects of augmentations. For augmentation $\mathcal{T}^s$ in student network of Mugs/DINO/iBOT, we implement it by strong or weak augmentation mentioned at the beginning of Sec. 4; for augmentation $\mathcal{T}^t$ in teacher, we always use weak augmentation. See implementations details in Appendix B, especially for iBOT. We pretrain all methods for 1,00 epochs on ImageNet-1K. Table 9 shows four observations. 1) Under weak or strong augmentation, Mugs always outperforms DINO and iBOT. 2) For all three methods, strong augmentation slightly improves their performance under weak augmentation, showing the effectiveness of our strong augmentation technique on ViTs. 3) Mugs using weak augmentation surpasses iBOT with both weak augmentation and random mask augmentation. 4) Under weak augmentation, Mugs improves 1.5% over DINO which means it is the multi-granular supervisions of Mugs that contributes this 1.5% improvement. Then by using strong augmentation, Mugs surpasses DINO using weak augmentation by 2.2%, showing strong augmentation only contributes 0.7% improvement over DINO. So compared with the strong augmentation, the multi-granular supervision framework of Mugs largely contributes to Mugs and is the key factor to the significant improvement of Mugs over DINO and iBOT.

Table 9: Augmentation effects to linear probing accuracy (%) on ImageNet. † denotes that we replace vanilla augmentation in method and run the new one.

weak aug.			strong aug.			weak aug.+rand. mask
DINO	iBOT†	Mugs	DINO†	iBOT†	Mugs	iBOT
74.2	74.9	75.7	74.7	75.4	76.4	75.3

Finally, we evaluate Mugs without multi-crop augmentation, i.e. using two crops of size 224×224 for pretraining. Table 9 in Appendix A shows that Mugs also surpasses the SoTAs, including DINO and iBOT, on ViTs under the same setting, which also demonstrates the superiority of Mugs.

5 Conclusion

In this work, we propose Mugs to learn multi-granular features via three complementary granular supervisions. Instance discrimination supervision distinguishes different instances to learn fine-grained features. Local-group discrimination supervision considers the local-group around an instance and then discriminates different local-groups to extract higher-level fine-grained features. Group discrimination supervision clusters similar samples and local-groups into one cluster to capture coarse-grained global group semantics. Experimental results testify the advantages of Mugs.

REFERENCES

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv preprint arXiv:2202.03555*, 2022.

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

M. Caron, P. Bojanowski, A. Joulin, and Matthijs M. Douze. Deep clustering for unsupervised learning of visual features. In *Proc. European Conf. Computer Vision*, pp. 132–149, 2018a.

M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Proc. Conf. Neural Information Processing Systems*, 2020a.

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

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

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 13–18 Jul 2020a. URL https://proceedings.mlr.press/v119/chen20j.html.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020b.

X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020c.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

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*, pp. 248–255. Ieee, 2009.

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.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9588–9597, 2021.

J. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Pires, Z. Guo, and M. Azar. Bootstrap your own latent: A new approach to self-supervised learning. In *Proc. Conf. Neural Information Processing Systems*, 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. *arXiv preprint arXiv:2006.07733*, 2020b.

K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. IEEE Conf. 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. *arXiv preprint arXiv:2111.06377*, 2021a.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021b.

R. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

N. Komodakis and S. Gidaris. Unsupervised representation learning by predicting image rotations. 2018.

Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, et al. Mst: Masked self-supervised transformer for visual representation. *Advances in Neural Information Processing Systems*, 34, 2021.

S. Lin, P. Zhou, Z. Hu, S. Wang, R. Zhao, Y. Zheng, L. Lin, E. Xing, and X. Liang. Prototypical graph contrastive learning. *arXiv preprint arXiv:2106.09645*, 2021.

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 *eccv*, 2014.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018.

M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. European Conf. Computer Vision*, pp. 69–84. Springer, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proc. Int'l Conf. Machine Learning*, 2020.

Xudong Wang, Ziwei Liu, and Stella X Yu. Unsupervised feature learning by cross-level instance-group discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12586–12595, 2021.

Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. *arXiv preprint arXiv:2112.09133*, 2021.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *eccv*, 2018.

Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*, 2021a.

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.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *cvpr*, 2017.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021a.

Pan Zhou, Caiming Xiong, Xiaotong Yuan, and Steven Hoi. A theory-driven self-labeling refinement method for contrastive representation learning. In *Proc. Conf. Neural Information Processing Systems*, 2021b.

APPENDIX

This supplementary document provides more additional experimental results and the pretraining & fine-tuning details for the ICLR submission entitled "Mugs: A Multi-Granular Self-supervised Learning Framework". It is structured as follows. Appendix A provides more extra experimental results, including 1) the comparison among SoTAs without the multi-crop augmentation strategy in Appendix A.2, 2) fine-tuning comparison on ViT-L/16 in Appendix A.3, 3) more T-SNE clustering visualization results in Appendix A.4, 4) more attention visualization results in Appendix A.5, 5) more visualization results on object detection and segmentation in Appendix A.6.

Appendix B provides more experimental details for Sec. 4 in manuscript. Specifically, Appendix B.1 gives more pretraining details, including implementations of weak and strong augmentations, loss construction under multi-crop setting, hyper-parameter settings for pretraining and the pretraining cost. Then Appendix B.2 introduces more details for fine-tuning and semi-supervised learning in Sec. 4.1 in manuscript. Next, in Appendix B.3, we present more details for downsteam tasks, including transfer learning, object detection & instance segmentation, and semantic segmentation. Finally, Appendix B.4 tells us more implementation details of DINO and iBOT under weak and strong agumentation settings to complement Sec. 4.3 in manuscript.

A MORE EXPERIMENTAL RESULTS

Due to space limitation, we defer more experimental results to this appendix. Here we first investigate the performance of Mugs without multi-crop augmentations which is widely used in several representative works, and further compare it with other methods, include iBOT and DINO under the same setting. Then we present more visualization results, including T-SNE clustering visualization, attention visualization of multi-heads in ViT, and object detection and segmentation visualization. We hope these visualization results can help readers intuitively understand the learnt features by Mugs.

A.1 MORE PRETRAINING DETAILS

We pretrain Mugs on the training data of ImageNet-1K (Deng et al., 2009) without labels. Following (Zhou et al., 2021a), we pretrain ViT-S/16 for 800 epochs, ViT-B/16 for 400 epochs, and ViT-L for 250 epochs. Following DINO and iBOT, we use symmetric training loss, i.e. $\frac{1}{2}(\mathcal{L}(\boldsymbol{x}_1, \boldsymbol{x}_2) + \mathcal{L}(\boldsymbol{x}_2, \boldsymbol{x}_1))$. We use AdamW optimizer (Loshchilov & Hutter, 2018) with a momentum of 0.9, a weight decay of 0.1, and a cosine schedule (Loshchilov & Hutter, 2016). For data augmentation, we adopt weak augmentation in DINO (Caron et al., 2021) to implement $\mathcal{T}^{\text{t}}$ in teacher, and use strong augmentation (mainly including AutoAugment (Cubuk et al., 2018)) in DeiT (Touvron et al., 2021) as the augmentation $\mathcal{T}^{\text{s}}$ in student. Following conventional multi-crop setting (Caron et al., 2020b; 2021; Zhou et al., 2021a), we crop each image into 2 large crops of size 224 and 10 extra small crops of size 96. For both large crops, we feed each of them into teacher and use its output to supervise the student's output from the other 11 crops. For two-crop setting, Table 9 in Appendix A reports the results and shows the superiority of Mugs over SoTAs.

For Mugs, we follow MoCo to set $\tau_{\text{in}} = \tau_{\text{lg}} = 0.2$ in the infoNCE loss, and follow DINO to set $\tau'_{\text{g}} = 0.1$ and linearly warm up τ_{g} from 0.04 to 0.07. We set the neighbor number $k = 8$, and set $\rho = 0.9$ in group discrimination. Mugs has almost the same training cost with DINO, *e.g.* about 27 hours with 8 A100 GPUs for 100 pretraining epochs on ViT-S/16, as our projection/prediction heads and transformers $g_{\text{transformer}}$ are much smaller than the backbone.

Table 9: **Linear probing accuracy (%) and k-NN accuracy (%)** on ImageNet-1K without multi-crop augmentation (left) and with multi-crop augmentation (right). "Epo" is the effective pretraining epochs adjusted by number of views processed by the models following (Zhou et al., 2021a).

Method	Para.	Epo.	Lin.	k-NN
DINO	21	3200	73.7	70.0
iBOT	21	3200	76.2	72.4
Mugs	21	3200	**76.9**	**73.1**

Method	Para.	Epo.	Lin.	k-NN
DINO	21	3200	77.0	74.5
iBOT	21	3200	77.9	75.2
Mugs	21	3200	**78.9**	**75.6**

Table 10: **Fine-tuning** classification accuracy (%) on ImageNet-1K. All methods are pretrained on ImageNet-1K. "Epo." is the effective pretraining epochs adjusted by number of views processed by the models following (Zhou et al., 2021a).

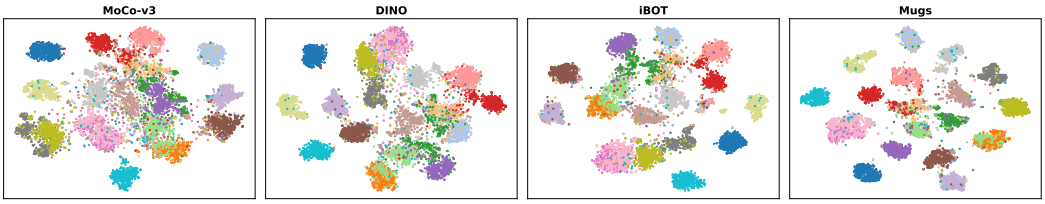
	Method	ViT-L/16 Epo.	ViT-L/16 Acc. (%)
	Supervised (Touvron et al., 2021)	—	83.1
reconstruction	BEiT (Bao et al., 2021)	800	85.2
	MAE (He et al., 2021a)	1600	85.9
	data2vec (Baevski et al., 2022)	1600	**86.6**
contrastive or clustering	DINO (Caron et al., 2021)	—	—
	iBOT (Zhou et al., 2021a)	1000	84.8
	MoCo-v3 (Chen et al., 2021)	600	84.1
	Mugs (ours)	1000	85.2

A.2 Comparison w/o and w/ Multi-Crop Augmentation

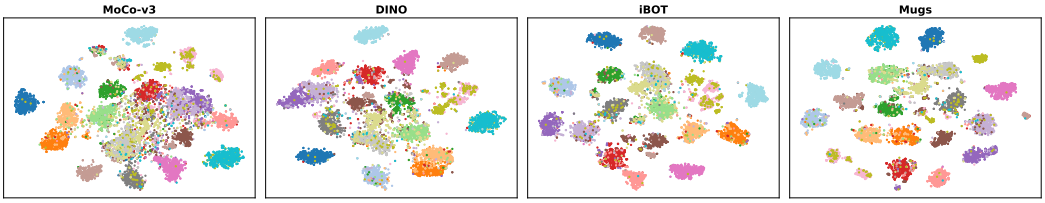
Here we first investigate the performance of Mugs without the multi-crop augmentation which is widely used in several representative works, and further compare it with other SoTA methods, include iBOT and DINO under the same setting. Specifically, for Mugs without multi-crop augmentation, it only uses two 224-sized crops for pretraining. The left table in Table 9 reports the results of all compared methods without multi-crop augmentation, while the right one summarizes the results under multi-crop augmentation setting. By comparison, one can observe that without multi-crop augmentation, Mugs still consistently achieves the highest accuracy under both linear probing setting and KNN setting. Specifically, Mugs improves the runner-up, namely iBOT, by respectively 0.8% and 0.5% under linear probing and KNN evaluation settings. More importantly, we can observe that Mugs without multi-crop augmentation even achieves very similar results as DINO with multi-crop augmentation. All these results are consistent with those results in Table 2 in the manuscript, and well demonstrate the superiority of Mugs over previous state-of-the-arts.

A.3 Comparison under Fine-tuning Setting

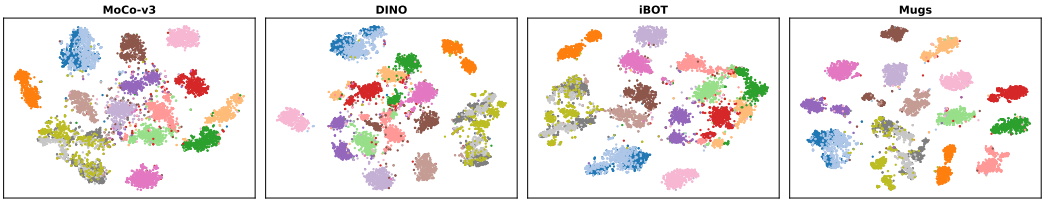
In the manuscript, we already compare Mugs with state-of-the-art approaches on the ViT-S/16 and ViT-B/16 under the fine-tuning setting. Due to limited space, we defer the comparison among Mugs and others on ViT-L/16 into Table 10. This setting allows us to optimize the pretrained backbone with a linear classifier. Following BEiT (Bao et al., 2021), DINO and iBOT, we use AdamW optimizer with layer-wise learning rate decay to train ViT-L for 50 epochs on ImageNet-1K. On ViT-L, Mugs achieves 85.2% top-1 accuracy, and surpasses all contrastive learning and clustering learning methods. One can also observe that on ViT-L, most of the reconstruction methods achieves higher accuracy than constrictive or clustering learning approaches, including iBOT and our Mugs. There are two possible reasons. Firstly, the reconstruction methods use much more computations for pretraining than constrictive or clustering learning approaches. Specifically, the reconstruction family always use 224×224-sized images to pretrain the model, while constrictive or clustering learning approaches uses multi-crop augmentations which contains two 224-sized images and ten 96-sized images. Since "Epo." in Table 10 is the effective pretraining epochs adjusted by number of views processed by the models (Zhou et al., 2021a) which means each 96-sized image equals to one 224-sized image in terms of the defined "epochs", with the same pretraining epochs, the computation cost of the reconstruction approaches is much more. Actually, from Table 10, the reconstruction methods have much more effective pretraining epochs than constrictive or clustering learning approaches, e.g. 1600 epochs in data2vec v.s. 1000 epochs in iBOT & Mugs, which further increases the training cost. Secondly, for large models, using small-sized images, e.g. ten 96-sized images in multi-crop augmentations, may lead to overfitting issue in contrastive or clustering learning approaches. Specifically, from Table 2 in manuscript and Table 10 here, once can observe that on relatively small models, such as ViT-S and ViT-B, SoTA contrastive learning or clustering methods, such as Mugs and iBOT, outperform the reconstruction methods, even though the formers have much less pretraining cost as mentioned above. But on large models, e.g. ViT-L, the superiority of SoTA contrastive or clustering learning methods disappears. One possible reason for these inconsistent observation is that large model needs more pretraining epochs for learning semantic features, and could suffer from over-fitting problem when using 96-sized crops, since 1) large model is capable to memory all images as demonstrated in many works; and 2) 96-sized crops may contain incomplete



(a) T-SNE visualization of 19 classes of insects, *e.g.* beetle, butterfly, stick, and cricket.



(b) T-SNE visualization of 20 classes of various monkeys, *e.g.* gibbon, siamang, patas, and gorilla.



(c) T-SNE visualization of 17 classes of various birds, *e.g.* junco, robin, jay, cock, and ostrich.

Figure 5: More T-SNE visualization of the learned features by ViT-B/16 trained by our Mugs. **Best viewed in color pdf file.**

semantics of the vanilla image and lead to over-fitting issue, especially under insufficient pretraining epochs. Note, as explained at the end of Sec. 2 in manuscript, this fine-tuning setting needs much higher extra training cost, and also destroys model compatibility for deployment. Therefore, in this work, we do not further push Mugs's limits on the large models which needs huge training cost as the reconstruction methods.

A.4 More T-SNE Visualization Results

Same with Fig. 4 in the manuscript, here we use T-SNE (Van der Maaten & Hinton, 2008) to reveal the feature differences among MoCo-v3, DINO, iBOT, and Mugs in Fig. 5. By comparison, Mugs often can scatter the samples from different classes more separately, while keeping the samples in the same class close in the feature space. This could means that our Mugs can better distinguish different classes than MoCo-v3, DINO and iBOT, and thus shows higher performance. The potential reason behind this observation is explained in manuscript. That is, instead of regards the class as a whole, Mugs utilizes its multi-granular supervisions to consider the multi-granular (hierarchical) data semantic structures and divides the whole class into several clusters for easily discriminating in the pretraining phase. Differently, MoCo-v3, DINO and iBOT ignore the multi-granular semantic structures and only uses one granular supervision which often could not well handle the challenging classes.

A.5 More Attention Visualization Results

Here same with Fig. 5 in the manuscript, we visualize more self-attention map of the 12 self-attention heads in ViT-B/16 pretrained by Mugs in Fig. 6. The first column denotes the vanilla images, while each column of the last 12 columns denote the self-attention score maps of each individual head. The second column combines the 12 self-attention score maps from 12 heads into one, and also sets a threshold to remove some noises via only keeping top attention score. From these visualizations, one can observe that by using Mugs for pretraining, the overall self-attention of 12 heads can capture the object shapes very well. For example, from the first bird image, it is even hard for human to get the

Figure 6: Self-attention visualization of ViT-B/16 pretrained by our Mugs. The images from left to right respectively denote the vanilla image, the overall self-attention score of all 12 heads in ViT-B, and the individual self-attention score of 12 heads. **Best viewed in color pdf file.**

bird location at the first glance, due to the similar color of the bird and the flowers. But the ViT-B/16 pretrained by Mugs still can well locate the bird and also capture the bird shape. Moreover, one can also compare the attention visualization of Mugs with state-of-the-arts, *e.g.* iBOT. In iBOT (Zhou et al., 2021a), Fig. 18 in their appendix also visualizes the self-attention map. By comparison, the

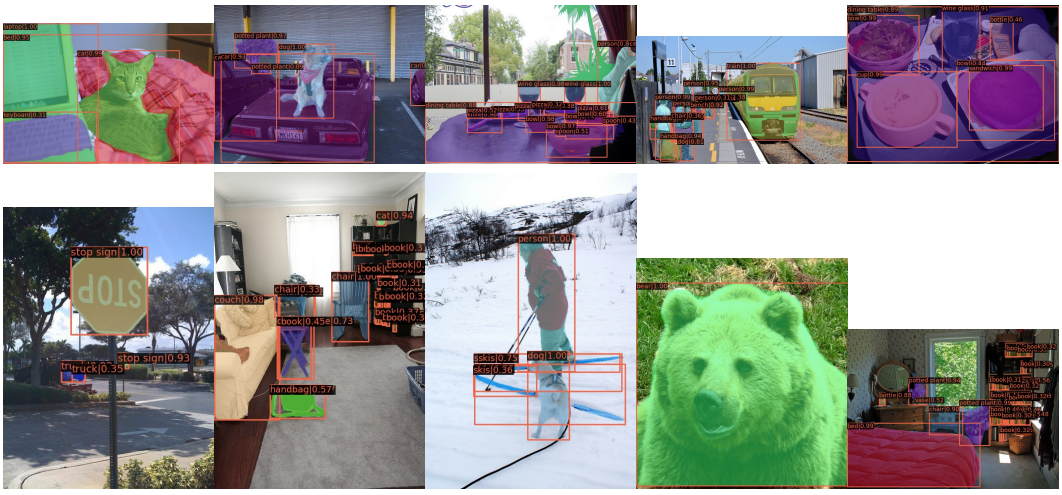


Figure 7: Object detection visualization of ViT-B/16 pretrained by Mugs. **Best viewed in color pdf file.**

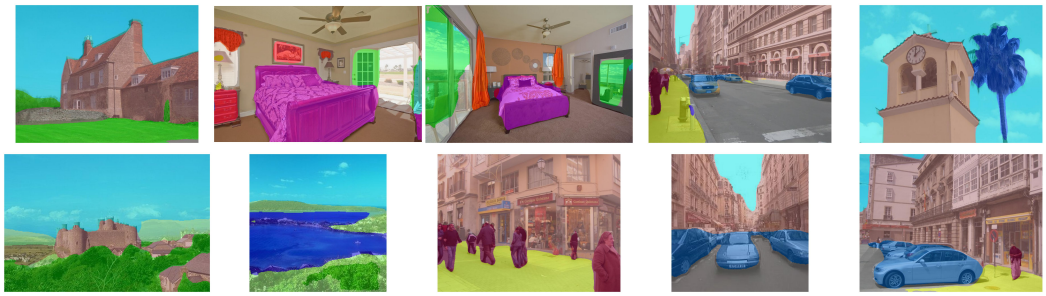


Figure 8: Semantic segmentation visualization of ViT-B/16 pretrained by Mugs. **Best viewed in color pdf file.**

model pretrained by Mugs can better separate the object from background. These results testify that ViT-B/16 pretrained by Mugs can capture semantics in data even without any manual labels.

A.6 More Visualization Results on Object Detection and Semantic Segmentation

In the manuscript, we already provide some object detection and segmentation examples in Fig. 5. Here we give more examples. Fig. 7 shows more object detection examples on the COCO datasets, where we use the ViT-B/16 pretrained by our Mugs. From these results, one can observe that Mugs not only accurately locate the objects in the images but also precisely recognizes these objects. For semantic segmentation on ADE20K, Fig. 8 visualizes more examples. We also can find that Mugs can capture the object shape accurately and thus well captures the semantics of an image.

B More Experimental Details

Due to space limitation, we defer more experimental details to this section. Here we first provide more details for pretraining. Then we present the experimental details for various kinds of downstream tasks. Finally, we also give the details for ablation study.

B.1 More Details for Pretraining

B.1.1 AUGMENTATION.

Here the weak augmentation in our teacher backbone refers to the augmentations used in many SSL works, *e.g.* DINO (Caron et al., 2021), which includes random crop, color jitter, Gaussian noise, horizontal flipping and gray scaling. The hyper-parameters in these augmentation operations are the same with those in DINO (Caron et al., 2021). The strong augmentation in our student backbone is the combination of AutoAugment (Cubuk et al., 2018) used in DeiT (Touvron et al., 2021) and the above weak augmentation. Specifically, for each image, with probability 0.5, we use AutoAugment (Cubuk et al., 2018) to augment it; otherwise, we use weak augmentation to crop it. We use this sampling strategy to avoid training collapse while keeping sufficient data diversity. Following DINO and iBOT, we always set the global crop scale as $[0.25, 1]$ and local crop scale as $[0.05, 0.25]$.

B.1.2 LOSS FOR MULTI-CROP SETTING.

Following conventional multi-crop setting (Caron et al., 2020b; 2021; Zhou et al., 2021a), we crop each image into 2 large crops of size 224 and 10 extra small crops of size 96. Then to construct the overall pretraining loss 6 in manuscript, we regard one large crop as $\boldsymbol{x}_1$ and respectively take the remaining 11 crops as $\boldsymbol{x}_2$. Then symmetrically, we view another large crop as $\boldsymbol{x}_1$ and respectively take the remaining 11 crops as $\boldsymbol{x}_2$. Finally, we average these loss to obtain the overall training loss.

B.1.3 HYPER-PARAMETER SETTINGS FOR PRETRAINING.

For all experiments, we use AdamW optimizer (Loshchilov & Hutter, 2018) with a momentum of 0.9 and a cosine learning rate schedule (Loshchilov & Hutter, 2016). We also linearly warm up the learning rate at the first 10 epochs from 10^{-6} to its base value, and then decay it with a cosine schedule (Loshchilov & Hutter, 2016). For ViT-S and ViT-B, we use a minibatch size of 1024, a base learning rate of 8×10^{-4}, and a weight decay of 0.1. For ViT-L, due to our limited computational resource, we use a minibatch size of 640, a base learning rate of 1.5×10^{-4}, and a weight decay of 0.08. For all experiments, the learning rate of the patch embedding layer is $5\times$ smaller than the base learning rate. This strategy is demontrated to be useful for stabling training in MoCo-v3 (Chen et al., 2021). For drop path rate, we set it as 0.1/0.2/0.4 for ViT-S/B/L respectively. We set clip gradient as 3.0 for ViT-S/B and 0.3 for ViT-L. For Mugs, we follow MoCo to set $\tau_{\text{in}} = \tau_{\text{lg}} = 0.2$ in the infoNCE loss, and follow DINO to set $\tau'_{\text{g}} = 0.1$ and linearly warm up τ_{g} from 0.04 to 0.07. We set the neighbor number $k = 8$, and set $\rho = 0.9$ to estimate the center $\boldsymbol{p}_{\text{ema}}$ in group discrimination. All these settings are almost the same as DINO for simplicity which reduces hyper-parameter tuning and saves computational budget.

B.1.4 PRETRAINING COST.

Mugs takes about 27 hours with 8 A100 GPUs for 100 pretraining epochs on ViT-S/16. This means that Mugs has almost the same training cost with DINO, since our projection/prediction heads and transformers $g_{\text{transformer}}$ are much smaller than the backbone. For ViT-B/16, Mugs needs about 24 hours on 16 A100 GPUs for 100 pretraining epochs. To training 100 epochs on ViT-L/16, Mugs takes about 48 hours on 40 A100 GPUs. For ViT-B and ViT-L, it is hard for us to compare with DINO, since it does not report the training time.

B.2 MORE TRAINING DETAILS FOR EVALUATION ON IMAGENET-1K

B.2.1 FINE-TUNING.

As mentioned in manuscript, we follow BEiT (Bao et al., 2021), DINO and iBOT, and use AdamW optimizer with layer-wise learning rate decay to train ViT-S/ViT-B/ViT-L for 200/100/50 epochs on ImageNet-1K. We set layer-wise learning rate decay as 0.55 and learning rate 1.2×10^{-3} for both ViT-S and ViT-B. For ViT-L, we use layer-wise learning rate decay 0.75 and learning rate 8.0×10^{-4}. For drop path rate, we set it as 0.1/0.2/0/3 for ViT-S/ ViT-B/ViT-L respectively. All these hyper-parameters are around at the suggested ones in BEiT and iBOT.

B.2.2 SEMI-SUPERVISED LEARNING.

Following DINO and iBOT, we consider two settings: 1) training a logistic regression classifier on frozen features; and 2) fine-tuning the whole pretrained backbone. For logistic regression classifier, we use AdamW optimizer with total minibatch size 1024 and weight decay 0.05, under both 1% and 10% training data settings. We sweep the learning rate $\{0.03, 0.06, 0.10, 0.2\}$. For fine-tuning 1000 epochs on ViT-S/16, we also use AdamW optimizer with total minibatch size 1024 and weight decay 0.05 under both 1% and 10% training data settings. We respectively set learning rate as 2×10^{-6} and 5×10^{-6} for 1% and 10% training data.

B.3 MORE DETAILS FOR DOWNSTEAM TASKS

B.3.1 TRANSFER LEARNING.

We pretrain the model on ImageNet-1K, and then fine-tune the pretrained backbone on various kinds of other datasets with same protocols and optimization settings in DINO and iBOT. Specifically, following DINO and iBOT, for both ViT-S and ViT-B, we always use AdamW optimizer with a minibatch size of 768. We fine-tune the pretrained model 360 epochs on INat_{18} and INat_{19}, and 1000 epochs on Cif_{10}, Cif_{100}, Flwrs and Car. For all datasets, we sweep the learning rate $\{7.5 \times 10^{-6}, 1.5 \times 10^{-5}, 3.0 \times 10^{-5}, 7.5 \times 10^{-5}, 1.5 \times 10^{-4}\}$. For we set weight decay as 2×10^{-2} for CIFAR10 and CIFAR100 on ViT-B, and use a weight decay of 5×10^{-2} for all remaining experiments. For example, on the INat_{18} dataset, we use a learning rate of 3.0×10^{-5}/1.5×10^{-5} for ViT-S/ViT-B; on the INat_{19} dataset, we set learning rate as 7.5×10^{-5}/3.0×10^{-5} for ViT-S/ViT-B. For more hyper-parameters, please refer to the hyper-parameter configure file in our released code.

B.3.2 OBJECT DETECTION & INSTANCE SEGMENTATION.

For fairness, we follow DINO and iBOT, and fine-tune the pretrained backbone via a multi-scale strategy, namely resizing image at different scales. Please refer to iBOT for more details. We use AdamW optimizer with a learning rate of 2×10^{-4}, a weight decay of 0.05 to fine-tune with $1\times$ schedule, i.e. 12 epochs with the learning rate decayed by $10\times$ at epochs 9 and 11. We sweep a layer decay rate of $\{0.65, 0.75, 0.8, 0.9\}$ and finally choose 0.8 because of its good performance. For test, we do not use multi-scale strategy.

B.3.3 SEMANTIC SEGMENTATION.

For semantic segmentation, we follow DINO and iBOT, and fine-tune the pretrained backbone, and fine-tune the pretrained backbone by using 512×512-sized images for 1.6×10^4 iterations. We use AdamW optimizer with a learning rate of 2×10^{-4}, a weight decay of 0.05 and a layer decay rate of 0.9 to fine-tune. For this task, we do not use any multi-scale strategy for training and test. We sweep the learning rate $\{2 \times 10^{-5}, 3 \times 10^{-5}, 4 \times 10^{-5}, 5 \times 10^{-5}\}$ and finally choose 3×10^{-5} because of its good performance.

B.4 MORE DETAILS FOR ABLATION STUDY

Here we provide the implementation details for DINO and iBOT under different augmentations. As mentioned in Sec. 4.3 in the manuscript, for the augmentation $\mathcal{T}^s$ in student network of Mugs/DINO/iBOT, we implement it by strong or weak augmentation as mentioned at the beginning of Sec. 4; for augmentation $\mathcal{T}^t$ in teacher, we always use weak augmentation.

We first consider weak augmentation setting. For DINO, there is no any change, since its vanilla version uses the weak augmentation. For Mugs, we only replace the strong augmentation used in the student network with the weak augmentation. For iBOT, it has two losses, the proposed masked image modeling (MIM) loss and the clustering loss in DINO. To construct the MIM loss, iBOT needs to randomly mask the patches of input in the student network. But to build the clustering loss, it actually does not require random masks on input patches, but still uses the random masks in practice which actually increases the data augmentations for clustering loss. In this case, for fair comparison, we remove the random masks and only perform weak augmentation to construct the

clustering loss in iBOT. Note, we still preserve the random masks for MIM loss to ensure MIM is the vanilla one in iBOT.

We then consider strong augmentation setting. For DINO, we only replace the weak augmentation used in the student network with our strong augmentation. For iBOT, same as the above for weak augmentation, we does not change the augmentation in MIM loss. But for building the clustering loss, we also remove the random masks and only perform strong augmentation.