Mugs: A Multi-Granular Self-Supervised Learning Framework

Pan Zhou[1] **Yichen Zhou**[1,2] **Chenyang Si**[1] **Weihao Yu**[1,2] **Teck Khim Ng**[2] **Shuicheng Yan**[1]
[1]Sea AI Lab [2]National University of Singapore
{zhoupan,zhouyc,sicy,yuweihao,yansc}@sea.com ngtk@comp.nus.edu.sg

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 group level. 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.

1 Introduction

The family of self-supervised learning (SSL) approaches [4, 8, 9, 12, 19, 21, 24] 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 [29], is elaborately designed to generate pseudo labels of unlabeled visual data which are then utilized to train a model. 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 [4, 8, 18, 19, 21, 26, 44], and is showing greater potential than supervised learning approaches for representation learning.

Motivation. In practice, various downstream tasks in SSL field often require different granular features, *e.g.* 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 [16] 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, *e.g.* Labrador and poodle in the dog category. But this important multi-granularity requirement is ignored in the current state-of-the-art SSL approaches, including contrastive learning family [21, 23] and clustering learning family [3, 7].

NeurIPS 2022 Workshop SSLTheoryPractice

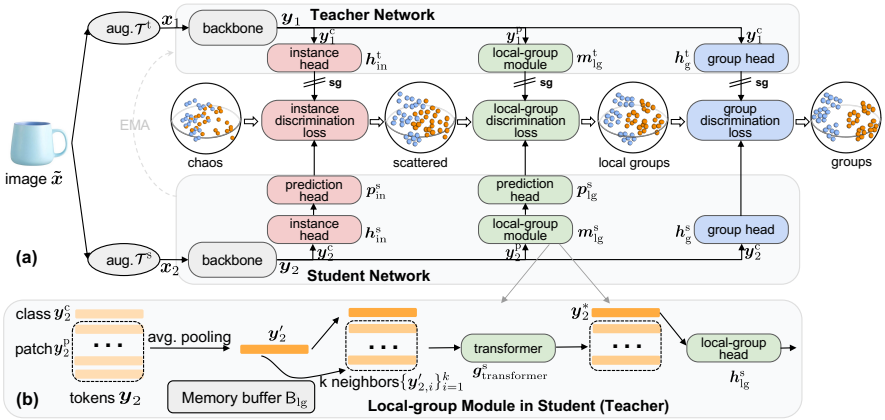


Figure 1: **Overall framework of Mugs**. **(a)** shows the overall framework. For each image, Mugs respectively feeds its two crops into backbones of student and teacher. Next, it uses three granular supervisions from instance, local-group and group levels. "sg" denotes stop-gradient. **(b)** shows the 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.

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). 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", 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. LGDS provides complementary instance supervision to IDS, since it enforces different crops of the same image to have highly similar neighbors, which is an extra challenging alignment; 2) it encourages highly similar instances to constitute small local-groups and scatters these groups separately, enhancing discrimination. 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 supervisions, Mugs learns multi-granular features which often enjoy better generality and transferability on diverse downstream tasks than single-granular features.

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 [43], by a large margin 1.1%. Besides, on several downstream tasks, *e.g.* detection, Mugs also beats previous SoTAs.

2 Multi-granular self-supervised learning

Overall Framework. 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. 1 (a), given an image $\tilde{\boldsymbol{x}}$, Mugs uses augmentations $\mathcal{T}^{\text{t}}$ and $\mathcal{T}^{\text{s}}$ 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.

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 together and pushes other crops away via the following InfoNCE (MoCo) loss [30]

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

where $\boldsymbol{y}_1^{\text{c}}$/ $\boldsymbol{y}_2^{\text{c}}$ is class token in $\boldsymbol{y}_1$/$\boldsymbol{y}_2$, $\boldsymbol{z}_1=h_{\text{in}}^{\text{t}}(\boldsymbol{y}_1^{\text{c}})$, $\boldsymbol{z}_2=p_{\text{in}}(h_{\text{in}}^{\text{s}}(\boldsymbol{y}_2^{\text{c}}))$ with heads h_{in}^{t} and h_{in}^{s}. Buffer B_{in} stores the negative instances of $\boldsymbol{z}_2$, and is updated by the minibatch features $\{\boldsymbol{z}_1\}$ of teacher.

Local-group discrimination supervision (LGDS). As aforementioned, 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.

As shown in Fig. 1 (a), for crop $\boldsymbol{x}_1$ of image $\widetilde{\boldsymbol{x}}$, 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 $\boldsymbol{x}_2$ into student to obtain $\boldsymbol{y}_2$ with class/patch token $\boldsymbol{y}_2^{\text{c}}$/$\boldsymbol{y}_2^{\text{p}}$. Next, Mugs respectively averages the patch tokens $\boldsymbol{y}_1^{\text{p}}$ and $\boldsymbol{y}_2^{\text{p}}$ to obtain $\boldsymbol{y}_1'$ and $\boldsymbol{y}_2'$ shown in Fig. 1 (b). Meanwhile, Mugs uses a buffer B_{lg} to store historical $\{\boldsymbol{y}_1'\}$ and $\{\boldsymbol{y}_2'\}$. Next, for $\boldsymbol{y}_1'$ 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 buffer B_{lg}. Finally, it 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 $g_{\text{transformer}}^{\text{t}}(\boldsymbol{y}_1';\{\boldsymbol{y}_{1,i}'\}_{i=1}^k)$ is a 2-layered vanilla ViT with output (class) token $\boldsymbol{y}_1^*$. Since $\boldsymbol{y}_1^*$ comes from $\boldsymbol{y}_1'$ and its neighbors $\{\boldsymbol{y}_{1,i}'\}_{i=1}^k$ which together constitute a local group of $\boldsymbol{y}_1'$, $\boldsymbol{y}_1^*$ is called "local group feature". 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.

LGDS benefits Mugs from two aspects. **1)** It provides complementary instance supervision to the above IDS. It brings two local-group features $\boldsymbol{y}_1^*$ and $\boldsymbol{y}_2^*$ from $\widetilde{\boldsymbol{x}}$ close. 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 neighbors. Thus, besides the crops themselves, their corresponding neighbors should also be well aligned, which is an extra challenging alignment problem 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. It aims to cluster 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 $\boldsymbol{p}_i^{\text{t}}=\frac{\exp(\sigma(h_{\text{g}}^{\text{t}}(\boldsymbol{y}_1^{\text{c}}))\cdot\boldsymbol{c}_i/\tau_{\text{g}})}{\sum_{i=1}^m\exp(\sigma(h_{\text{g}}^{\text{t}}(\boldsymbol{y}_1^{\text{c}}))\cdot\boldsymbol{c}_i/\tau_{\text{g}})}$ and $\boldsymbol{p}_i^{\text{s}}=\frac{\exp(h_{\text{g}}^{\text{s}}(\boldsymbol{y}_2^{\text{c}})\cdot\boldsymbol{c}_i/\tau_{\text{g}}')}{\sum_{i=1}^m\exp(h_{\text{g}}^{\text{s}}(\boldsymbol{y}_2^{\text{c}})\cdot\boldsymbol{c}_i/\tau_{\text{g}}')}$. Here the function σ in [7] is to sharpen the soft pseudo label $\boldsymbol{p}^{\text{t}}$. Next, similar to a supervised classification task, Mugs employs the cross-entropy loss but with soft labels as its training loss $\mathcal{L}_{\text{group}}(\boldsymbol{x}_1,\boldsymbol{x}_2)=-\sum_{i=1}^m\boldsymbol{p}_i^{\text{t}}\log(\boldsymbol{p}_i^{\text{s}})$.

Now we discuss the co-effects of the above supervisions. IDS pulls the crops of the same image close and scatters the instance features separately on the spherical surface as shown in Fig. 1 (a), thus learning instance-level fine-grained features. LGDS first provides complementary supervision for IDS by encouraging crops of the same instance to have highly similar neighbors. Then, as shown in the third sphere in Fig. 1 (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 local-group features. To avoid similar local-groups to be

Table 1: **Linear probing and k-NN** accuracy (%) on 1K. "1K" is short for ImageNet-1K.

	Method	Pre. data	Pre. Epoch	Lin.	k-NN
ResNet-50	MoCo-v3 [14]	1K	1600	74.6	—
	SimCLR [10]	1K	1600	69.3	—
	InfoMin Aug [33]	1K	1600	73.0	—
	SimSiam [13]	1K	1600	71.3	—
	BYOL [20]	1K	2000	74.3	—
	SwAV [6]	1K	2400	75.3	65.7
	DeepCluster [5]	1K	2400	75.2	—
	DINO [7]	1K	3200	75.3	67.5
ViT-S	MoCo-v3 [14]	1K	3200	73.4	—
	SwAV [6]	1K	3200	73.5	66.3
	DINO [7]	1K	3200	77.0	74.5
	iBOT [43]	1K	3200	77.9	75.2
	Mugs (ours)	1K	3200	**78.9**	**75.6**
ViT-B	MoCo-v3 [14]	1K	1200	76.7	—
	DINO [7]	1K	1600	78.2	76.1
	iBOT [43]	1K	1600	79.5	77.1
	Mugs (ours)	1K	1600	**80.6**	**78.0**
ViT-L	MoCo-v3 [14]	1K	1200	77.6	—
	iBOT [43]	1K	1000	81.0	78.0
	Mugs (ours)	1K	1000	**82.1**	**80.3**
	iBOT [43]	22K	200	82.3	72.9

Table 2: **Fine-tuning** accuracy (%) on 1K. All are pretrained on 1K.

Method	ViT-S/16		ViT-B/16	
	Epo.	Acc. (%)	Epo.	Acc. (%)
Supervised [34]	—	79.9	—	81.8
BEiT [2]	800	81.4	800	83.4
MAE [22]	—	—	1600	83.6
SimMIM [39]	—	—	1600	83.8
MaskFeat [36]	—	—	1600	84.0
data2vec [1]	—	—	1600	84.2
MoCo-v3 [14]	600	81.4	600	83.2
DINO [7]	3200	82.0	1600	83.6
iBOT [43]	3200	82.3	1600	83.8
Mugs (ours)	3200	**82.6**	1600	**84.3**

Table 3: **Semi-supervised** accuracy (%) on 1K.

Method	Arch.	logistic reg.		fine-tuning	
		1%	10%	1%	10%
SimCLRv2 [11]	RN50	—	—	57.9	68.1
BYOL [20]	RN50	—	—	53.2	68.8
SwAV [6]	RN50	—	—	53.9	70.2
DINO [7]	ViT-S/16	64.5	72.2	60.3	74.3
iBOT [43]	ViT-S/16	65.9	73.4	61.9	75.1
Mugs (ours)	ViT-S/16	**66.9**	**74.0**	**66.8**	**76.8**

scattered randomly or far away, GDS brings similar samples together and thus pulls similar local-groups close, as illustrated by the last sphere in Fig. 1 (a). It is responsible to capture the coarse-grained group features. 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.

3 Experiments

Due to space limitation, we defer the experimental details into Appendix B. We follow the standard SSL pretraining setting and use Mugs to train ViT [17, 25, 32, 40, 41] on ImageNet-1K [16].

Linear Probing & KNN. For linear probing, Table 1 shows that by pretraining on ImageNet-1K, Mugs improves corresponding SoTAs on ViT-S and ViT-B by at least 1.0%. Notably, Mugs sets a new SoTA 82.1% on ViT-L by using ImageNet-1K, even comparable to the accuracy 82.3% pretrained on ImageNet-22K. For KNN, Mugs achieves the highest top-1 accuracy on all backbones.

Fine-tuning. Table 2 shows that 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.

Semi-supervised learning. Table 3 shows that with 1% or 10% training data, Mugs always surpasses previous SoTAs. With 1% labeled data, Mugs improves iBOT by a significant 4.9% accuracy.

Transfer learning. Table 4 shows our Mugs surpasses SoTAs in most cases.

More extra results. Appendix B gives more results on detection, segmentation, and visualization.

Table 4: **Transfer learning** accuracy (%) 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. [7]	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 [2]	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 [22]	—	—	—	—	—	—	—	—	75.4	80.5	—	—
MoCo-v3 [14]	—	—	—	—	—	—	98.9	90.5	—	—	97.7	—
DINO [7]	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 [43]	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

4 Conclusion

In this work, we propose Mugs to learn multi-granular features via three complementary granular supervisions. instance discrimination supervision (IDS), local-group discrimination supervision (LGDS), and group discrimination supervision (GDS). 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 on several benchmark tasks.

References

[1] 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.

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

[3] 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, 2018.

[4] 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*, 2020.

[5] 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, 2018.

[6] 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*, 2020.

[7] 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.

[8] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *Proc. Int'l Conf. Machine Learning*, 2020.

[9] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. Big self-supervised models are strong semi-supervised learners. In *Proc. Conf. Neural Information Processing Systems*, 2020.

[10] 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 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

[11] 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*, 2020.

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

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Towards sustainable self-supervised learning. *arXiv preprint arXiv:2210.11016*, 2022.

[19] 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*, 2020.

[20] 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*, 2020.

[21] 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.

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

[23] 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.

[24] J. Li, P. Zhou, C. Xiong, R. Socher, and S. CH Hoi. Prototypical contrastive learning of unsupervised representations. In *Int'l Conf. Learning Representations*, 2021.

[25] Yuxuan Liang, Pan Zhou, Roger Zimmermann, and Shuicheng Yan. Dualformer: Local-global stratified transformer for efficient video recognition. In *European Conference on Computer Vision*, pp. 577–595. Springer, 2022.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

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

[31] 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.

[32] Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng Yan. Inception transformer. *arXiv preprint arXiv:2205.12956*, 2022.

[33] 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.

[34] 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.

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

[36] 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.

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

[38] 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*, 2021.

[39] 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*, 2021.

[40] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10819–10829, 2022.

[41] Weihao Yu, Chenyang Si, Pan Zhou, Mi Luo, Yichen Zhou, Jiashi Feng, Shuicheng Yan, and Xinchao Wang. Metaformer baselines for vision. *arXiv preprint arXiv:2210.13452*, 2022.

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

[43] 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*, 2021.

[44] 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*, 2021.

A Appendix

This supplementary document provides more additional experimental results and the pretraining & fine-tuning details for the submission entitled "Mugs: A Multi-Granular Self-supervised Learning Framework". It is structured as follows. Appendix B provides more extra experimental results.

B More Experimental Results

Due to space limitation, we defer more experimental results to this appendix. 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 [17]. 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 [16]. For augmentation, we adopt weak augmentation in DINO to implement $\mathcal{T}^{\text{t}}$ in teacher, and use strong augmentation (mainly including AutoAugment [15]) in DeiT [34] 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 B reports the results and shows superiority of Mugs over SoTAs.

We set the neighbor number $k = 8$, and thus use transformers $g^{\text{t}}_{\text{transformer}}$ and $g^{\text{s}}_{\text{transformer}}$. 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.

B.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 DINO and iBOT, and use SGD with different learning rates for different models. Table 1 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% pretrained on ImageNet-22K.

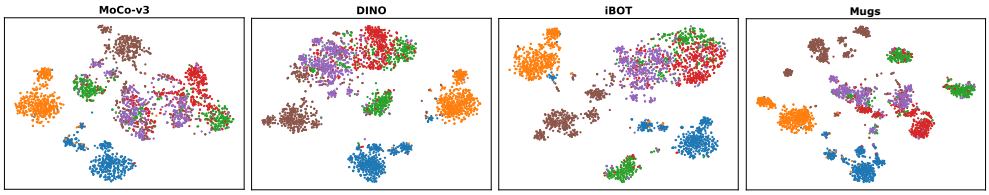
KNN. Table 1 shows that for all backbones, Mugs achieves the highest top-1 accuracy on ImageNet-1K. It respectively makes 0.4%, 0.9%, and 2.3% improvement on ViT-S, ViT-B and ViT-L over the runner-up, showing the advantages of multi-granular representation in Mugs.

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 2 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 prob-

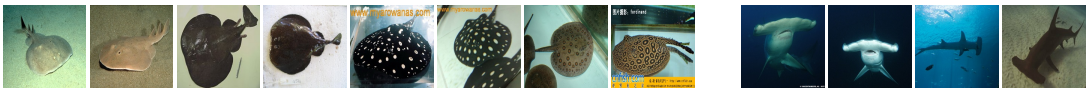
Table 9: **Linear probing accuracy (%) and k-NN accuracy (%)** on ImageNet-1K without multi-crop augmentation (left) and with multi-crop augmentation (right). "Epo" is the effective pretraining epochs adjusted by number of views processed by the models following [43].

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**



(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 2: 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.

ing performance and thus are included in Table 1. Moreover, 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 3 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.

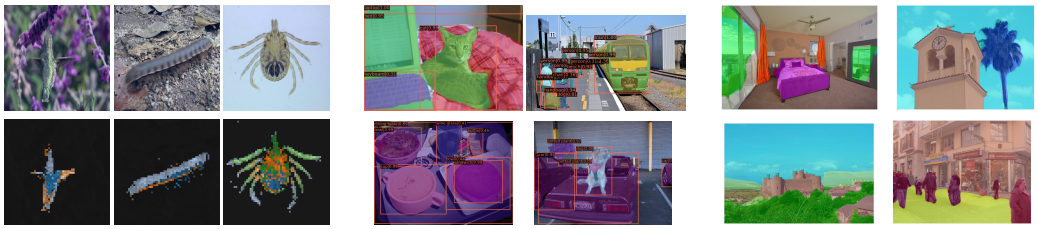
Result Analysis. Fig. 2 uses T-SNE [35] to reveal the feature differences among MoCo-v3, DINO, iBOT, and Mugs, in which each color denotes a unique class. The last subfigure in Fig. 2 (a) shows that for one class, Mugs often divides it into several clusters in the feature space, *e.g.* 4 clusters for brown, 4 for purple, 6 for red, and 5 for blue, and scatters these small clusters in a big class. We further visualize two clusters of Mugs in Fig. 2 (b) and (c): the four clusters in (b) of electric ray ("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 contrast, MoCo-v3, DINO and iBOT often do not show this multi-granular feature structure in Fig. 2 (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. 3 (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 B.5.

B.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 4 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 4 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 [27]. For fairness, we use the same protocol in iBOT. Besides SSL approaches, *e.g.* MoBY [38], we also compare supervised baselines, Swin-T/7 [28] with similar model size as ViT-S/16. Table 10 shows that on detection, Mugs makes 0.4 AP^b improvement over the runner-up, i.e. iBOT. Fig. 3 (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 [42]. Following iBOT, we stack the task layer in UPerNet [37] and fine-tune



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

Figure 3: Visualization of pretrained ViT-B/16 (a) and ViT-S/16 (b) & (c) by Mugs.

Table 10: **Object detection** (Det.) & **instance segmentation** (ISeg.) on COCO & **semantic seg.** (SSeg.) on ADE20K.

	Arch.	Param.	Det.	ISeg.	SSeg.
			AP^b	AP^m	mIoU
Sup. [43]	Swin-T	29	48.1	41.7	44.5
MoBY [38]	Swin-T	29	48.1	41.5	44.1
Sup. [43]	ViT-S/16	21	46.2	40.1	44.5
iBOT [43]	ViT-S/16	21	49.4	42.6	45.4
Mugs (ours)	ViT-S/16	21	**49.8**	**43.0**	**47.4**

Table 11: **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 [7]	62.3	60.7	63.9
iBOT [43]	62.4	60.8	64.0
Mugs	**63.1**	**61.4**	**64.9**

the whole backbone. Table 10 reports the mean intersection over union (mIoU) on all semantic categories. Mugs consistently outperforms the compared SoTAs by significant 2.0 mIoU. Fig. 3 (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 11 reports the mean region similarity $\mathcal{J}_m$ and mean contour-based accuracy $\mathcal{F}_m$ on the DAVIS-2017 video segmentation dataset [31] by using ViT-B/16. Mugs enjoys better feature transferability than DINO and iBOT even for video segmentation.

B.3 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 1 in the manuscript, and well demonstrate the superiority of Mugs over previous state-of-the-arts.

Table 12: 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

B.4 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 13. This setting allows us to optimize the pretrained backbone with a linear classifier. Following BEiT [2], 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

Table 13: **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 [43].

	Method	ViT-L/16 Epo.	ViT-L/16 Acc. (%)
	Supervised [34]	—	83.1
reconstruction	BEiT [2]	800	85.2
	MAE [22]	1600	85.9
	data2vec [1]	1600	**86.6**
contrastive or clustering	DINO [7]	—	—
	iBOT [43]	1000	84.8
	MoCo-v3 [14]	600	84.1
	Mugs (ours)	1000	85.2

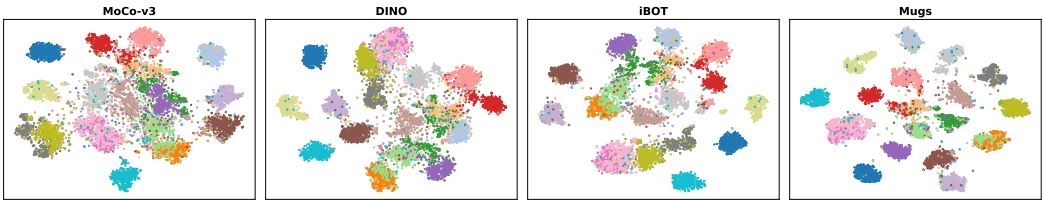
constrictive or clustering learning approaches. Specifically, the reconstruction family always use 224×224-sized images to pretrain the model, while constrictive or clustering learning approaches uses multi-crop augmentations which contains two 224-sized images and ten 96-sized images. Since "Epo." in Table 13 is the effective pretraining epochs adjusted by number of views processed by the models [43] 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 13, 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 1 in manuscript and Table 13 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 semantics of the vanilla image and lead to over-fitting issue, especially under insufficient pretraining epochs. Note, 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.

B.5 More T-SNE Visualization Results

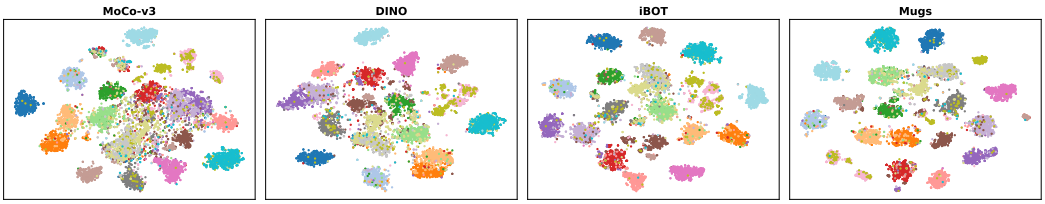
Same with Fig. 2 in the manuscript, here we use T-SNE [35] to reveal the feature differences among MoCo-v3, DINO, iBOT, and Mugs in Fig. 5. By comparison, Mugs often can scatter the samples from different classes more separately, while keeping the samples in the same class close in the feature space. 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.

B.6 More Attention Visualization Results

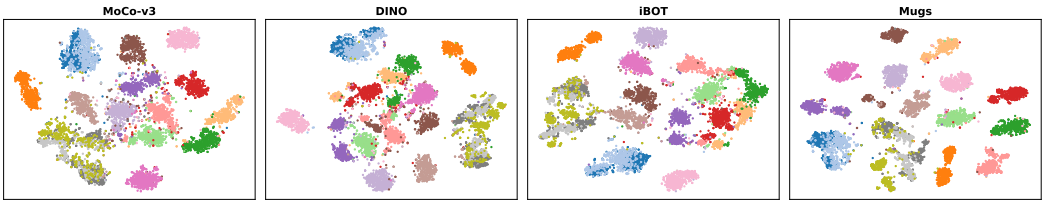
Here same with Fig. 3 in the manuscript, we visualize more self-attention map of the 12 self-attention heads in ViT-B/16 pretrained by Mugs in Fig. 6. The first column denotes the vanilla images, while each column of the last 12 columns denote the self-attention score maps of each individual head. The 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



(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.**

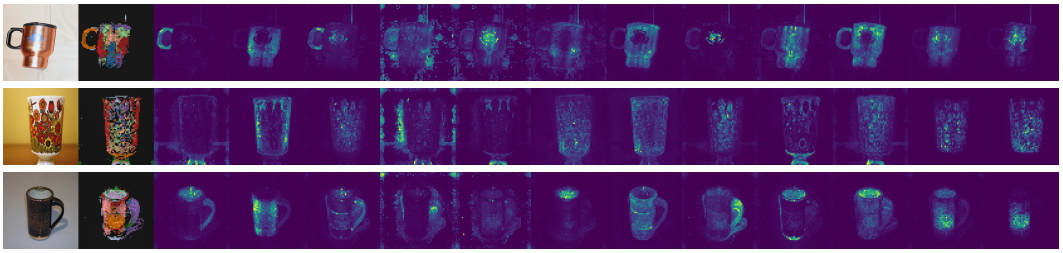


Figure 6: Self-attention visualization of ViT-B/16 pretrained by our Mugs. The images from left to right respectively denote the vanilla image, the overall self-attention score of all 12 heads in ViT-B, and the individual self-attention score of 12 heads. **Best viewed in color pdf file.**

the object shapes very well. For example, from the first bird image, it is even hard for human to get the 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 [43], Fig. 18 in their appendix also visualizes the self-attention map. By comparison, the 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.

B.7 More Visualization Results on Object Detection and Semantic Segmentation

In the manuscript, we already provide some object detection and segmentation examples in Fig. 3. 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

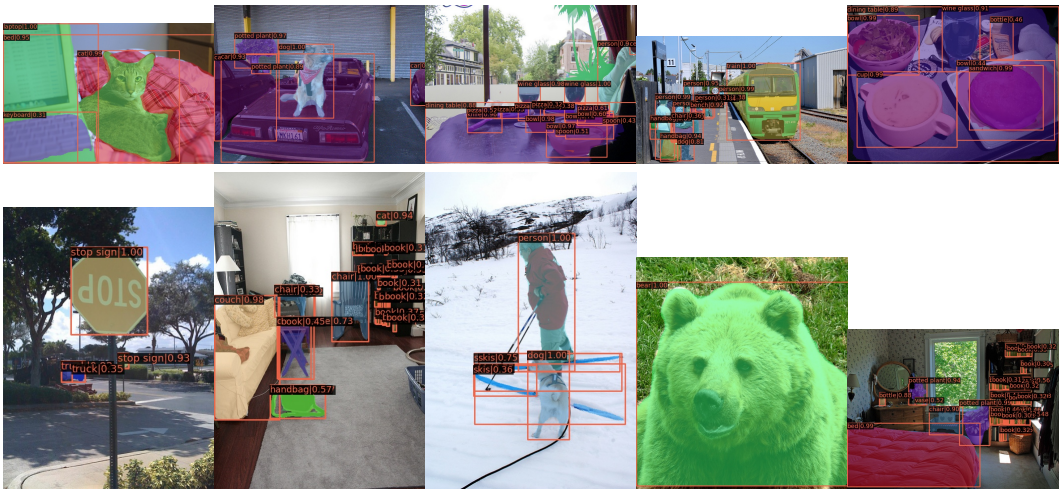


Figure 7: Object detection visualization of ViT-B/16 pretrained by Mugs. **Best viewed in color pdf file.**

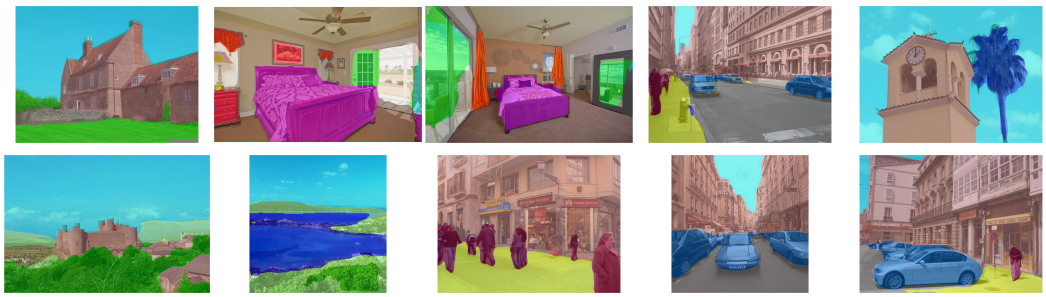


Figure 8: Semantic segmentation visualization of ViT-B/16 pretrained by Mugs. **Best viewed in color pdf file.**

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.