Supplementary material: Exploring the Limits of Deep Image Clustering using Pretrained Models

Nikolas Adaloglou*
adaloglo@hhu.de
Felix Michels*
felix.michels@hhu.de
Hamza Kalisch
hakal104@hhu.de
Markus Kollmann
markus.kollmann@hhu.de

Heinrich Heine University
Düsseldorf, Germany

1 Proof of Theorem 1

We will first show that

$$q^*(c|x) = \arg\max_{q(c|x)} \mathbb{E}_{x,x' \sim p(x,x')}[\text{pmi}(x,x')], \tag{1}$$

is bounded and leads to the correct joint distribution.

Lemma 1. *The mutual information*

$$\text{I}(x;x') = \int p(x,x') \log \frac{p(x,x')}{p(x)p(x')} \text{d}x\text{d}x'$$

is an upper bound for the expected pointwise mutual information. In particular,

$$\begin{aligned} &\mathbb{E}_{x,x' \sim p(x,x')}[\text{pmi}(x,x')] \\ =&\text{I}(x;x') - \text{KL}(p(x,x') \parallel q(x,x')), \end{aligned}$$

where KL *is the Kullback–Leibler divergence.*

Proof.

$$\begin{aligned} & \mathbb{E}_{x,x' \sim p(x,x')}[\text{pmi}(x,x')] \\ = & \int p(x,x') \log \frac{q(x,x')}{p(x)p(x')} \text{d}x\text{d}x' \\ = & \int p(x,x') \log \left(\frac{p(x,x')}{p(x)p(x')} \frac{q(x,x')}{p(x,x')} \right) \text{d}x\text{d}x' \\ = & \int p(x,x') \log \frac{p(x,x')}{p(x)p(x')} \text{d}x\text{d}x' \\ & - \int p(x,x') \log \frac{p(x,x')}{q(x,x')} \text{d}x\text{d}x' \\ = & \text{I}(x;x') - \text{KL}(p(x,x') \parallel q(x,x')). \end{aligned}$$

Lemma 1 already states that the objective is maximized if and only if $q^*(x,x') = p(x,x')$ and therefore

$$\begin{aligned}\mathrm{pmi}(x,x') &= \sum_{c=1}^{C} \frac{q^*(c|x)q^*(c|x')}{q^*(c)} \\ &= \frac{q^*(x,x')}{p(x)p(x')} = \frac{p(x,x')}{p(x)p(x')} = [c_x = c_{x'}]p(c_x)^{-1}.\end{aligned}$$

If $c_x \neq c_{x'}$, we have

$$0 \leq q^*(\hat{c}_x|x)q^*(\hat{c}_x|x')/q^*(\hat{c}_x) \leq \mathrm{pmi}(x,x') = 0.$$

Since $q^*(\hat{c}_x|x) > 0$, this implies $q^*(\hat{c}_x|x') = 0$ and therefore $\hat{c}_x \neq \hat{c}_{x'}$. Furthermore, from the pigeonhole principle it follows that $q^*(c|x) = 0$ for $c \neq c_x$ which both implies that $q^*(c|x)$ is one-hot as well as $\hat{c}_x = \hat{c}_{x'}$ if $c_x = c_{x'}$, therefore concluding the proof. □

2 Further discussion points

2.1 Fine-tuning the pretrained backbone with TEMI

Given a pretrained backbone network, fine-tuning the backbone simultaneously with training randomly initialized heads gave bad results. However, fine-tuning the backbone simultaneously with fine-tuning the already trained head with TEMI, yielded superior performance but only when the pretraining dataset was different from the downstream dataset, e.g. 67.1 → 70.9 for CIFAR100 using DINO ViT-B/16 pretrained on ImageNet as the backbone model.

2.2 Additional computational complexity from multiple heads

In theory, the computational time complexity of TEMI by adding multiple heads is linear given a sequential implementation. In practice, due to GPU-related optimizations, it's much faster. In fact, training on a single Nvidia A100 GPU takes only 4 GB of memory with 50 heads on CIFAR100 and training takes just about 45 minutes because we precompute the feature representations, while training with just one head takes about 5 minutes.

2.3 Are multiple heads necessary?

The idea of using multiple heads is inspired by previous works, such as SCAN [5] and SSCN [1]. The proposed PMI objective does not require multiple heads by design. As shown in Table 3, we experimentally observed an initial gain of 0.8% by adding independent heads (PMI and WMI setup with 50 heads). Importantly, one of our core novelties lies in the combination of the teacher predictions from multiple heads, Eq. (10) in the main text, which provides superior results compared to having independent heads (Table 3 in the main text). Overall, we find the reported performances saturate quickly with more heads and are already close to the maximum for 16 heads on CIFAR100. Based on our first results on CIFAR100, we fixed the number of heads to 50 for all models and datasets.

2.4 Contrastive versus non-contrastive self-supervised pretraining for image clustering.

The performance gap between contrastive (MoCoV3 ViT-B) and non-contrastive (DINO ViT-B) backbones likely originates from the homogeneous distribution of examples in feature space as part of the contrastive learning objective, which likely attenuates the necessary structure in feature space for image clustering [3, 6].

3 Additional implementation details.

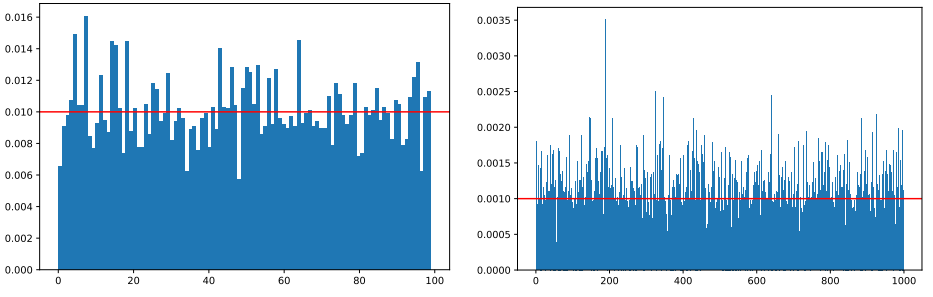
To enforce reproducibility, the means and standard deviations are reported for all our experiments and metrics, computed over 3 independent runs with different seeds. For a fair comparison with SCAN, we tune its entropy regularization hyperparameter, λ, based on a grid search and use the value $\lambda = 4$. Crucially, we found that some pretrained models (i.e. MSN) produce unnormalized features. For that reason, we standardize the features of all models before feeding them to the clustering heads. For the linear probing experiments, we trained a linear layer using the Adam [9] optimizer with a learning rate of 10^{-3} and weight decay of 10^{-3}.

3.1 How to choose β for a new dataset?

Here we provide a more detailed explanation of Fig. 3 (in the main text) on how to pick $\beta \in (0.5, 1]$ without access to ground-truth data. First, the motivation behind β is to avoid the imbalanced growth of clusters during training. The closer β is to 0.5 the more balanced the clusters (clusters contain a similar number of examples). The reason is that the loss contribution to assign each training sample a single class is reduced for smaller β. However, for $\beta = 0.5$, each sample occupies all clusters with equal probability. Consequently, we have to impose $\beta > 0.5$ but β should be sufficiently close to 0.5. We take for β the value when the conditional entropy, $E_x[\sum_c -q(c|x)logq(c|x)]$ (Fig. 3 in the main text, green line), is starting to become constantly low. We experimentally found 0.6 to work consistently well across models and datasets. An exception is CIFAR20, where we used $\beta = 0.55$ since superclasses are conceptually a form of under-clustering.

3.2 A Note on CIFAR100 VS CIFAR20

We observe that previous works have established the CIFAR20 as a clustering benchmark. However, we believe that the CIFAR20 superclasses are not an ideal benchmark for image clustering. In the reported results, one can easily notice that all models perform worse in CIFAR20 than in CIFAR100. Examples that justify the performance gap include a) clocks, computer keyboards, lamps, telephones, and televisions are grouped into household electrical devices, b) bridges, castles, houses, roads, and skyscrapers are grouped into large man-made outdoor things, and c) bears, leopards, lions, and wolfs are grouped into carnivores. These examples illustrate that the superclasses are not separable from the pixel information alone. To this end, we would like to encourage future works to adopt CIFAR100 as a benchmark for image clustering.



(a) CIFAR100 using TEMI DINO ViT-B/16 (b) ImageNet using TEMI MSN ViT-L/16

Figure 1: **Histogram of cluster assignments on different datasets**. The horizontal red line illustrates the ideal histogram, where all clusters would be uniformly utilized. We also compute the KL divergence between the predictions and the uniform distribution on CIFAR100 and ImageNet, which is $1.5 \cdot 10^{-2}$ and $5 \cdot 10^{-2}$, respectively. The predictions would be uniform in the extreme case where the KL divergence is 0.

Methods	NMI(%)	ACC(%)	ARI(%)
TEMI DINO ViT-B/16	**76.9±0.45**	**67.1±1.30**	**53.3±1.02**
TEMI MSN ViT-L/16	73.0±0.20	61.4±0.16	47.4±0.42
(natural language) supervised pretraining			
TEMI CLIP ViT-L/14	79.9±0.23	73.7±0.92	61.2±0.75
TEMI Sup. ViT-L/16	85.2±0.34	81.8±0.73	70.6±0.89
supervised baselines			
Probing DINO ViT-B/16	85.7	85.3	73.6
Probing MSN ViT-L/16	84.6	84.4	71.9
Probing CLIP ViT-L/14	87.4	87.1	76.5
Probing Sup. ViT-L/16	86.0	86.3	75.0

Table 1: **Clustering performance metrics on on the CIFAR100 dataset.** All methods use models pretrained on external data.

Datasets	ImageNet		CIFAR100	
Methods	TEMI	k-means	TEMI	k-means
self-supervised methods				
MAE ViT-B/16	9.09±0.05	4.93	7.78±0.10	7.11
MAE ViT-L/16	27.81±0.13	12.45	19.56±0.17	12.05
MAE ViT-H/16	22.34±0.11	10.18	17.64±0.19	11.31
MOCOv3 ViT-S/16	16.73±0.19	12.23	16.58±0.16	13.63
MOCOv3 ViT-B/16	54.10±0.08	47.64	63.51±0.53	49.94
DINO Resnet50	45.20±0.23	32.07	45.34±0.41	34.21
DINO ViT-S/16	56.84±0.25	51.84	61.69±0.75	50.17
DINO ViT-B/16	58.08±0.26	52.26	**67.11±1.30**	**57.01**
MSN ViT-S/16	58.53±0.39	55.58	63.06±0.89	49.96
MSN ViT-B/16	60.82±0.06	57.56	65.57±1.23	50.60
MSN ViT-L/16	**61.56±0.28**	**58.08**	61.40±0.15	54.08
natural language supervised methods				
CLIP Resnet50	45.93±0.11	34.41	34.06±0.72	25.96
CLIP ViT-B/16	56.68±0.24	45.86	60.74±0.79	45.84
CLIP ViT-L/14	**63.99±0.38**	**54.12**	**73.70±0.92**	**54.55**
supervised methods				
Resnet50	72.60±0.18	65.69	49.77±0.43	40.28
ConvNext S	77.67±0.41	71.85	57.31±0.20	43.19
ConvNext B	78.23±0.12	73.67	58.31±0.76	43.20
ConvNext L	**79.77±0.20**	**76.98**	59.43±0.24	47.94
ViT-S/16	64.72±0.14	60.32	60.60±0.97	50.65
ViT-B/16	69.23±0.27	64.48	63.36±0.43	51.72
ViT-L/16	77.12±0.21	74.91	**81.77±0.73**	**70.06**

Table 2: **Benchmarking various models with the introduced objective versus k-means.** We report the clustering accuracy (ACC) in %

Datasets	ImageNet 50			ImageNet 100			ImageNet 200		
Methods	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)
SCAN (Resnet50)	82.2	76.8	66.1	80.8	68.9	57.6	77.2	58.1	47.0
Propos (Resnet50)	82.8	-	69.1	83.5	-	63.5	80.6	-	53.8
TEMI DINO ViT-B/16	86.10±0.54	80.01±1.26	70.93±1.24	85.65±0.30	75.05±1.11	65.45±1.11	85.20±0.21	73.12±0.72	62.13±0.59
TEMI MSN ViT-L/16	**88.14±0.55**	**84.87±1.16**	**76.46±1.17**	**88.53±0.56**	**82.86±0.73**	**74.08±1.20**	**86.65±0.32**	**77.96±0.71**	**66.70±0.71**
(natural language) supervised pretraining									
TEMI CLIP ViT-L/14	92.32±0.38	88.27±0.53	82.78±0.94	90.06±0.53	83.43±1.98	75.81±1.36	88.39±0.16	77.76±0.37	69.41±0.23
TEMI Sup. ViT-L/16	95.75±0.60	95.12±1.61	91.40±1.88	94.95±0.21	92.50±0.23	87.95±0.31	93.94±0.02	90.37±0.14	84.05±0.09
supervised baselines									
Probing DINO ViT-B/16	95.10	95.76	91.64	93.29	92.74	86.30	91.64	89.48	80.61
Probing MSN ViT-L/16	94.21	94.92	90.03	93.00	92.42	85.74	91.36	89.02	79.88
Probing CLIP ViT-L/14	98.72	98.96	97.88	96.61	96.16	92.73	95.09	93.57	88.00
Probing Sup. ViT-L/16	97.77	98.12	96.21	96.13	95.76	91.90	95.07	93.60	88.02

Table 3: **Clustering performances on ImageNet subsets.** All subsets were evaluated on their respective validation splits, as detailed in Table 4.

Dataset	Classes	Train images	Val images	Size
CIFAR10	10	50,000	10,000	32×32
CIFAR100	100	50,000	10,000	32×32
CIFAR20	20	50,000	10,000	32×32
STL10	10	5,000	8,000	96×96
ImageNet-50	50	64,274	2,500	224×224
ImageNet-100	100	128,545	5,000	224×224
ImageNet-200	200	256,558	10,000	224×224
ImageNet	1000	1,281,167	50,000	224×224

Table 4: **An overview of the number of classes and the number of samples on the considered datasets.** The train set is used for training, while the validation split is used to compute the clustering performance metrics. The selected classes on the ImageNet [2] subsets (ImageNet-50, ImageNet-100, and ImageNet-200) can be found in SCAN [5].

config	value
optimizer	AdamW
base learning rate	10^{-4}
weight decay	10^{-4}
optimizer momentum	β_1, β_2=0.9, 0.999
batch size	512, 1024 (ImageNet)
learning rate schedule	constant
softmax temperature τ	0.1
β	0.6, 0.55 (CIFAR20)
cluster heads	50
warmup epochs	20, 10 ImageNet
training epochs	200, 800 (STL10)
teacher momentum	0.996
augmentation	None

Table 5: **Hyperparameters for training the clustering heads.**

config	value
optimizer	Adam
learning rate	10^{-3}
weight decay	10^{-3}
optimizer momentum	β_1, β_2=0.9, 0.999
batch size	256
learning rate schedule	cosine decay
training epochs	100
augmentation	None

Table 6: **Hyperparameters for linear probing.**

4 Randomly sampled images

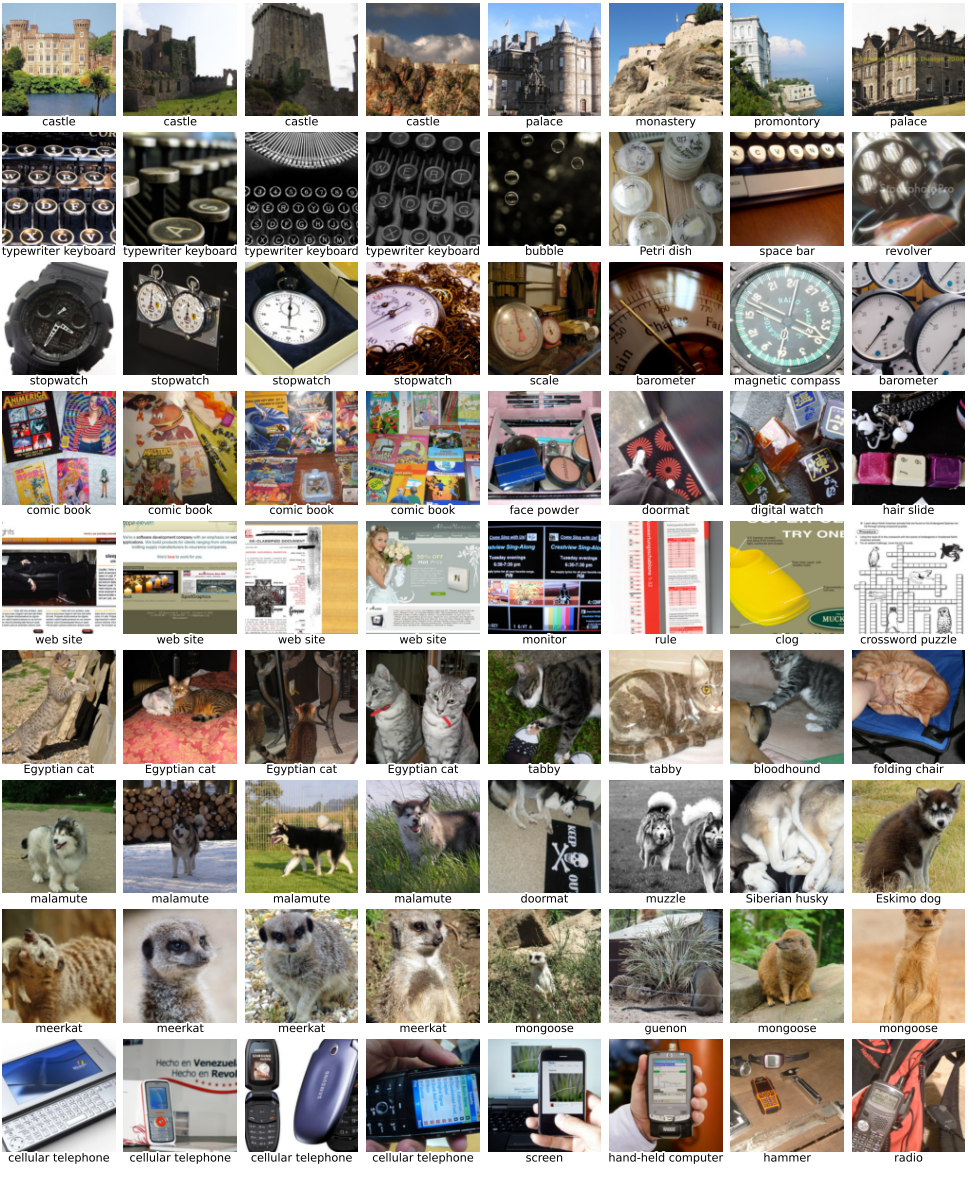


Figure 2: **Randomly sampled images from the ImageNet dataset that are assigned in the same cluster using the TEMI MSN ViT-L/16 model**. The ground-truth label is indicated in the text under the image. The images in each row are assigned to the same cluster. The first four columns correspond to correctly classified images while the last four are examples of misclassified images.

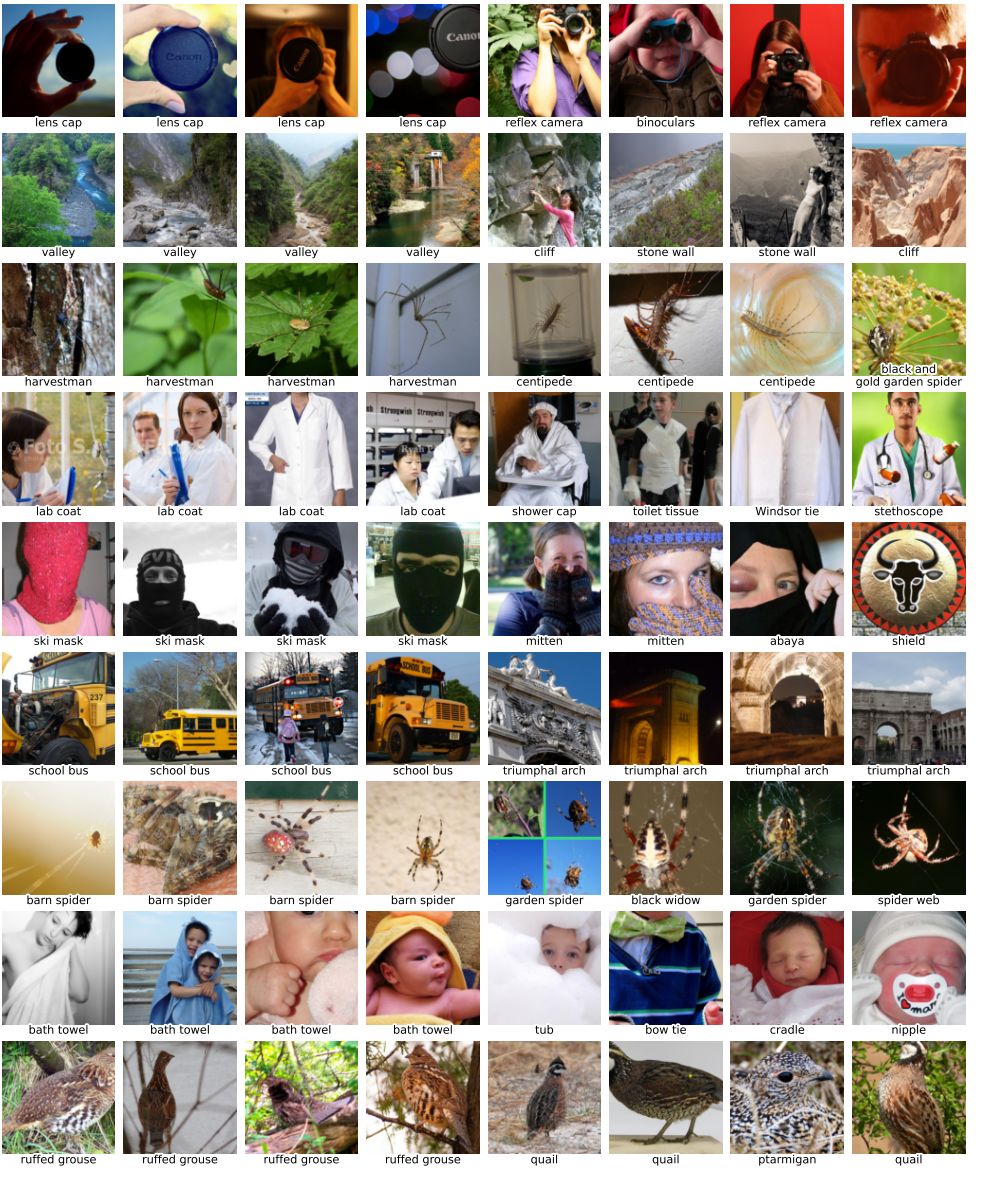


Figure 3: **More randomly sampled images from the ImageNet dataset that are assigned in the same cluster.**

References

[1] Elad Amrani, Leonid Karlinsky, and Alex Bronstein. Self-supervised classification network. In *European Conference on Computer Vision*, pages 116–132. Springer, 2022.

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

[3] Zhizhong Huang, Jie Chen, Junping Zhang, and Hongming Shan. Learning representation for clustering via prototype scattering and positive sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[4] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[5] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European conference on computer vision*, pages 268–285. Springer, 2020.

[6] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.