HCSC: Hierarchical Contrastive Selective Coding
Supplementary Material

1. More Implementation Details

Linear classification. For performance comparison, we follow the learning configuration of PCL [12] to train the linear classifier with an SGD optimizer (weight decay: 0; momentum: 0.9; batch size: 256) for 100 epochs. The learning rate is initialized as 5.0 and decayed by a factor of 0.1 at the 60th and 80th epoch.

KNN evaluation. We follow NPID [18] to design a KNN classifier which predicts the label of each sample by aggregating the labels of its nearest neighbors. Specifically, given a test image x, we first extract its embedding z using the pre-trained encoder. This embedding vector is compared against the embeddings of all other images in the dataset, and a cosine similarity score $s_{\cos}(z, z_i)$ is computed for each image pair. According to these similarity scores, we select the top K nearest neighbors of the test image, denoted as $\mathcal{N}_K(x)$. On such basis, we compute the unnormalized likelihood $p_c(x)$ that the test image belongs to class c via a weighted voting:

$$p_c(x) = \sum_{x_i \in \mathcal{N}_K(x)} \mathbb{1}(y_i = c) \exp\left(s_{\cos}(z, z_i)/\tau_{\text{KNN}}\right), \quad (1)$$

where $\mathbb{1}(y_i = c)$ is an indicator function judging whether the sample x_i belongs to class c, and the temperature parameter τ_{KNN} is set as 0.07 following NPID. Based on these likelihoods, the KNN classifier predicts the category of x as $y = \arg\max_{c \in C} p_c(x)$. As in NPID, the final result of KNN evaluation is reported as the highest classification accuracy over $K \in \{10, 20, 100, 200\}$.

Semi-supervised learning. In this experiment, we follow NPID [18] to fine-tune the image encoder and linear classifier with an SGD optimizer (weight decay: 0; momentum: 0.9; batch size: 256) for 70 epochs. The learning rate is initialized as 0.005 and decayed by a factor of 0.1 at the 30th and 60th epoch.

Transfer learning. This experiment involves two types of transfer learning tasks, *i.e.* object classification and object detection. We strictly follow the fine-tuning paradigms of MoCo [7] on these two types of tasks.

For object classification, our model is evaluated on PASCAL VOC [6] and Places205 [20] datasets. We follow the standard dataset splits of VOC07 and Places205 to perform training and testing. On both datasets, following SwAV [2], we keep the pre-trained encoder fixed and learn a linear layer for classification. On PASCAL VOC, the linear classifier is trained for 100 epochs by an SGD optimizer (weight decay: 0; momentum: 0.9; batch size: 16), and the initial learning rate of 0.05 is adjusted by a cosine annealing scheduler [14]. On Places205, we train the linear classifier with an SGD optimizer (weight decay: 0; momentum: 0.9; batch size: 256) for 100 epochs, and the initial learning rate of 3.0 is adjusted by a cosine annealing scheduler.

For object detection, we evaluate our model on PASCAL VOC [6] and COCO [13] datasets. On PASCAL VOC, the training and validation splits of VOC07+12 is used for training, and the test split of VOC07 is used for evaluation. Faster-RCNN-C4 [15] serves as the object detector. We initialize its ResNet-50 backbone with the weights pre-trained by our HCSC approach, and the whole detection model is fine-tuned for 24,000 iterations by an SGD optimizer (weight decay: 1×10^{-4}; momentum: 0.9; batch size: 16). The initial learning rate of 0.02 is warmed up for 100 iterations and decayed by a factor of 0.1 at the 18,000th and 22,000th iteration. On COCO, the detection model is trained on the train2017 subset for 180,000 iterations, and it is then evaluated on the val2017 subset. An identical SGD optimizer as in PASCAL VOC experiment is employed, and the initial learning rate of 0.02 is warmed up for 100 iterations and decayed by a factor of 0.1 at the 120,000th and 160,000th iteration.

Clustering evaluation. Following PCL [12], the clustering evaluation with 25,000 and 1,000 clusters are respectively performed. For the experiment using 25,000 clusters, we train an HCSC model with three prototype hierarchies 25000-10000-1000, and the bottom hierarchy with 25,000 prototypes are used for evaluation. For the experiment using 1,000 clusters, an HCSC model with three prototype hierarchies 3000-2000-1000 is trained, and we utilize the top hierarchy with 1,000 prototypes for evaluation.

2. More Results of Linear Classification

We notice that the fine-tuning configuration vary across previous works when performing linear classification on

Table 1. Performance comparison on linear classification under different learning configurations.

Method	Config	Initial lr	Scheduler	Top1-Acc
PCL v2 [12]	PCL [12]	5.0	step(0.1, [60,80])	67.6
HCSC	PCL [12]	5.0	step(0.1, [60,80])	**69.2**
AdCo [9]	AdCo [9]	10.0	cosine	68.6
HCSC	AdCo [9]	10.0	cosine	**68.9**
MoCo v2 [4]	MoCo v2 [4]	30.0	step(0.1, [60,80])	**67.5**
HCSC	MoCo v2 [4]	30.0	step(0.1, [60,80])	67.3

ImageNet [5]. Therefore, in Tab. 1, we further evaluate our HCSC model under the configurations from three different works, *i.e.* PCL [12], AdCo [9] and MoCo v2 [4]. Under the learning configurations of PCL and AdCo, the performance difference of HCSC is merely 0.3%, and it outperforms these two approaches on their respective configurations. These results verify the robustness of our method when varying the initial learning rate between 5.0 and 10.0 and changing between a step scheduler decaying twice and a cosine annealing scheduler. On the configuration of MoCo v2, HCSC suffers an obvious performance decrease and performs worse than MoCo v2. This negative result illustrates that too high initial learning rate, like 30.0 in MoCo v2's configuration, will hamper the effectiveness of HCSC during downstream fine-tuning.

3. Zero-Shot Classification on CUB

In this section, we study a more difficult transfer learning problem, *i.e.* directly transferring the encoder learned on ImageNet [5] to a fine-grained classification dataset, Caltech-UCSD-Birds (CUB) [17], without learning a task-specific classifier. Therefore, this problem can be regarded as a **cross-domain zero-shot classification** problem, and it evaluates whether a self-supervised learning method can capture fine-grained semantic structures by pre-training on a general-purpose database, like ImageNet.

Evaluation details. We evaluate model's zero-shot classification performance on CUB with the standard KNN evaluation protocol. Specifically, a KNN classifier is employed to predict the label of each sample by aggregating the labels of its nearest neighbors. The implementation details of such a KNN classifier is specified in the KNN evaluation part of Sec. 1. We report the highest accuracy of the KNN classifiers over $K \in \{10, 20, 100, 200\}$, which follows NPID [18].

Results. Tab. 2 presents the performance comparison among different approaches on this task. Under both the configurations with and without multi-crop augmentation, HCSC clearly outperforms other baseline methods. This superior performance demonstrates that, by pre-training with HCSC, the image encoder can well capture the fine-grained semantic structures underlying an image dataset, and such a capability can even be transferred to other datasets.

Table 2. Performance comparison on zero-shot classification. This experiment transfers the encoder learned on ImageNet to CUB.

Method	KNN-Top1-Acc
MoCo [7]†	19.5
MoCo v2 [4]†	23.1
SimCLR [3]†	23.9
PIC [1]†	18.2
PCL v2 [12]†	22.3
AdCo [9]†	22.9
HCSC	**26.9**
SwAV* [2]†	26.2
AdCo* [9]†	30.6
HCSC*	**31.5**

* With multi-crop augmentation.
† Evaluated by us with officially released model weights.

Table 3. Performance of models under different training epochs. The results are reported on linear and KNN evaluation.

Method	Epochs	Batch size	Top1-Acc	KNN-Top1-Acc
NPID [18]	200	256	58.5	46.8
LocalAgg [21]	200	128	58.8	-
MoCo [7]	200	256	60.8	45.0†
SimCLR [3]	200	256	61.9	57.4†
MoCo v2 [4]	200	256	67.5	55.8†
CPC v2 [16]	200	512	67.6	-
PCL v2 [12]	200	256	67.6	58.1†
PIC [1]	200	512	67.6	54.7†
MoCHi [11]	200	512	67.6	57.5†
DetCo [19]	200	256	68.6	58.9†
AdCo [9]	200	256	68.6	57.2†
HCSC	200	256	**69.2**	**60.7**
SwAV* [2]	200	256	72.7	62.4†
AdCo* [9]	200	256	73.2	66.3†
HCSC*	200	256	**73.3**	**66.6**
DeepCluster-v2 [2]	400	4096	70.2	62.4†
SeLa-v2 [2]	400	4096	67.2	57.9†
SwAV [2]	400	4096	70.1	61.3†
HCSC	400	256	**71.0**	**64.1**
DeepCluster-v2* [2]	400	4096	74.3	66.0†
SeLa-v2* [2]	400	4096	71.8	61.7†
SwAV* [2]	400	256	74.3	64.3†
SwAV* [2]	400	4096	**74.6**	65.0†
HCSC*	400	256	74.1	**69.9**
MoCo v2 [4]	800	256	71.1	61.8†
HCSC	800	256	**72.0**	**64.5**

* With multi-crop augmentation.
† Evaluated by us with officially released model weights.

4. Model Zoo

To make this project a more solid contribution, we train a comprehensive set of models, including longer training epochs, single- and multi-crop settings and more backbone architectures, and we will continually release corresponding

Table 4. Per-epoch running time comparison (batch size: 256).

Method	*w/o* multi-crop	*w/* multi-crop
SwAV [2]	27min 53s	44min 30s
HCSC (non-parallel)	26min 22s	44min 59s
HCSC (parallel)	**21min 11s**	**39min 22s**

codes and model weights to the community.

4.1. Models of Longer Training

In Tab. 3, we give comprehensive comparisons among various methods under different training epochs, and this table will be continually extended according to our progress on training longer epochs models. The current results show that, our HCSC method preserves its superiority over previous state-of-the-art approaches on 400 and 800 epochs training for both *w/* and *w/o* multi-crop augmentation.

4.2. Models with Different Architectures

This part of works are in progress.

5. Time Complexity Analysis

HCSC involves an extra hierarchical K-means step for each epoch. Here, we compare it with another clustering-based method, SwAV [2]. In each training step, SwAV performs three iterations of Sinkhorn-Knopp algorithm to update clustering assignments, which has a time complexity of $\mathcal{O}(KM)$ (K: batch size; M: number of prototypes). After amortizing the cost of hierarchical K-means to all training steps within an epoch, our HCSC method has an extra time complexity of $\mathcal{O}(NM_1+M_1M_2+M_2M_3)/T = \mathcal{O}(KM_1)$ for each step (N: dataset size; M_l: number of prototypes at the l-th hierarchy; T: training steps per epoch). Therefore, when it holds that $M \approx M_1$, SwAV and HCSC have comparable extra computation. In the first two rows of Tab. 4, we compare the per-epoch running time of SwAV (with 3000 prototypes) and the vanilla HCSC (with 3000-2000-1000 hierarchical prototypes), which makes $M = M_1$. The comparable cost of time supports the analysis above.

To further enhance the efficiency of HCSC, we employ faiss [10], a library for efficient similarity search and clustering, to perform the hierarchical K-means step. Thanks for the high parallelism of faiss, the improved HCSC model, *i.e.* HCSC (parallel), achieves much better computational efficiency than the vanilla HCSC, *i.e.* HCSC (non-parallel), as shown in the last two rows of Tab. 4.

6. Analysis on Contrastive Selective Coding

Here, we analyze the proposed instance-wise and prototypical contrastive selective coding from two perspectives: (1) how it can select more diverse positive pairs with similar semantics, and (2) how it can select more precise negative pairs with truly distinct semantics. Though the pre-training stage is unsupervised, the labels and label hierarchies of the pre-training database, ImageNet [5], are publicly available to enable us to perform this analysis.

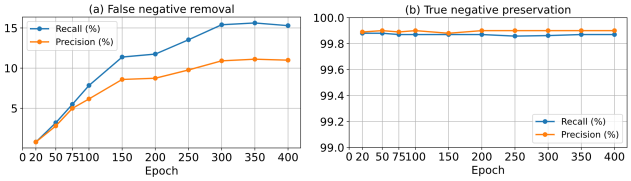

Figure 1. Performance of our negative sample selection scheme.

Table 5. Adjusted Mutual Information (AMI) between prototypes and ImageNet labels on the 1st, 2nd and 3rd label hierarchy (count from bottom to top).

Prototype Config	1st hierarchy	2nd hierarchy	3rd hierarchy
6000	0.543	0.535	0.506
3000-2000-1000	**0.582**	**0.588**	**0.566**

6.1. Analysis on Positive Pair Selection

In this study, we aim to verify that our method can better include **images and their corresponding prototypes at higher ImageNet label hierarchy** as positive pairs. In Tab. 5, we report the adjusted mutual information (AMI) between prototypes and the ImageNet labels at three hierarchies. Compared with the prototypes with a single hierarchy, the prototypes with three hierarchies can better capture the semantics on all three label hierarchies. Hence, the positive image-prototype pairs selected based on our hierarchical prototypes are more semantically diverse.

6.2. Analysis on Negative Sample Selection

This study seeks to measure the effectiveness of our negative sample selection scheme. In Fig. 1, we plot the precision and recall of false negatives and true negatives along training. This recording shows **stably growing false negative removal** and **constantly high true negative preservation**, which verifies that the proposed scheme can keep most of the correct negative samples and, at the same time, eliminate more and more false negatives as the representation quality improves.

7. More Visualization Results

7.1. Visualization of Hierarchical Semantics

In Fig. 2, we visualize the images assigned to the prototypes in a substructure of hierarchical prototypes. The semantics of the images assigned to the prototype at top hierarchy are most diverse, which represents the coarse-grained semantics of "human interacting with animals or items". By comparison, the images assigned to the prototypes at bottom hierarchy express finer-grained semantics, *e.g.* "hu-

man catching snakes", "human interacting with birds" and "human catching fish". These results illustrate that the proposed hierarchical prototypes can indeed capture hierarchical semantic structures.

7.2. Visualization of Feature Representations

In Fig. 3, we use t-SNE [8] to visualize the representations of ImageNet [5] images learned by three methods, *i.e.* MoCo v2 [4], PCL v2 [12] and the proposed HCSC, in which the first 20 classes of ImageNet are visualized following PCL [12]. The image representations learned by MoCo v2 are not separable among many classes. By comparison, PCL v2 derives more separable representations among different classes, while it confuses the image representations of class 7, 19 and 20. HCSC produces more separable feature representations among these three classes, and the representations from all 20 classes are best separated under our approach. These visualization results demonstrate that HCSC can derive discriminative feature representations which benefit various downstream tasks.

Figure 2. Visualization of a typical substructure of hierarchical prototypes.

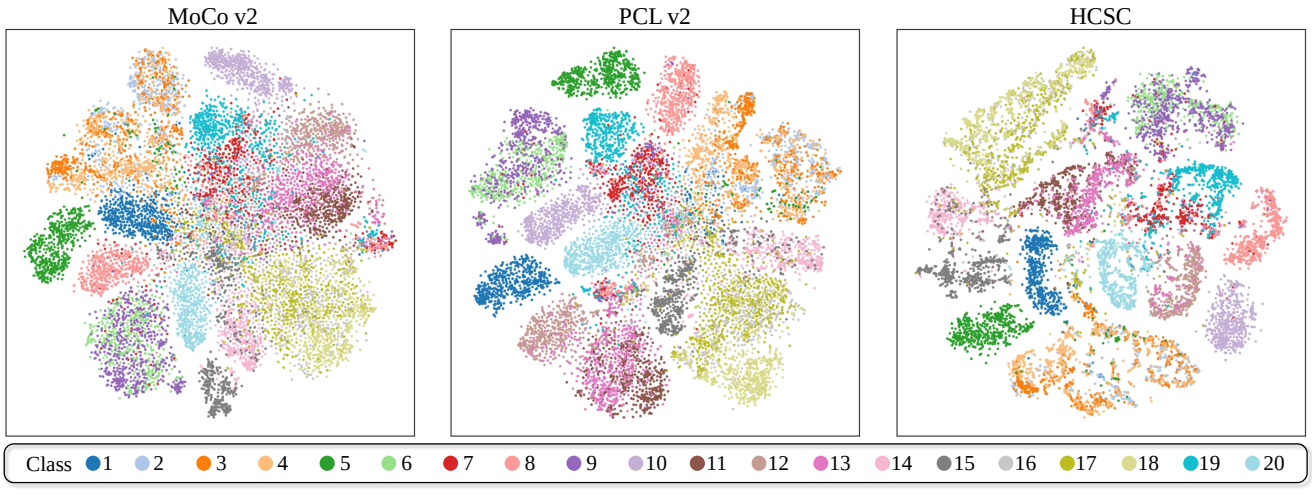

Figure 3. The t-SNE visualization of the learned representations for ImageNet training samples from the first 20 classes.

References

[1] Yue Cao, Zhenda Xie, Bin Liu, Yutong Lin, Zheng Zhang, and Han Hu. Parametric instance classification for unsupervised visual feature learning. *arXiv preprint arXiv:2006.14618*, 2020. 2

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, 2020. 1, 2, 3

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020. 2

[4] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020. 2, 4

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009. 2, 3, 4

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 1

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2

[8] Geoffrey Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems*, 2002. 4

[9] Qianjiang Hu, Xiao Wang, Wei Hu, and Guo-Jun Qi. Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. In *IEEE International Conference on Computer Vision*, 2021. 2

[10] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019. 3

[11] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *arXiv preprint arXiv:2010.01028*, 2020. 2

[12] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven C. H. Hoi. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*, 2021. 1, 2, 4

[13] 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 *European Conference on Computer Vision*. Springer, 2014. 1

[14] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 1

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015. 1

[16] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. 2

[17] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2

[18] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE International Conference on Computer Vision*, 2018. 1, 2

[19] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *IEEE International Conference on Computer Vision*, 2021. 2

[20] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, 2014. 1

[21] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *IEEE International Conference on Computer Vision*, 2019. 2