Supplementary Materials: Unsupervised Feature Learning by Cross-Level Instance-Group Discrimination

Xudong Wang
UC Berkeley / ICSI
xdwang@eecs.berkeley.edu

Ziwei Liu
S-Lab, NTU
ziwei.liu@ntu.edu.sg

Stella X. Yu
UC Berkeley / ICSI
stellayu@berkeley.edu

We provide further details on Kitchen-HC construction, implementation details, and various choices and experiments we have explored to validate our approach.

1. Kitchen-HC Dataset Construction

The original multi-view RGB-D kitchen dataset [7] is comprised of densely sampled views of several kitchen counter-top scenes with annotations in both 2D and 3D. The viewpoints of the scenes are densely sampled and objects in the scenes are annotated with bounding boxes and in the 3D point cloud. Kitchen-HC is constructed from multi-view RGB-D dataset Kitchen by extracting objects in their 2D bounding boxes. The customized Kitchen-HC dataset has 11 categories with highly correlated samples (from different viewing angles) and 20.8K / 4K / 14.4K instances for training / validation / testing. Fig. A.1 shows sample images in the original RGB-D Kitchen dataset from which our Kitchen-HC data are constructed (See samples used in Fig. 1).

Figure A.1: Samples of multi-view RGB-D dataset Kitchen [7]. Instances of the same category captured from different perspectives are highly correlated. The high-correlation dataset Kitchen-HC is constructed from Kitchen by extracting objects in their bounding boxes.

2. Implementation Details

We use SGD as our optimizer, with weight decay 0.0001 and momentum 0.9. We follow MoCo and NPID [17, 9] and use only standard data augmentation methods for experiments on NPID+CLD and MoCo+CLD: random cropping, resizing, horizontal flipping, color and grayscale transformation, unless otherwise noticed.

1. **ImageNet-{100 [15], ILSVRC-2012 [5] , Long-tail [12]}**. For ILSVRC-2012 and ImageNet-LT, we use mini-batch size 256, initial learning rate 0.03, on 8 RTX 2080Ti GPUs. For ImageNet-100, we use batch size 512 and a larger initial learning rate of 0.8 on 8 GPUs, and apply the same setting to baselines and our methods. Training images are randomly cropped and resized to 224×224. For experiments on MoCov2+CLD with an MLP projection head, we extend the original augmentation in [9] by including the blur augmentation and apply cosine learning rate scheduler to further improve the performance on recognition as in [2]. BYOL+CLD is implemented based on OpenSelfSup [21] benchmark. For experiments on InfoMin+CLD and BYOL+CLD, we follow the same training recipe with InfoMin and BYOL [16, 8] for fair comparisons.

2. **CIFAR-{10, 100, 10-LT, 100-LT}, Kitchen-HC**. As [17], we use mini-batch size 256, initial learning rate 0.03 on 1 GPU for CIFAR [11] and Kitchen-HC. The number of epochs is 200 for CIFAR and 80 for Kitchen-HC. Training images are randomly cropped and resized to 32×32.

3. **STL-10 [3]**. Following [18], we use mini-batch size 256, initial learning rate 0.03, on 2 GPUs. Baseline models and baselines with CLD are trained on "train+unlabelled" split (105k samples), and tested on "test" split (5k samples). Training images are randomly cropped and resized to 96×96.

4. **Transfer learning on object detection**. We use Faster R-CNN with a backbone of R50-C4, with tuned syn-

chronized batch normalization layers [13] as the detector. As in [9], the detector is fine-tuned for 24k iterations for the experiment on Pascal VOC *trainval07+12* and 9k iterations for the experiment on Pascal VOC *trainval07*. The image scale is [480, 800] pixels during training and 800 at inference. NPID+CLD and MoCo+CLD use the same hyper-parameters as in MoCo [9]. The VOC-style evaluation metric [6] AP_{50} at IoU threshold is 50% and COCO-style evaluation metric AP are used.

5. **Semi-supervised learning**. To make fair comparisons with baseline methods, we use OpenSelfSup [21] benchmark to implement baseline results and ours. We follow [20] and fine-tune the pre-trained model on two subsets for semi-supervised learning experiments, i.e. 1% and 10% of the labeled ImageNet-1k training datasets in a class-balanced way. The necks or heads are removed and only the backbone CNN is evaluated by appending a linear classification head.

   We apply greedy search on a list of hyper-parameter settings with the base learning rate from {0.001, 0.01, 0.1} and the learning rate multiplier for the head from {1, 10, 100}. We choose the optimal hyper-parameter setting for each method. Empirically, all baselines and their alternatives with CLD obtain the best performance with a learning rate of 0.01 and a learning rate multiplier for the head of 100. We train the network for 20 epochs using SGD with weight decay 0.0001 and a momentum of 0.9, and a mini-batch of 256 on 4 GPUs. The learning rate is decayed by 5 times at epoch 12 and 16 respectively.

3. Which Clustering Method to Use?

We have tried two popular clustering methods: k-Means clustering and spectral clustering, both implemented in Pytorch for fast performance on GPUs.

k-Means clustering [1, 10] aims to partition n representations into k groups, each representation belongs to the cluster with the nearest cluster centroid, serving as a prototype of the cluster. We use spherical k-Means clustering which minimizes: $\sum(1-\cos(f_i, u_{c(i)}))$ over all assignments c of objects i to cluster ids $c(i) \in \{1, ..., k\}$ and over all prototypes $u_1, ..., u_k$ in the same feature space as the feature vector f_i representing the objects. We use binary cluster assignment, where the cluster membership $m_{ij} = 1$ if item i is assigned to cluster j and 0 otherwise. The following k-means objective can be solved using the standard Expectation-Maximization algorithm [4]:

$$\begin{aligned}\Phi(M, \{u_1, ..., u_C\}) &= \sum_{i,j} m_{ij}(1-\cos(f_i, u_{c(i)})) \\ &= \sum_{i,j} m_{ij}(1-\frac{f_i \cdot u_{c(i)}}{||f_i|| \cdot ||u_{c(i)}||}).\end{aligned} \tag{1}$$

Spectral clustering [14, 19] treats data points as nodes of a graph.

1. For feature $f_i \in \mathbb{R}^{d\times 1}$ of N samples, we build a weighted gragh $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, with weight measuring pairwise feature similarity: $w_{i,j} = \frac{f_i \cdot f_j}{||f_i|| \cdot ||f_j||}$.
2. Let $\mathbf{D}$ be the $N \times N$ diagonal degree matrix with $d_{ij} = \sum_{j=1}^{n} w_{ij}$ and $\mathbf{L}$ be the normalized Laplacian matrix:

$$\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}} \tag{2}$$

3. We compute the k largest eigenvalues of $\mathbf{L}$ and use the corresponding k row-normalized eigenvectors E_i as the globalized new feature [14, 19]. We apply EM to find the cluster centroids.

group	spectral	k-Means
10	77.1%	78.9%
64	74.5%	76.3%
128	72.6%	73.4%
256	70.5%	70.8%

Table A.1: Top-1 kNN accuracies on Kitchen-HC under different group numbers for different clustering methods.

NPID+CLD	Subspace	Cross-augmentation	CIFAR-10	CIFAR-100
✗			80.8%	51.6%
✓	share		82.7%	53.3%
✓	separate		84.2%	55.0%
✓	separate	✓	86.5%	57.5%

Table A.2: Ablation study on various components of our method, i.e. adding the cross-level discrimination, projecting the representation to two different spaces, and using cross-augmentation comparison between x_i and x'_i. kNN top-1 accuracy is reported here.

Table A.1 shows that k-means clustering achieves better performance on Kitchen-HC and outperforms optimal spectral clustering result by 1.8% when the group number is 10. However, as the group number increases, the performance difference becomes negligible.

4. Are Separate Feature and Group Branches Necessary?

Intuitively, instance grouping and instance discrimination are at odds with each other. Our solution is to formulate

the feature learning on a common representation, forking off two branches where we can impose grouping and discrimination separately. Table A.2 shows that projecting the representation to different spaces and jointly optimize the two losses increase top-1 kNN accuracy by 1.5% and 1.8% on CIFAR-10 and CIFAR-100 respectively.

5. How Effective Is Cross-Augmentation Comparisons?

Instance-level discrimination presumes each instance is its own class and any other instance is a negative. The groups needed for any group-level discrimination have to be built upon local clustering results extracted from the current feature in training, which are fluid and unreliable.

Our solution is to seek the most certainty among all the uncertainties: We presume stable grouping between one instance and its augmented version, and our cross-level discrimination compares the former with the groups derived from the latter. We roll the three processes: instance grouping, invariant mapping, and instance-group discrimination all into one CLD loss.

Table A.2 shows that our cross-augmentation comparison increases the top-1 accuracy by more than 2% on recognition task. It demands the feature not only to be invariant to data augmentation, but also to be respectful of natural grouping between individual instances, often aligning better with downstream semantic classification.

6. How Sensitive Are Hyper-parameters Weight λ and Temperature T?

λ controls the relative importance of CLD with respect to instance-level discrimination, and helps strike a balance between the caveates of noisy initial grouping and the benefits it brings with coarse-grained repulsion between instances and local groups. Table A.3 shows that, at a fixed group number, $\lambda = 0.25$ achieves optimal performance, and a larger λ generally leads to worse performance and even decreases top-1 accuracy by 3.1% at $\lambda = 3$.

	NPID+CLD		MoCo+CLD	
	top-1 (%)	top-5 (%)	top-1 (%)	top-5 (%)
$\lambda = 0$	75.3	92.4	77.6	93.8
$\lambda = 0.1$	78.8	94.4	80.3	95.0
$\lambda = 0.25$	**79.7**	**95.1**	**81.7**	**95.7**
$\lambda = 0.50$	78.9	94.4	80.5	95.2
$\lambda = 1.0$	78.8	94.5	80.1	94.8
$\lambda = 3.0$	76.6	93.2	78.4	94.1

Table A.3: Top-1 and top-5 linear classification accuracies (%) on ImageNet-100 with different λ's. The backbone network is ResNet-50.

T is known to critical for discrminative learning and can be sometimes tricky to choose. Table A.4 shows that the best performance is achieved at $T = 0.2$ for both CIFAR and ImageNet-100. With local grouping built into our CLD method, we find the sensitivity of T is greatly reduced.

$T(T_I = T_G)$	0.07	0.1	0.2	0.3	0.4	0.5
CIFAR-100	57.9%	57.8%	58.1%	58.1%	57.6%	57.2%
ImageNet-100	79.3%	79.6%	81.7%	80.7%	79.4%	79.0%

Table A.4: Linear (ImageNet-100) and kNN (CIFAR-100) evaluations for models trained with different choices of temperature T, $T_I = T_G$ for simplicity.

7. Is A Larger Memory Bank Always Better for Discriminative Learning?

A larger memory bank includes more negatives and is known to deliver a better discriminator. However, we cannot simply adjust the memory bank size according to NMI or retrieval accuracy in order to deliver the best performance on downstream classification.

Fig. A.2 compares NMI and retrieval accuracies under different negative prototype numbers. If there are too many negatives, the model would focus on repelling negative instances, ignoring the commonality between instances; if there are too few negatives, the model would be subject to random fluctuations from batch to batch, affecting optimization and convergence. However, neither the number of negatives (i.e. infoNCE-k) to obtain the best retrieval accuracy nor the number of negatives to achieve the best NMI score can deliver the best downstream classification task. To deliver optimal performance at downstream classification task, there is a trade-off between local mutual information (evaluated by retrieval task) and global mutual information (evaluated by Normalized Mutual Information).

8. Sample Retrievals

Fig. A.3 shows our near-perfect sample retrievals on ImageNet-100 using $f_I(x)$ in our NPID + CLD model. On the contrary, NPID seems to be much more sensitive to textural appearance (e.g., Rows 1,4,6,7), first retrieve those with similar textures or colors. CLD is able to retrieve semantically similar samples. Our conjecture is that by gathering similar textures into groups, CLD can actually find more informative feature that contrasts between groups. For example, the 5th query image is a Chocolate sauce, which has similar texture with Grouper fish. NPID incorrectly retrieves many images from the Grouper Fish class, but CLD successfully captures the semantic information of the query image, and retrieves instances with the same semantic information.

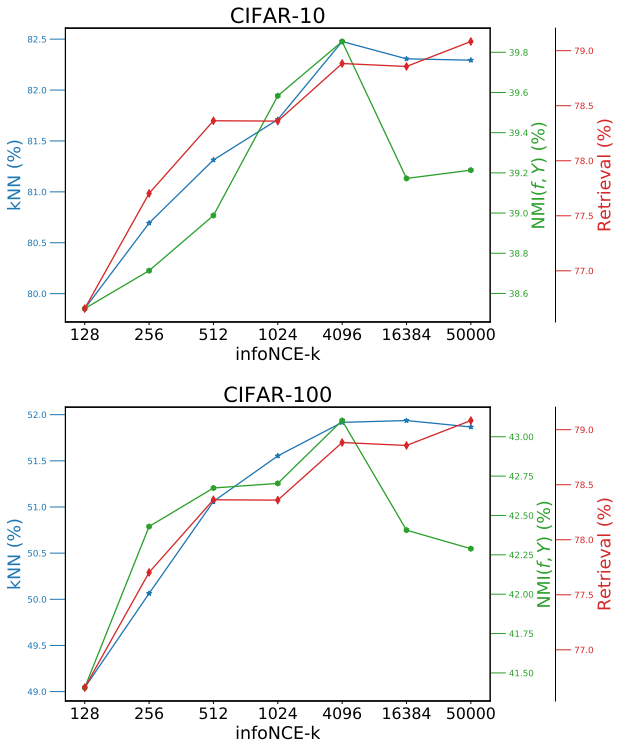

Figure A.2: MoCo trained with different memory bank sizes are evaluated with NMI, retrieval and kNN accuracy. While a larger memory bank improves the retrieval performance, the classification accuracy and NMI score do not always increase: the NMI score drops sharply due to a large negative/positive ratio. There is a trade-off for best performance at downstream classification.

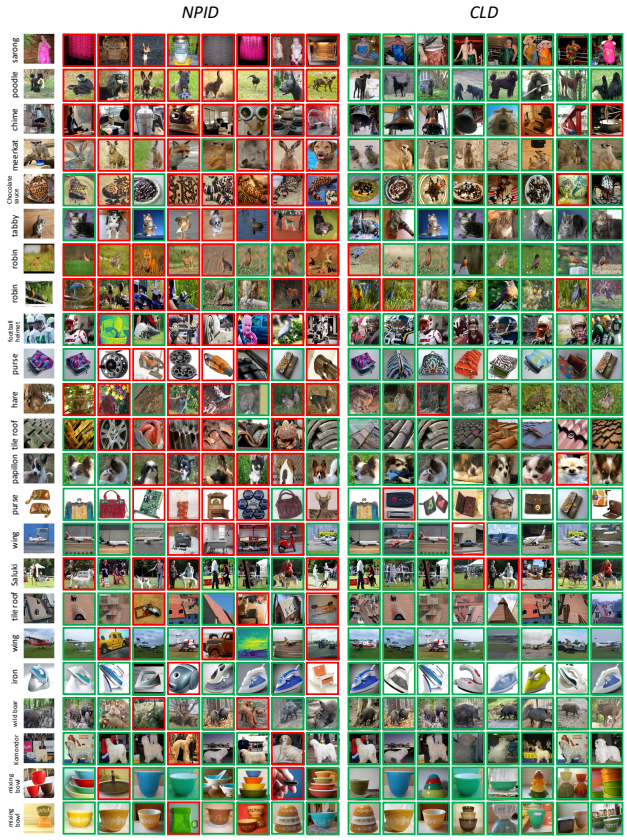

Figure A.3: Comparisons of top **retrieves** by NPID (Columns 2-9) and NPID+CLD (Columns 10-17) according to f_I for the query images (Column 1) from the ImageNet validation set. The results are sorted by NPID's performance: Retrievals with the same category as the query are outlined in green and otherwise in red. NPID seems to be much more sensitive to textural appearance (e.g., Rows 1,4,5,7), first retrieve those with similar textures or colors. Integrated with CLD, NPID+CLD is able to retrieve semantically similar samples. (Zoom in for details)

References

[1] Christian Buchta, Martin Kober, Ingo Feinerer, and Kurt Hornik. Spherical k-means clustering. *Journal of Statistical Software*, 2012.

[2] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[3] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011.

[4] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1977.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

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

[7] Georgios Georgakis, Md Alimoor Reza, Arsalan Mousavian, Phi-Hung Le, and Jana Košecká. Multiview rgb-d dataset for object instance detection. In *3DV*, 2016.

[8] 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.

[9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[10] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, 2002.

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009.

[12] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019.

[13] Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun. Megdet: A large mini-batch object detector. In *CVPR*, 2018.

[14] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 2000.

[15] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[16] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020.

[17] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

[18] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, 2019.

[19] Stella X. Yu and Jianbo Shi. Multiclass spectral clustering. In *CVPR*, 2003.

[20] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE international conference on computer vision*, pages 1476–1485, 2019.

[21] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Dahua Lin, and Chen Change Loy. OpenSelfSup: Open mmlab self-supervised learning toolbox and benchmark. 2020.