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

Abstract

Unsupervised feature learning has made great strides with contrastive learning based on instance discrimination and invariant mapping, as benchmarked on curated class-balanced datasets. However, natural data could be highly correlated and long-tail distributed. Natural between-instance similarity conflicts with the presumed instance distinction, causing unstable training and poor performance.

Our idea is to discover and integrate between-instance similarity into contrastive learning, not directly by instance grouping, but by cross-level discrimination (CLD) between instances and local instance groups. While invariant mapping of each instance is imposed by attraction within its augmented views, between-instance similarity could emerge from common repulsion against instance groups.

Our batch-wise and cross-view comparisons also greatly improve the positive/negative sample ratio of contrastive learning and achieve better invariant mapping. To effect both grouping and discrimination objectives, we impose them on features separately derived from a shared representation. In addition, we propose normalized projection heads and unsupervised hyper-parameter tuning for the first time.

Our extensive experimentation demonstrates that CLD is a lean and powerful add-on to existing methods such as NPID, MoCo, InfoMin, and BYOL on highly correlated, long-tail, or balanced datasets. It not only achieves new state-of-the-art on self-supervision, semi-supervision, and transfer learning benchmarks, but also beats MoCo v2 and SimCLR on every reported performance attained with a much larger compute. CLD effectively brings unsupervised learning closer to natural data and real-world applications. Our code is publicly available at: https://github.com/frank-xwang/CLD-UnsupervisedLearning.

1. Introduction

Representation learning aims to extract latent or semantic information from raw data. Typically, a model is first trained on a large-scale annotated dataset [34] and then tuned on a small-scale dataset for a downstream task [25]. As the model gets bigger and deeper [26, 29], more annotated data are needed; supervised pre-training is no longer viable.

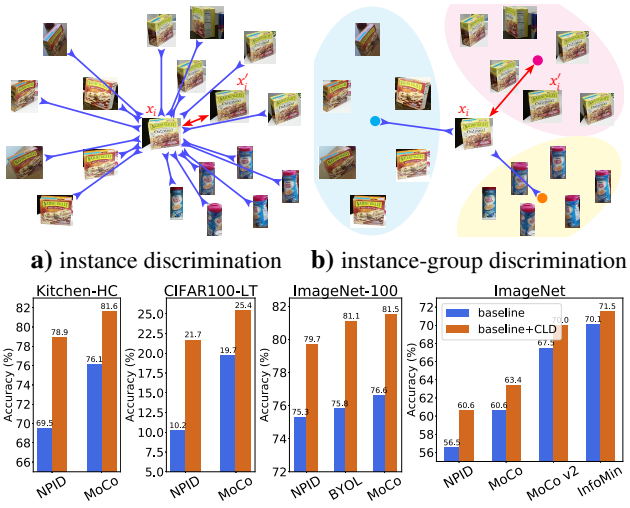
Self-supervised learning [13, 44, 63, 41, 14, 39] gets around labeling with a pre-text task which does not require annotations and yet would be better accomplished with semantics. For example, to predict the color of an object from its grayscale image does not require labeling; however, doing it well would require a sense of what the object is. The biggest drawback is that pre-text tasks are domain-specific and hand-designed, and they are not directly related to downstream semantic classification.

Unsupervised contrastive learning has emerged as a direct winning alternative [53, 64, 58, 6, 24]. The training objective and the downstream classification are aligned on discrimination, albeit at different levels of granularities: training is to discriminate known individual instances, whereas testing is to discriminate unknown groups of instances.

Contrastive learning approaches have made great strides with two ideas: invariant mapping [23] and instance discrimination [53]. That is, the learned representation should be **1)** stable for certain transformed versions of an instance, and **2)** distinctive for different instances. Both aspects can be formulated without labels, and the feature learned appears to automatically capture semantic similarity, as benchmarked by downstream classification on standard datasets such as CIFAR100 and ImageNet [6]. However, these datasets are curated with distinctive and class-balanced instances, whereas natural data could be highly correlated within the class (e.g., repeats) and long-tail distributed across classes.

Natural between-instance similarity demands instance grouping not instance discrimination, where *all the instances are presumed different*. Consequently, feature learning by instance discrimination is unstable and under-performing without instance grouping, whereas instance grouping based on the feature learned without instance discrimination is easily trapped into degeneracy. Ad-hoc tricks [3, 4] and mutual information maximization with a uniform class distribution prior [32] have been used to prevent feature degeneracy.

We propose to discover and integrate between-instance similarity into contrastive learning, not directly by instance grouping, e.g., by imposing group-level discrimination as



a) instance discrimination **b)** instance-group discrimination

c) Our cross-level discrimination boosts existing SOTA's

Figure 1: Our unsupervised feature learning discovers similar instances and integrates grouping into instance-level discrimination, outperforming the state-of-the-art (SOTA) classifiers on highly correlated, long-tail, or balanced datasets. **a)** Instance discrimination presumes all instances distinctive: Instance x_i attracts ($\leftrightarrow$) its augmented version x_i' and repels ($\not\leftrightarrow$) all other instances including those highly similar ones. **b)** We propose cross-level discrimination (CLD) between instance x_i and local groups of alternative views $\{x_j'\}$. x_i attracts ($\leftrightarrow$) the group centroid that x_i' belongs to and repels ($\not\leftrightarrow$) other group centroids. Visually similar instances tend to attract/repel the same group centroids and are thus mapped closer. **c)** Our CLD can be added to existing methods such as NPID [53], MoCo [24], MoCo v2 [7], InfoMin [49] and BYOL [21]. It consistently provides a significant performance boost on highly correlated (HC), long-tail (LT), and standard balanced ImageNet datasets.

DeepCluster [3, 4] or by regulating instance-level discrimination based on the grouping outcome as Local Aggregation (LA) [64], but by imposing cross-level discrimination (CLD) between instances and local instance groups.

Contrastive learning is built upon dual forces of attraction and repulsion [23]. Existing methods generally assume repulsion between different instances and attraction within *known groupings* of instances, e.g., between augmented views of the same data instances [53, 64, 24], or between data captured from different times, views, or modalities of the same physical instances [42, 1, 50, 48].

Feature learning with between-instance similarity calls for attraction within *unknown groupings*, not the universal between-instance repulsion (Fig. 1a). An chicken-and-egg challenge is to discover such groupings for feature learning while the feature for the groupings is still to be developed.

Our key insight is that grouping could result from not just attraction, but also common repulsion. While invariant mapping is achieved by within-instance similarity from attraction across augmented views, between-instance similarity can emerge from repulsion against common instance groups, the centroids of which are more stable in the developing feature space. That is, to discover the most discriminative feature that also respects natural instance grouping, we desire each instance to attract the closest group related by augmentation and repel groups of other instances that are far from it.

In our approach (Fig. 1b), between-instance similarity, unknown *a priori*, is not captured directly as attraction between instances, but by more likely common attraction and repulsion between each instance and instance group centroids. By pulling an instance towards and pushing it against more stable instance groups, *similar* instances get mapped closer in the feature space. To effect both grouping and discrimination objectives on feature learning, we also impose them on features separately derived from a shared representation.

Such an interplay between attraction and repulsion has been utilized to model perceptual popout [60, 2], as well as simultaneous image segmentation and depth segregation [59, 38]. However, those works are prior to deep learning and aim at grouping pixels based on certain fixed pixel-level feature such as edges, whereas our work aims at learning the image-level feature discriminatively.

We add CLD to popular state-of-the-art (SOTA) unsupervised feature learning approaches (Fig. 1c), e.g., NPID [53], MoCo [24], InfoMin [49] (all three based on instance discrimination), and BYOL [21] (focusing only on invariant mapping without instance discrimination). CLD delivers a significant performance boost not only on highly correlated, long-tail, and balanced datasets, but also on all the self-supervision, semi-supervision, and transfer learning benchmarks under fair comparison settings [53, 24, 62].

Our work makes three major contributions. **1)** We extend unsupervised feature learning to natural data with high correlation and long-tail distributions. **2)** We propose cross-level discrimination between instances and local groups, to discover and integrate between-instance similarity into contrastive learning. We also propose normalized projection heads and unsupervised hyper-parameter tuning. **3)** Our experimentation demonstrates that adding CLD to existing methods has an negligible overhead and yet delivers a significant boost. It achieves new SOTA on all the benchmarks, and beats MoCo v2 [7] and SimCLR [6] on every reported performance attained with a much larger compute.

2. Related Works

Unsupervised representation learning [13, 44, 63, 41, 14, 35, 31, 19, 61] aims to learn features transferable to downstream tasks. Our work is closely related to contrastive learning and unsupervised feature learning with grouping.

Contrastive learning maps positive samples closer and negative samples apart in the feature space [53, 39, 48, 24, 7, 6].

Positive samples come from augmented views of each instance, whereas negative ones come from different instances. The key distinction among existing methods lies in how these samples are obtained and maintained during learning.

Batch methods [6] draw samples from the current mini-batch with the same encoder, updated end-to-end with back-propagation. **Memory-bank methods** [53, 39] draw samples from a memory bank that stores the prototypes of all the instances computed previously. **Hybrid methods** [24, 7] encode positive samples by a momentum-updated encoder and maintain negative samples in a queue.

Instance discrimination methods presume distinctive instances. Their performance drops on natural data that are highly correlated or long-tail distributed, e.g., consecutive frames in a video, or different views of the same instance. Note that our setting is *completely unsupervised* and different from learning representation across views [1, 50, 48]: We have mixed data without any object or view labels.

Feature learning with grouping exploits natural organization of data [54, 55, 4, 64]. Unlike self-supervised learning [44, 41, 19], it does not require domain knowledge [3].

Earlier works restrict learning to linear feature transformations. DisCluster [10, 12] and DisKmeans [57] iteratively apply K-means to generate cluster labels and then use linear discriminant analysis (LDA) to select the most discriminative subspace. [56] applies LDA along with spectral clustering [52]. [40] uses linear regression as a regularization term to handle out-of-sample data in spectral clustering.

Nonlinear feature transformations have also been studied. [47] applies a deep sparse autoencoder to a normalized graph similarity matrix and performs K-means on the latent representation. [51] implements t-SNE embedding with a deep neural network. Deep Embedded Clustering [54] simultaneously learns cluster centroids and feature mapping such that centroid-based soft assignments in the embedding matches a desirable target distribution.

Recent works jointly optimize the feature and the cluster assignment. **DeepCluster** [3, 4] gets pseudo-class labels from global clustering and applies supervised learning to iteratively fine-tune the model, whereas our CLD incorporates local clustering into contrastive metric learning. **Local Aggregation (LA)** [64] identifies a local neighbourhood of each instance through clustering, and restricts instance-level discrimination within individual neighbourhoods, whereas CLD looks beyond local neighbourhoods and conducts cross-level instance-group discrimination. **PCL** [36] is a concurrent work that compares instance features with group centroids which are obtained through global clustering per epoch, whereas our CLD uses local clustering per batch and compares instance-group features within the batch. Global clusters not only takes more time to compute during training, but conceptually also do not align with classes in downstream tasks. Empirically, PCL gains much over MoCo but not over MoCo v2 [36]. **SegSort** [30] extends representation learning from classification to segmentation. It learns a feature per pixel, and assumes that all the pixels in the same region form a cluster in the feature space. SegSort uses *one* common feature and contrasts each *pixel* with cluster centroids in the feature from the *same*-view, whereas our CLD uses *two* separate features and contrasts each *image* with cluster centroids in the feature from a *different* view.

Discussions. While clustering on a fixed feature is well studied [17], clustering with an adapting feature is a tricky model selection problem: **1)** Clustering could fall into trivial solutions where most samples are assigned to a single cluster, trapping feature learning into degeneracy [3]. **2)** Without any external supervision, it is unclear how to ensure that the learned feature captures latent semantics.

Our work combines contrastive learning and grouping in a single framework, by expanding discrimination between instances to that between instances and local groups. Discrimination prevents feature learning from degeneracy, while grouping improves stability and helps instance-level discrimination see beyond the finest granularity. With these two aspects integrated, our CLD significantly improves the learned representation for downstream classification.

3. Learning with Cross-Level Discrimination

Given n images, we regard instance x_i as a *view* obtained by a certain transformation (e.g. cropping) of the i-th image. Let x_i and x_i' denote two different *views* of the i-th instance.

Contrastive learning [23, 53, 24, 48, 42, 6] aims to learn a mapping function f such that in the $f(x)$ feature space, instance x_i is **1)** close to positive sample x_i' **(invariant mapping)**, and **(2)** far from negative sample x_j (with $j \neq i$) of any other instances **(instance discrimination)**.

We model f by a convolutional neural network (CNN) with parameters θ, mapping x onto a d-dimensional hypersphere such that $\|f(x)\| = 1$. Let f, f^+, f^- denote the feature for an instance and its positive / negative samples respectively. We optimize θ by minimizing loss C over all n instances so that f attracts f^+ and repels f^-.

instance-centric contrastive loss:

$$C\left(f_i, f_i^+, f_{\neq i}^-\right) = -\log \frac{\exp \frac{<f_i, f_i^+>}{T}}{\exp \frac{<f_i, f_i^+>}{T} + \sum_{j \neq i} \exp \frac{<f_i, f_j^->}{T}} \quad (1)$$

Temperature T is a hyperparameter regulating what distance is close. C is the noise contrastive estimation (NCE) [22] of softmax instance classification loss [53], and it can be viewed as maximizing a lower bound of mutual information (MI) between samples of the same instances [43, 23, 42].

Implementation of $(f_i, f_i^+, f_{\neq i}^-)$ during training. For sample x_i, the self feature is $f_i = f(x_i)$, whereas positive feature f_i^+ and negative feature $f_{\neq i}^-$ come from a memory bank v that holds the representative feature for $\{x_i\}_{i=1}^n$. It

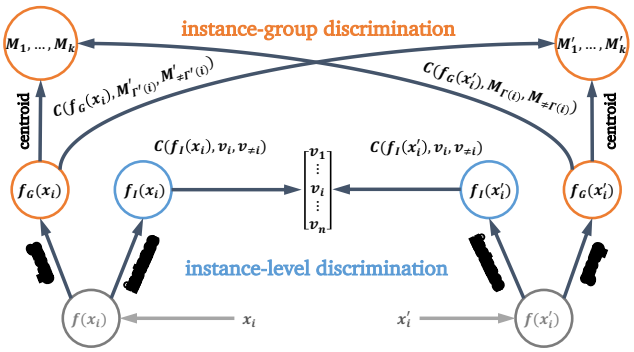


Figure 2: Method overview. Our goal is to learn representation $f(x)$ given image x and its alternative view x' from data augmentation. We fork two branches from f: fine-grained *instance branch* f_I and coarse-grained *group branch* f_G. All the computation is mirrored and symmetrical with respect to different views of the same instance. **1) Instance Branch**: We apply contrastive loss (*two bottom C's*) between $f_I(x_i)$ and a global memory bank $\{v_i\}$, which holds the prototype for x_i, computed from the average feature of the augmented set of x_i. **2) Group Branch**: We perform *local* clustering of $f_G(x_i)$ for a batch of instances to find k centroids, $\{M_1, \ldots, M_k\}$, with instance i assigned to centroid $\Gamma(i)$. Their counterparts in the alternative view are $f_G(x'_i)$, M', and Γ'. **3) Cross-Level Discrimination**: We apply contrastive loss (*two top C's*) between feature $f_G(x_i)$ and centroids M' according to grouping Γ', and vice versa for x'_i. **4)** Two similar instances x_i and $x_j$would be pushed apart by the instance-level contrastive loss but pulled closer by the cross-level contrastive loss, as they repel common negative groups. Forces from branches f_I and f_G act on their common feature basis f, organizing it into one that respects both instance grouping and instance discrimination.

is computed as the average feature of all the augmented versions of x_i seen so far [53, 6]. It could also be encoded by a parametric model as in MoCo [24]. Existing methods apply C at the instance level, between instance feature f_I and its average v: $C\left(f_I(x_i), v_i, v_{\neq i}\right)$ (Fig. 2 instance branch).

Pros and cons of instance-level contrastive learning. Contrastive learning has greatly closed the gap with supervised classification [53, 42, 24, 6]. However, there are 4 caveats.

1. It focuses on within-instance similarity by data augmentation, oblivious of between-instance similarity.
2. It focuses on discrimination at the finest instance level, oblivious of natural groups which often underlie downstream tasks' discrimination at a coarser semantic level.
3. It presumes distinctive instances, whereas non-curated data could contain repeats, redundant observations of the same instance, and long-tail distributed instances across classes in the downstream task. For feature f_i, its negative features $\{f_i^-\}$ would thus contain highly correlated samples which f_i should ideally attract rather than repel.
4. Each instance has a high positive/negative imbalance ratio (1 vs. rest); the more negatives, the larger the signal to noise ratio [45], and the better the performance [28, 48]. However, the model also leans towards more instance discrimination than invariant mapping, reducing robustness.

Feature grouping. To overcome these caveates, we step beyond individual instances and discover how they might be related. We acknowledge the natural grouping of instances by finding *local* clusters within a batch of samples. Which specific clustering method to use is not as critical; we apply spherical K-means to the unit-length feature vectors.

Local clustering could be rather noisy, especially at the early stage of learning. Instead of imposing group-level discrimination, we validate local groupings across views and impose consistent discrimination between individual instances and their cross-view local groups.

Group branch. Grouping and discrimination are opposite in nature. To effect both objectives, we fork two branches (just one FC layer each) from feature f: fine-grained instance branch f_I and coarse-grained group branch f_G (Fig. 2). We first extract f_G at the instance level in a batch, then compute k local cluster centroids $\{M_1, \ldots, M_k\}$ and assign each instance to its nearest centroid. Clustering assignment $\Gamma(i) = j$ means that instance i is assigned to centroid j.

Cross-level discrimination. Natural groups identified in the group branch allows the expansion of positive samples from augmented versions of an individual instance to like-kind *other* instances. We also expand negative samples from other instances to groups of their like-kind instances. We apply *local* (i.e., batch-wise) contrastive loss across views between instance feature $f_G(x'_i)$ and group centroids M, i.e., $C\left(f_G(x'_i), M_{\Gamma(i)}, M_{\neq\Gamma(i)}\right)$ and vice versa for $f_G(x_i)$ (Fig. 2). Intuitively, if local clustering Γ separates $\{x_i\}$ well, when x_i is replaced by its alternative view x'_i, it should still be close to x_i's centroid $M_{\Gamma(i)}$ and far from other centroids $M_{\neq\Gamma(i)}$. That is, instances and their local clusters should retain their grouping relationships across views.

Comparisons across levels, instances, views are beneficial:

1. For instances clustered in the same group, instance feature $f_G(x_i)$ and $f_G(x_j)$ would be attracted to the same group centroid M or M' and are thus drawn closer.
2. For similar instances x_i and x_j not in the same cluster, they likely repel common group centroids, thereby pulling instance features $f_G(x_i)$ and $f_G(x_j)$ closer.
3. CLD discriminates at instance *and* group levels, more in line with coarser discrimination at downstream tasks.
4. Comparisons between f_G and M not only avoid direct repulsion between similar instances, but also greatly improves the positive/negative ratio for invariant mapping. For example, the ratio on ImageNet is $\frac{1}{4096}$ for NPID [53]'s set-wise NCE vs. $\frac{1}{255}$ for CLD's batch-wise NCE.

5. Cross-view comparisons between x_i and x'_i focus the model more on invariant mapping.

Probabilistic interpretation of CLD. Our CLD objective can be understood as minimizing the cross entropy between hard clustering assignment p_{ij} (as *ground-truth*) based on $f_G(x_i)$ and soft assignment q_{ij} predicted from $f_G(x'_i)$ in a different view. Since $p_{ij}=1$ only when $j=\Gamma(i)$, we have a loss that validates local groupings across different views:

$$-E_p[\log q] = \sum_i C(f_G(x'_i), M_{\Gamma(i)}, M_{\neq\Gamma(i)}; T_G). \quad (2)$$

Total contrastive learning loss. We add CLD to instance discrimination (with temperatures T_I, T_G, weight λ) in symmetrical terms over views x_i and x'_i:

$$L(f; T_I, T_G, \lambda) = \sum_i \underbrace{C(f_I(x_i), v_i, v_{\neq i}; T_I) + C(f_I(x'_i), v_i, v_{\neq i}; T_I)}_{\text{instance-level discrimination}}$$
$$+\lambda \sum_i \underbrace{C(f_G(x'_i), M_{\Gamma(i)}, M_{\neq\Gamma(i)}; T_G) + C(f_G(x_i), M'_{\Gamma'(i)}, M'_{\Gamma'(i)}; T_G)}_{\text{cross-level discrimination}}$$

We analyze why two feature branches are better than one branch, where $f_I=f_G$ and M is simply the group centroids of $f_I(x_i)$ or v. In that case, while the instance discriminiation term would repel x_i against any other instances $\{x_j\}$, the CLD term would make x_i attract *some other* instances $\{x_j\}$ in the same group of x_i through their group centroid. Minimizing the two terms would lead to opposite effects no matter what the local clustering is. Basing instance feature f_I and group feature f_G as separate branches off feature f would force f to be discriminative enough for the instance branch yet loosely similar enough for the group branch.

Normalized projection head. Existing methods derive instance feature $f_I(x)$ by mapping the latent feature $f(x)$ onto a unit hypersphere with first a projection head and then normalization. NPID [53] and MoCo [24] use one FC layer as a linear projection head. MoCo v2 [7], SimCLR [6], and BYOL [21] use a multi-layer perceptron (MLP) head; it is better for large datasets and worse for small datasets.

We propose to normalize both the FC layer weights W and the shared feature vector f so that projecting f onto W simply calculates their cosine similarity. The t-th component of normalized feature $N(x_i)$ (where $N=f_I$ or $N=f_G$) is:

$$N_t(x_i) = < \frac{W_t}{\|W_t\|}, \frac{f(x_i)}{\|f(x_i)\|} > . \quad (3)$$

Normalized linear (NormLinear) or MLP (NormMLP) projection heads bring additional gains to CLD. Empirically, they help reduce feature variance from data augmentation.

4. Experiments

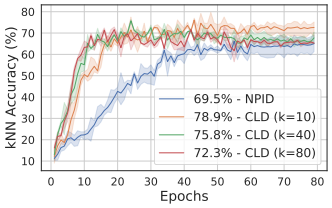
We use ResNet-50 for ImageNet data and ResNet-18 otherwise. We compare linear classification accuracies on ImageNet, and follow NPID on using kNN accuracies ($k=200$) for all the small-scale benchmarks. The kNN accuracies are higher and more fitting for metric learning. Results marked by † are obtained with released code.

We consider 3 types of datasets. **1) High-correlation**: Kitchen-HC is constructed by extracting objects in their bounding boxes from the multi-view RGB-D Kitchen dataset [18]. It has 11 categories with highly correlated samples and 20.8K / 4K / 14.4K instances in train / validation / test sets. **2) Long-tail**: CIFAR10-LT, CIFAR100-LT and ImageNet-LT [37]. **3) Major benchmarks**: CIFAR [33], STL10 [9], ImageNet-100 [48], ImageNet [11]. Following [58], we train models on 5K samples in the *train* set and 100K samples in the *unlabeled* set, and test on the *test* set of STL10.

4.1. Benchmarking Results

Results on high-correlation data. Having highly correlated instances breaks the instance discrimination presumption and causes slow or unstable training. Accuracies in Fig. 3 and feature visualization in Fig. 4 indeed show that CLD is much better and fast converging towards a more distinctive feature representation. At Epoch 10, CLD outperforms by 40% (23% vs. 63%). CLD outperforms NPID by 9.4%, when the number of groups used in local clustering is closer to the number of semantic classes in the downstream classification. Likewise, MoCo + CLD outperforms its counterpart MoCo by 5.5%.

Results on long-tailed data. Table 1 shows that CLD outperforms baselines by a large margin on CIFAR10-LT and



Kitchen-HC	kNN Accuracy
NPID	69.5
NPID + CLD	78.9 **(+9.4)**
MoCo	76.1
MoCo + CLD	**81.6** **(+5.5)**

Figure 3: **Left**: CLD is more accurate and fast converging than NPID on Kitchen-HC, esp. when the number of groups is closer to the number of classes 11. The average top-1 kNN accuracy of 5 runs is reported. **Right**: CLD outperforms NPID or MoCo on **high correlation dataset** Kitchen-HC.

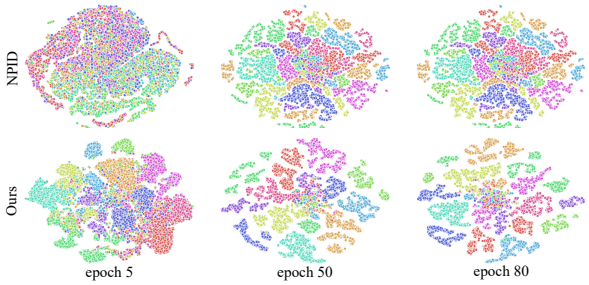


Figure 4: CLD has earlier and better separation between classes (indicated by the dot color) than NPID in the **t-SNE visualization** of instance feature $f_I(x_i)$ on Kitchen-HC.

	CIFAR10-LT		CIFAR100-LT		ImageNet-LT		
	top1	top5	top1	top5	many/med/few	top1	top5
Unsupervised							
NPID [53]	32.3	74.8	10.2	29.8	47.5/21.3/6.6	29.5	51.1
NPID + CLD	41.1	78.9	21.7	44.3	52.4/25.0/8.3	32.7	55.6
vs. baseline	**+8.8**	**+4.1**	**+11.5**	**+14.5**	**+4.9/+3.7/+1.7**	**+3.2**	**+4.5**
MoCo [24]	34.2	76.7	19.7	42.6	48.1/21.3/6.9	29.9	51.8
MoCo + CLD	**43.1**	**80.4**	**25.4**	**50.0**	**53.1/24.9/9.4**	**33.3**	**57.3**
vs. baseline	**+8.9**	**+3.7**	**+5.7**	**+7.4**	**+5.0/+3.6/+2.5**	**+3.4**	**+5.5**
Supervised							
CE	-	-	-	-	40.9/10.7/0.4	20.9	-
OLTR [37]	-	-	-	-	43.2/35.1/18.5	35.6	-

Table 1: CLD outperforms unsupervised baselines on **long-tailed datasets**, approaching supervised cross-entropy (CE) and OLTR [37]. The kNN (linear) classifiers are used for CIFAR (ImageNet-LT). CLD is significantly better than supervised CE on many-shot (100+), medium-shot ([20, 100)), few-shot (20−), and gets close to OLTR.

kNN accuracies	STL10	CIFAR10	CIFAR100	ImageNet100
DeepCluster	-	67.6	-	-
Exemplar [15]	79.3	76.5	-	-
Inv. Spread [58]	81.6	83.6	-	-
CMC [48]	-	-	-	79.2
NPID [53]	79.1	80.8	51.6	75.3
NPID + CLD	83.6	86.7	57.5	79.7
vs. baseline	**+4.5**	**+5.9**	**+5.9**	**+3.6**
MoCo [24]	80.8	82.1	53.1	76.6
MoCo + CLD	**84.3**	**87.5**	**58.1**	**81.5**
vs. baseline	**+3.5**	**+5.4**	**+5.0**	**+4.9**
BYOL [21]	-	-	-	75.8
BYOL + CLD	-	-	-	81.1
vs. baseline	-	-	-	**+4.7**

Table 2: **On self-supervised learning on small/medium-sized benchmarks**: STL10, CIFAR10, CIFAR100 and ImageNet-100, CLD delivers consistent gains as an add-on to various methods which use either standard contrastive loss (e.g. MoCo [24]) or without negative pairs (e.g. BYOL [21]). On ImageNet-100, we use our re-implemented code for baselines as they are better than those in CMC [48]. All baselines and their CLD add-on's are optimized with the same training recipe for fair comparisons. For small- and medium-sized datasets, the nonlinear multi-layer perceptron (MLP) head performs worse than a linear projection head.

CIFAR100-LT. On ImageNet-LT, CLD outperforms NPID by 4.5% per top-5 accuracy, with the largest relative gain (24%) on few-shot classes; Our unsupervised CLD even significantly outperforms supervised plain Cross-Entropy (CE) by 8-14% and is catching up closely with supervised long-tail classifier OLTR (33.3% vs. 35.6%).

Results on major benchmarks. Table 2 shows that CLD outperforms SOTA on STL10, CIFAR10, CIFAR100 and ImageNet-100. On ImageNet, Table 3 shows that CLD consistently outperforms baselines under fair comparison settings: 200 training epochs, standard augmentations [53], and comparable model sizes. Adding CLD to InfoMin instead of MoCo produces 7.7% gain, by using an MLP projection head over feature $f(x)$, a cosine learning scheduler, extra data augmentation [7, 6, 49], and a JigSaw branch as in PIRL [39]. Fig. 5 shows CLD retrievals less distracted by textures.

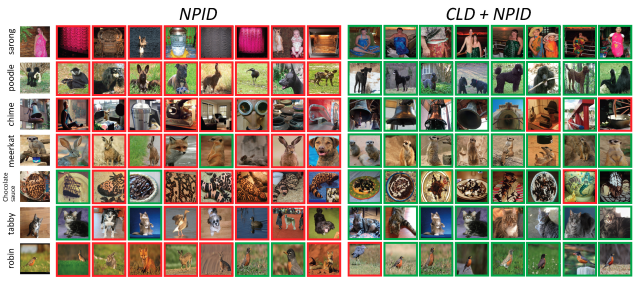


Figure 5: CLD top retrievals according to f_I (Columns 10-17) are less distracted by textures than NPID (Columns 2-9) for query images (Column 1) from the ImageNet validation set. Results are sorted by NPID's performance. Correct retrievals, those in the same category as the query, are outlined in green and wrong ones in red. NPID seems more sensitive to textural appearance (e.g., Rows 1,4,5,7), first retrieve those with similar textures or colors.

Results on semi-supervised learning. Table 4 shows that CLD utilizes annotations far more efficiently, outperforming SOTA (InfoMin) by 6.1% with only 1% labeled samples. Baselines and CLDs follow OpenSelfSup benchmarks [62] for fair comparisons. Baseline results are copied from [62].

Transfer learning for object detection. We test the feature transferability by fine-tuning an ImageNet trained model for Pascal VOC object detection [16]. Table 5 shows that CLD not only outperforms its supervised learning counterpart by more than 6%(3%) in terms of AP in VOC07(VOC07+12), but also surpasses current SOTA of MoCo and MoCo v2.

4.2. Further Analysis

Why CLD performs better on long-tailed data? CLD groups similar samples and uses coarse-grained group prototypes instead of instance prototypes. There are two consequences. **1)** The positive to negative sample ratio is greatly increased from the instance branch to our group branch. For example, while each instance is compared against 4,096 negatives (as in MoCo), it is only compared against k negative centroids in our group branch, where $k \leq 256$ – our batch size. The importance of positives increases from $\frac{1}{4096}$ to $\frac{1}{k}$. CLD thus achieves better invariant mapping for all the classes, head or tail. However, the increased ratio is more important for tail classes, as they don't have so many instances to rely on as head classes. **2)** The imbalance between head and tail classes in the negatives is also reduced in our group branch. While the distribution of instances in a random mini-batch is long-tailed, it would be more flattened across classes after clustering. The tail-class negatives would be better represented in the NCE loss. Fig. 6 shows that indeed CLD has clearer class separation than MoCo.

Methods	Architecture	#epoch	#GPU	top-1
NPID [53]	R50-Linear (24M)	200	8	56.5
w/ CLD	R50-Linear (24M)	200	8	60.6
MoCo [24]	R50-Linear (24M)	200	8	60.6
w/ CLD	R50-Linear (24M)	200	8	63.4
w/ CLD	R50-NormLinear (24M)	200	8	63.8
MoCo v2 [7]	R50-MLP (28M)	200	8	67.5
w/ CLD	R50-MLP (28M)	200	8	69.2
w/ CLD	R50-NormMLP (28M)	200	8	70.0
BYOL† [21]	R50-MLP (28M)	100	128	66.5
w/ CLD‡	R50-NormMLP (28M)	100	8	69.1
InfoMin [49]	R50-MLP (28M)	100	8	67.4
w/ CLD	R50-MLP (28M)	100	8	69.5
w/ CLD	R50-NormMLP (28M)	100	8	70.1
InfoMin [49]	R50-MLP (28M)	200	8	70.1
w/ CLD	R50-MLP (28M)	200	8	70.6
w/ CLD	R50-NormMLP (28M)	200	8	**71.5**
SimCLR† [6]	R50-MLP (28M)	100	128	66.5
SwAV† [5]	R50-MLP (28M)	100	128	66.5
BYOL† [21]	R50-MLP (28M)	100	128	66.5
SimSiam† [8]	R50-MLP (28M)	100	8	68.1
SimCLR [6]	R50-MLP (28M)	200	8	61.9
SimCLR† [6]	R50-MLP (28M)	200	128	68.3
SwAV† [5]	R50-MLP (28M)	200	128	69.1
BYOL† [21]	R50-MLP (28M)	200	128	70.6
MoCo v2 [7]	R50-MLP (28M)	200	8	67.5
SimSiam† [8]	R50-MLP (28M)	200	8	70.0
PIRL [39]	R50-Linear (24M)	800	32	63.6
CMC [48]	$R50_{L+ab}$-Linear (47M)	280	8	64.1
CPC v2 [27]	R170-Linear (303M)	200	32	65.9
SimCLR [6]	R50-MLP (28M)	800	128	69.3
MoCo v2 [7]	R50-MLP (28M)	800	8	71.1
SwAV [5]	R50-MLP (28M)	400	128	70.1
SimSiam† [8]	R50-MLP (28M)	800	8	71.3

Table 3: **On self-supervised learning on ImageNet**, our CLD and NormMLP can be added to improve existing methods and achieve SOTA under 100-/200-epoch pre-training settings. Note that our experiments with CLD are conducted with 8 RTX 2080Ti GPUs, whereas PIRL, SimCLR, BYOL and SwAV require batch size 4,096 and 128/512 GPUs/TPUs for their original reported performance. All the results follow the standard linear evaluation protocol as used in [53, 24, 7, 49], except those marked by † (all copied from [8]): The linear classifier training of SwAV [5], BYOL [21] and SimSiam [8] uses base $lr = 0.02$ with a cosine decay scheduler, batch size 4096 with a LARS optimizer, giving these methods about 1% additional gain [8]. All the baseline results are from either their original papers or [8]. For BYOL+CLD results marked by ‡, the target network is updated once every 16 steps and uses batch size 256.

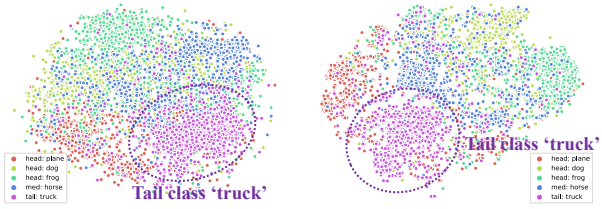
How many groups shall CLD use? The ideal number of groups depends on the level of instance correlation, the number of classes, and the batch size. Table 7 shows that for CIFAR100, CLD is best when the number of groups is close to the number of classes, although CLD already outperforms MoCo at 10 groups. For ImageNet, the instance correlation is low; since the number of classes of 1,000 is larger than the batch size that our 8 GPUs can afford, we just choose the largest number of groups possible. We expect continuous gain with more groups and larger batches afforded by more GPUs. Nevertheless, our model wins with its merit of the CLD idea instead of a large compute.

Methods	Model	Label fraction	
		1%	10%
random initialization	ResNet50	1.6	21.8
rotation [19]	ResNet50	19.0	53.9
DeepCluster [3]	ResNet50	33.4	52.9
NPID [53]	ResNet50	28.0	57.2
MoCo [24]	ResNet50	33.2	60.1
SimCLR [6]	ResNet50	36.3	58.5
MoCo v2 [7]	ResNet50	38.7	61.6
InfoMin † [49]	ResNet50	39.7	62.3
MoCo v2 + CLD	ResNet50	44.4	63.6
InfoMin + CLD	ResNet50	**45.8**	**64.4**
vs. SOTA	ResNet50	**+6.1**	**+2.1**

Table 4: Top-1 accuracy of **semi-supervised learning** (1% and 10% label fractions) on ImageNet. CLD greatly improves SOTA. Baselines and CLD follow training recipes of OpenSelfSup benchmark [62] for fair comparisons, and apply the best performing hyper-parameter setting for each method. † denotes re-implemented results with [62].

Methods	VOC07		VOC07+12	
	AP_{50}	AP	AP_{50}	AP
supervised	74.6	42.4	81.3	53.5
JigSaw [20]	-	-	82.7	53.3
LocalAgg [64]	69.1	-	-	-
MoCo [24]	74.9	46.6	81.5	55.9
MoCo v2 [7]	-	-	82.0	56.4
SimCLR [6]	75.2	-	-	-
NPID + CLD	75.7	47.2	82.0	56.4
MoCo + CLD	76.8	48.3	82.4	56.7
MoCo v2 + CLD	77.6	49.3	82.7	57.0
InfoMin + CLD	**77.9**	**49.8**	**83.0**	**57.2**
vs. SOTA	**+2.7**	**+3.2**	**+1.0**	**+0.8**

Table 5: **Transfer learning** results on object detection: We fine-tune on Pascal VOC *trainval07+12* or *trainval07*, and test on VOC *test2007*. The detector is Faster R-CNN with ResNet50-C4. MoCo v2 model is pre-trained for 200 epochs. Note that our model outperforms SOTA methods without using an MLP head. Baseline results are copied from [24, 7].



(a) MoCo (b) MoCo+CLD

Figure 6: **t-SNE feature visualization** of (a) MoCo (b) MoCo + CLD on CIFAR10-LT. Tail class embedding is more compact and better separated from head classes. Head and medium-shot classes also have cleaner separation.

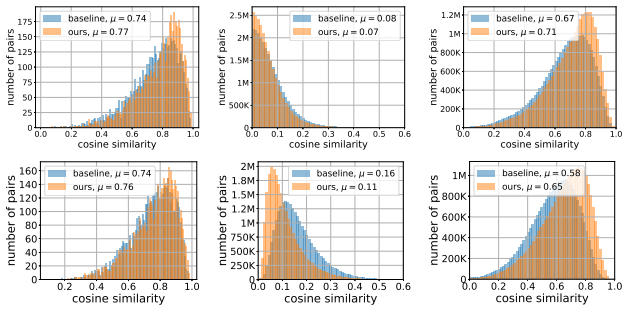
Similarity among positives / negatives? We measure fea-

CIFAR10		retrieval	NMI	kNN
NPID	f_I	75.1	57.7	80.8
CLD	f_I	**78.6**	63.5	**86.7**
	f_G	75.6	**69.0**	81.4
CIFAR100				
NPID	f_I	48.7	36.1	51.6
CLD	f_I	**50.2**	43.8	**57.5**
	f_G	48.8	**49.4**	51.8

Table 6: The **feature quality** of f_I and f_G evaluated by retrieval, normalized mutual information and kNN.

# groups	top-1
baseline	53.1
10	55.2
20	55.4
60	56.7
80	57.4
100	57.7
128	58.1

Table 7: **#groups vs. Accuracy** on CIFAR100 for CLD.



(a) **pos pairs:** A_{ii} (b) **neg pairs:** A_{ij} (c) **difference:** A^{Δ}_{ij}

Figure 7: CLD has more (dis)similar instances in positive(negative) pairs than baseline MoCo, creating a larger similarity gap. Columns 1-3 are the **histograms of cosine similarities** between positive and negative pairs and their differences per the linear projection layer for $f_I(x_i)$ (Row 1) and $f(x_i)$ (Row 2) on ImageNet100.

ture (cosine) similarity as $A_{ij}(f) =< \frac{f(x_i)}{\|f(x_i)\|}, \frac{f(x'_j)}{\|f(x_j)\|} >$, with A_{ii} ($A_{i,j\neq i}$) for positive (negative) pairs, and their gap is $A^{\Delta}_{ij} = A_{ii} - A_{ij}$. Fig. 7 shows that CLD has higher (lower) similarities between positives (negatives) than MoCo, creating larger gaps of A^{Δ}_{ij}, especially on $f(x_i)$ (Fig. 7 Row 2) – the common feature shared by our instance and group branches, making f a better discriminator than MoCo. It in turn improves f_I (Fig. 7 Row 1), the instance branch that runs parallel to the group branch f_G.

Mutual information characterization? We use kNN classification accuracy, Normalized Mutual Information (NMI), and retrieval accuracy R to compare features. $\text{NMI}(f, Y) = \frac{I(C|f,Y)}{\sqrt{H(C|f)H(Y)}}$ reflects global MI between feature f and downstream classification labels Y, where C is cluster labels predicted from k-Means clustering of f (k assuming the number of classes), $H(\cdot)$ is entropy, and $I(C|f;Y)$ is the MI between Y and C [46]. The top-1 retrieval accuracy $R(f, Y)$ reflects instance-level mutual information.

Table 6 shows that f_I is more accurate than f_G at retrievals and downstream classification. While f_G has higher NMI, its kNN accuracy is worse than f_I. That is, maximizing global MI would not deliver better downstream classification; maximizing instance-level MI is also important.

Unsupervised hyper-parameter tuning? Unsupervised learning is meant to draw inference from unlabeled data. However, its hyper-parameters such as our weight λ and temperature T are often selected by labeled data in the downstream task. Self-supervised feature learning benchmarks pass as a supervised shallow feature learner with a few hyper-parameters. We explore unsupervised hyper-parameter selection based entirely on the unlabeled data.

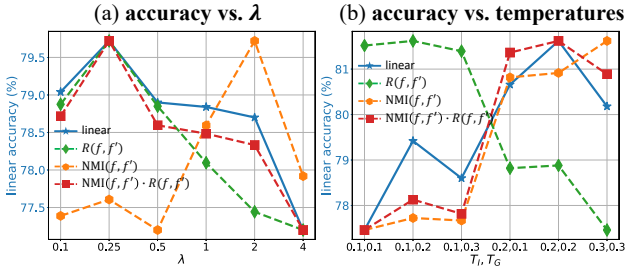


Figure 8: **Unsupervised hyper-parameter tuning** on ImageNet-100, for weight λ (left) and for the temperatures $T_{\text{I}}, T_{\text{G}}$ used in CLD (right). Unsupervised evaluation metric $\text{NMI}(f, f') \cdot R(f, f')$ ranks models similarly as supervised linear classification, corroborating our idea that both global mutual information and augmentation-invariant local information are important for downstream performance. Each curve is individually normalized.

We study how the supervised linear accuracy at the downstream can be indicated by unsupervised metrics such as NMI and R between feature $f(x)$ and $f' = f(x')$. Fig. 8 shows that the linear accuracy is well indicated by $R(f, f')$ for λ and by $\text{NMI}(f, f')$ for temperatures, but neither alone is sufficient. Their product $\text{NMI}(f, f') \cdot R(f, f')$ turns out to be a promising unsupervised evaluation metric.

5. Summary

We extend unsupervised learning to natural data with correlation and long-tail distributions by integrating local clustering into contrastive learning. It discovers between-instance similarity not by direct attraction and repulsion at the instance or group level, but cross-level between instances and groups. Their batch-wise and cross-view comparisons greatly improve the positive/negative sample ratio for achieving more invariant mapping. We also propose normalized projection heads and unsupervised hyper-parameter tuning.

Our extensive experimentation and analysis shows that CLD is a lean and powerful add-on to existing SOTA methods, delivering a significant performance boost on all the benchmarks and beating MoCo v2 and SimCLR on every reported performance with a much smaller compute.

Acknowledgments. This work was supported, in part, by Berkeley Deep Drive, US Government Fund through Etegent Technologies on Low-Shot Detection and Semi-supervised Detection, Texas Advanced Computing Center, and NTU NAP and A*STAR via Industry Alignment Fund.

References

[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NeurIPS*, 2019.

[2] Elena Bernardis and Stella X. Yu. Finding dots: Segmentation as popping out regions from boundaries. In *CVPR*, 2010.

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[4] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019.

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33, 2020.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

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

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020.

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

[10] Fernando De la Torre and Takeo Kanade. Discriminative cluster analysis. In *ICML*, 2006.

[11] 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.

[12] Chris Ding and Tao Li. Adaptive dimension reduction using discriminant analysis and k-means clustering. In *ICML*, 2007.

[13] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[14] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *ICLR*, 2017.

[15] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *TPAMI*, 2015.

[16] 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.

[17] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data clustering: theory, algorithms, and applications*. SIAM, 2007.

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

[19] Spyros Gidaris, Praveer Singh, Nikos Komodakis, et al. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[20] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. *arXiv preprint arXiv:1905.01235*, 2019.

[21] 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.

[22] Michael U Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 2012.

[23] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

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

[25] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[27] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

[28] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[29] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[30] Jyh-Jing Hwang, Stella X. Yu, Jianbo Shi, Maxwell D Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *ICCV*, 2019.

[31] Simon Jenni and Paolo Favaro. Self-supervised feature learning by learning to spot artifacts. In *CVPR*, 2018.

[32] Xu Ji, João F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019.

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

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[35] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *CVPR*, 2017.

[36] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020.

[37] 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.

[38] Michael Maire, Stella X. Yu, and Pietro Perona. Object detection and segmentation from joint embedding of parts and pixels. In *ICCV*, 2011.

[39] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. *arXiv preprint arXiv:1912.01991*, 2019.

[40] Feiping Nie, Zinan Zeng, Ivor W Tsang, Dong Xu, and Changshui Zhang. Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering. *IEEE Transactions on Neural Networks*, 2011.

[41] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

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

[43] Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 2003.

[44] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[45] Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A Alemi, and George Tucker. On variational bounds of mutual information. *arXiv preprint arXiv:1905.06922*, 2019.

[46] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 2002.

[47] Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. In *AAAI*, 2014.

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

[49] 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.

[50] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019.

[51] Laurens Van Der Maaten. Learning a parametric embedding by preserving local structure. In *Artificial Intelligence and Statistics*, 2009.

[52] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 2007.

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

[54] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, 2016.

[55] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, 2016.

[56] Yi Yang, Dong Xu, Feiping Nie, Shuicheng Yan, and Yueting Zhuang. Image clustering using local discriminant models and global integration. *TIP*, 2010.

[57] Jieping Ye, Zheng Zhao, and Mingrui Wu. Discriminative k-means for clustering. In *NIPs*, 2008.

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

[59] Stella X. Yu and Jianbo Shi. Segmentation with pairwise attraction and repulsion. In *ICCV*, 2001.

[60] Stella X. Yu and Jianbo Shi. Understanding popout through repulsion. In *CVPR*, 2001.

[61] Xiaohang Zhan, Xingang Pan, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised learning via conditional motion propagation. In *CVPR*, 2019.

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

[63] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

[64] Chengxu Zhuang, Alex Lin Zhai, Daniel Yamins, , et al. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, 2019.