CLUSTER CONTRAST FOR UNSUPERVISED VISUAL REPRESENTATION LEARNING

Nikolaos Giakoumoglou and Tania Stathaki

Department of Electrical and Electronic Engineering, Imperial College London, London, UK
{nikos, t.stathaki}@imperial.ac.uk

ABSTRACT

We introduce Cluster Contrast (CueCo), a novel approach to unsupervised visual representation learning that effectively combines the strengths of contrastive learning and clustering methods. Inspired by recent advancements, CueCo is designed to simultaneously scatter and align feature representations within the feature space. This method utilizes two neural networks, a query and a key, where the key network is updated through a slow-moving average of the query outputs. CueCo employs a contrastive loss to push dissimilar features apart, enhancing inter-class separation, and a clustering objective to pull together features of the same cluster, promoting intra-class compactness. Our method achieves 91.40% top-1 classification accuracy on CIFAR-10, 68.56% on CIFAR-100, and 78.65% on ImageNet-100 using linear evaluation with a ResNet-18 backbone. By integrating contrastive learning with clustering, CueCo sets a new direction for advancing unsupervised visual representation learning.

***Index Terms*—** Self-supervised learning, contrastive learning, clustering, unsupervised learning, representation learning

1. INTRODUCTION

Self-supervised learning focuses on extracting features without relying on manual annotations, increasingly closing the performance gap with supervised training techniques in computer vision. Recent state-of-the-art methods in contrastive learning treat each image and its variations as distinct classes, yielding representations that can discriminate between different images while achieving invariance to image transformations. Key to these methods are two fundamental components: (a) a contrastive loss mechanism [1] and (b) a series of image transformations [2]. Both elements are crucial for the performance of the resulting models [2, 3].

Unlike contrastive learning, which primarily focuses on intra-image invariance, clustering-based methods enable exploration of similarities between different images [4]. While initial assessments of these methods were generally conducted on smaller datasets [5], recent approaches have advanced clustering-based representation learning to larger scales [4]. Specifically, these methods generate pseudo-labels through clustering, which then serve as supervision for subsequent training [6]. Recent innovations have simplified this process by performing clustering and network updates simultaneously, enhancing the adaptation of the model to evolving data characteristics [7].

In this work, we introduce Cluster Contrast (CueCo), a novel self-supervised learning framework designed to enhance visual representation learning by leveraging the synergistic effects of contrastive learning and clustering methodologies (see Figure 1). Inspired by the "push-pull" dynamics observed in physical forces, CueCo utilizes a dual mechanism that optimizes the feature space: the contrastive component, leveraging the InfoNCE loss [8], effectively "pushes" dissimilar representations apart, akin to repulsive forces, ensuring that distinct classes are well-separated. Concurrently, the clustering component "pulls" similar representations closer, analogous to attractive forces, promoting tighter clusters of like data points.

We make the following **contributions**: **(a)** We introduce a novel unsupervised method that employs a dual mechanism to optimize the feature space through "push-pull" dynamics, effectively leveraging both contrastive and feature space clustering losses to refine feature discriminability; **(b)** We provide a comprehensive theoretical analysis of the objectives of our framework, illustrating how the integration of contrastive loss with feature space clustering losses cohesively enhances feature representation; **(c)** We demonstrate state-of-the-art performance on CIFAR-10, CIFAR-100, and ImageNet-100, underscoring the effectiveness of CueCo.

2. RELATED WORK

2.1. Contrastive Methods

Contrastive learning employs instance discrimination, where each image is treated as a unique class. This approach pulls together anchor and "positive" samples while pushing apart "negative" samples [2]. Positive pairs are generated through multiple views like augmentations [2, 9], patches [8], or teacher-student models [10, 11]. Training typically uses InfoNCE-based objectives [8, 2]. To address computational constraints, methods like MoCo [9] introduce memory structures. Recent approaches eliminate negative samples [10, 12] or use regularization [13, 14, 15] to prevent collapse, while others enhance learning through synthetic negatives [16].

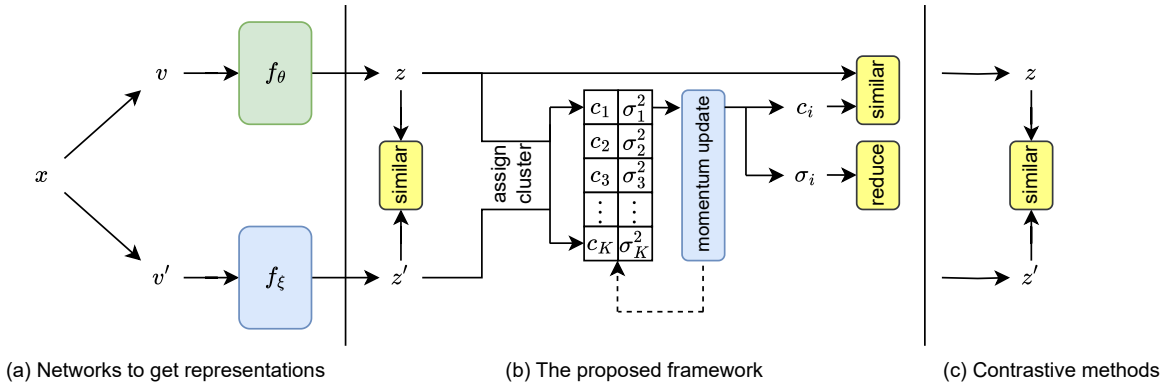

Fig. 1. CueCo framework. Like typical contrastive methods, CueCo processes two augmented views, v and v', of the same image x through encoders f_θ and f_ξ, producing feature vectors $\mathbf{z} = f_\theta(v)$ and $\mathbf{z}' = f_\xi(v')$. In addition to that, CueCo: (a) enforces inter-class separation via contrastive loss, (b) improves intra-class cohesion through clustering, and (c) integrates both for refined feature representation.

2.2. Clustering Methods

Clustering approaches like DeepCluster [6] use K-means to form groups and learn features iteratively. While ODC [17] and CoKe [18] attempt to improve this process, the inherent delay in supervisory signals remains problematic. SwAV [4] reframes grouping as pseudo-labelling with optimal transport, though their equipartition constraints may reduce grouping effectiveness. Alternative approaches include prototype-based methods [19], momentum clusters [7], and contrastive clustering [20] which treats rows and columns of feature matrices as instance and cluster representations directly. However, our approach differs by introducing separate instance-level and cluster-level queues with both soft and hard assignments, providing more flexible feature learning.

3. METHODOLOGY

3.1. Contrastive Learning

Contrastive learning serves as the backbone of our framework, primarily focusing on scattering features across the feature space to enhance their discriminability. Contrastive learning seeks to differentiate between similar and dissimilar data pairs. We utilize a dual-crop strategy, similar to the methods in [2, 9], with random data augmentations. Given an image x, and two distributions of image augmentation $\mathcal{T}$ and $\mathcal{T}'$, we create two augmented views of the same image using the transformation $t \sim \mathcal{T}$ and $t' \sim \mathcal{T}'$, i.e., $v = t(x)$ and $v' = t'(x)$. Two encoders, f_θ and f_ξ, generate the vectors $\mathbf{z} = f_\theta(v)$ and $\mathbf{z}' = f_\xi(v')$ respectively. The learning objective minimizes a contrastive loss using the InfoNCE criterion [8]:

$$\mathcal{L}_1(\mathbf{z}, \mathbf{z}') = -\log \frac{\exp(\mathbf{z}^\top \cdot \mathbf{z}'/\tau)}{\exp(\mathbf{z}^\top \cdot \mathbf{z}'/\tau) + \sum_{k=1}^{K} \exp(\mathbf{z}^\top \cdot \mathbf{z}_k/\tau)} \tag{1}$$

where, $\mathbf{z}'$ is f_ξ's output from the same augmented image as $\mathbf{z}$, and $\{\mathbf{z}_k\}_{k=1}^K$ includes outputs from different images, representing negative samples. The temperature parameter τ adjusts scaling for the ℓ_2-normalized vectors $\mathbf{z}$ and $\mathbf{z}'$. This loss function scatters the feature representations by forcing them to occupy distinct points in the feature space, reducing the likelihood of different instances collapsing into indistinguishable points. Following [9, 21], we maintain a queue of negative keys, refreshing only the queries and positive keys in each training batch. A momentum encoder updates the base encoder to ensure consistent representations across updates $\xi \leftarrow m \cdot \xi + (1-m) \cdot \theta$, where m is the momentum coefficient [9, 10].

3.2. Feature Space Clustering

Beyond contrastive learning, our goal is to develop an encoder that groups images of the same class into distinct regions of the feature space. While contrastive learning spreads the feature vectors of different classes, we introduce an objective to pull together features of the same class, forming tight clusters. Since true class labels are not available during pretraining, we apply a clustering algorithm, such as K-means, to infer pseudo-classes. We then gather pre-softmax features, termed activation vectors, from images classified under the same pseudo-class. With this, we can store a running average class prototype, or mean activation vector, $\mathbf{c_i}$, given a key $\mathbf{z}' = f_\xi(x)$ that

belongs to the set S_i of features assigned to i-th cluster, along the per-class variance σ_i^2 via sum of squares as follows:

$$\mathbf{c}_i = \frac{1}{|S_i|} \sum_{\mathbf{z}' \in S_i} \mathbf{z}' \quad \text{and} \quad \sigma_i^2 = \frac{1}{|S_i|} \sum_{\mathbf{z}' \in S_i} (\mathbf{z}' - \mathbf{c}_i)^2 \tag{2}$$

In the clustering-based approach to learning representations, the first step involves assigning each feature vector $\mathbf{z}$ to a cluster centroid. This assignment is determined by:

$$\mathbf{c}_{i[\mathbf{z}]} = \underset{\mathbf{c}_l}{\operatorname{argmin}} \|\mathbf{z} - \mathbf{c}_l\| \tag{3}$$

where $\mathbf{c}_l$ are the available centroids and $\mathbf{c}_i$ is the normalized mean vector of the cluster to which $\mathbf{z}$ is closest in terms of Euclidean distance. After this assignment, the learning objectives can be applied to refine the feature vectors with respect to their clusters.

The first clustering objective, the centroid contrastive loss, aligns features with the centroid of their assigned cluster, ensuring they remain distinct from centroids of other clusters. This is achieved with an InfoNCE objective [8] that measures the similarity between the features and their respective cluster centroids:

$$\mathcal{L}_2(\mathbf{z}, \mathbf{c}_{i[\mathbf{z}]}) = -\log \frac{\exp(\mathbf{z}^\top \cdot \mathbf{c}_{i[\mathbf{z}]}/\tau)}{\exp(\mathbf{z}^\top \cdot \mathbf{c}_{i[\mathbf{z}]}/\tau) + \sum_{l=1}^{L} \exp(\mathbf{z}^\top \cdot \mathbf{c}_l/\tau)} \tag{4}$$

where $\mathbf{c}_{i[\mathbf{z}]}$ is the centroid to which $\mathbf{z}$ has been assigned, $\mathbf{c}_l$ are the centroids of other clusters, and τ is a temperature parameter that adjusts the ℓ_2 normalized vectors (can be different from the temperature in contrastive loss). This loss encourages each feature vector $\mathbf{z}$ to align closely with its own cluster centroid while remaining distinct from the centroids of other clusters, effectively refining the granularity of the learned representations.

The second objective, namely the variance loss, minimizes the squared Euclidean distance between features and their corresponding cluster centroid, scaled by the variance of the cluster:

$$\mathcal{L}_3(\mathbf{z}, \mathbf{c}_{i[\mathbf{z}]}) = \frac{\left\|\mathbf{z} - \mathbf{c}_{i[\mathbf{z}]}\right\|^2}{2 \cdot \sigma_{i[\mathbf{z}]}^2 + \epsilon} \tag{5}$$

where σ_i^2 is the variance of the cluster to which $\mathbf{z}$ is assigned and $\epsilon = 10^{-6}$ is a small constant to prevent division by zero. This objective ensures that not only do the features align with the centroids, but they also conform to the cluster's overall distribution, promoting uniformity within clusters and enhancing the robustness of the clustering.

3.3. Final Objective

The encoder minimizes a combination of the loss functions:

$$\mathcal{L} = \lambda_1 \cdot \mathcal{L}_1 + \lambda_2 \cdot \mathcal{L}_2 + \lambda_3 \cdot \mathcal{L}_3 \tag{6}$$

where $\lambda_1, \lambda_2, \lambda_3$ are coefficients that balance the impact of each loss component on the representation learning. After training, only the convolutional encoder of f_θ is as in [9], utilizing f_ξ during training to compute mean vectors and variances. The encoder f_ξ maintains a slow-moving average of f_θ, which stabilizes the feature representations over time and ensures a consistent interpretation of the feature space. This consistency is critical for maintaining reliable cluster centroids and standard deviations, thereby enhancing the overall robustness and accuracy of the clustering mechanism. Both the centroid contrastive and variance losses during the forward pass are computed using the outputs from f_θ, as the parameters need to be updated based on the current, directly observed representations, rather than the averaged historical data.

3.4. Towards Online Feature Space Clustering

Using an offline clustering algorithm like K-means is restrictive because it requires that the algorithm be applied at the end of each epoch to use the resulting cluster centroids and standard deviations for training objectives. Moreover, once established, the clustering remains unchanged for the duration of the epoch, rendering it static. This approach becomes particularly impractical for large datasets. To make the clustering algorithm even more tractable, we apply K-means in the queue to get the cluster centers and cluster standard deviations. Furthermore, cluster centroids need to reflect the most current instances to facilitate gradient propagation and must be updated in sync with f_θ using a differentiable function [7]. To address these challenges, we introduce a momentum grouping scheme, similar to [7]. We initialize the cluster features $\{\mathbf{c}_1, \mathbf{c}_2, \dots \mathbf{c}_L\}$ randomly or via a method such as K-means and throughout the training process, we continuously update the centroids $\mathbf{c}_i$ and standard deviations σ_i at each iteration as follows:

$$\mathbf{c}_i \leftarrow \beta_1 \cdot \mathbf{c}_i + (1 - \beta_1) \cdot \left(\frac{1}{|S_i|} \sum_{\mathbf{z} \in S_i} \mathbf{z}' \right) \tag{7}$$

$$\sigma_i^2 \leftarrow \beta_2 \cdot \sigma_i^2 + (1 - \beta_2) \cdot \left(\frac{1}{|S_i|} \sum_{\mathbf{z} \in S_i} (\mathbf{z}' - \mu_i)^2 \right) \tag{8}$$

where μ_i is the mean vector of features in cluster i and β_1, β_2 represent the momentum ratios. The momentum grouping method dynamically assigns each instance to the nearest centroid and updates the centroid and standard deviation features in a momentum-based manner. Consequently, the centroid and standard deviation features consistently represent the most recent visual characteristics of the instances.

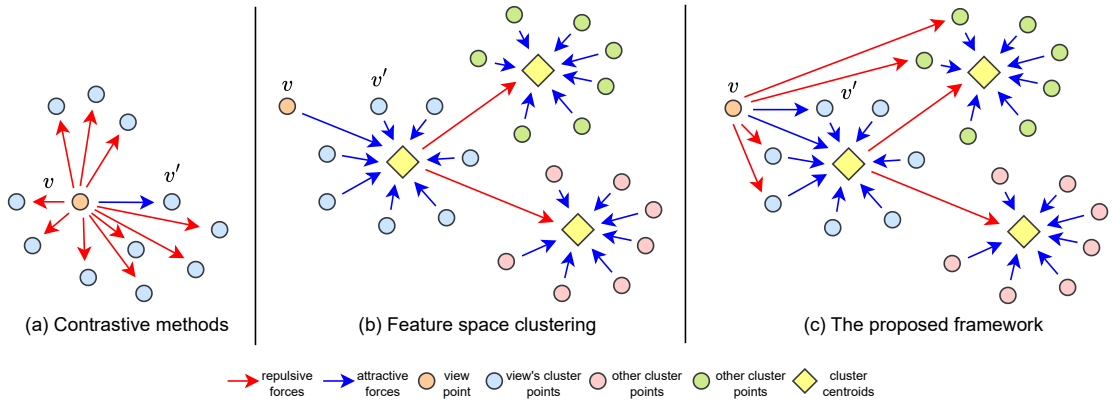

Fig. 2. Visualization of the repulsive and attractive forces. (a) Illustrates the contrastive objective, emphasizing class separation (Section 3.1). (b) Applies feature space clustering to improve intra-class compactness (Section 3.2). (c) Demonstrates the CueCo framework, merging (a) and (b).

3.5. Intuition on Behavior

The dynamics within CueCo resemble the forces in electromagnetism, where embeddings are metaphorically "charged" to repel or attract each other based on the model's losses and architecture. The contrastive loss acts as a repulsive force, pushing apart embeddings of dissimilar instances to expand the feature space, akin to like charges repelling. In contrast, clustering-driven losses act as an attractive force, pulling together embeddings of similar instances to form cohesive clusters, similar to oppositely charged particles attracting. As illustrated in Figure 2, repulsive forces ensure wide separation among classes, while attractive forces (e.g., with $\lambda_2, \lambda_3 > 0$) pull embeddings into tightly-knit groups. The ultimate goal is to reach an equilibrium after t training steps, where embeddings $\mathbf{z}_{t,Ti} = f(x_i; w_t)$ achieve optimal separation between dissimilar classes and cohesion within similar classes, resulting in robust, accurate representations.

Table 1. Linear top-1 and top-5 accuracies (%) obtained through linear evaluation protocol on CIFAR-10, CIFAR-100, and ImageNet-100 datasets using ResNet-18 as the backbone network. Results adapted from [22].

Method	CIFAR-10		CIFAR-100		ImageNet-100	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
BYOL [10]	92.61	99.82	70.18	91.36	80.09	94.99
DINO [11]	89.19	99.31	66.38	90.18	74.84	92.92
SimSiam [12]	90.51	99.72	65.86	89.48	77.04	94.02
MoCo-v2 [21]	92.94	99.79	69.54	91.49	78.20	95.50
MoCo-v3 [23]	93.10	99.80	68.83	90.57	80.86	95.18
ReSSL [24]	90.63	99.62	65.83	89.51	76.59	94.41
VICReg [14]	90.07	99.71	68.54	90.83	79.22	95.06
SwAV [4]	89.17	99.68	64.67	88.52	74.28	92.84
SimCLR [2]	90.74	99.75	65.39	88.58	77.48	93.42
BT [13]	89.57	99.73	69.18	91.19	78.62	94.72
CueCo (ours)	91.40	99.79	68.56	91.05	78.65	94.03

4. EXPERIMENTS

4.1. Implementation Details

We evaluate CueCo on CIFAR-10, CIFAR-100 [25], and ImageNet-100 [26]. Each input image is transformed twice to generate two different views using the same augmentation as used in [10]. Our encoder f_θ includes a ResNet-18 backbone (adapted by replacing the 7×7 convolution (`conv1`) with a 3×3 convolution and removing the max pooling layer), a 2-layer MLP projection head, and a prediction head [23]. The encoder f_ξ shares the backbone and projection head but excludes the prediction head and is updated as a moving average of f_θ [9, 10]. Both MLPs have hidden layers of 4096-d with ReLU [27], 256-d output layers without ReLU, and batch normalization [28]. For pretraining we use the SGD optimizer [29] with a base learning rate of 0.3 (decreased to 0 with cosine schedule), momentum of 0.9, and weight decay of 10^{-4}. Clustering is frozen for the first 313 iterations to prevent instability, and cluster features are reset every 1,000 iterations to avoid collapse and maintain balanced cluster distributions (following [4, 7]). We train CIFAR-10/1000 for 1000 epochs and ImageNet-100 for 400 epochs. We use a FIFO queue of size 16,384 from which we calculate the cluster centroids and variances (but updated via momentum to reduce computational overhead), and the momentum coefficient for the moving average encoder starts at 0.996 and increases to 1.0 using a cosine schedule. All temperatures are set to $\tau = 0.2$, and loss weights are $\lambda_1 = 1$, $\lambda_2 = 0.1$, and $\lambda_3 = 0.01$.

Table 2. Unsupervised image classification results on CIFAR-10 and CIFAR-100 datasets. We report standard clustering metrics including Normalized Mutual Information (NMI), Adjusted Mutual Information (AMI), Adjusted Rand Index (ARI), and clustering accuracy (ACC). All methods are reimplemented.

Method	CIFAR-10				CIFAR-100			
	NMI	AMI	ARI	ACC	NMI	AMI	ARI	ACC
MoCo-v2 [21] (repr.)	60.96	60.88	28.95	63.51	51.77	45.84	9.95	31.72
SimCLR [2] (repr.)	69.03	68.98	53.14	74.50	50.75	44.60	11.15	32.17
CueCo (ours)	**69.33**	**69.01**	**53.87**	**75.06**	**52.37**	**46.31**	**11.35**	**33.82**

Table 3. Ablation studies on CIFAR-100 evaluating the impact of different loss terms. We report linear evaluation accuracies (top-1, top-5), k-nearest neighbor accuracies (20-NN, 100-NN), and clustering metrics (NMI, AMI, ARI, ACC). Checkmarks indicate which loss terms are active. All methods are reimplemented.

λ_1	λ_2	λ_3	Top-1	Top-5	20-NN	100-NN	NMI	AMI	ARI	ACC
✓			67.9	90.7	66.5	66.4	50.5	44.4	8.3	31.1
✓	✓		66.9	90.6	63.9	62.4	**54.6**	**48.6**	**14.9**	**34.5**
✓		✓	68.4	90.7	66.7	66.8	51.2	45.1	8.6	32.3
✓	✓	✓	**68.5**	**91.0**	**66.8**	**67.0**	52.3	46.3	11.3	33.8

4.2. Linear Evaluation

We evaluate CueCo's representations by training a linear classifier on top of the frozen features from the ResNet-18 encoder. The linear classifier is trained for 100 epochs using a learning rate of 3.0 with a cosine learning rate scheduler. Training minimizes the cross-entropy loss with an SGD optimizer, momentum of 0.9, and weight decay of 1×10^{-6}, using a batch size of 256. We report top-1 and top-5 accuracies as percentages on the test set in Table 1, comparing our results to previous state-of-the-art self-supervised methods from [22]. Our method performs competitively with state-of-the-art self-supervised approaches, while not being heavily tuned.

4.3. Unsupervised Image Classification

We evaluate our approach on the task of unsupervised image classification for CIFAR-10/100. We report the standard clustering metrics: Normalized Mutual Information (NMI), Adjusted Normalized Mutual Information (AMI), Adjusted Rand-Index (ARI), and Clustering Accuracy (ACC), as in [30]. We compare our approach with reproductions of MoCo-v2 [21] and SimCLR [2], using the same hyperparameter settings for a fair comparison. As shown in Table 2, our method, CueCo, consistently outperforms the compared methods across all metrics on both CIFAR-10 and CIFAR-100.

4.4. Ablation Study

We investigate the importance of each objective in our method, focusing on contrastive, centroid contrastive, and variance reduction losses. Table 3 presents results on CIFAR-100 using different loss combinations. Contrastive loss alone provides a strong baseline, while adding centroid contrastive loss improves clustering metrics. Incorporating variance reduction loss further enhances linear evaluation and unsupervised classification. These findings highlight the effectiveness of our multi-objective approach in producing versatile and discriminative representations by balancing instance-level distinctiveness and class-level coherence.

5. CONCLUSION

CueCo advances unsupervised visual representation learning by integrating contrastive learning with momentum clustering, creating a "push-pull" dynamic that simultaneously enhances inter-class separation and intra-class cohesion. The framework demonstrates competitive performance on benchmark datasets while particularly excelling in unsupervised image classification metrics. By balancing these complementary forces, CueCo establishes a promising direction for self-supervised learning that produces more discriminative representations with improved class structure without requiring labeled data.

6. REFERENCES

[1] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006, vol. 2, pp. 1735–1742.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," 2020.

[3] Ishan Misra and Laurens van der Maaten, "Self-

supervised learning of pretext-invariant representations," 2019.

[4] 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*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, Eds. 2020, vol. 33, pp. 9912–9924, Curran Associates, Inc.

[5] Junyuan Xie, Ross Girshick, and Ali Farhadi, "Unsupervised deep embedding for clustering analysis," 2016.

[6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze, "Deep clustering for unsupervised learning of visual features," 2019.

[7] Bo Pang, Yifan Zhang, Yaoyi Li, Jia Cai, and Cewu Lu, "Unsupervised visual representation learning by synchronous momentum grouping," 2022.

[8] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," 2019.

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

[10] 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, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko, "Bootstrap your own latent: A new approach to self-supervised learning," 2020.

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin, "Emerging properties in self-supervised vision transformers," 2021.

[12] Xinlei Chen and Kaiming He, "Exploring simple siamese representation learning," 2020.

[13] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny, "Barlow twins: Self-supervised learning via redundancy reduction," 2021.

[14] Adrien Bardes, Jean Ponce, and Yann LeCun, "Vicreg: Variance-invariance-covariance regularization for self-supervised learning," 2022.

[15] Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell, "Representation learning via invariant causal mechanisms," 2020.

[16] Nikolaos Giakoumoglou and Tania Stathaki, "Synco: Synthetic hard negatives in contrastive learning for better unsupervised visual representations," 2024.

[17] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew Soon Ong, and Chen Change Loy, "Online deep clustering for unsupervised representation learning," 2020.

[18] Qi Qian, Yuanhong Xu, Juhua Hu, Hao Li, and Rong Jin, "Unsupervised visual representation learning by online constrained k-means," 2022.

[19] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C. H. Hoi, "Prototypical contrastive learning of unsupervised representations," 2021.

[20] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng, "Contrastive clustering," 2020.

[21] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He, "Improved baselines with momentum contrastive learning," 2020.

[22] Victor Guilherme Turrisi da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci, "solo-learn: A library of self-supervised methods for visual representation learning," *Journal of Machine Learning Research*, vol. 23, no. 56, pp. 1–6, 2022.

[23] Xinlei Chen, Saining Xie, and Kaiming He, "An empirical study of training self-supervised vision transformers," 2021.

[24] Mingkai Zheng, Shan You, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu, "Ressl: Relational self-supervised learning with weak augmentation," 2021.

[25] Alex Krizhevsky, "Learning multiple layers of features from tiny images," pp. 32–33, 2009.

[26] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[27] Vinod Nair and Geoffrey Hinton, "Rectified linear units improve restricted boltzmann machines vinod nair," 06 2010, vol. 27, pp. 807–814.

[28] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[29] Sebastian Ruder, "An overview of gradient descent optimization algorithms," 2017.

[30] Elad Amrani, Leonid Karlinsky, and Alex Bronstein, "Self-supervised classification network," 2022.