8-2022

Unsupervised Contrastive Representation Learning for Knowledge Distillation and Clustering

Fei Ding
feid@g.clemson.edu

Unsupervised Contrastive Representation Learning for Knowledge Distillation and Clustering

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Computer Science

by
Fei Ding
August 2022

Accepted by:
Dr. Feng Luo, Committee Chair
Dr. Yin Yang
Dr. Kai Liu
Dr. Nianyi Li

Abstract

Unsupervised contrastive learning has emerged as an important training strategy to learn representation by pulling positive samples closer and pushing negative samples apart in low-dimensional latent space. Usually, positive samples are the augmented versions of the same input and negative samples are from different inputs. Once the low-dimensional representations are learned, further analysis, such as clustering, and classification can be performed using the representations. Currently, there are two challenges in this framework. First, the empirical studies reveal that even though contrastive learning methods show great progress in representation learning on large model training, they do not work well for small models. Second, this framework has achieved excellent clustering results on small datasets but has limitations on the datasets with a large number of clusters such as ImageNet. In this dissertation, our research goal is to develop new unsupervised contrastive representation learning methods and apply them to knowledge distillation and clustering.

The knowledge distillation transfers knowledge from high-capacity teachers to small student models and then improves the performance of students. And the representational knowledge distillation methods try to distill the knowledge of representations from teachers to students. Current representational knowledge distillation methods undesirably push apart representations of samples from the same class in their correlation objectives, leading to inferior distillation results. Here, we introduce the Dual-level Knowledge Distillation (DLKD) by explicitly combining knowledge alignment and knowledge correlation instead of using one single contrastive objective. We show that both knowledge alignment and knowledge correlation are necessary to improve distillation performance. The proposed DLKD is task-agnostic and model-agnostic and enables effective knowledge transfer from supervised or self-supervised trained teachers to students. Experiments demonstrate that DLKD outperforms other state-of-the-art methods in a large number of experimental settings including different (a) pretraining strategies (b) network architectures (c) datasets (d) tasks.

Currently, the two-stage framework is widely used in deep learning-based clustering, namely, learning representation first, then clustering algorithms, such as K-means, are usually performed on representations to obtain cluster assignment. However, the learned representation may not be optimized for clustering in this two-stage framework. Here, we propose Contrastive Learning-based Clustering (CLC), which uses contrastive learning to directly learn cluster assignment. We decompose the representation into two parts: one encodes the categorical information under an equipartition constraint, and the other captures the instance-wise factors. We theoretically analyze the proposed contrastive loss and reveal that CLC sets different weights for the negative samples while learning cluster assignments. Therefore, the proposed loss has high expressiveness that enables us to efficiently learn cluster assignments. Experimental evaluation shows that CLC achieves overall state-of-the-art or highly competitive clustering performance on multiple benchmark datasets. In particular, we achieve 53.4% accuracy on the full ImageNet dataset and outperform existing methods by large margins (+ 10.2%).

Dedication

I would like to dedicate this work to my family, especially my parents, my wife, and my lovely daughter.

Acknowledgments

First and foremost, I would like to express my heartfelt appreciation to my advisor Dr. Feng Luo for his invaluable advice, continuous support, and endless patience during my Ph.D. study. He has inspired me to become an independent researcher and helped me think about how to do research in a right and better way. His immense knowledge and plentiful experience have encouraged me all the time in my research and daily life. Without his tremendous understanding and support over the past six years, it would be impossible for me to complete my study. My sincere thanks must also go to the members of my committee: Dr. Yin Yang, Dr. Kai Liu, and Dr. Nianyi Li. They generously took their time to provide valuable comments toward improving my work.

I am also grateful to the faculties and staff in the School of Computing at Clemson University. Thank you, Dr. Rong Ge, for providing the comprehensive guidance to build the power consumption measurement device for the IoT project. Thank you, Dr. Hongxin Hu and Dr. Long Cheng, for your pointed sharp observations and criticisms during the IoT paper discussions. Dr. Ilya Safro, thanks for constructive suggestions for the graph paper. Special thanks go to my research collaborators: Dr. Hai Xiao, Dr. Yin Yang, Dr. Venkat Krovi, Dr. Jianhua Tong, and Dr. Gang Li. It was a great pleasure working with you. Thanks to Dr. Mark Smotherman and Adam Rollins for being so supportive in completing my M.S. and Ph.D. programs.

I would like to express my gratitude to my friends, lab mates, and research team - Dr. Yunsheng Wang, Yu Li, Linxiong Liu, Nushrat Humaira, Dr. Bo Wu, Dr. Hongda Li, Dr. Justin Sybrandt, Mingqi Li, Dan Zhang, Melanie Lambert, Bradley Sanders, Xiaoyan Han, Ximei Zhai, and Yunpeng Wu for a cherished time spent together in the lab, and for their unfailing emotional support.

Last, I deeply thank my family, especially my parents, wife, and daughter for their unconditional support and timely encouragement all through my studies.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Given a machine learning task and enough data with labels, supervised learning can achieve satisfactory performance, but requires manually collecting huge amounts of high-quality labels, which is expensive and does not easily scale up. Considering that the amount of unlabeled data far exceeds the amount of labeled data, recent research in deep learning has focused on unsupervised representational learning. The goal of unsupervised representation learning is to use various pretext tasks to obtain feature representations from data without expensive manual labels. These learned representations can capture good semantic or structural meanings, which are beneficial to various downstream tasks.

There are numerous pretext tasks in the literature, including the patch context prediction [29, 82], solving jigsaw puzzles [85, 86], predicting rotations [39], adversarial training [30, 31], and so on. The autoencoding, which ensures an approximate one-to-one mapping between individual inputs and feature representations, can also be considered as a kind of pretext task to learn latent representation. This prevents the learned representation from collapsing to a single point, while similar functionality is achieved in contrastive learning by negative samples. The implementation of autoencoding usually includes two neural networks: an encoder infers the latent variable given the input and a decoder maps the latent variable to the data space. One of the popular pretext tasks is instance contrastive learning [32]. The key point of contrast learning is to create two random views from each training sample, called the positive and anchor sample, and select one of the other training samples as the negative. Usually, positive samples are the augmented versions of the same input and negative samples are from different inputs.

Once the low-dimensional representations are learned, further analysis, such as clustering, and classification can be performed using the representations. Currently, there are two challenges in this framework. First, the empirical studies reveal that even though contrastive learning methods show great progress in representation learning on large model training, they do not work well for small models. Second, this framework has achieved excellent clustering results on small datasets but has limitations on the datasets with a large number of clusters such as ImageNet. In this dissertation, our research goal is to develop new unsupervised contrastive representation learning methods and apply them to knowledge distillation and clustering.

1.1 Contrastive Representation Learning

The instance-wise contrastive objectives [115, 102, 51, 14, 10, 70, 17] have emerged as an important training strategy to learn well-clustered representation via pulling positive samples closer and pushing negative samples apart in the representation space. This Instance-level method considers each sample in the dataset as its own class. Wu *et al.* [115] first propose to use a memory bank to store previously-calculated features and utilize the noise contrastive estimation to learn feature representations. He *et al.* [51] continue to improve the training strategy by introducing momentum updates, which enables to build a large and consistent representations. SimCLR [14] is another representative line of works, which apply a large batch instead of the memory bank. In a sense, we can still consider that the autoencoding uses positive samples, which do not come from data augmentation, but the sample reconstruction. A group of contrastive learning methods that rely only on positive samples [44, 10, 126] has recently emerged. For example, they come from the augmentation of inputs. These methods avoid the need for negative samples by regularizing the dataset-level statistics of the feature representation.

While contrastive learning methods have made great progress in training large models, it does not apply to small models [36]. The most likely reason is that smaller models with fewer parameters cannot effectively capture instance-level discriminative information with large amounts of samples. In addition, most of the existing methods still suffer from a major limitation [70] due to the unsupervised setting. Some negative pairs from the same class should be closer in the representation space, but are undesirably pushed apart by the contrastive objective.

Knowledge Distillation (KD) provides a promising solution to build lightweight models by transferring knowledge from high-capacity teachers with additional supervision signals [8, 56]. Existing KD methods focus on either knowledge alignment or knowledge correlation according to whether the transferred knowledge comes from an individual sample or across samples. The original KD minimizes the KL-divergence loss between the probabilistic outputs of teacher and student networks. This objective aims to transfer the dark knowledge [56], *i.e.*, the assignments of relative probabilities to incorrect classes. Our analysis demonstrates that this logit matching solution performs knowledge alignment for an individual sample. Recently, CRD [97] has been proposed to learn the structural representational knowledge based on the contrastive objective. SEED [36] is another contrastive distillation method to encourage the student to learn from self-supervised pretrained teachers. By analyzing these recent representational knowledge distillation methods, we find that their correlation objectives undesirably push apart representations of samples from the same class, leading to inferior distillation results.

1.2 Clustering

As an important unsupervised learning method, clustering has been widely used in many computer vision applications, such as image segmentation [22], visual features learning [9], and 3D object recognition [107]. Clustering becomes difficult when processing large amounts of high-semantic and high-dimensional data samples [80]. For example, an image usually consists of thousands of pixels, massive images need to be processed in a reasonable time, and images containing the same object may not have any similarities from the pixel level. To overcome these challenges, many latent space clustering approaches such as DEC [117], DCN [119], and ClusterGAN [81], have been proposed. In these latent space clustering methods, the original high-dimensional data is first projected to low-dimensional feature representation, then clustering algorithms, such as K-means [76], are performed on the latent space. To avoid learning the random discriminative representations, their training objectives are usually coupled with data reconstruction loss or data generation constraints, which allows to rebuild or generate the input samples from the latent space. These objectives force the latent space to capture all key factors of variations and similarities, which are essential for reconstruction or generation. Therefore, these learned low-dimensional representations are not just related to clusters, and may not be the optimal latent representations for clustering. In addition, IIC [61] and IMSAT [59] propose to learn the clusters assignment by maximizing the mutual information between features of images and their augmented version. Since these clustering methods rely on network initialization, they may focus on low-level features, such as color and texture, which are prone to degenerate solutions [9]. It's still difficult to effectively integrate low-dimensional representation learning and clustering algorithm. The performance of distance-based clustering algorithms, such as K-means [76], are highly dependent on the selection of proper similarity and distance measures. Although constructing latent space can alleviate the problem of computing the distance between high-dimensional data, defining a proper distance in latent space is still central to obtaining superior clustering performance.

Much of the recent research also combines representation learning and clustering together. DeepCluster [9] and Self-labelling [4] propose to combine clustering and representation learning together as the pretext task. Although their goals are to learn good representations from unlabeled samples, we can also consider them as clustering methods that can be trained in an end-to-end manner. In general, these methods perform continuous iterative optimization of the clusters by obtaining supervised signals from the most confident samples. However, these methods that rely on the initial feature representation of the network are prone to degenerate solutions. The current state-of-the-art clustering method is SCAN [104], which proposes a two-step approach including feature representation learning and clustering. First, SCAN uses a contrastive representation learning task to obtain semantically meaningful representation. Second, it uses the assumption that nearest neighbors tend to belong to the same class as a prior for learning clusters assignment. But this assumption is not always true, considering the representation quality of existing contrastive learning methods. Therefore, SCAN shows excellent results on small datasets but still has limitations on large datasets, such as ImageNet.

1.3 Research Questions

The two-stage framework is widely used in deep learning-based clustering, namely, learning representation first, then clustering algorithms, such as K-means [76], are usually performed on representations to obtain cluster assignment. However, there are two challenges in this framework.

- The empirical studies reveal that even though contrastive learning methods show great progress in representation learning on large model training, they do not work well for small models.

- This framework has achieved excellent clustering results on small datasets but has limitations on the datasets with a large number of clusters such as ImageNet.

In this dissertation, our research goal is to develop new unsupervised contrastive representation learning methods and apply them to knowledge distillation (Chapter 2) and clustering (Chapter 3). In addition, there are several scenarios where data augmentation is difficult to implement, such as single-cell RNA sequencing data. We propose to achieve clustering via unsupervised conditional generation, which directly learns cluster assignments from disentangled latent space without additional clustering methods (Chapter 4). We identify the future works and conclude the dissertation in Chapter 5.

Chapter 2

Dual-level Knowledge Distillation via Knowledge Alignment and Correlation

To improve feature representations on small models, we employ knowledge distillation which provides a promising solution by transferring knowledge from high-capacity teachers. By analyzing the recent representational knowledge distillation methods, we find that their correlation objectives undesirably push apart representations of samples from the same class, leading to inferior distillation results. Thus, we introduce the Dual-level Knowledge Distillation (DLKD) by explicitly combining knowledge alignment and correlation together instead of using one single contrastive objective. We show that both knowledge alignment and correlation are necessary to improve the distillation performance. The proposed DLKD is task- agnostic and model-agnostic, and enables effective knowledge transfer from supervised or self-supervised pretrained teachers to students. Experiments demonstrate that DLKD outperforms other state-of-the-art methods on a large number of experimental settings including different (a) pretraining strategies (b) network architectures (c) datasets (d) tasks.

2.1 Related Work

Knowledge Distillation. Hinton *et al.* [56] first propose KD to transfer dark knowledge from the teacher to the student. The softmax outputs encode richer knowledge than one-hot labels and can provide extra supervisory signals. SRRL [121] performs knowledge distillation by leveraging the teacher's projection matrix to train the student's representation via L2 loss. However, these works rely on a supervised pretrained teacher (with logits), and they may not be suitable for self-supervised pretrained teachers. SSKD [118] is proposed to combine the self-supervised auxiliary task and KD to transfer richer dark knowledge, but it cannot be trained in an end-to-end training way. Similar to logits matching, intermediate representation [92, 124, 123, 101, 53] are widely used for KD. FitNet [92] proposes to match the whole feature maps, which is difficult and may affect the convergence of the student in some cases. Attention transfer [124] utilizes spatial attention maps as the supervisory signal. In flow-based distillation [123], inter-layer flow matrices of the teacher are computed to guide the learning of the student. AB [53] proposes to learn the activation boundaries of the hidden neurons in the teacher. SP [101] focuses on transferring the similar (dissimilar) activations between the teacher and student. However, most of these works depend on certain architectures, such as convolutional networks. Since these distillation methods involve knowledge matching in an individual sample, they are related to knowledge alignment. Our work also includes the knowledge alignment objective, but doesn't rely on pretraining strategies or network architectures.

2.1.1 Knowledge alignment and self-supervised learning

Self-supervised learning [88, 5, 14, 51, 10] focuses on learning low-dimensional representations by the instance discrimination, which usually requires a large number of negative samples. Recently, BYOL [44] and DINO [11] utilize the momentum encoder to avoid collapse without negatives. The momentum encoder can be considered as the mean teacher [96], which is built dynamically during the student training. For distillation, the teacher is pretrained and fixed during distillation. Although different views (augmented images) are passed through networks in self-supervised learning, they are from the same original sample for feature alignment. These self-supervised methods perform knowledge alignment between the student and the momentum teacher during each iteration. In particular, DINO focuses on local-to-global knowledge alignment based on multi-crop augmentation.

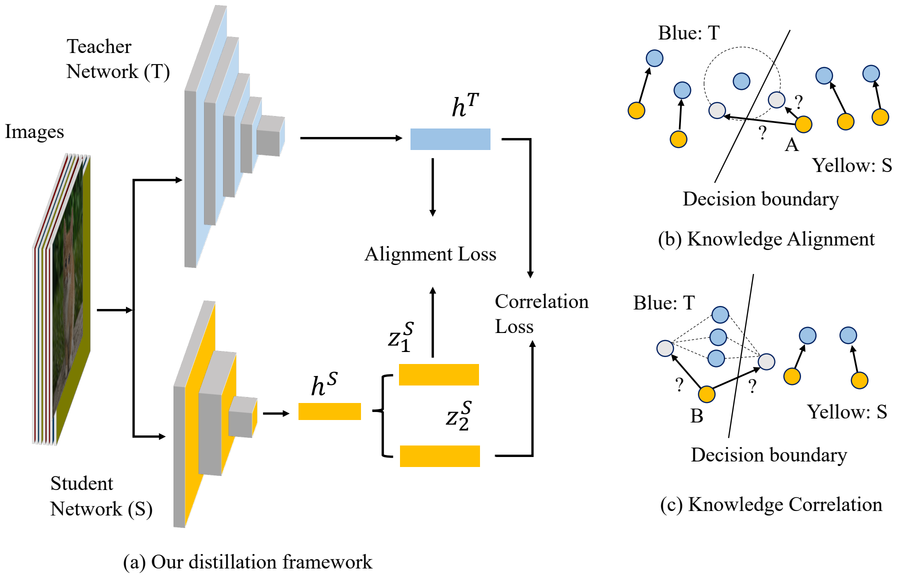
2.1.2 Relational Knowledge distillation

Besides knowledge alignment, another research line of KD focuses on transferring relationships between samples. DarkRank [19] utilizes cross-sample similarities to transfer knowledge for metric learning tasks. Also, RKD [89] transfers distance-wise and angle-wise relations of different feature representations. Recently, CRD [97] is proposed to apply contrastive objective for structural knowledge distillation. However, it randomly draws negative samples and inevitably selects false negatives, hence leading to a suboptimal solution. SEED [36] is another contrastive distillation method to transfer relational knowledge between different samples from a self-supervised pretrained teacher. It only considers knowledge correlation between the sample and a queue. But due to the use of a large queue, it cannot effectively transfer knowledge between different semantic samples. Our work proposes an effective knowledge correlation objective.

2.2 Method

To uncover the relationships between existing distillation methods, we reformulate the standard KD and CRD objectives and identify distillation methods as knowledge alignment or knowledge correlation according to whether the transferred knowledge comes from an individual sample or across samples. We find that standard KD indirectly performs knowledge alignment through the class prototypes, while CRD applies a distillation objective similar to self-supervised contrastive loss [88, 5, 14] which can be decomposed into knowledge alignment and correlation. Therefore, both KD and CRD include the knowledge alignment objective and CRD has an extra correlation objective. However, we find that the knowledge correlation objective of CRD aims to distribute the negative samples (samples from different instances) more uniformly, which undesirably pushes apart samples from the same class and results in inferior distillation performance. Thus, it's necessary to propose a novel knowledge correlation objective. Besides, the standard KD method relies too much on specific pretraining strategies and network architectures, which requires a more general distillation solution to effectively combine knowledge alignment and correlation together.

In this chapter, we extract the common part of the existing distillation methods and propose a L2-based knowledge alignment objective. We find that a spindle-shaped transformation plays a pivotal role in knowledge alignment. Then, we introduce an effective knowledge correlation objective to capture structural knowledge of the teacher. Both of our alignment and correlation objectives



(a) Our distillation framework

(b) Knowledge Alignment

(c) Knowledge Correlation

Figure 2.1: The overview of knowledge alignment and correlation. (a) our distillation framework: $\mathbf{h}^T$ and $\mathbf{h}^S$ indicate representations of the teacher and student. $\mathbf{z}_1^S$ and $\mathbf{z}_2^S$ are two different transformations for distillation. Knowledge alignment (b) focuses on direct feature matching, and knowledge correlation (c) captures relative relationship between samples. The blue (the teacher) and yellow (the student) circles represent different samples. ? indicates that A and B samples could be mapped to different locations (gray circles). Given the decision boundary, different mappings lead to different classification results. The dotted circle in (b) indicates possible feature alignment results and dotted lines in (c) indicate that two different mappings share the same relationship between samples. (b) and (c) illustrate the necessity of knowledge alignment and correlation. It could not achieve the optimal distillation via one single objective.

focus on the feature representation. Therefore, our method is independent of the specific pretraining tasks or architectures, which provides a more flexible knowledge distillation. We demonstrate that knowledge alignment and correlation are necessary to improve the distillation performance. In particular, knowledge correlation can serve as an effective regularization to enable the student to learn generalized representations. We identify the proposed method as Dual-Level Knowledge Distillation (DLKD) to emphasize that it effectively combines both knowledge alignment and correlation, as shown in Figure 2.1. Besides, we introduce an optional supervised distillation objective by leveraging the labels, which can indirectly transfer the category-wise structural knowledge between networks. To summarize, our main contributions are as follows:

- We introduce a novel knowledge distillation method, Dual-Level Knowledge Distillation (DLKD), which provides a general and model-agnostic solution to transfer richer representational knowledge between networks.

- We define a general knowledge quantification metric to measure and evaluate the consistency of visual concepts in the learned representation.
- We show that knowledge alignment and correlation can provide effective supervisory signals for knowledge distillation, and allow students to learn more generalized representations.
- We demonstrate that DLKD consistently outperforms state-of-the-art methods over a large set of experiments including different pretraining strategies (supervised, self-supervised), network architectures (vgg, ResNets, WideResNets, MobileNets, ShuffleNets), datasets (CIFAR-10/100, STL10, ImageNet, Cityscapes) and tasks (classification, segmentation, self-supervised learning).

2.3 Dual-Level Knowledge Distillation

2.3.1 Reformulating KD and CRD

Given a pair of teacher and student networks, $f_\eta^T(\cdot)$ and $f_\theta^S(\cdot)$, the distillation methods train the student via extra supervisory signals from the supervised or self-supervised pretrained teacher. $f_\eta^T(\cdot)$ and $\mathbf{h}^T$ denote the feature extractor and representation vector of the teacher. Take the supervised teacher as an example, besides $f_\eta^T(\cdot)$, there is also a projection matrix $\mathbf{W}^T \in \mathbb{R}^{D\times K}$ to map the feature representation to K category logits, where D is the feature dimensionality. We denote by $s(\cdot)$ the softmax function and the standard KD loss [56] can be written as:

$$\begin{aligned}
\mathcal{L}_{\text{KD}} &= -\sum_{k=1}^{K} s(\mathbf{W}_k^T\mathbf{h}^T)\log s(\mathbf{W}_k^S\mathbf{h}^S) \\
&= -\sum_{k=1}^{K} s(\mathbf{W}_k^T\mathbf{h}^T)[\log s(\mathbf{W}_k^S\mathbf{h}^S) + \log s(\mathbf{W}_k^T h_\varphi(\mathbf{h}^S)) \\
&- \log s(\mathbf{W}_k^T h_\varphi(\mathbf{h}^S))] = -\sum_{k=1}^{K} s(\mathbf{W}_k^T\mathbf{h}^T)\log s(\mathbf{W}_k^T h_\varphi(\mathbf{h}^S)) \\
&+ \sum_{k=1}^{K} s(\mathbf{W}_k^T\mathbf{h}^T)\log\frac{s(\mathbf{W}_k^T h_\varphi(\mathbf{h}^S))}{s(\mathbf{W}_k^S\mathbf{h}^S)},
\end{aligned} \tag{2.1}$$

where $h_\varphi(\cdot)$, $\mathbf{h}^S$ and $\mathbf{W}_k^S$ are trainable, $\mathbf{h}^T$ and $\mathbf{W}_k^T$ are frozen. $h_\varphi(\cdot)$ represents a feature transformation function of aligning the student's representation to the teacher's representation. We observe that when $\mathbf{h}^T = h_\varphi(\mathbf{h}^S)$, the first loss item achieves the optimal solution, and the second loss item becomes the KL divergence between softmax distributions. In other words, the standard KD objective is related to knowledge alignment, and can minimize the discrepancy between networks' outputs indirectly through the class prototypes $\mathbf{W}^T$ and $\mathbf{W}^S$. Recently, CRD shows that indirect learning of the teacher's knowledge is not sufficiently effective, and proposes the contrastive representation distillation. Inspired by [110], the softmax formulation of CRD's objective can be reformulated into two parts:

$$\mathcal{L}_{\text{CRD}} = -\mathbf{z}_i^S \mathbf{z}_i^T / \tau + \log \left(\exp\left(\mathbf{z}_i^S \mathbf{z}_i^T / \tau\right) + \sum_{j=1}^{N} \exp\left(\mathbf{z}_i^S \mathbf{z}_j^T / \tau\right) \right), \tag{2.2}$$

where $\mathbf{z}_i^S$ and $\mathbf{z}_i^T$ are the positive representation pair of the teacher (T) and student (S) from the sample x_i. τ is the temperature parameter, N indicates the total number of negative samples, and j indicates the $j^{th}(j \neq i)$ negative sample of $\mathbf{z}_i^S$. Intuitively, the first term encourages the outputs of the teacher and student for the same sample to be similar (alignment), while the second term encourages representations of samples from negatives to be more dissimilar (correlation). However, because negative samples usually are randomly chosen as long as they are different from x_i, the second term causes many negative samples from the same class (false negatives) be undesirably pushed apart in the representation space.

The distinction of knowledge alignment and correlation provides a novel viewpoint to analyze different distillation methods by reformulating their objectives. From the above analysis, we find that both KD and CRD contain the knowledge alignment objective. We also find that although CRD considers transferring the relationship between samples, it's not optimal due to the problem of false negatives. Here, we propose a novel knowledge correlation objective to capture structural knowledge of samples. And we apply two independent objectives to perform knowledge alignment and correlation respectively. Both of the proposed objectives are calculated at the feature level, which allows our method to be extended to new pretraining strategies and architectures.

2.3.2 Knowledge Alignment

A well-trained teacher already encodes excellent representational knowledge, *i.e.*, categorical knowledge. The stronger supervision is necessary for better matching between the teacher's representation ($f_\eta^T(x)$) and the transformation of the student's representation ($h_\varphi(f_\theta^S(x))$). To meet the requirement of knowledge alignment ($\mathbf{h}^T = h_\varphi(\mathbf{h}^S)$), we propose a L2-based knowledge alignment objective:

$$\mathcal{L}_{\text{Align}} = \mathbb{E}_x \left[\left\| h_\varphi(f_\theta^S(x)) - f_\eta^T(x) \right\|_2^2 \right]. \tag{2.3}$$

This objective forces the student to directly mimic the teacher's representation, thus can provide stronger supervisory signals of inter-class similarities than the standard KD loss [56]. Eq. 2.3 applies the feature representation (penultimate layer) to perform knowledge alignment. Our method is better than previous FitNet loss which matches whole feature maps and may cause training to become difficult or even fail when $h_\varphi(\cdot)$ is only regarded as dimensionality matching. In section 2.6, we confirm that appropriate representation capability of $h_\varphi(\cdot)$ plays a key role in knowledge alignment.

The knowledge alignment can be further expressed as:

$$\mathcal{L}_{\varphi,\theta} = \mathbb{E}_x \left[l\left(h_\varphi(f_\theta^S(x)), g_\phi(f_\eta^T(x)) \right) \right], \tag{2.4}$$

where $l(\cdot,\cdot)$ loss function is used to penalize the difference between networks in different outputs. This is a generalization of existing KD objectives [56, 92, 123, 124, 121]. For example, Hinton *et al.* [56] calculate KL-divergence between f^T and f^S in which the linear functions h_φ and g_ϕ map representations to logits. SRRL [121] utilizes the teacher's pre-trained projection matrix W^T to enforce the teacher's and student's feature to produce the same logits via the L2 loss. These methods rely on the logits of the classification task. In contrast, our method is task-agnostic. Although knowledge alignment is the common part of the existing distillation methods, it doesn't ensure that the teacher's knowledge is fully transferred, as it neglects the structural knowledge between different samples.

2.3.3 Knowledge Correlation

The pretrained teacher also encodes the knowledge of rich relationships between samples, and knowledge correlation allows the student to learn a structure of the representation space similar to the teacher. Here, we propose a novel knowledge correlation objective to capture structural knowledge from the teacher. To be specific, we calculate the relational scores for each (N+1)-tuple samples as the cross-sample relational knowledge. The correlation objective can be expressed as

$$\mathcal{L}_{\text{Corr}} = \sum_{i=1}^{N} l(\psi\left(f_\eta^T(\tilde{x}_i), f_\eta^T(x_1), .., f_\eta^T(x_N)\right), \psi\left(f^S(\tilde{x}_i), f^S(x_1), .., f^S(x_N)\right)), \tag{2.5}$$

where N is the batch size, ψ is the relational function that measures the relational scores between the augmented $\tilde{x}_i$ and samples $\{x_i\}_{i=1:N}$. $l(\cdot,\cdot)$ is a loss function. The samples in each batch have different semantic similarities, and ψ needs to assign higher scores to samples with similar semantic meaning and lower relational scores otherwise. Here, we apply the cosine similarity to measure the semantic similarity between representations, and transform them to softmax distribution for the knowledge correlation objective. All similarities between $\{\tilde{x}_i\}_{i=1:N}$ and $\{x_i\}_{i=1:N}$ can be written as matrix $\mathcal{A}$. For the teacher network, $\mathcal{A}_{i,j}$ is calculated by the representations $\mathbf{h}^S$. For the student network, we also apply a transformation function to the representation $\mathbf{z}^S$ for loss calculation.

We apply the softmax function as the relational function ψ and KL-divergence loss as $l(\cdot,\cdot)$ to transfer these relationships from the teacher to the student.

$$\mathcal{L}_{\text{Corr}} = \sum_{i}^{N}\sum_{j}^{N} -\frac{\exp\left(\mathcal{A}_{i,j}/\tau\right)}{\sum_j \exp\left(\mathcal{A}_{i,j}/\tau\right)} \cdot \log \frac{\exp\left(\mathcal{A}_{i,j}/\tau\right)}{\sum_j \exp\left(\mathcal{A}_{i,j}/\tau\right)} \tag{2.6}$$

where τ is the temperature parameter to soften peaky distributions and $f(\cdot)$ is the teacher or student network.

We also compare our knowledge correlation objective with other relational distillation objectives. RKD [89] proposes distance-wise and angle-wise losses for relational knowledge distillation. The former has a significant difference in scales and makes training unstable. The latter utilizes a triplet of samples to calculate angular scores (O(N^3) complexity. Our KL-based solution achieves high-order property with O(N^2) complexity. SEED [36] is proposed to transfer knowledge from a self-supervised pretrained teacher by leveraging similarity scores between a sample and a queue. However, the large queue results in sparse softmax outputs due to lots of dissimilar samples, which makes it not effective to transfer knowledge between different semantic samples. We directly calculate mutual relationships in each batch and utilize KL divergence loss, which does not require additional queue and large-size batch, thus has high computation efficiency.

2.3.4 Supervised Knowledge Distillation

Both above objectives are related to feature representations and therefore independent of specific pretraining tasks. Here, we also propose an additional distillation objective for supervised pretrained teachers based on the InfoNCE loss. We overcome the false negative problem in CRD by leveraging the true labels to construct positives from the same category and negatives from different categories. There are two kinds of anchors (teacher and student anchor) in distillation:

$$\mathcal{L}_{\text{Sup}}^{T/S} = -\frac{1}{C}\sum_{i=1}^{N}\sum_{j=1}^{2N} \mathbb{1}_{i\neq j} \cdot \mathbb{1}_{\mathbf{y}_i=\mathbf{y}_j} \cdot \log \frac{\exp\left(\mathbf{z}_i \cdot \mathbf{z}_j/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{i\neq k} \cdot \exp\left(\mathbf{z}_i \cdot \mathbf{z}_k/\tau\right)}, \tag{2.7}$$

where $C = 2N_{y_i} - 1$ and N_{y_i} is the number of images with the label y_i in the minibatch. The feature vectors $\mathbf{z}$ are transformed from $\mathbf{h}^T$ or $\mathbf{h}^S$ via MLP heads. $\mathbf{z}_i$ is the anchor representation of the teacher or student. $\mathbf{z}_j$ and $\mathbf{z}_k$ represent positive and negative features, respectively. When $\mathbf{z}_i$ is from the teacher, $\mathbf{z}_j$ and $\mathbf{z}_k$ are from the student, vice versa. This objective provides categorical similarities to encourage a student to map samples from the same category into close representation space and samples from different categories be far away. Our formulation is similar to the supervised contrastive loss [62], with the difference that our objective requires fixed anchors for knowledge transfer.

2.3.5 DLKD objective

The total distillation objective for any pretraining teacher is a linear combination of knowledge alignment and correlation objectives:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{Align}} + \lambda_2 \mathcal{L}_{\text{Corr}}, \tag{2.8}$$

where λ_1 and λ_2 are balancing weights. For the supervised pretrained teacher, we also add the above supervised distillation loss $\mathcal{L}_{\text{Sup}}$ and the standard cross-entropy loss $\mathcal{L}_{\text{CE}}$ with different balancing weights. This objective forces a student network to learn multiple facets of representational knowledge from a teacher, as shown in Figure 2.1.

2.4 Knowledge Quantification Metric

To evaluate the distillation performance, it's necessary to understand the representation knowledge by quantifying the knowledge encoded in networks. Cheng *et al.* [20] proposed to quantify the visual concepts of networks on foreground and background, which requires annotations of the object bounding box. However, these kinds of ground-truth bounding boxes are not always available. Here, we define more general metrics to explain and analyze the knowledge encoded in networks based on the conditional entropy.

Let $\mathbf{X}$ denotes a set of input images. The conditional entropy $H(\mathbf{X}|\mathbf{z}=f(x))$ measures how much information from the input image x to the representation $\mathbf{z}$ is discarded during the forward propagation [45, 20]. A perturbation-based method [45] is proposed to approximate $H(\mathbf{X}|\mathbf{z})$. The perturbed input $\tilde{x}$ follows Gaussian distribution with the assumption of independence between pixels, $\tilde{x} \sim \mathcal{N}\left(x, \mathbf{\Sigma} = \operatorname{diag}\left(\sigma_1^2, \ldots, \sigma_n^2\right)\right)$, where n denotes the total number of pixels. Therefore, the image-level conditional entropy $H(\mathbf{X}|\mathbf{z})$ can be decomposed into pixel-level entropy H_i ($H(\mathbf{X}|\mathbf{z}) = \sum_{i=1}^{n} H_i$), where $H_i = \log \sigma_i + \frac{1}{2}\log(2\pi e)$. High pixel-wise entropy H_i indicates that more information is discarded through layers. The pixels with low pixel-wise entropy are more related with the representation, thus the low-entropy pixels can be considered as reliable visual concepts.

We define two general quantification metrics from the view of knowledge quantification and consistency: average and IoU. The average entropy $\bar{H} = \frac{1}{n}\sum_i H_i$ of the image indicates how much information is discarded in the whole input. A smaller $\bar{H}$ indicates that the network utilizes more pixels to compute feature representation from the input. However, more visual concepts don't always lead to the optimal feature representation, which might result in the over-fitting issue [6]. Ideally, a well-learned network is supposed to encode more robust and reliable knowledge. Thus, we measure the knowledge consistency by the IoU metric, which quantifies the consistency of visual concepts between two views of the same image, *i.e.*, two augmented images x_1 and x_2.

$$\begin{aligned} &\mathrm{IoU} = \mathbb{E}_{x \in \mathcal{X}}\left[\frac{\sum_{i \in x_1 \cap x_2}\left(S_{\text{concept}}^1\left(x_i\right) \cap S_{\text{concept}}^2\left(x_i\right)\right)}{\sum_{i \in x_1 \cap x_2}\left(S_{\text{concept}}^1\left(x_i\right) \cup S_{\text{concept}}^2\left(x_i\right)\right)}\right], \\ &\text{where}, S_{\text{concept}}(x) = \mathbb{1}\left(\bar{H} > H_i\right), \end{aligned} \tag{2.9}$$

where $\mathbb{1}$ is the indicator function, and $S_{\text{concept}}(x)$ denotes the set of visual concepts (pixels with lower entropy than $\bar{H}$). $i \in x_1 \cap x_2$ denotes the same pixels of two augmented images. These same pixels are supposed to obtain similar visual concepts and keep a good consistency between augmented images. We choose the ratio between number of visual concepts overlap and number of visual concepts union (IoU) to measure the knowledge consistency of the learned representations. Our IoU metric meets the requirements of generality and coherency [20], and can be used to quantify and analyze the visual concepts without relying on specific architectures, tasks or datasets.

2.5 DLKD and mutual information bound

Considering the representations of teacher and student in terms of T and S ($T = f_\eta^T(x)$, $S = f_\theta^S(x)$), we define a distribution q with binary variable C to denote whether a pair of representations $(f_\eta^T(x_i), f_\theta^S(x_j))$ is drawn from the joint distribution $p(T,S)$ or the product of marginals $p(T)p(S)$: $q(T,S|C=1) = p(T,S)$, $q(T,S|C=0) = p(T)p(S)$. The joint distribution indicates positive pairs from close representation space, and the product of marginals indicates negative pairs from far representation space. CRD only considers the same input provided to $f_\eta^T(\cdot)$ and $f_\theta^S(\cdot)$ as the positives, and samples drawn randomly from the training data as the negatives, which leads to sampling bias problem [21].

Given N_p positive samples and N_n negative samples, we consider the positives in T and S from $p(T,S)$ are empirically related and semantically similar, *e.g.*, representations of the same sample, augmented sample, and samples from the same category, and the negative samples are drawn empirically from different categories. The contrastive-based distillation methods aim to encourage student's representations to be close to teacher's representations in positives, and those of negatives to be more orthogonal. Then, the priors can be written as: $q(C=1) = N_p/(N_p + N_n)$, $q(C=0) = N_n/(N_p+N_n)$. According to the Bayes' rule, the posterior $q(C=1|T,S)$ can be written as:

$$q(C=1|T,S) = \frac{p(T,S)}{p(T,S) + p(T)p(S)(N_n/N_p)}, \tag{2.10}$$

$$\begin{aligned} \log q(C=1|T,S) &= -\log\left(1 + (N_n/N_p)\frac{p(T)p(S)}{p(T,S)}\right) \\ &\leq -\log(N_n/N_p) + \log\frac{p(T,S)}{p(T)p(S)}. \end{aligned} \tag{2.11}$$

Taking expectation over both sides w.r.t. $q(T,S|C=1)$, we have the mutual information bound as follows:

$$I(T;S) \geq \log(N_n/N_p) + \mathbb{E}_{q(T,S|C=1)} \log q(C=1|T,S). \tag{2.12}$$

The first term $\log(N_n/N_p)$ is constant for the given dataset. Previous studies [97] suggest that a larger batch size can obtain a better lower bound. But our analysis indicates that the influence factor is the ratio of negative and positive samples, which depends on the training data. The second term is to maximize the expectation w.r.t. the student parameters to increase the lower found. But the true distribution $q(C=1|T,S)$ is intractable. We note that this equation is similar to the InfoNCE loss [88], which provides a tractable estimator.

When the teacher's representation $\mathbf{z}_i^T$ and the student's representation $\mathbf{z}_i^S$ form a positive pair, we can relate our knowledge alignment objective to the dot product of positive samples in the InfoNCE through Eq. 2.13, where we maximize the similarity of teacher and student's representations via knowledge alignment.

$$\mathcal{L}_{\text{Align}} = -\frac{\mathbf{z}_i^S \cdot \mathbf{z}_i^T}{\left\|\mathbf{z}_i^S\right\| \cdot \left\|\mathbf{z}_i^T\right\|} = \frac{1}{2} \cdot \left\|\mathbf{z}_i^S - \mathbf{z}_i^T\right\|_2^2 - 1. \tag{2.13}$$

For the knowledge correlation objective, it doesn't directly align representations between networks. Instead, it considers the relationship between an anchor $\mathbf{z}_i^T$ and the j^{th} sample $\mathbf{z}_j^T$ in the teacher by the softmax function:

$$\psi\left(\mathbf{z}_i^T, \mathbf{z}_j^T\right) = \frac{\exp\left(\mathbf{z}_i^T \mathbf{z}_j^T / \tau\right)}{\sum_{k=1}^{N} \exp\left(\mathbf{z}_i^T \mathbf{z}_k^T / \tau\right)}. \tag{2.14}$$

In practice, we convert the relationships between all samples in the batch to the softmax distribution. Then we apply KL-divergence loss to transfer the relationships from the teacher to the student. Because the teacher already encodes the relational knowledge between samples, our knowledge correlation objective encourages the student to learn the similar relationships between samples. Thus it enables the student to map samples from the same category to be closer, and indirectly models the binary classification problem, which is related to $q(C=1|T,S)$. Because the objectives for knowledge alignment and correlation don't rely on an explicit definition of positives/negatives, it's applicable in supervised/self-supervised pretrained teachers.

2.6 Experiments

In this section, we first compare our method with state-of-the-art methods in the knowledge distillation tasks (supervised, structured and self-supervised knowledge distillation). Then we conduct an ablation study to verify each loss of DLKD via classification accuracy and knowledge quantification metric. We also perform experiments to evaluate the transferability of representations and the performance under a few-shot scenario.

Table 2.1: Distillation performance comparison between similar architectures. It reports Top-1 accuracy (%) on CIFAR100 test dataset. We denote the best and the second-best results by **Bold** and underline. The results of all compared methods are from [118].

Teacher Student	wrn40-2 wrn16-2	wrn40-2 wrn40-1	resnet56 resnet20	resnet32×4 resnet8×4	vgg13 vgg8
Teacher	76.46	76.46	73.44	79.63	75.38
Student	73.64	72.24	69.63	72.51	70.68
KD [56]	74.92	73.54	70.66	73.33	72.98
Fitnets [92]	75.75	74.12	71.60	74.31	73.54
AT [124]	75.28	74.45	71.78	74.26	73.62
FT [64]	75.15	74.37	71.52	75.02	73.42
SP [101]	75.34	73.15	71.48	74.74	73.44
VID [1]	74.79	74.20	71.71	74.82	73.96
RKD [89]	75.40	73.87	71.48	74.47	73.72
AB [53]	68.89	75.06	71.49	74.45	74.27
CRD [97]	76.04	75.52	71.68	75.90	74.06
SSKD [118]	76.04	76.13	71.49	76.20	75.33
DLKD (ours)	**77.20**	**76.74**	**72.34**	**77.11**	**75.40**

Network architectures. We adopt vgg [94] ResNet [52], WideResNet [125], MobileNet [57], and ShuffleNet [129] as teacher-student combinations to evaluate the supervised KD on CIFAR100 dataset [67] and ImageNet dataset [27]. Their implementations are from [97]. For structured KD, we implement DLKD based on [73] and evaluate it on Cityscapes dataset [24]. The teacher model is the PSPNet architecture [130] with a ResNet101 and the student model is set to ResNet18. For self-supervised KD, the teachers are pretrained via MoCo-V2 [16] or SwAV [10] and we directly download the pretrained weights for our evaluation. The student network is set to smaller ResNet networks (ResNet18, 34). We also perform the transferability evaluation of representations on STL10 dataset [23] and TinyImageNet dataset [25, 27].

Implementation details. Our implementation is mainly to verify the effectiveness of DLKD. We follow the same training strategy based on the existing solutions without any tricks. For supervised KD, we use the SGD optimizer with the momentum of 0.9 and the weight decay of 5×10^{-4} in CIFAR100. All the students are trained for 240 epochs with a batch size of 64. The initial learning rate is 0.05 and then divided by 10 at the 150th, 180th and 210th epochs. In ImageNet, we follow the official implementation of PyTorch [1] and adopt the SGD optimizer with a 0.9 momentum and 1×10^{-4} weight decay. The initial learning rate is 0.1 and is decayed by 10 at the 30th, 60th, and 90th epoch in a total of 100 epochs. For these two datasets, we apply normal data augmentation

[1] `https://github.com/pytorch/examples/tree/master/imagenet`

Table 2.2: Distillation performance comparison between different Architectures. It reports Top-1 accuracy (%) on CIFAR100 test dataset. We denote the best and the second-best results by **Bold** and underline. The results of all compared methods are from [118].

Teacher Student	vgg13 MobileV2	ResNet50 MobileV2	ResNet50 vgg8	resnet32×4 ShuffleV1	resnet32×4 ShuffleV2	wrn40-2 ShuffleV1
Teacher	75.38	79.10	79.10	79.63	79.63	76.46
Student	65.79	65.79	70.68	70.77	73.12	70.77
KD [56]	67.37	67.35	73.81	74.07	74.45	74.83
Fitnets [92]	68.58	68.54	73.84	74.82	75.11	75.55
AT [124]	69.34	69.28	73.45	74.76	75.30	75.61
FT [64]	69.19	69.01	73.58	74.31	74.95	75.18
SP [101]	66.89	68.99	73.86	73.80	75.15	75.56
VID [1]	66.91	68.88	73.75	74.28	75.78	75.36
RKD [89]	68.50	68.46	73.73	74.20	75.74	75.45
AB [53]	68.86	69.32	74.20	76.24	75.66	76.58
CRD [97]	68.49	70.32	74.42	75.46	75.72	75.96
SSKD [118]	71.53	72.57	75.76	78.44	78.61	77.40
DLKD (ours)	**72.52**	**73.18**	**76.15**	**78.89**	**79.54**	**78.01**

Table 2.3: Top-1 and Top-5 error rates (%) on ImageNet. We denote the best and the second-best results by **Bold** and underline.

	Teacher	Student	SP	KD	AT	CRD	SSKD	SRRL [121]	DLKD
Top-1	26.70	30.25	29.38	29.34	29.30	28.83	28.38	28.27	**27.88**
Top-5	8.58	10.93	10.20	10.12	10.00	9.87	9.33	9.40	**9.30**

methods, such as rotation with four angles, *i.e.*, $0°, 90°, 180°, 270°$. To perform structured KD, the student is trained with an SGD optimizer with the momentum of 0.9 and the weight decay of 5×10^{-4} for 40000 iterations. The training input is set to 512×512, and normal data augmentation methods, such as random scaling and flipping, are used during the training. The self-supervised KD is trained by an SGD optimizer with the momentum of 0.9 and the weight decay of 1×10^{-4} for 200 epochs. More detailed training information can be found in the compared methods(CRD [97], SKD [73] and SEED [36]). The temperature τ in $\mathcal{L}_{\text{Corr}}$ and $\mathcal{L}_{\text{Sup}}$ is set to be 0.5 and 0.07. For the balancing weights, we set $\lambda_1 = 10$ and $\lambda_2 = 20$ according to the magnitude of the loss value. During supervised KD, we set the weights of $\mathcal{L}_{\text{Sup}}$ and $\mathcal{L}_{\text{CE}}$ loss to be 0.5 and 1.0. All models are trained using Tesla V100 GPUs on an NVIDIA DGX2 server.

2.6.1 Supervised knowledge distillation

CIFAR100. DLKD is compared with the existing distillation methods, as shown in Table 2.1 and Table 2.2. Following CRD [97] and SSKD [118], Table 2.1 and Table 2.2 compare teacher-student pairs with similar and different architectures. Our method achieves a large improvement compared with KD and CRD methods, which validates the effectiveness of combination of knowledge alignment and correlation. SSKD is an improved KD method combined with contrast learning, yet only applicable to supervised pretrained teachers for classification tasks, and is more complex which requires two steps. In contrast, our method is simpler, meanwhile still achieve better distillation results and can be applied to supervised and self-supervised pretrained teachers. For similar-architecture comparisons, DLKD increases the performance of the students by an average of 0.66% compared to the other best methods. Taking the teacher resnet32 × 4 as an example, two different types of student networks resnet8 × 4 and ShuffleV2 achieve 77.11% and 79.54% performance respectively. This demonstrates that DLKD can break through the architecture-specific limitation to achieve excellent performance. Notably, we find that DLKD enables the student to obtain better performance than the teacher in three out of five pairs. While comparing the teacher-student pairs with different architectures, DLKD also enables the student to learn better than the teacher.

ImageNet. We further conduct the experiment (teacher: ResNet34, student: ResNet18) on ImageNet. As shown in Table 2.3, our DLKD achieves the best classification performances for both Top-1 and Top-5 error rates, which demonstrate the efficiency and scalability on the large-scale dataset.

2.6.2 Structured Knowledge Distillation

Semantic segmentation can be considered as a structured prediction problem, with different levels of similarities among pixels. To transfer the structured knowledge from the teacher to the student, it's also necessary to perform the pixel-level knowledge alignment and correlation in the feature space. The former encourages the student to learn similar feature representations for each pixel from the teacher, even though their receptive fields (convolutional networks) are different. The latter focuses on maintaining the similarity between pixels belonging to the same class, and the dissimilarity of pixels between different classes. SKD [73] proposes to transfer pair-wise similarities among pixels in the feature space. IFVD [111] proposes to transfer similarities between each pixel and its corresponding class prototype. In contrast, our distillation method can achieve better distillation results than the existing structured KD methods (Table 2.4).

Table 2.4: The segmentation performance comparison on Cityscapes val dataset. Teacher: ResNet101 and Student:ResNet18.

Method	val mIoU (%)	Params (M)
Teacher	78.56	70.43
Student	69.10	13.07
SKD [73]	72.70	13.07
IFVD [111]	74.54	13.07
DLKD (ours)	**75.73**	13.07

Table 2.5: Top-1 k-NN classification accuracy(%) on ImageNet. + and ∗indicates the teachers pretrained by MoCo-V2 and SwAV.

Teacher	ResNet18	ResNet34
Supervised	69.5	72.8
Self-supervised	36.7	41.5
R-50$^+$ + SEED	43.4	45.2
R50x2* + SEED	55.3	58.2
R50x2* + Ours	**56.4**	**59.6**

2.6.3 Self-supervised knowledge distillation

We evaluate the self-supervised distillation with the k-NN nearest neighbor classifier (k=10) as in SEED [36], which does not require any hyperparameter tuning, nor augmentation. Table 2.5 shows the distillation results from different teacher-student pairs. The results of all compared methods are from [36]. The first two rows show the supervised training and self-supervised (MoCo-V2) training baseline results. The k-NN accuracy of self-supervised pretrained ResNet-50(R-50) and ResNet-50w2(R50x2) are 61.9% and 67.3% [11]. We apply the same pretrained R50x2 teacher as [36], to train students (ResNet18 and ResNet34) using the same training strategy. The results show that our solution can further improve the classification accuracy of students.

Table 2.6: ImageNet test accuracy(%) using linear classification. + and ∗indicates the teachers pretrained by MoCo-V2 and SwAV.

Methods		ResNet18		ResNet34	
	Top-1	Top-1	Top-5	Top-1	Top-5
Supervised		69.5		72.8	
Self-supervised		52.5	77.0	57.4	81.6
R-50$^+$ + SEED	67.4	57.9	82.0	58.5	82.6
R50x2* + SEED	77.3	63.0	84.9	65.7	86.8
R50x2* + Ours	77.3	**65.8**	**86.5**	**67.9**	**87.7**

We also evaluate the self-supervised KD by linear classification following previous works in SEED [36]. We apply the SGD optimizer and train the linear classifier for 100 epochs. The weight decay is set to be 0, and the learning rate is 30 at the beginning then reduced to 3 and 0.3 at 60 and 80 epochs. Table 2.6 reports the Top-1 and Top-5 accuracy and indicates that our method also works well in self-supervised settings.

Table 2.7: Distillation performance comparison of different $h_\varphi(\cdot)$ on the resnet32×4 and ShuffleV2. It reports Top-1 accuracy (%) on CIFAR100 test dataset. It denotes multiples of $\dim(\mathbf{z}_T)$.

Hidden size	0.25 ×	0.5 ×	1 ×	2 ×	4 ×	8 ×	16 ×	32 ×	64 ×
Top-1	78.54	78.63	78.58	78.62	78.43	78.57	**79.01**	78.81	78.66

Table 2.8: Ablation study of DLKD. It reports Top-1 accuracy (%) of two teacher-student pairs on CIFAR100 test dataset.

Teacher Student	resnet32×4 resnet8×4	resnet32×4 ShuffleV2
$\mathcal{L}_{\text{Align}}$	76.59	79.01
$\mathcal{L}_{\text{Corr}}$	74.94	76.06
$\mathcal{L}_{\text{Sup}}$	74.73	75.98
$\mathcal{L}_{\text{Align}} + \mathcal{L}_{\text{Sup}}$	76.99	79.26
$\mathcal{L}_{\text{Corr}} + \mathcal{L}_{\text{Sup}}$	75.90	77.35
$\mathcal{L}_{\text{Align}} + \mathcal{L}_{\text{Corr}}$	76.90	79.17
All	**77.11**	**79.54**

Table 2.9: Ablation study of DLKD. Top-1 accuracy (%) of linear evaluation on two datasets using learned representation on CIFAR100 dataset (teacher: resnet32×4, student: resnet8×4).

Datset	STL10	TinyImageNet
$\mathcal{L}_{\text{Align}}$	75.86	40.50
$\mathcal{L}_{\text{Corr}}$	73.73	36.70
$\mathcal{L}_{\text{Align}} + \mathcal{L}_{\text{Corr}}$	77.48	42.17
All	**77.95**	**42.32**

2.6.4 Ablation Study

Section 2.3 demonstrates that it's crucial to set suitable modelling capability for the transformation function $h_\varphi(\cdot)$. We apply 2-layer MLPs to implement $h_\varphi(\cdot)$ for knowledge alignment and correlation on student's output, which is widely used in self-supervised learning [14, 44]. We set different dimensions for the hidden layer to model different capabilities in knowledge alignment, which only include $\mathcal{L}_{\text{Align}}$ and $\mathcal{L}_{\text{CE}}$ losses. Table 2.7 shows the comparison results of different multiples of the student representation's dimension ($\dim(\mathbf{z}_T)$). A spindle-shaped MLP (16 times) can achieve the best alignment results. We have not found similar trends in the knowledge correlation, and we directly set all dimensions to $\dim(\mathbf{z}_S)$. For the additional $\mathcal{L}_{\text{Sup}}$ and $\mathcal{L}_{\text{CE}}$ losses, only linear projections are used.

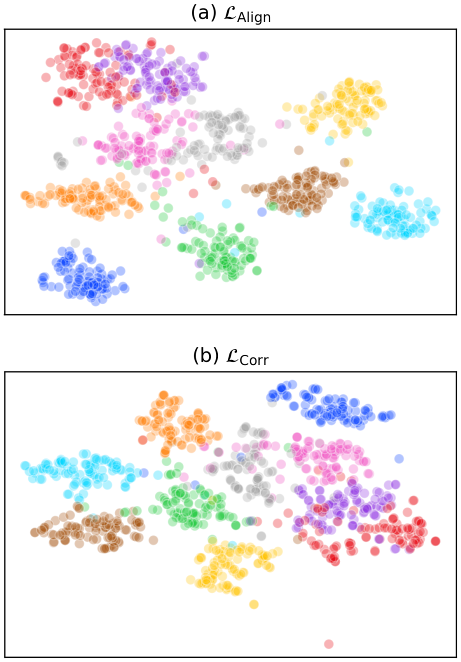
To verify the importance of the transformation function $h_\varphi(\cdot)$, we apply 2-layer multi-layered perceptron (MLP), which is widely used in self-supervised learning [14, 44], for $\mathcal{L}_{\text{Align}}$ and $\mathcal{L}_{\text{Corr}}$ on student's output. We set different dimensions for the hidden layer to model different capabilities. Table 2.7 compares different multiples of the student representation's dimension ($\dim(\mathbf{z}_T)$), and shows that the choice of representation's dimension is important to achieve the optimal performance. A spindle-shaped MLP (16 times) can achieve best alignment results. For $\mathcal{L}_{\text{Corr}}$, we have not observed similar trends and directly set all dimensions to $\dim(\mathbf{z}_S)$. For the additional $\mathcal{L}_{\text{Sup}}$ and $\mathcal{L}_{\text{CE}}$ losses, we apply linear projections.

We also perform the ablation study to examine the effectiveness of each distillation objective, $\mathcal{L}_{\text{Align}}$, $\mathcal{L}_{\text{Corr}}$ and $\mathcal{L}_{\text{Sup}}$. The students are trained via different combinations of these objectives, as shown in Table 2.8. We find that combinations of objectives can obtain better results than single objective, indicating that multiple supervisory signals can improve the representation quality of the student. And among these objectives, $\mathcal{L}_{\text{Align}}$ plays a more important role than others in knowledge distillation. To demonstrate that $\mathcal{L}_{\text{Corr}}$ is also critical in distillation, we compare the transferability of learned representations by using $\mathcal{L}_{\text{Align}}$ and $\mathcal{L}_{\text{Corr}}$, as shown in Table 2.9. We find that $\mathcal{L}_{\text{Corr}}$ can boost the performance of transfer learning by capturing structural knowledge between samples, which is helpful to learn generalized representations.

To visually understand the different roles of $\mathcal{L}_{\text{Align}}$ and $\mathcal{L}_{\text{Corr}}$, we perform t-SNE visualization on cifar100 dataset (randomly select 10 categories from 100 categories), as shown in Figure 2.2. $\mathcal{L}_{\text{Align}}$ tends to make the student learn representations with the large margin between different classes. In contrast, $\mathcal{L}_{\text{Corr}}$ enables the student to capture better intra-class structure for certain classes. It's necessary to combine them to improve the distillation performance.

2.6.5 Transferability of representations

We also examine whether the representational knowledge learned by DLKD can be transferred to the unseen datasets. We perform six comparisons with three teacher-student pairs. The students are fixed to extract feature representations of STL10 and TinyImageNet datasets (all images resized to 32×32). We then compare the quality of the learned representations by training linear classifiers to perform 10-way and 200-way classification. As shown in Table 2.10, DLKD achieves a significant performance improvement compared to multiple baseline methods, demonstrating the superior transferability of learned representations. Notably, most distillation methods improve the quality of the student's representations on STL10 and TinyImageNet. The reason why the teacher performs worse on these two datasets may be that the representations learned by the teacher are biased towards the training dataset and are not generalized well. In contrast, DLKD encourages the student to learn more generalized representations.



(a) $\mathcal{L}_{\text{Align}}$

(b) $\mathcal{L}_{\text{Corr}}$

Figure 2.2: The t-SNE visualization of student's representations: (a) $\mathcal{L}_{\text{Align}}$ loss and (b) $\mathcal{L}_{\text{Corr}}$ loss (teacher: resnet32×4, student: resnet8×4). $\mathcal{L}_{\text{Align}}$ enables the student to learn representations with the large margin between different classes. $\mathcal{L}_{\text{Corr}}$ enables the student to learn better intra-class structure.

2.6.6 Quantification of knowledge consistency

Table 2.12 compares the knowledge consistency of student networks trained by different distillation methods. It verifies that representation distillation can learn more reliable knowledge, compared with other distillation methods. Table 2.11 shows the average score $\bar{H}$ of pixel-level conditional entropy as mentioned in Section 2.4. It indicates that the representation of lower $\bar{H}$ tends to achieve better classification performance. A lower $\bar{H}$ also means that the network focuses on more visual concepts to compute the feature representation. Our method has a lower $\bar{H}$, indicating that the student can learn richer representational knowledge from the teacher. Further, we utilize the IoU score to quantify the knowledge consistency and evaluate the reliability of visual concepts, as shown in Table 2.12. We show that both of the average and IoU scores can provide additional insights about the knowledge distillation, in addition to classification accuracy.

Table 2.10: Classification accuracy (%) of STL10 (10 classes) and TinyImageNet (200 classes) using linear evaluation on the representations from CIFAR100 trained networks. We denote compared results from [118] by *. We denote the best and the second-best results by **Bold** and underline.

Dataset	STL10			TinyImageNet		
Teacher	resnet32×4	vgg13	wrn40-2	resnet32×4	vgg13	wrn40-2
Student	resnet8×4	vgg8	ShuffleV1	resnet8×4	vgg8	ShuffleV1
Teacher	70.45	64.45	71.01*	31.92	27.20	31.69
Student	71.26	67.48	71.58*	35.31	30.87	32.43*
KD [56]	71.29	67.81	73.25*	33.86	30.87	32.05*
Fitnets [92]	72.93	67.16	73.77*	37.86	31.20	33.28*
AT [124]	73.46	71.65	73.47*	36.53	33.23	33.75*
FT [64]	74.29	69.93	73.56*	38.25	32.73	33.69*
SP [101]	72.06	68.43	72.28	35.05	31.55	34.74
VID [1]	73.35	67.88	72.56	37.38	31.12	35.62
CRD [97]	73.39	69.20	74.44*	37.13	33.04	34.30*
SSKD [118]	74.39	71.24	74.74*	37.83	34.87	34.54*
DLKD	**77.95**	**74.49**	**77.43**	**42.31**	**38.74**	**42.48**

Table 2.11: Quantification of representational knowledge. It reports average scores of two students trained by different distillation methods on CIFAR100 test dataset.

Teacher	resnet32×4	resnet32×4
Student	resnet8×4	ShuffleV2
KD	0.4400	0.6307
CRD	0.1460	0.4454
$\mathcal{L}_{\text{Align}}$	0.0934	0.1641
$\mathcal{L}_{\text{Corr}}$	0.2533	0.4288
$\mathcal{L}_{\text{Sup}}$	0.2746	0.3816
DLKD	**0.0887**	**0.1622**

Table 2.12: Quantification of knowledge consistency. It reports IoU scores (0.0 - 1.0) of students trained by different distillation methods on CIFAR100 dataset, and higher is better.

Teacher	resnet32×4	resnet32×4
Student	resnet8×4	ShuffleV2
KD	0.4647	0.2769
CRD	0.7288	0.4612
$\mathcal{L}_{\text{Align}} + \mathcal{L}_{\text{Corr}}$	0.7394	0.7449
DLKD	**0.7512**	**0.7528**

2.6.7 Teacher-Student similarity

DLKD can encourage the student to learn richer structured representational knowledge under the dual-level supervisory signals of the teacher. Thus, we conduct the similarity analysis between the teacher's and the student's representations to further understand the contrastive representation distillation. We calculate the CKA-similarity [66] (RBF Kernel) between the teacher and student networks, as shown in Figure 2.3. Combined with Table 2.10, we find that forcing students to be

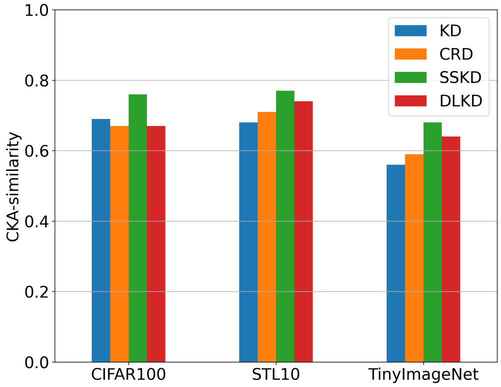


Figure 2.3: CKA-similarity between the representations from the teacher (vgg13) and student (vgg8) networks.

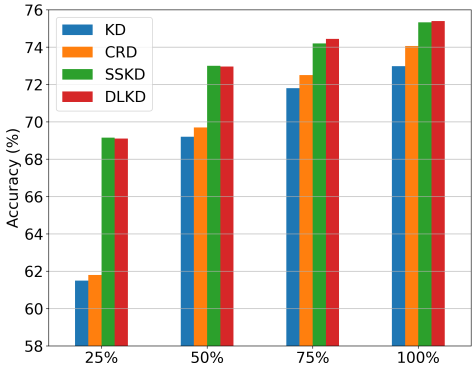


Figure 2.4: Top-1 accuracy on CIFAR100 test data under a few-shot scenario. The student network is trained with only 25%, 50%, 75% and 100% of the available training data.

more similar to teachers does not guarantee that students can learn more general representations.

2.6.8 Few-Shot Scenario.

DLKD enables the student to learn enough representational knowledge from the teacher, instead of relying entirely on labels. It's necessary to investigate the performance of DLKD under limited training data. We randomly sample 25%, 50%, 75%, and 100% images from CIFAR100 train set to train the student network and test on the original test set. The comparisons of different methods (Figure 2.4), show that DLKD maintains superior classification performance in all proportions. As the training set size decreases, dual-level supervisory signals in DLKD serve as an effective regularization to prevent overfitting.

2.7 Conclusion

In this work, we summarize the existing distillation methods as knowledge alignment and correlation and propose an effective and flexible dual-level distillation method called DLKD, which focuses on learning individual and structural representational knowledge. We further demonstrate that our solution can increase the lower bound on mutual information between distributions of the teacher and student representations. We conduct thorough experiments to demonstrate that our method achieves state-of-the-art distillation performance under different experimental settings. Further analysis of student's representations shows that DLKD can improve the transferability of learned representations. We also demonstrate that our method can work well with limited training data in the few-shot scenario. Due to the hardware limitation, we have not carried out more systematic hyperparameter tuning, which can be done in future works to further obtain better performance.

Chapter 3

CLC: Cluster Assignment via Contrastive Representation Learning

Image clustering has been widely used in many computer vision, such as such as image segmentation [61] and visual features learning [70, 10]. Since images are usually high-semantic and high-dimensional, it is difficult to achieve better performance when clustering on large-scale datasets. Earlier clustering studies [117, 119, 59, 61] focus on end-to-end training solutions. For example, IMSAT [59] and IIC [61] develop clustering methods from a mutual information maximization perspective, and DEC [117] and DCN [119] perform clustering on initial features obtained from autoencoders. Since these methods rely on network initialization and are likely to focus on low-level non-semantic features [104], such as color and texture, they are prone to cluster degeneracy solutions. The recently developed clustering methods usually consist of two key steps: representation learning and cluster assignment. Representation learning aims to learn semantically meaningful features, *i.e.*, samples from the same category are projected to similar features so that all samples are linearly separable. One popular representation learning is self-supervised contrastive learning [32, 115, 51, 14] that greatly improve the learned representation. To obtain categorical information without labels, additional step such as K-means clustering [117, 9, 70] or training of a classifier [104] is required for cluster assignment. K-means clustering [117, 9, 70] is widely used for clustering on learned features.

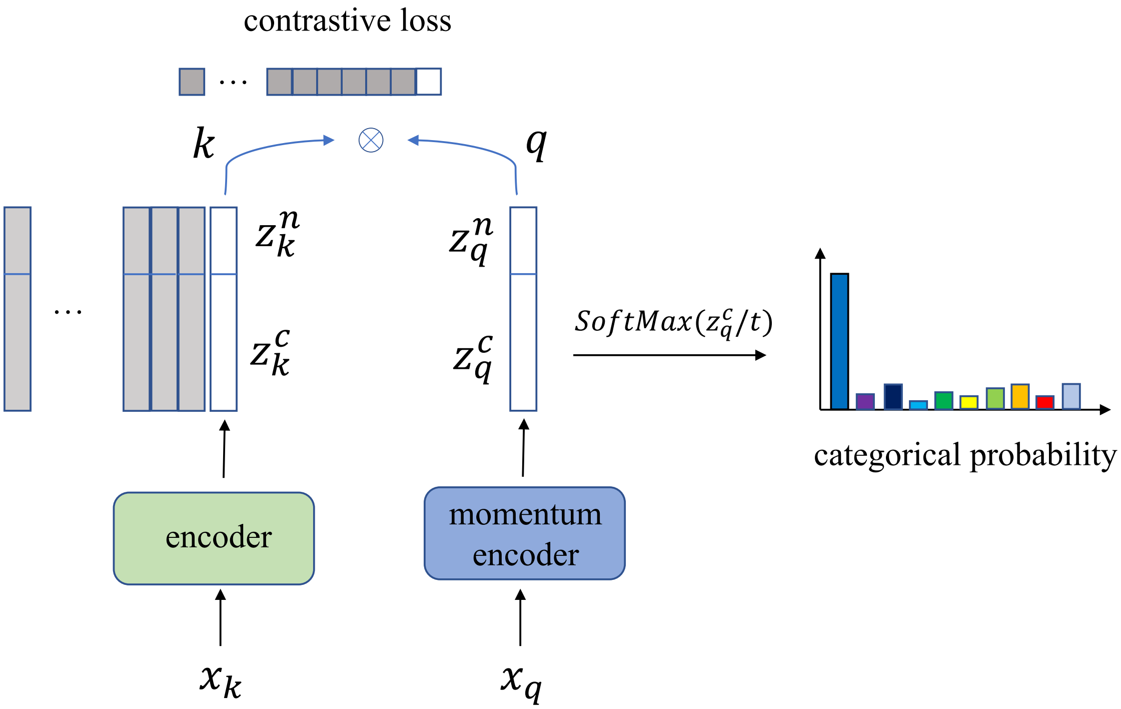


Figure 3.1: The training framework of CLC, illustrated by MoCo [51]. The dot product between q and k is written as $q \cdot k = \mathbf{z}_q^n \cdot \mathbf{z}_k^n + \mathbf{z}_q^c \cdot \mathbf{z}_k^c$. After training, we obtain the categorical probability by applying softmax function on $\mathbf{z}^c$.

It requires the proper selection of distance measures, thus suffering from the uneven assignment of clusters and leading to a degenerate solution [9]. SCAN [104] proposes a novel objective function to train a classifier instead of using K-means. Its performance relies heavily on the feature quality such that nearest neighbors of each sample in the feature space belong to the same category. Due to the presence of noisy nearest neighbors, there is still room for improvement in clustering performance on large-scale datasets.

In this Section, we propose Contrastive Learning based Clustering (CLC), a novel clustering method that directly encodes the categorical information into the part of the representation. Specifically, we formulate contrastive learning as a proxy task to learn cluster assignments, which enables us to take advantage of the powerful contrastive learning frameworks. Figure 3.1 shows an illustration of CLC. First, each representation is decomposed into two parts: $\mathbf{z}^c$ and $\mathbf{z}^n$, where $\mathbf{z}^c$ represents categorical information (logits) and $\mathbf{z}^n$ is used for capturing instance-wise factors. Then, $\mathbf{z}^c$ and $\mathbf{z}^n$ are concatenated together for the training of typical contrastive learning. After training, we can obtain cluster assignments from $\mathbf{z}^c$. To avoid the collapse of assignment, we introduce the equipartition constraint on $\mathbf{z}^c$ to ensure that the clusters are evenly assigned. We demonstrate that this constraint can enforce $\mathbf{z}^c$ to encode the categorical information. By considering $\mathbf{z}^c$ as part of the representation, which can handle well a large number of clusters, we achieve the efficient learning of cluster assignment through contrastive learning.

3.1 Related work

Self-supervised learning applies various pretext tasks to obtain feature representations from images without any manual annotation. There are numerous pretext tasks in the literature, including Autoencoding [105, 117, 50], patch context prediction [29, 82], solving jigsaw puzzles [85, 86], predicting rotations [39], adversarial training [30, 31], and so on. More recently, instance-wise contrastive learning [32, 115, 51, 14], has become an important research area due to its excellent performance in representation learning.

Contrastive learning. The instance-wise contrastive learning considers each sample as its own class. Wu *et al.* [115] first propose to utilize a memory bank and the noise contrastive estimation for learning. He *et al.* [51] continue to improve the training strategy by introducing momentum updates. SimCLR [14] is another representative work that applies a large batch instead of the memory bank. SwAV [10] and PCL [70] bridges contrastive learning with clustering to improve representation learning. There are also some recent works [44, 17, 11] that consider only the similarity between positive samples. Although the typical contrastive loss enables learning a latent space where each instance is well-differentiated, it cannot deal well with the uniformity of the hard negative samples. In our work, the proposed contrastive loss with self-adjusting negative weights solves this problem well. Once the feature representation is obtained, cluster assignment is either obtained by K-means clustering [117, 9, 70] or training an additional component [37, 104]. However, they still cannot achieve promising clustering results on a large-scale dataset, *e.g.*, ImageNet, which requires to develop an efficient objective function that jointly learns cluster assignment and representation learning.

Clustering. Another main line of recent research is jointly learning feature representation and cluster assignment in an alternating or simultaneous manner. Earlier studies (*e.g.* IMSAT [59], IIC [61]) focus on learning a clustering method from a mutual information maximization perspective. Since these methods may only focus on low-level features, such as color and texture, they don't achieve excellent clustering results. DeepCluster [9] performs clustering and representation learning alternatingly, which is further improved with online clustering [128]. Self-labelling [4] is another simultaneous learning method by maximizing the information between labels and data indices. However, due to the lack of a powerful representation learning framework, many of these methods cannot achieve superior clustering performance. We propose to utilize the powerful contrastive learning framework as a proxy task to learn clustering. Also, the introduction of cluster assignment task allows the contrastive learning equipped with the mechanism of self-adjusting negative weights.

3.2 Method

Our goal is to learn cluster assignments via contrastive learning. There are several representation learning methods [9, 4, 70, 10] that jointly learn clustering and feature representation. For example, SwAV [10] proposes an online codes assignment to learn feature representation by comparing prototypes corresponding to multiple views. These methods either require representation learning and clustering to be performed alternately or require learning additional prototypes, which may be not efficient enough for cluster assignment without labels. It's still necessary to propose a novel objective function to directly obtain cluster assignment. Our method learns cluster assignment via self-adjusting contrastive loss and improves the clustering performance on the large-scale dataset. The proposed contrastive loss can be combined with many contrastive learning methods for cluster assignment. Our method demonstrates that contrastive learning can not only achieve remarkable performance in representation learning, but also high efficiency for cluster assignment.

Our method can be interpreted as instance-wise contrastive learning with self-adjusting weights, which learns to set different weights to distinguish different negative samples. Typical contrastive learning methods aim to force positive pairs to achieve high similarity (dot product) and negative pairs to achieve low similarity on $\mathbf{z}^n$. Our method adjusts the order of magnitude corresponding to each dimension of $\mathbf{z}^c$ to distinguish between intra-class and inter-class samples. For example, positive pairs yield high similarity scores because they are from the same instance, and negative pairs from different categories yield low similarity scores due to different semantics. Note that negative pairs from the same category yield moderate similarity scores during the instance-wise setting. We demonstrate that the similarity of $\mathbf{z}^c$ provides a mechanism to adjust different weights for negative samples and improves the uniformity property of negative samples in the representation space, therefore beneficial for representation learning. Our contributions can be summarized as follows:

- We propose CLC, a Contrastive Learning based Clustering method that encodes categorical information into the part of the representation. It considers contrastive learning as a proxy task to efficiently learn cluster assignments.

- We apply the equipartition constraint on part of the representation to enforce cluster assignment. The proposed objective function plays a key role in learning both categorical and instance-wise information simultaneously. The typical contrastive learning method forms a special case of our method.

- We provide a theoretical analysis to demonstrate that CLC adjusts different weights for negative samples through learning cluster assignments. With a gradient analysis, we show that the larger weights tend to concentrate more on hard negative samples.

- The clustering experiments show that CLC outperforms existing methods in multiple benchmarks, and in particular, achieves significant improvements on ImageNet. CLC also contributes to better representation learning results.

3.3 Weighted Instance-wise Contrastive Learning

Given a training set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, contrastive learning aims to map $\mathbf{X}$ to $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ with $\mathbf{z}_i = h(f(\mathbf{x}_i))$, such that $\mathbf{z}_i$ can represent $\mathbf{x}_i$ in representation space. $f(\cdot)$ denotes a feature encoder backbone and the projection head $h(\cdot)$ usually is a multi-layer perceptron. The objective function of contrastive learning, such as InfoNCE [102, 14, 51], can be formulated as:

$$\mathcal{L}_{\text{InfoNCE}}(\mathbf{x}_i) = -\log\left[\frac{\exp(s_{i,i}/\tau)}{\sum_{k\neq i}\exp(s_{i,k}/\tau) + \exp(s_{i,i}/\tau)}\right] \tag{3.1}$$

$$= -\log\left[\frac{\exp(s^n_{i,i}/\tau)}{\sum_{k\neq i}(\exp((s^c_{i,k} - s^c_{i,i})/\tau)\cdot\exp(s^n_{i,k}/\tau)) + \exp(s^n_{i,i}/\tau)}\right]. \tag{3.2}$$

where τ is a temperature hyper-parameter, the positive similarity $s_{i,i}$ is calculated by two augmented versions of the same image, and the negative similarity $s_{i,j}(j \neq i)$ compares different images. In our settings, $\mathbf{z}_i$ consists of $\mathbf{z}_i^c$ and $\mathbf{z}_i^n$. The similarity $s_{i,j}$ can be written as: $s_{i,j} = \mathbf{z}_i \cdot \mathbf{z}_j = \mathbf{z}_i^c \cdot \mathbf{z}_j^c + \mathbf{z}_i^n \cdot \mathbf{z}_j^n$. Let $s^c_{i,j} = \mathbf{z}_i^c \cdot \mathbf{z}_j^c$, $s^n_{i,j} = \mathbf{z}_i^n \cdot \mathbf{z}_j^n$, then we can re-write the standard contrastive loss (equation 3.1) as equation 3.2. More details can be found in supplementary materials.

Note that $s^n_{i,i}$ can be considered as instance-wise positive similarity, $s^n_{i,k}$ can be considered as instance-wise negative similarity, and $\exp((s^c_{i,k} - s^c_{i,i})/\tau)$ is a learnable coefficient for each negative sample. Thus, we obtain a more expressive contrastive loss. Considering that τ is a hyperparameter, the value of this coefficient is mainly determined by $s^c_{i,k}$ and $s^c_{i,i}$, which are further calculated by $\mathbf{z}^c_i$ and $\mathbf{z}^c_k$. This coefficient learns to set different weights for each negative sample, *i.e.*, larger weight on hard negative (intra-class) samples and smaller weight on inter-class negative samples. In this way, the part of representation $\mathbf{z}^c$ plays a role in adjusting different penalties on different negative samples. In contrast, the typical contrastive loss sets all negative samples to the same coefficient with a value of 1.

To make the above coefficient work as expected, we need to add some constraints to $\mathbf{z}^c$. First, the constraint should ensure that the value range of coefficients is bounded. This requirement can be satisfied by performing normalization on $\mathbf{z}^c$ and $\mathbf{z}^n$ separately. More importantly, the constraint should satisfy that $\mathbf{z}^c$ does not collapse to the same assignment, otherwise our method will degrade to typical contrastive learning. Here, we introduce the equipartition constraint on $\mathbf{z}^c$ which encourages it to encode the semantic structural information, while also avoiding its collapse problem. $\mathbf{z}^c$ is expected to represent the probability over clusters $\mathcal{C} = \{1, \dots, K\}$ after softmax function.

3.4 Equipartition constraint for clustering

Given the logits $\mathbf{z}^c$, we can obtain the categorical probabilities via the softmax function: $\mathbf{p}\left(y \mid \mathbf{x}_i\right) = \operatorname{softmax}\left(\mathbf{z}^c_i/t\right)$, where t is the temperature to rescale the logits score. To avoid degeneracy, we add the equipartition constraint on cluster assignment to enforce that the clusters are evenly assigned in a batch. The pseudo-assignment $\mathbf{q}\left(y \mid \mathbf{x}_i\right) \in \{0, 1\}$ is used to describe the even assignment of $\mathbf{z}^c$. We denote B logits in a batch by $\mathbf{Z}^c = [\mathbf{z}^c_1/t, \dots, \mathbf{z}^c_B/t]$, and the pseudo-assignment by $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_B]$.

The equipartition constraint has been used in previous self-supervised learning studies [4, 10] for representation learning. Asano *et al.* [4] propose to solve the matrix $\mathbf{Q}$ by restricting the transportation polytope on the entire training dataset. SwAV [10] improves the above solution to calculate the online prototypes for contrastive learning, and achieves excellent representation learning results. Unlike SwAV, which uses the similarity of features and prototypes as input to obtain pseudo-assignment, whose assignment can be interpreted as the probability of assigning each feature to a prototype, we consider the representation $\mathbf{z}^c$ as logits to directly obtain assignments without any prototypes. Here, we propose to adopt a similar solution to optimize $\mathbf{Q}$ directly from the logits matrix $\mathbf{Z}^c$,

$$\max_{\mathbf{Q} \in \mathcal{Q}} \operatorname{Tr}\left(\mathbf{Q}^{\top} \mathbf{Z}^c\right) + \varepsilon H(\mathbf{Q}), \tag{3.3}$$

where $\mathcal{Q} = \left\{\mathbf{Q} \in \mathbb{R}_{+}^{K \times B} \mid \mathbf{Q} \mathbf{1}_B = \frac{1}{K} \mathbf{1}_K, \mathbf{Q}^{\top} \mathbf{1}_K = \frac{1}{B} \mathbf{1}_B\right\}$ and H denotes the entropy regularization. $\mathbf{1}_B$ and $\mathbf{1}_K$ denote the vector of ones in dimension B and K. $H(\mathbf{Q}) = -\sum_{ij} \mathbf{Q}_{ij} \log \mathbf{Q}_{ij}$. The parameter ε is used to control the smoothness of $\mathbf{Q}$. These constraints ensure that all samples in each batch are evenly divided into K clusters. We also set ε to be small to avoid a trivial solution [10].

The solution of equation 3.3 can be written as: $\mathbf{Q}^* = \operatorname{Diag}(\mathbf{u}) \exp\left(\frac{\mathbf{Z}^c}{\varepsilon}\right) \operatorname{Diag}(\mathbf{v})$, where $\mathbf{u}$ and $\mathbf{v}$ are two scaling vectors such that $\mathbf{Q}$ is a probability matrix [4, 26]. The vectors $\mathbf{u}$ and $\mathbf{v}$ can be computed using Sinkhorn-Knopp algorithm [26] through several iterations. In practice, by using GPU, 3 iterations are fast enough and can ensure satisfactory results [10]. Once we obtain the solution $\mathbf{Q}^*$, we directly apply its soft assignment to constrain $\mathbf{z}^c$ by minimizing the following cross-entropy loss:

$$\mathcal{L}_{\mathrm{CE}}\left(\mathbf{z}^c, \mathbf{q}\right) = -\sum_k \mathbf{q}^{(k)} \log \mathbf{p}^{(k)}. \tag{3.4}$$

3.5 Gradients Analysis

Here, we perform a gradient analysis to understand the properties of the proposed contrastive loss. Because the equipartition constraint is not related to negative similarity $s_{i,j}^n (j \neq i)$, for convenience, our analysis focuses on the negative gradients. Considering that the magnitude of positive gradient $\frac{\partial \mathcal{L}(\mathbf{x}_i)}{\partial s_{i,i}^n}$ is equal to the sum of all negative gradients, we can also indirectly understand the property of the positive gradient through negative gradients. The gradient with respect to the negative similarity $s_{i,j}^n (j \neq i)$ is formulated as:

$$\frac{\partial \mathcal{L}\left(\mathbf{x}_i\right)}{\partial s_{i,j}^n} = \frac{\lambda_{i,j}^c}{\tau} \cdot \frac{\exp(s_{i,j}^n / \tau)}{\sum_{k \neq i}(\lambda_{i,k}^c \cdot \exp(s_{i,k}^n / \tau)) + \exp(s_{i,i}^n / \tau)}, \tag{3.5}$$

where $\lambda_{i,j}^c = \exp((s_{i,j}^c - s_{i,i}^c)/\tau)$. Without the loss of generality, the hyperparameter τ can be considered as a constant.

From equation 3.5, we observe that $\lambda_{i,j}^c$ is proportional to negative gradients. A larger $\lambda_{i,j}^c$ leads to the corresponding sample to receive more attention during the optimization. Since $\lambda_{i,j}^c$

depends mainly on $\mathbf{z}_i^c$ and $\mathbf{z}_j^c$, we need to analyze them separately according to whether samples belong to the same category or not. Due to the equipartition constraint, $\mathbf{z}^c$ is encouraged to encode the categorical information. Thus, the similarity $s_{i,j}^c (j \neq i)$ of the same category is greater than the similarity of different categories. The intra-class $\lambda_{i,j}^c$ is also greater than the inter-class $\lambda_{i,j}^c$. In other words, the gradient tends to concentrate more on samples of the same category, which are often considered as hard negative samples. In this way, the categorical information of $\mathbf{z}^c$ can contribute to the optimization of $\mathbf{z}^n$ so that all samples tend to be uniformly distributed.

Our overall objective, namely CLC, is defined as

$$\mathcal{L} = \mathcal{L}_{\text{InfoNCE}} + \alpha \mathcal{L}_{\text{CE}}. \tag{3.6}$$

In addition to the loss weight α, there are two temperature hyperparameters: τ and t. We observe that the choice of temperature values has a crucial impact on the clustering performance. In general, the relationship of temperatures satisfies: $0 < t \leq \tau \leq 1$. We refer to Section 3.7.2 for the concrete analysis.

Because samples with the highly confident prediction (close to 1) can be considered to obtain pseudo labels, our method can optionally include the confidence-based cross-entropy loss [104], which can be gradually added to the overall objective, or be used for fine-tuning on the pretrained model. SCAN applies this loss to correct for errors introduced by noisy nearest neighbors, while we aim to encourage the model to produce a smooth feature space, thus helping assign proper clusters for boundary samples. We only consider well-classified samples, *i.e.*, $p_{\max} >$ threshold (0.99), and perform strong data augmentation on them. This encourages different augmented samples to output consistent cluster predictions through the cross-entropy loss, also known as self-labeling [104].

Algorithm 1 provides PyTorch-like pseudo-code to describe how we compute the objective (equation 3.6).

3.6 Experiments

In this section, we evaluate CLC on multiple benchmarks, including training models from scratch and using self-supervised pretrained models. We follow the settings in MoCo [51] and choose the same backbone network as the baseline methods, to ensure that our performance gains are

Algorithm 1: PyTorch-like Pseudo-code for CLC.

```
# model: includes base_encoder, momentum_encoder and MLP heads
# sinkhorn-knopp: implementation details can be found in supplementary materials
# T and t: temperatures for contrastive loss and cross entropy loss
# alpha: weight of the loss term
# K: dimension of zc (number of clusters), C: dimension of zn

for x in loader: # load a minibatch x with N samples
    x_q = aug(x) # a randomly augmented version
    x_k = aug(x) # another randomly augmented version

    # no gradient to k
    q, k = model.forward(x_q, x_k) # compute features: N x (K + C)

    zc_q = normalize(q[:, :K], dim=1) # normalize zc: N x K
    zn_q = normalize(q[:, K:], dim=1) # normalize zn: N x C
    q = cat([zc_q, zn_q], dim=1)

    zc_k = normalize(k[:, :K], dim=1) # normalize zc: N x K
    zn_k = normalize(k[:, K:], dim=1) # normalize zn: N x C
    k = cat([zc_k, zn_k], dim=1)

    # compute assignments with sinkhorn-knopp
    with torch.no_grad():
        q_q = sinkhorn-knopp(zc_q)
        q_k = sinkhorn-knopp(zc_k)

    # convert logits to probabilities
    p_q = Softmax(zc_q / t)
    p_k = Softmax(zc_k / t)

    # compute the equipartition constraint
    cross_entropy_loss = - 0.5 * mean(q_q * log(p_k) + q_k * log(p_q))

    loss = contrastive_loss(q, k, T) + alpha * cross_entropy_loss

    # SGD update: network and MLP heads
    loss.backward()
    update(model.params)
```

from the proposed objective function. We first compare our results to the state-of-the-art clustering methods, where we find that our method is overall the best or highly competitive in many benchmarks. Then, we quantify the representation learned by the proposed contrastive loss, and the results show that it can also improve the representation quality. All experimental details can be found in the supplementary material.

3.6.1 Experimental setup

Datasets. We perform the experimental evaluation on CIFAR10 [67], CIFAR100-20 [67], STL10 [23] and ImageNet [27]. Some prior works [117, 61, 12] use the full dataset for both training and evaluation. Here, we follow the experimental settings in SCAN [104], which trains and evaluates the model using train and val split respectively. This helps us to understand the generalizability of our method on unseen samples. All datasets are processed using the same augmentations in MoCo v2 [16]. We report the results with the mean and standard deviation from 5 different runs.

Implementation details. We apply the standard ResNet [52] backbones (ResNet-18 and ResNet-

Table 3.1: State-of-the-art comparison: We report the averaged results (Avg ± Std) for 5 different runs after the clustering and self-labeling steps. All the baseline results are from [104]. We train and evaluate the model using the train and val split respectively, which is consistent with the SCAN [104].

Dataset	**CIFAR10**			**CIFAR100-20**			**STL10**		
Metric	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
K-means [109]	22.9	8.7	4.9	13.0	8.4	2.8	19.2	12.5	6.1
SC [127]	24.7	10.3	8.5	13.6	9.0	2.2	15.9	9.8	4.8
JULE [120]	27.2	19.2	13.8	13.7	10.3	3.3	27.7	18.2	16.4
DAE [106]	29.7	25.1	16.3	15.1	11.1	4.6	30.2	22.4	15.2
AE [7]	31.4	23.4	16.9	16.5	10.0	4.7	30.3	25.0	16.1
GAN [91]	31.5	26.5	17.6	15.1	12.0	4.5	29.8	21.0	13.9
DEC [117]	30.1	25.7	16.1	18.5	13.6	5.0	35.9	27.6	18.6
ADC [49]	32.5	–	–	16.0	–	–	53.0	–	–
DeepCluster [9]	37.4	–	–	18.9	–	–	33.4	–	–
DAC [12]	52.2	40.0	30.1	23.8	18.5	8.8	47.0	36.6	25.6
IIC [61]	61.7	51.1	41.1	25.7	22.5	11.7	59.6	49.6	39.7
Pretext [14] + K-means	65.9 ± 5.7	59.8 ± 2.0	50.9 ± 3.7	39.5 ± 1.9	40.2 ± 1.1	23.9 ± 1.1	65.8 ± 5.1	60.4 ± 2.5	50.6 ± 4.1
SCAN [104]	81.8 ± 0.3	71.2 ± 0.4	66.5 ± 0.4	42.2 ± 3.0	44.1 ± 1.0	26.7 ± 1.3	75.5 ± 2.0	65.4 ± 1.2	59.0 ± 1.6
SCAN [104] + selflabel	87.6 ± 0.4	78.7 ± 0.5	75.8 ± 0.7	45.9 ± 2.7	46.8 ± 1.3	30.1 ± 2.1	**76.7 ± 1.9**	**68.0 ± 1.2**	**61.6 ± 1.8**
Supervised	93.8	86.2	87.0	80.0	68.0	63.2	80.6	65.9	63.1
CLC	83.1 ± 0.6	73.4 ± 0.7	68.7 ± 0.6	44.0 ± 1.2	46.3 ± 1.1	28.2 ± 1.0	71.2 ± 1.5	64.3 ± 1.3	55.3 ± 1.5
CLC + selflabel	**89.0 ± 0.3**	**80.6 ± 0.4**	**78.4 ± 0.5**	**49.2 ± 0.8**	**50.4 ± 0.5**	**34.6 ± 0.6**	75.2 ± 0.8	66.8 ± 0.5	59.4 ± 0.6

50) each with a MLP projection head. The dimensionality of $\mathbf{z}^c$ is determined by the number of clusters, and the dimensionality of $\mathbf{z}^n$ is set to 256 on the ImageNet and 128 on the other datasets. On the smaller datasets, our implementation is based on the Lightly library [95]. The parameters are trained through the SGD optimizer with a learning rate of 6e-2, a momentum of 0.9, and a weight decay of 5e-4. The loss term is set to $\alpha = 5.0$, and two temperature values are set to $\tau = 0.15$ and $t = 0.10$. We train the models from scratch for 1200 epochs using batches of size 512. For ImageNet, we adopt the implementation from MoCo v2. To speed up training, we directly initialize the backbone with the released pretrained weights like SCAN, and only train the MLP head. The weights are updated through SGD optimizer with a learning rate of 0.03, a momentum of 0.9, and a weight decay of 1e-4. The three hyperparameters are set to $\alpha = 1.0$, $\tau = 0.40$ and $t = 0.20$. We train the network weights for 400 epochs using a batch size of 256.

Equipartition constraint. Most of the Sinkhorn-Knopp implementation are directly from SwAV work [10]. The regularization parameter is set to $\epsilon = 0.05$ and the number of Sinkhorn iterations is set to 3 for all experiments.

3.6.2 Comparison with state-of-the-art methods

We first evaluate CLC's clustering performance on three different benchmarks. We report the results of clustering accuracy (ACC), normalized mutual information (NMI) and adjusted rand index

Table 3.2: Clustering results for 100 and 200 selected classes from ImageNet validation data. All the baseline results are from SCAN. Both our method and SCAN are based on MoCo v2's pretraining, and for a fair comparison, we follow the same settings as MoCo v2.

ImageNet	**100 Classes**				**200 Classes**			
Metric	Top-1	Top-5	NMI	ARI	Top-1	Top-5	NMI	ARI
K-means	59.7	-	76.1	50.8	52.5	-	75.5	43.2
SCAN	66.2	**88.1**	78.7	**54.4**	56.3	80.3	75.7	44.1
CLC	**67.0**	83.4	**79.0**	53.9	**61.4**	**80.6**	**77.6**	**47.6**

(ARI) in Table 3.1. CLC outperforms other clustering methods in two of the benchmarks and is on par with state-of-the-art performance in another benchmark. Our method further reduces the gap between clustering and supervised learning on CIFAR-10. On CIFAR100-20, the reason why there is still a large gap with supervised learning is due to the ambiguity caused by the superclasses (mapping 100 classes to 20 classes). On the STL10 dataset, we train the model on the train+unlabeled split from scratch due to the small size of the train split. However, the exact number of clusters in the train+unlabeled split is unknown. We choose the number of clusters equal to 10 for the evaluation on the test dataset. Nevertheless, our method still achieves competitive clustering results, which demonstrates that our method is also applicable to datasets with an unknown number of clusters. Note that the results of other state-of-the-art methods are based on the pretrained representations from contrastive learning, while our clustering results are obtained by training the model from scratch in an end-to-end manner. This also confirms that $\mathbf{z}^c$ efficiently encodes the categorical information. In Section 3.6.4, we further verify that the presence of $\mathbf{z}^c$ enables us to learn better feature representation.

3.6.3 ImageNet clustering

ImageNet - subset. We first test our method on ImageNet subsets of 100 and 200 classes, which is consistent with SCAN. All compared methods apply the same pre-trained weights from MoCo v2 [16]. We fix the ResNet-50 (R50) backbone and train the MLP projection head for 400 epochs with the same settings as MoCo v2. Table 3.2 shows that our method can further improve the clustering accuracy, where the clustering results of K-means and SCAN are from SCAN. For example, it has achieved a much higher accuracy of 61.4% than SCAN (56.3%) on the subset of 200 classes. This also implies that CLC is applicable to large-scale datasets with a large number of clusters.

Table 3.3: Comparison with other clustering methods on the full ImageNet dataset (1000 classes). We obtain the clustering results on the MoCo V3 pretrained weights by using the code provided by SCAN (*). All compared methods are based on the ResNet-50 backbone.

Method	**Backbone**	**ACC**	**NMI**	**ARI**
MoCo V2 + SCAN [104]	R50	39.9	72.0	27.5
MoCo V3 + SCAN*	R50	43.2	70.9	-
MoCo V3 + CLC	R50	**53.4**	**76.3**	**34.7**

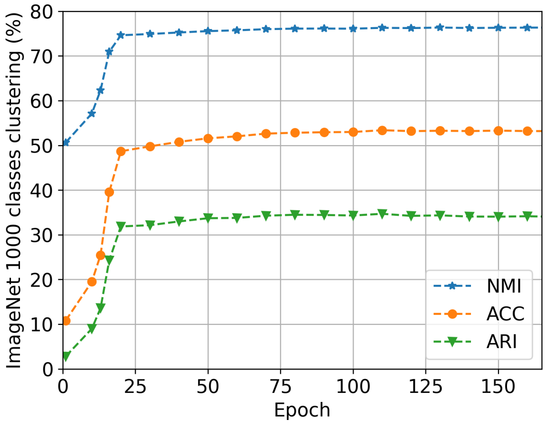


Figure 3.2: Clustering results of CLC ($t = 0.1$) on full ImageNet dataset (1000 classes) up to 200 epochs.

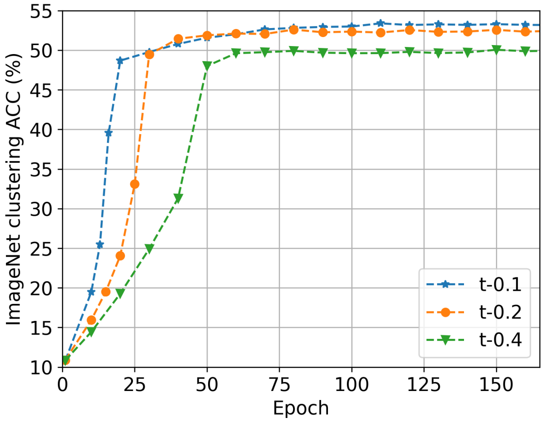


Figure 3.3: Comparison of clustering accuracy at different temperatures on full ImageNet dataset.

ImageNet - full. We further consider the clustering evaluation on the full ImageNet dataset. We apply our method to the latest contrastive learning studies, such as MoCo v3 [18], to uncover its potential in clustering tasks. We load the pretrained R50 backbone from MoCo v3 and only train two MLP heads for 200 epochs with the same settings as MoCo V3. Table 3.3 compares our method against SCAN on three metrics. CLC consistently outperforms the baseline method in all metrics. In particular, it achieves significant performance improvements in terms of accuracy (53.4%) compared to SCAN (39.9%). Although previous studies [104] find that there may be multiple reasonable ways to cluster images in ImageNet based on their semantics, without a priori knowledge, it's still challenging to cluster images in ImageNet according to their true labels. But CLC still achieves promising clustering results, which demonstrates the advantages of the proposed method. Figure 3.2 shows the training efficiency of our method, which can converge in a small number of epochs.

Table 3.4: Image classification with linear classifiers. We report the top-1 accuracy for ImageNet and Places205, mAP for VOC dataset. All the baseline results are from [70] and [10]. Our results are obtained by directly applying the evaluation code [58] on the pretrained R50 backbone.

Method	Architecture	#pretrain		Dataset	
		epochs	ImageNet	VOC07	Places205
Supervised	R50	-	76.5	87.5	53.2
SimCLR [14]	R50-MLP	200	61.9	–	–
MoCo v2 [51]	R50-MLP	200	67.5	84.0	50.1
PCL v2 [70]	R50-MLP	200	67.6	85.4	50.3
CLC	R50-MLP	200	**68.0**	**91.8**	**52.0**

3.6.4 Linear evaluation

We follow the same settings as MoCo v2 to enable a reasonable evaluation of the benefits due to the introduction of $\mathbf{z}^c$. We perform the same data augmentation and training strategy to train the model on ImageNet training data for 200 epochs from scratch. Then, we fix the R50 backbone and train a linear classifier to evaluate the learned feature encoder on three datasets: ImageNet, VOC07 [35], and Places205 [133]. Table 3.4 shows that the proposed contrastive objective achieves competitive results on these linear classification tasks. Especially, for the transfer learning on VOC07 and Places205, it demonstrates that our method achieves better generalizability to the downstream tasks than other methods. PCL v2 [70] is another method that utilizes clustering to improve representation learning. Although our main purpose of introducing $\mathbf{z}^c$ is for clustering, it can also improve the quality of feature representation. This demonstrates that the expressiveness of the model is improved by the negative coefficients due to the introduction of $\mathbf{z}^c$.

3.6.5 Transfer to Object Detection

We fine-tune the whole network following the experiment settings in detectron2, which are consistent with the other methods [51, 70]. Table 3.5 and Table 3.6 show that CLC is overall better than MoCo v2 on COCO and VOC datasets.

3.7 Analysis

The quantitative evaluation in Section 3.6 demonstrates that CLC not only outperforms other clustering methods on multiple benchmarks, but also improves the representation quality. Here,

Table 3.5: Transfer learning results to object detection tasks on COCO dataset. The detection model is fine-tuned on COCO train2017 dataset and evaluated on COCO val2017 dataset. All the baseline results are from [70].

Method	Architecture	#pretrain epochs	bbox AP	bbox AP_{50}	bbox AP_{75}	segm AP	segm AP_{50}	segm AP_{75}
Supervised	R50	-	40.0	59.9	43.1	34.7	56.5	36.9
MoCo v2 [51]	R50	200	40.7	60.5	44.1	35.4	57.3	37.6
CLC	R50	200	**40.8**	**60.6**	**44.3**	**35.5**	57.3	**38.0**

Table 3.6: Transfer learning results to object detection tasks on VOC dataset. The detection model is fine-tuned on VOC07+12 trainval dataset and evaluated on VOC07 test dataset. The baseline results are from [51].

Method	Architecture	#pretrain epochs	VOC AP_{50}	VOC AP	VOC AP_{75}
Supervised	R50	-	81.3	53.5	58.8
MoCo v2 [51]	R50	200	82.4	**57.0**	63.6
CLC	R50	200	**82.6**	56.8	**63.7**

we further analyze the proposed objective to understand how it improves learning semantic and instance-wise information.

3.7.1 Ablation studies

$\mathbf{z}^c$ plays a crucial role in our method. We perform ablation studies on $\mathbf{z}^c$ to understand the importance of each technique. The evaluation results are reported in Table 3.7, where the training is performed from scratch for 1200 epochs. First, we find that the lack of the equipartition constraint leads to a degenerate solution for cluster assignment, but has almost no effect on the training of representation learning. Second, we avoid the use of normalization on $\mathbf{z}^c$ and consider it as a regular logit like in classification. Our experiment shows that the loss becomes nan and the training fails, so the normalized $\mathbf{z}^c$ enables the corresponding dot product is bounded. Finally, we set the temperature t to 1.0 (without t) and find that its value has an important impact on the clustering performance, which is analyzed in detail in Section 3.7.2. We also train the model only using the equipartition constraint, and find that the model can not achieve optimal clustering results without the weighted InfoNCE loss. It also indicates that the weighted InfoNCE loss adjusts different weights of negatives to encourage $\mathbf{z}^c$ to capture more categorical information.

Table 3.7: Ablation studies of $\mathbf{z}^c$ on CIFAR10 including without the equipartition constraint, the normalization, temperature t (default 1.0) and the InfoNCE loss. We compare the clustering results and linear evaluation on the pretrained backbone separately.

Settings	Clustering (%)	Linear evaluation (%)
Without constraint	27.3	86.0
Without norm	-	-
Without t	71.1	88.0
Without InfoNCE	67.0	86.0
Full setup	**83.0**	**88.8**

Table 3.8: The mean and standard deviation of similarity scores for $\mathbf{z}^c$ and $\mathbf{z}^n$ / $\mathbf{z}$ from a category perspective. Augmented: samples from the same instance, Same: samples from the same category, Different: samples from different categories.

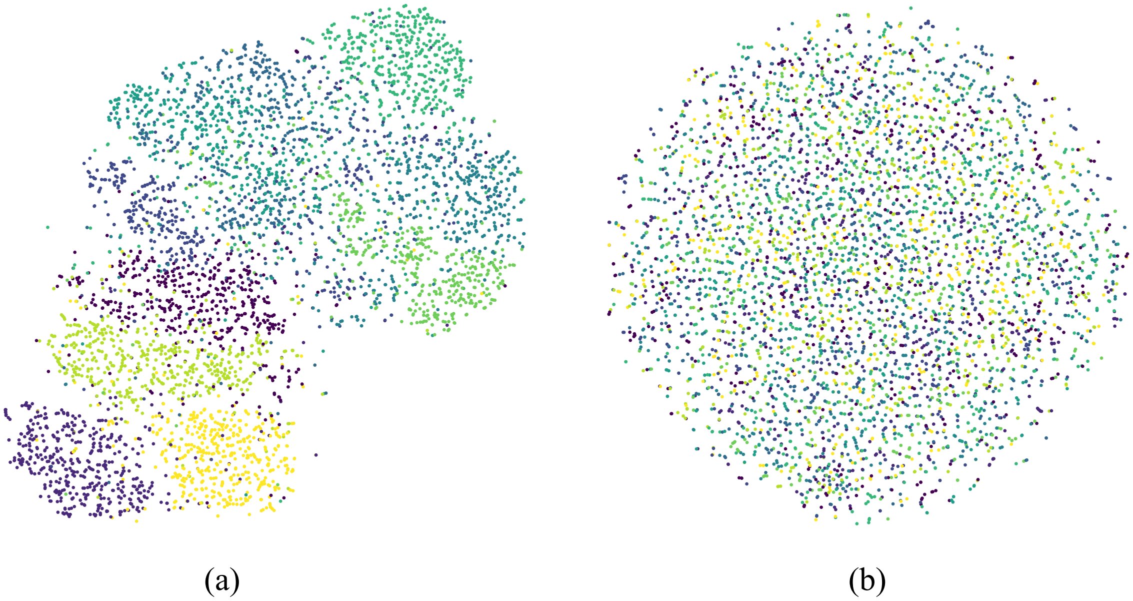
Category	MoCo v2	CLC	
	$\mathbf{z}$	$\mathbf{z}^c$	$\mathbf{z}^n$
Augmented	0.781 ± 0.188	0.872 ± 0.154	0.720 ± 0.199
Same	0.062 ± 0.175	0.504 ± 0.237	0.004 ± 0.183
Different	-0.006 ± 0.115	-0.033 ± 0.323	-0.001 ± 0.167

3.7.2 Temperature analysis

Our method involves two temperatures: τ and t. Previous work [108] shows that temperature τ plays an important role in controlling the penalty strength of negative samples. Here, we consider τ as a constant and focus on the effect of t on the clustering results, as shown in Figure 3.3. Since t plays a role in the scaling of logits in the calculation of cross-entropy loss, a small t reduces the difficulty of matching pseudo-assignment, and achieves better clustering results faster. In contrast, a larger t value leads to the training of $\mathbf{z}^c$ becoming difficult. It is the presence of t that balances the degree of difficulty between the instance-level discrimination and cluster assignment tasks. t also allows $\mathbf{z}^c$ to be converted into proper softmax probabilities, as verified in the self-labeling experiments in Section 3.6.2. The results demonstrate that similar clustering results can be achieved for a range of t, such as 0.1 or 0.2.

3.7.3 Latent space analysis

Table 3.8 shows the statistics of similarity scores (dot product), where MoCo v2 has only $\mathbf{z}$ to compute the contrastive loss, and our method has $\mathbf{z}^c$ and $\mathbf{z}^n$. Both methods satisfy the requirements of instance-wise contrastive learning well, where positive samples (Augmented) have high similarity



(a) (b)

Figure 3.4: The t-SNE visualization of $\mathbf{z}$ / $\mathbf{z}^n$ on CIFAR10 test dataset. (a): MoCo v2, (b): CLC (ours). Colors indicate different categories.

scores of $\mathbf{z}$ / $\mathbf{z}^n$ and negative samples (Same or Different) have low similarity scores of $\mathbf{z}$ / $\mathbf{z}^n$. In CLC, since $\mathbf{z}^c$ encodes the categorical information, the similarity score of $\mathbf{z}^c$ is also able to distinguish well between samples from different categories. As analyzed in Section 3.3, it provides a weight adjustment mechanism for different negative samples so that it can handle negative samples of different hardness well. In contrast, MoCo sets the weight of all negative samples to 1, which tends to learn larger dot products for positive samples. The previous study [110] summarizes two key properties of contrastive loss: alignment and uniformity. In other words, MoCo is more concerned with alignment for the optimization purpose.

We further analyze the uniformity properties of $\mathbf{z}$ / $\mathbf{z}^n$ of MoCo and CLC using t-SNE [103] and the results are shown in Figure 3.4. Compared to $\mathbf{z}$ learned by MoCo, our method tends to uniformly distribute points over the latent space without preserving any category-related information. Although MoCo achieves instance-level differentiation, we can still observe that points of the same category are clustered together. The main reason is that the typical contrastive loss cannot deal with the hard negative problem well and samples of the same category aren't distributed evenly. And the proposed method enables us to learn a uniformly distributed space due to the mechanism of self-adjusting negative weights. This also shows that the MLP projection head in our method works well as the role of transforming in two representation spaces. Therefore, we decompose the original $\mathbf{z}$ into two separate parts: $\mathbf{z}^c$ related to the categorical information, thus focusing on clustering, and $\mathbf{z}^n$ related to instance-wise information, thus focusing on alignment and uniformity. Our method also demonstrates that the MLP projection head plays a role in the transformation from the linearly separable feature space to the instance-wise representation space.

3.8 Conclusion

This work aims to learn cluster assignments based on contrastive learning in a more efficient way. The existing state-of-the-art clustering methods usually require two steps where representation learning and clustering are decoupled, preventing them from achieving superior results on large-scale datasets. In our work, we decompose the representation into two separate parts: one focuses on clustering and the other part focuses on contrastive learning. Experiments on multiple benchmarks demonstrate that our method not only achieves excellent clustering performance, but also improves contrastive learning. Note that CLC can be combined with over-clustering, vision transformers, advanced augmentation and training strategies. Due to the limitations of our computational resources, we will explore these techniques in future work.

Chapter 4

Clustering by Directly Disentangling Latent Space

To overcome the high dimensionality of data, learning latent feature representations for clustering has been widely studied. Recently, ClusterGAN combined GAN with an encoder to learn a mixture of one-hot discrete and continuous latent variables, and achieved remarkable clustering performance. However, the performance of ClusterGAN decreases when it is applied to complex data. In this paper, we analyze the reasons for performance degeneracy in ClusterGAN. We show that minimizing the cycle-consistency loss of continuous latent variables in ClusterGAN trends to generate trivial latent features. Moreover, the objective of ClusterGAN doesn't include a real conditional distribution term, which makes it difficult to be generalized to real data. Therefore, we propose Disentangling Latent Space Clustering (DLS-Clustering), a new clustering mechanism that directly learns cluster assignments from disentangled latent spacing without additional clustering methods. We enforce the inference network (encoder) and the generator of GAN to form an encoder-generator pair in addition to the generator-encoder pair. We train the encoder-generator pair using real data, which can estimate the real conditional distribution. Moreover, the encoder-generator pair competes with the generator-encoder pair during optimization, which can avoid the triviality of continuous latent variables. Furthermore, we utilize a weight-sharing procedure to disentangle the one-hot discrete and the continuous latent variables generated from the encoder. This process enforces the disentangled latent space to match the independence of GAN inputs. Eventually, the one-hot discrete latent variables can be directly expressed as clusters and the continuous latent variables represent remaining unspecified factors. Experiments on benchmark datasets of different types demonstrate that our method outperforms existing state-of-the-art methods.

In summary, our contributions in this section are as follows:

(1) We propose a new clustering approach called DLS-Clustering, which can directly obtain cluster assignments through a weight-sharing procedure to disentangle latent space.

(2) We introduce an MMD-based regularization to enforce the inference network and the generator of standard GAN to form a encoder-generator pair, which enables the encoder to learn the real data conditional distribution.

(3) We combine the encoder-generator pair with the generator-encoder pair to form two cycle-consistencies, which help avoid the triviality on continuous latent variable.

(4) We evaluate DLS-Clustering with different types of benchmark datasets, and achieve superior clustering performance in most cases.

4.1 Related Work

Latent space clustering. A general method to avoid the curse of dimensionality in clustering is mapping data samples to in a low-dimensional latent space and performing clustering on latent space. Several pioneering works propose to utilize an encoding architecture [120, 59, 9, 4] to learn the low-dimensional representations. To obtain clustering assignments, several additional clustering algorithms, such as K-means, are performed on the latent space. IMSAT [12] and IIC [61] combine representation learning and clustering together via information maximizing. Most recent latent space clustering methods are based on Autoencoder [117, 28, 47, 119, 122], which enables reconstructing data samples from the low-dimensional representation. For example, Deep Embedded Clustering (DEC) [117] proposes to pre-train an Autoencoder with the reconstruction objective to learn low-dimensional embedded representations. Then, it discards the decoder and continues to train the encoder for the clustering objective through a well-designed regularizer. DCN [119] proposes a joint dimensionality reduction and K-means clustering approach, in which the low-dimensional representation is obtained via the Autoencoder. Because the learned latent representations are closely related to the reconstruction objective, these methods still do not achieve the desired clustering results. Recently, ClusterGAN [81] integrated GAN with an encoder network for clustering by creating a non-smooth latent space. However, its discrete and continuous latent variables are not completely disentangled. Thus, the one-hot encoded discrete variables cannot effectively represent clusters.

Disentanglement of latent space. Learning disentangled representation can reveal the factors of variation in the data [6]. Generally, existing disentangling methods can be mainly categorized into two different types. The first type of disentanglement involves separating the latent representations into two [79, 48, 132, 90] or three [41] parts. For example, Mathieu *et al.* [79] introduce a conditional VAE with adversarial training to disentangle the latent representations into label relevant and the remaining unspecified factors. Meanwhile, two-step disentanglement methods based on Autoencoder [48] or VAE [132] are also proposed. In those two-step methods, the first step is to extract the label relevant representations by training a classifier. Then, label irrelevant representations are obtained mainly via the reconstruction loss. All of these methods improve the disentanglement results by leveraging (partial) label information to minimize the cross-entropy loss. The second type of disentanglement, such as β-VAE [55], FactorVAE [63] and β-TCVAE [13], learns to separate each dimension in latent space without supervision. Although most of the disentanglement learning methods [90, 33, 34] have been proposed based on Autoencoder, especially VAEs [65], VAEs usually can not achieve high-quality generation in real-world scenarios, which is related to the training objective [38]. In our method, the proposed method integrates the Autoencoder and GAN, and separates the latent variables into two parts without any supervision. The discrete latent variables directly represent clusters, and the other continuous latent variables summarize the remaining unspecified factors of variation.

4.2 Method

Given a collection i.i.d. samples $\mathbf{x} = \{x^i\}_{i=1}^N$ (*e.g.*, images) drawn from an unknown data distribution P_{x}, where x^i is the i-th data sample and N is the size of the dataset, the standard GAN [42, 46] consists of two components: the generator G_θ and the discriminator D_ψ. G_θ defines a mapping from the latent space $\mathcal{Z}$ to the data space $\mathcal{X}$ and D_ψ can be considered as a mapping from the data space $\mathcal{X}$ to the probability of one sample being real or not. To achieve unsupervised conditional generation, we need to introduce an inference network E_ϕ to obtain the latent variables given the data sample.

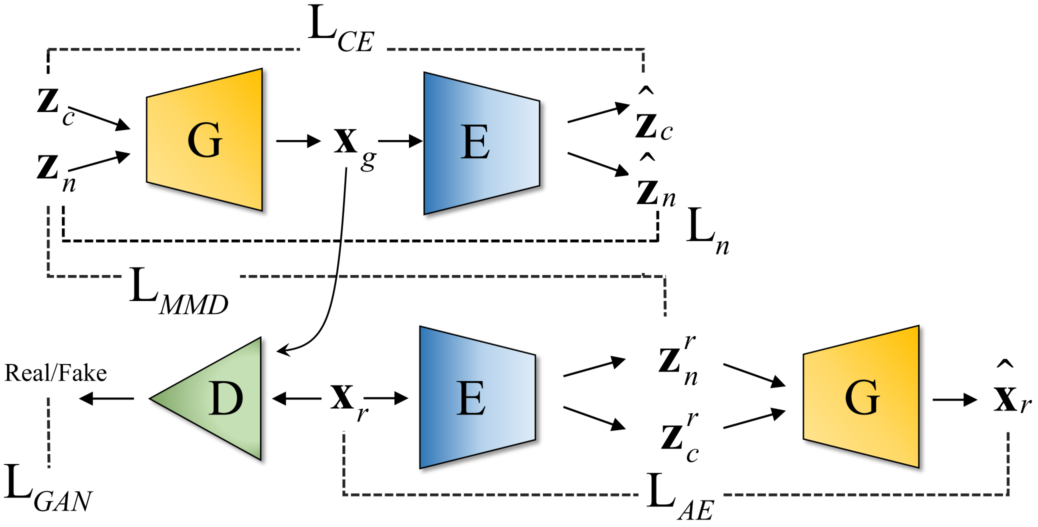


Figure 4.1: The architecture of DLS-Clustering (G: generator, E: encoder, D: discriminator). The latent representations are separated into one-hot discrete latent variables $\mathbf{z}_c$ and other factors of variation $\mathbf{z}_n$. The $\mathbf{z}_c$ and $\mathbf{z}_n$ are concatenated and fed into the G_θ for generation and the E_ϕ maps the samples ($\mathbf{x}_g$ and $\mathbf{x}_r$) back into latent space. The D_ψ is adopted for the adversarial training in the data space. Note that all generators share the same parameters and all encoders share the same parameters.

In this section, we first conduct a comprehensive analysis of ClusterGAN [81], and observe that there is a key loss item missing in the objective. To address this issue, we introduce an MMD-based regularization to enforce the inference network and the generator of standard GAN to form a deterministic Autoencoder. Meanwhile, the method enables us to disentangle the latent space $\mathbf{z}$ into the one-hot discrete latent variables $\mathbf{z}_c$, and the continuous latent variables $\mathbf{z}_n$ in an unsupervised manner. $\mathbf{z}_c$ naturally represents the categorical cluster information; $\mathbf{z}_n$ is expected to contain information of other variations. our goal is to learn a general method to project the data to the latent space, which is divided into the one-hot discrete latent variables directly related to clusters and the remaining unspecified continuous latent variables.

4.3 Unsupervised Conditional Generation

ClusterGAN [81] provides a new clustering method using GANs, which utilizes a joint distribution of discrete and continuous latent variables as the prior of GANs. Although it focuses on projecting the data to the latent space for clustering, it can be generalized to an unsupervised conditional generation framework. And the optimization is based on the combination of original

GAN loss, cycle-consistency loss, and cross-entropy loss.

$$\begin{aligned} &\min_{G,E}\max_{D}\mathcal{L}_{\text{Clus}}(G,D,E) = \\ &\underbrace{\mathbb{E}_{\mathbf{x}\sim P_{\text{x}}}[q(D_\psi(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}_c\sim P_{\text{c}},\mathbf{z}_n\sim P_{\text{n}}}[q(1-D_\psi(G_\theta(\mathbf{z}_c,\mathbf{z}_n)))]}_{\textcircled{1}} \\ &-\lambda_n \underbrace{\mathbb{E}_{\mathbf{z}_c\sim P_{\text{c}},\mathbf{z}_n\sim P_{\text{n}}}[c(E_\phi(G_\theta(\mathbf{z}_c,\mathbf{z}_n))_n,\mathbf{z}_n)]}_{\textcircled{2}} \\ &-\lambda_c \underbrace{\mathbb{E}_{\mathbf{z}_c\sim P_{\text{c}},\mathbf{z}_n\sim P_{\text{n}}}[c(E_\phi(G_\theta(\mathbf{z}_c,\mathbf{z}_n))_c,\mathbf{z}_c)]}_{\textcircled{3}}, \end{aligned} \tag{4.1}$$

where P_{x} is the real data distribution, P_{c} is the prior distribution of $\mathbf{z}_c$, and P_{n} is the prior distribution of $\mathbf{z}_n$. $c(\cdot,\cdot)$ is any measurable cost function, λ_n and λ_c are hyperparameters balancing these losses. For the original GAN [42], the function q is chosen as $q(t) = \log t$, and the Wasserstein GAN [46] applies $q(t) = t$. This adversarial density-ratio estimation [100] enforces Q_{x} to match P_{x}, as shown in term $\textcircled{1}$, $\mathcal{L}_{\text{GAN}}$. The term $\textcircled{2}$ and $\textcircled{3}$ are two constraints to the generator G_θ and the encoder E_ϕ, which correspond to the cycle-consistency of $\mathbf{z}_n$ and the cross-entropy loss on $\mathbf{z}_c$.

To analyze this clearly, the term $\textcircled{2}$ can be written as:

$$\begin{aligned} \mathcal{L}_n(G,E) &= -\mathbb{E}_{(\mathbf{x},\mathbf{z}_n)\sim Q_{xc}}[c(E_\phi(\mathbf{x})_n,\mathbf{z}_n)] \\ &= \mathbb{E}_{\mathbf{z}_c\sim P_{\text{c}},\mathbf{z}_n\sim P_{\text{n}}}[||E_\phi(G_\theta(\mathbf{z}_c,\mathbf{z}_n)) - \mathbf{z}_n||]. \end{aligned} \tag{4.2}$$

Thus, this loss term attempts to keep the cycle-consistency of $\mathbf{z}_n$ during optimization. After adding the recovery of $\mathbf{z}_n$, the information from $\mathbf{z}_n$ can be utilized for generation to a certain extent. However, since the dimension of $\mathbf{x}$ is much larger than the dimensions of $\mathbf{z}_c$ and $\mathbf{z}_n$, this constraint may become trivial for the generator-encoder (G-E) pair, and result in the generation of low-diversity samples.

The term $\textcircled{3}$ is the cross-entropy loss on $\mathbf{z}_c$:

$$\mathcal{L}_{\text{CE}}(G,E) = -\mathbb{E}_{(\mathbf{x},\mathbf{z}_c)\sim Q_{xc}}[\log(Q^E(\mathbf{z}_c|\mathbf{x}))], \tag{4.3}$$

where $Q^E(\mathbf{z}_c|\mathbf{x})$ is used to denote the conditional distribution induced by E_ϕ. $Q_{\mathbf{z}_c|\mathbf{x}}$ is the conditional distribution specified by the generator G. Therefore, minimizing loss term $\mathcal{L}_{\text{CE}}(G,E)$ is equivalent to minimizing the KL divergence between $Q_{\mathbf{z}_c|\mathbf{x}}$ and $Q^E_{\mathbf{z}_c|\mathbf{x}}$. However, ClusterGAN ignores the real data conditional distributions $P_{\mathbf{z}_c|\mathbf{x}}$ in the objective, which usually requires real category information to estimate. Even when the marginal distributions P_x and Q_x match perfectly through the term ①, ClusterGAN still can not guarantee that two conditional distributions $P_{\mathbf{z}_c|\mathbf{x}}$ and $Q^E_{\mathbf{z}_c|\mathbf{x}}$ are well matched. Only minimizing $\mathcal{L}_{\text{CE}}(G, E)$ makes G tend to generate data that are far from the decision boundaries of E_ϕ. In other words, the generated images for each category may be easily distinguishable by E_ϕ, but have low intra-class diversity. It is thus essential to incorporate $P_{\mathbf{z}_c|\mathbf{x}}$ in the objective function.

4.4 The Encoder-Generator Pair

Our above analysis of ClusterGAN reveals that simply adding an encoder cannot effectively achieve conditional generation, which has two main problems: trivial continuous latent variables recovery and missing real conditional distribution term, $P_{\mathbf{z}_c|\mathbf{x}}$. Therefore, we present to enforce E and G to form an Autoencoder (E-G pair) by introducing a distance-based regularizer. The real conditional distribution $P_{\mathbf{z}_c|\mathbf{x}}$ can also be estimated properly in an unsupervised manner. We define the following objective:

$$\begin{aligned} \min_{G,E} \mathcal{L}_{\text{E-G}}(G, E) = \\ \mathbb{E}_{Q_\phi(\mathbf{z}_n, \mathbf{z}_c|\mathbf{x})} \left[\log P_\theta(\mathbf{x}|\mathbf{z}_n, \mathbf{z}_c)\right] + \lambda \cdot \mathcal{D}_z\left(Q_z, P_z\right), \end{aligned} \tag{4.4}$$

where $\lambda > 0$ is a hyperparameter, $\mathcal{D}_z$ is an arbitrary divergence between Q_z and P_z, which encourages the encoded distribution Q_z to match the prior P_z. Because the latent variables $\mathbf{z} = (\mathbf{z}_c, \mathbf{z}_n)$, and the prior distribution $P_z(\mathbf{z}_c, \mathbf{z}_n) = P_c(\mathbf{z}_c)P_n(\mathbf{z}_n)$, these constraints can be added by simply penalizing the discrete variables part and the continuous variables part separately.

The constraint of continuous variables $\mathbf{z}_n$ can be considered to apply similar regularizations in the generative Autoencoder model like AAE [77] and WAE [98]. The former uses the GAN-based density-ratio trick to estimate the KL-divergence of distributions [100], and the latter minimizes the distance between distributions based on Maximum Mean Discrepancy (MMD) [43, 71]. We choose adversarial density-ratio estimation for modeling the data space because it can handle complex distributions. MMD-based regularizer is stable for optimization and works well with multivariate normal distributions [100]. Therefore, we choose MMD to quantify the distance between the prior distribution $P_n(\mathbf{z}_n)$ and the posterior distribution $Q_n(\mathbf{z}_n|\mathbf{x})$. Compared with WAE, we only penalize the continuous latent variables $\mathbf{z}_n$, not the whole latent variable. The regularizer $\mathcal{D}_z$ based on MMD is expressed as:

$$\begin{aligned} \mathcal{L}_{\text{MMD}}(E) = \frac{1}{N(N-1)} \sum_{\ell \neq j} k\left(z_n^{\ell}, z_n^{j}\right) + \\ \frac{1}{N(N-1)} \sum_{\ell \neq j} k\left(\hat{z}_n^{\ell}, \hat{z}_n^{j}\right) - \frac{2}{N^2} \sum_{\ell, j} k\left(z_n^{\ell}, \hat{z}_n^{j}\right), \end{aligned} \tag{4.5}$$

where $k(\cdot,\cdot)$ can be any positive definite kernel, $\{z_n^1, \ldots, z_n^N\}$ are sampled from the prior distribution $P_n(\mathbf{z}_n)$, $\hat{z}_n^i$ is sampled from the posterior distribution $Q_n(\mathbf{z}_n|\mathbf{x})$ and x^i is sampled from the real data samples for $i = 1, 2, \ldots, N$.

The constraint of $\mathbf{z}_c$ can't be applied explicitly without labels. Instead, we use a mean absolute error (MAE) criterion to estimate the encoding distribution $Q_\phi(\mathbf{z}|\mathbf{x})$ and the decoding distribution $P_\theta(\mathbf{x}|\mathbf{z})$, which are taken to be deterministic and can be replaced by E_ϕ and G_θ, respectively.

$$\mathcal{L}_{\text{AE}}(E, G) = \mathbb{E}_{\mathbf{x} \sim P_{\text{x}}}[|\mathbf{x} - G_\theta(E_\phi(\mathbf{x}))|]. \tag{4.6}$$

4.5 Disentangling Latent Space for Clustering

In addition to the encoder-generator pair, it also necessary to emphasize the generator-encoder pair for the disentanglement between discrete and continuous latent variables, as shown in Figure 4.1. Most of the existing methods [48, 132, 90] leverage labels to achieve the disentanglement of various factors. This work attempts to encourage independence between $Q_n(\mathbf{z}_n|\mathbf{x})$ and $Q_c(\mathbf{z}_c|\mathbf{x})$ as much as possible without labels.

We sample the latent variables $\mathbf{z} = (\mathbf{z}_c, \mathbf{z}_n)$ from the discrete-continuous prior, through the generator-encoder pair, it should output the identical discrete and continuous latent variables $(\hat{\mathbf{z}}_c, \hat{\mathbf{z}}_n)$. It enforces the generator to take advantage of extra information from $\mathbf{z}_c$. Besides, the recovery of latent variables ensure that outputs of the encoder E_ϕ are conditionally independent. When E_ϕ maps the real data sample $\mathbf{x}$ to latent representations $\mathbf{z}_c^r$ and $\mathbf{z}_n^r$, which are expected to be conditionally independent. The cross-entropy loss (Eq. 4.3) between $\mathbf{z}_c$ and $\hat{\mathbf{z}}_c$ can ensure that the latent variables $\hat{\mathbf{z}}_c$ only contain class-related information. Besides, to ensure that the latent variables $\hat{\mathbf{z}}_c$ or $\hat{\mathbf{z}}_c^r$ don't contain any class-related information, it is necessary to apply additional regularizers to penalize $\hat{\mathbf{z}}_n$ and $\hat{\mathbf{z}}_n^r$, which are related to the loss $\mathcal{L}_n$ and $\mathcal{L}_{\text{MMD}}$.

The objective function of our approach is integrated into the following form:

$$\mathcal{L} = \mathcal{L}_{\text{GAN}} + \mathcal{L}_{\text{AE}} + \beta_1\mathcal{L}_{\text{MMD}} + \beta_2\mathcal{L}_n + \beta_3\mathcal{L}_{\text{CE}}. \tag{4.7}$$

where the regularization coefficients β_1 to $\beta_3 \geq 0$, balancing the weights of different loss terms. Each term of Eq. 4.7 plays a different role for three components: generator G_θ, discriminator D_ψ, and encoder E_ϕ. Both $\mathcal{L}_{\text{GAN}}$ and $\mathcal{L}_{\text{AE}}$ are related to G_θ and E_ϕ, which constrain the whole latent variables. The $\mathcal{L}_{\text{GAN}}$ term is also related to D_ψ, which focuses on distinguishing the true data samples from the fake samples generated by G_θ. $\mathcal{L}_{\text{MMD}}$ and $\mathcal{L}_n$ are related to continuous latent variables, and $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_c$ are related to discrete latent variables. All these loss terms are used to ensure that our algorithm disentangles the latent space generated from encoder into cluster information and remaining unspecified factors. The training procedure of DLS-Clustering applies jointly updating the parameters of G_θ, D_ψ and E_ϕ, as described in 2. We empirically set $\beta_1 = \beta_2$ to enable a reasonable adjustment of the relative importance of continuous and discrete parts.

Algorithm 2: The training procedure of DLS-Clustering.

```
Input: $\theta$, $\psi$, $\phi$ initial parameters of $G_\theta$, $D_\psi$ and $E_\phi$, the dimension of latent code $d_n$,
       the number of clusters K, the batch size B, the number of critic iterations per
       end-to-end iteration M, the regularization parameters $\beta_1$ - $\beta_4$
Output: The parameters of $G_\theta$, $D_\psi$ and $E_\phi$
Data: Training data set $\mathbf{x}$
while not converged do
    for i=1, ..., M do
        Sample $\mathbf{z}_n \sim P(\mathbf{z}_n)$ a batch of random noise
        Sample $\mathbf{z}_c$ a batch of random one-hot vectors
        $\mathbf{z} \leftarrow (\mathbf{z}_c, \mathbf{z}_n)$
        $\mathbf{x}_g \leftarrow G_\theta(\mathbf{z})$
        Sample $\mathbf{x}_r \sim P_x$ a batch of the training dataset
        $\psi \leftarrow \nabla_\psi(D_\psi(\mathbf{x}_r) - D_\psi(\mathbf{x}_g))$
    Sample $\mathbf{z}_n \sim P(\mathbf{z}_n)$ a batch of random noise
    Sample $\mathbf{z}_c$ a batch of random one-hot vectors
    $\mathbf{z} \leftarrow (\mathbf{z}_c, \mathbf{z}_n)$
    $\mathbf{x}_g \leftarrow G_\theta(\mathbf{z})$
    $(\hat{\mathbf{z}}_c, \hat{\mathbf{z}}_n) \leftarrow E_\phi(\mathbf{x}_g)$, $(\mathbf{z}_c^r, \mathbf{z}_n^r) \leftarrow E_\phi(\mathbf{x}_r)$
    $\mathbf{z}' \leftarrow (\mathbf{z}_c, \mathbf{z}_n^r)$ , $\mathbf{z}^r \leftarrow (\mathbf{z}_c^r, \mathbf{z}_n^r)$
    $\mathbf{x}_g' \leftarrow G_\theta(\mathbf{z}')$ , $\hat{\mathbf{x}}_r \leftarrow G_\theta(\mathbf{z}^r)$
    $(\hat{\mathbf{z}}_c', \hat{\mathbf{z}}_n^r) \leftarrow E_\phi(\mathbf{x}_g')$
    $\theta \leftarrow \nabla_\theta(-D_\psi(G_\theta(z)) + ||\mathbf{x}_r - \hat{\mathbf{x}}_r||_2^2 + \beta_1\,\text{MMD}(\mathbf{z}_n^r, \mathbf{z}_n) + \beta_2||\mathbf{z}_n^r - \hat{\mathbf{z}}_n^r||_2^2 +$
      $\beta_3\mathcal{H}(\mathbf{z}_c, \hat{\mathbf{z}}_c)) + \beta_4\mathcal{H}(\mathbf{z}_c, \hat{\mathbf{z}}_c'))$
    $\phi \leftarrow \nabla_\phi(||\mathbf{x}_r - \hat{\mathbf{x}}_r||_2^2 + \beta_1\,\text{MMD}(\mathbf{z}_n^r, \mathbf{z}_n) + \beta_2||\mathbf{z}_n^r - \hat{\mathbf{z}}_n^r||_2^2 + \beta_3\mathcal{H}(\mathbf{z}_c, \hat{\mathbf{z}}_c)) + \beta_4\mathcal{H}(\mathbf{z}_c, \hat{\mathbf{z}}_c'))$
```

Table 4.1: The dimensions of $\mathbf{z}_c$ and $\mathbf{z}_n$ in DLS-Clustering for different datasets. Note that the dimension of one-hot discrete latent variables $\mathbf{z}_c$ is equal to the number of clusters.

Dataset	MNIST	Fashion-10	YTF	Pendigits	10x_73k	COIL-100
$\mathbf{z}_c$	10	10	41	10	8	100
$\mathbf{z}_n$	25	40	60	5	30	100

4.6 Experiments

In this section, we perform a variety of experiments to evaluate the effectiveness of our proposed method, including clusters assignment via $\mathbf{z}_c$ and visualization studies of $\mathbf{z}_n$. We also conduct ablation experiments to understand the contribution of various loss terms.

4.6.1 Data sets

The clustering experiments are carried out on six datasets: MNIST [68], Fashion-MNIST [116], YouTube-Face (YTF) [114], Pendigits [2], 10x_73k [131], and COIL-100 [83]. Both of the first two datasets contain 70k images with 10 categories, and each sample is a 28 × 28 grayscale image. YTF contains 10k face images of size 55 × 55, belonging to 41 categories. The Pendigits dataset contains a time series of (x, y) coordinates of hand-written digits. It has 10 categories and contains 10992 samples, and each sample is represented as a 16-dimensional vector. The 10x_73k dataset contains 73233 data samples of single-cell RNA-seq counts of 8 cell types, and the dimension of each sample is 720. The multi-view object image dataset COIL-100 has 100 clusters and contains 7200 images of size 128 × 128.

4.6.2 Implementation

We implement different neural network structures for G_θ, D_ψ, and E_ϕ to handle different types of data. For the image datasets (MNIST, Fashion-MNIST, and YTF), we employ the similar G_θ and D_ψ of DCGAN [91] with conv-deconv layers, batch normalization and leaky ReLU activations with a slope of 0.2. The E_ϕ uses the same architecture as D_ψ except for the last layer. For the Pendigits and 10x_73k datasets, the G_θ, D_ψ, and E_ϕ are the MLP with 2 hidden layers of 256 hidden units each. Table 4.2 summarizes the network structures of different datasets. The model parameters have been initialized following the random normal distribution. For the prior distribution of our method, we randomly generate the discrete latent code $\mathbf{z}_c$, which is equal to one of the elementary one-hot encoded vectors in $\mathbb{R}^K$, then we sample the continuous latent code from $\mathbf{z}_n \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{d_n})$, here $\sigma = 0.10$. The sampled latent code $\mathbf{z} = (\mathbf{z}_c, \mathbf{z}_n)$ is used as the input of G_θ to generate samples. The dimensions of $\mathbf{z}_c$ and $\mathbf{z}_n$ are shown in Table 4.1. We implement the MMD loss with RBF kernel [98] to penalize the posterior distribution $Q_\phi(\mathbf{z}_n|\mathbf{x})$. The improved GAN variant with a gradient penalty [46] is used in all experiments. To obtain the cluster assignment, we directly use the argmax over all softmax probabilities for different clusters. The following regularization parameters work well during all experiments: $\lambda = 10$, $\beta_1 = \beta_2 = 1$, $\beta_3 = \beta_4 = 10$. We implement the models in Python using the TensorFlow library and train them on one NVIDIA DGX-1 station.

Table 4.2: The structure summary of the generator (G), discriminator (D), and encoder (E) in DLS-Clustering for different datasets.

Dataset	Layer Type	G-1/D-4/E-4	G-2/D-3/E-3	G-3/D-2/E-2	G-4/D-1/E-1
MNIST	Conv-Deconv	$4 \times 4 \times 64$	$4 \times 4 \times 128$	-	-
Fashion-10	Conv-Deconv	$4 \times 4 \times 64$	$4 \times 4 \times 128$	-	-
YTF	Conv-Deconv	$5 \times 5 \times 32$	$5 \times 5 \times 64$	$5 \times 5 \times 128$	$5 \times 5 \times 256$
Pendigits	MLP	256	256	-	-
10x_73k	MLP	256	256	-	-

4.6.3 Evaluation of Disentanglement

We further explore the disentanglement capability of DLS-Clustering on dSprites dataset. We follow the same experimental settings and hyperparameters tuning as FactorVAE [63], InfoGAN [15] and InfoGAN-CR [72] for fair comparisons. We provide the experimental details in Appendix, and focus on explaining the results in this section. As shown in Table 4.3, our method also achieves excellent disentanglement performance. Compared with InfoGAN-CR, we implement the proposed double-cycle consistency to replace the contrastive regularizer (CR) based on the InfoGAN architecture, which has two latent variables. These consistencies force the generator to generate different samples while fixing one latent variable and changing another latent variable. This is beneficial for disentanglement, as it simulates the latent traversal experiments and encourages distinct changes in generated samples. In addition, ModelCentrality is proposed by [72] for unsupervised model selection to evaluate the trained models on an unlabelled dataset. It's naturally suitable for our unsupervised conditional generation settings.

Table 4.3: Comparison results based on different disentanglement metrics on the dSprites dataset.The score 1.0 denotes a perfect disentanglement. All the baseline results are from [72]. The proposed DLS-Clustering achieves desirable scores in most cases. The implementation of DLS-Clustering is based on the source code of InfoGAN-CR, and MC (ModelCentrality) denotes an unsupervised model selection scheme [72].

Model	FactorVAE	DCI	Modularity	MIG	BetaVAE
VAE	0.63 ± 0.06	0.30 ± 0.10	-	0.10	-
β-TCVAE	0.62 ± 0.07	0.29 ± 0.10	-	**0.45**	-
HFVAE	0.63 ± 0.08	0.39 ± 0.16	-	-	-
β-VAE	0.63 ± 0.10	0.41 ± 0.11	-	0.21	-
FactorVAE	0.82	-	-	0.15	-
FactorVAE (1.0)	0.79 ± 0.01	0.67 ± 0.03	0.79 ± 0.01	0.27 ± 0.03	0.79 ± 0.02
FactorVAE (10.0)	0.83 ± 0.01	0.70 ± 0.02	0.79 ± 0.0	0.40 ± 0.01	0.83 ± 0.0
FactorVAE (40.0)	0.82 ± 0.01	0.74 ± 0.01	0.77 ± 0.01	0.43 ± 0.01	0.84 ± 0.01
FactorVAE + MC	0.84 ± 0.0	0.73 ± 0.01	0.82 ± 0.0	0.37 ± 0.0	0.86 ± 0.0
IB-GAN	0.80 ± 0.07	0.67 ± 0.07	-	-	-
InfoGAN	0.82 ± 0.01	0.60 ± 0.02	0.94 ± 0.01	0.22 ± 0.01	0.87 ± 0.01
InfoGAN-CR + MC	0.92 ± 0.0	0.77 ± 0.0	**0.99 ± 0.0**	**0.45 ± 0.0**	0.99 ± 0.0
Ours + MC	**0.936 ± 0.0**	**0.790 ± 0.0**	0.985 ± 0.0	0.378 ± 0.0	**0.998 ± 0.0**

4.6.4 Evaluation of DLS-Clustering algorithm

To evaluate clustering results, we report two standard evaluation metrics: Clustering Purity (ACC) and Normalized Mutual Information (NMI). We compare DLS-Clustering with four clustering baselines: K-means [76], Non-negative Matrix Factorization (NMF) [69]. We also compare our method with the state-of-the-art clustering approaches based on GAN and Autoencoder, respectively. For GAN-based approaches, ClusterGAN [81] is chosen as it achieves the superior clustering performance compared to other GAN models (*e.g.*, InfoGAN). For Autoencoder-based methods such as DEC [117], DCN [119] and DEPICT [37], Dual Autoencoder Network (DualAE) [122] are used for comparison. In addition, the deep spectral clustering (SpectralNet) [93] and joint unsupervised learning (JULE) [120] are also included in the comparison.

Table 4.4 reports the best clustering metrics of different models from 5 runs. Our method achieves significant performance improvement on Fashion-10, YTF, Pendigits, and 10x_73k datasets than other methods. Particularly, while all other methods perform worse than K-means on the 16-dimensional Pendigit dataset, our method significantly outperforms K-means in both ACC (0.847 vs. 0.793) and NMI (0.803 vs. 0.730). DLS-Clustering achieves the best ACC result on YTF dataset while maintaining comparable NMI value. For MNIST dataset, DLS-Clustering achieves close to the best performance on both ACC and NMI metrics. To further evaluate the performance of DLS-Clustering on large numbers of clusters, we compare our clustering method with K-means on

Table 4.4: Comparison of clustering algorithms on five benchmark datasets. The results marked by (*) are from existing sklearn.cluster.KMeans package. The dash marks (-) mean that the source code is not available or that running released code is not practical, all other results are from [81] and [122]. SpecNet and ClusGAN mean SpectralNet and ClusterGAN.

Method	MNIST		Fashion-10		YTF		Pendigits		10x_73k	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
K-means	0.532	0.500	0.474	0.512	0.601	0.776	0.793*	0.730*	0.623*	0.577*
NMF	0.560	0.450	0.500	0.510	-	-	0.670	0.580	0.710	0.690
DEC	0.863	0.834	0.518	0.546	0.371	0.446	-	-	-	-
DCN	0.830	0.810	-	-	-	-	0.720	0.690	-	-
JULE	0.964	0.913	0.563	0.608	0.684	0.848	-	-	-	-
DEPICT	0.965	0.917	0.392	0.392	0.621	0.802	-	-	-	-
SpecNet	0.800	0.814	-	-	0.685	0.798	-	-	-	-
InfoGAN	0.890	0.860	0.610	0.590	-	-	0.720	0.730	0.620	0.580
ClusGAN	0.950	0.890	0.630	0.640	-	-	0.770	0.730	0.810	0.730
DualAE	**0.978**	**0.941**	0.662	0.645	0.691	**0.857**	-	-	-	-
Ours	0.975	0.936	**0.693**	**0.669**	**0.721**	0.790	**0.847**	**0.803**	**0.905**	**0.820**

Coil-100 dataset using three standard evaluation metrics: ACC, NMI, and Adjusted Rand Index (ARI). As shown in Table 4.7, DLS-Clustering achieves better performance on all three metrics.

4.6.5 Evaluation of Generation Quality

Table 4.5: Comparison of FID score to reveal the quality of generated samples from GAN methods (Lower is better).

Method	Ours	ClusterGAN	WGAN	InfoGAN
MNIST	**0.15**	0.81	0.88	1.88
Fashion	**0.67**	0.91	0.95	11.04

Table 4.6: Comparison of mean SSIM scores of 200 pairs to reveal the diversity of generated samples from GAN methods (Lower is better).

Class	0	1	2	3	4	5	6	7	8	9
ClusterGAN	0.362	0.599	0.263	**0.314**	0.315	0.282	0.351	**0.388**	0.340	0.427
Ours	**0.343**	**0.576**	**0.231**	0.316	**0.312**	**0.259**	**0.322**	0.392	**0.336**	**0.377**

To demonstrate the quality and diversity of generated samples from DLS-Clustering, we first calculate the Frechet Inception Distance (FID) [54] score of generated samples, as shown in Table 4.5. The FID scores on MINST and Fashion are significantly lower than those of ClusterGAN. Our method shows that the estimation of real conditional distribution can improve the quality of generated samples. Then we randomly sample 200 pairs of generated images from one category to calculate structural similarity (SSIM) [113, 112] for diversity evaluation on MNIST data. This evaluation method for diversity has also been used in AC-GAN [87]. The SSIM scores range between 0.0 and 1.0, and lower mean scores indicate that samples from the same class are less similar. As shown in Table 4.6, our method achieves lower SSIM scores on most classes, which demonstrates that it can enhance the diversity of generation. The diversity of generated images indicates that there exist different latent variables for generative factors, except the cluster information. To further understand these generative factors, we change the value of one single dimension from $[-0.5, 0.5]$ in $\mathbf{z}_n$ while fixing other dimensions and the discrete latent variables $\mathbf{z}_c$. As shown in Figure 4.2, the value changing leads to semantic changes in generated samples. The changed dimensions represent the tilt, style, and width factors of digits, which shows the potential to disentangle the latent space.

4.6.6 Evaluation on More Images

We also use the t-SNE [75] algorithm to visualize $\mathbf{z}_n$ of MNIST datasets and compare them to ClusterGAN and the original data. As shown in Figure 4.3, we can observe different categories in the original data. In ClusterGAN, there are still several distinguishable clusters. In contrast, our method can make these points more cluttered in latent space, which doesn't contain obvious category information in the $\mathbf{z}_n$. Therefore, our method demonstrates another excellent capability: all these informative continuous factors are independent of cluster information.

Table 4.7: The clustering results on the Coil-100 dataset, which has a large number of clusters (K=100).

Method	ACC	NMI	ARI
K-means	0.668	0.836	0.574
ClusterGAN	0.615	0.797	0.487
Our method	**0.822**	**0.911**	**0.764**

We first evaluate the scalability of DLS-Clustering to large numbers of clusters on the COIL-100 dataset(100 clusters). Here, we compare our clustering method with K-means on three standard evaluation metrics: ACC, NMI and Adjusted Rand Index (ARI). As shown in Table 4.7, DLS-Clustering achieves better performance on all three metrics. DLS-Clustering even gains an increase of 0.154 on ACC metric. We also perform image generation task on Coil-100 dataset, to further verify the generative performance, which involves mapping latent variables to the data space.

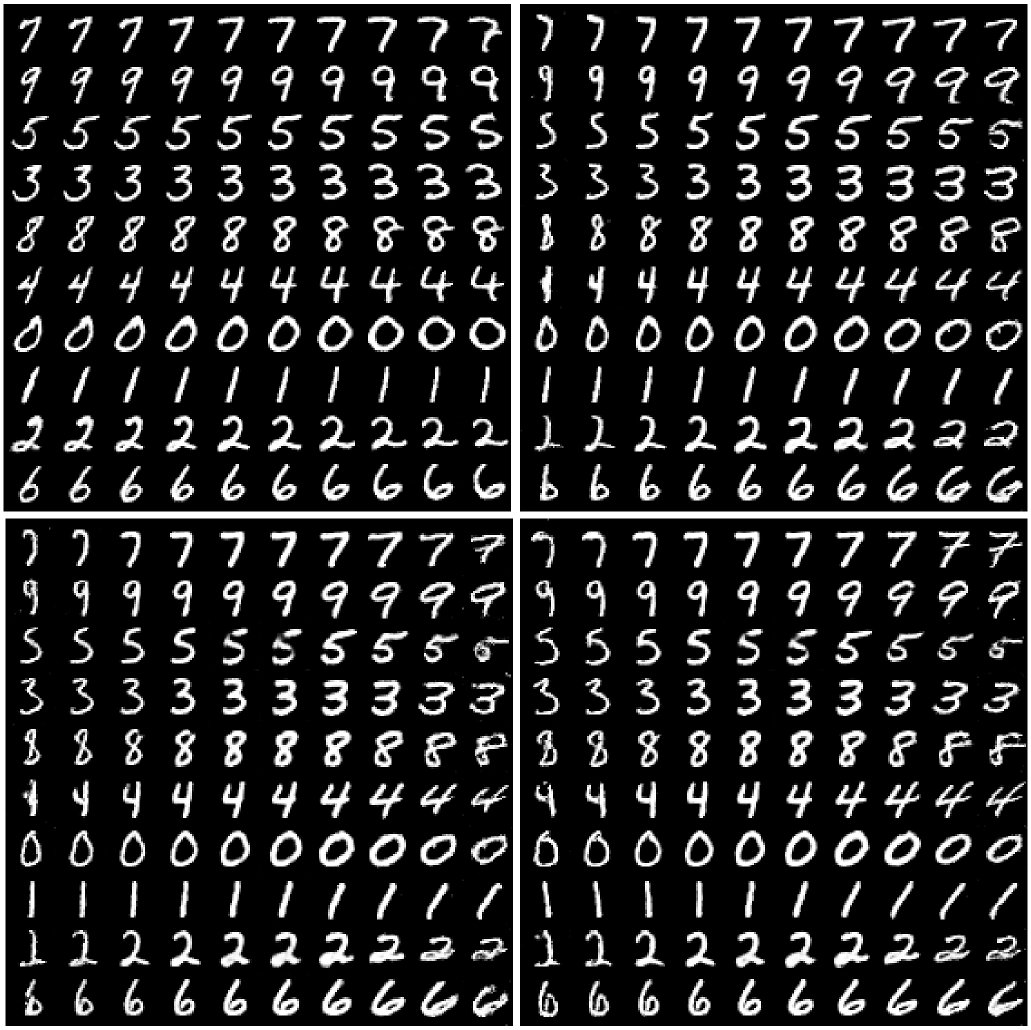


Figure 4.2: Samples generated on fixed discrete latent codes from the models trained on MNIST.

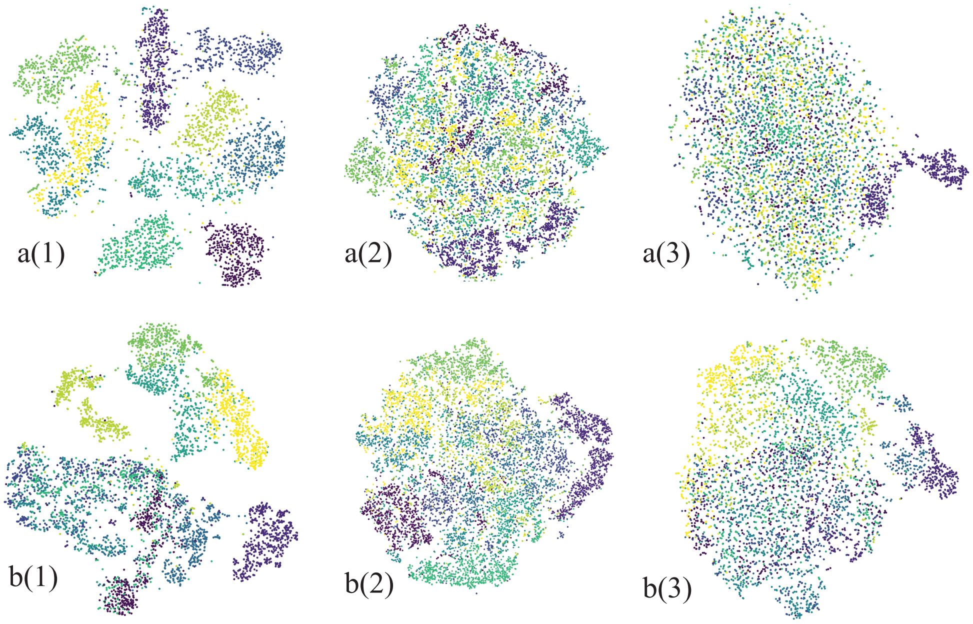


Figure 4.3: The t-SNE visualization of raw data (a), $\mathbf{z}_n$ of ClusterGAN (b) and DLS-Clustering (c) on MNIST dataset. The bulk of samples in the right part of a(3) is a small group of "1" images. The reason that they are not well mixed may be due to their low complexity.

Figure 4.4 shows the generated samples by fixing one-hot discrete latent variables, which are diverse and realistic. The continuous latent variables represent meaningful factors such as the pose, location and orientation information of objects. Therefore, the disentanglement of latent space not only provides the superior clustering performance, but also retains the remarkable ability of diverse and high-quality image generation.

Besides, we further evaluate the proposed method on more complex dataset: CIFAR-10. The implementation is based on Google compare-gan framework [1]. The spectral normalization is used on both generator and discriminator. We use the same class-conditional BatchNorm in the generator as Lucic *et al.* [74], to incorporate the category information from $\mathbf{z}_n$. For the encoder, we use the pre-trained SimCLR [14] model to improving training efficiency, and apply 2-layer MLP as project head to map the learned representations to $\mathbf{z}_n$ and $\mathbf{z}_c$. The self-supervised SimCLR model is pre-trained by following the official implementation [2]. Table 4.8 shows that DLS-Clustering achieves close to the best clustering performance on ACC. Because our method learns cluster memberships from conditional generation without labels, it's also necessary to evaluate the generation results of

[1] `https://github.com/google/compare_gan`

[2] `https://github.com/google-research/simclr`

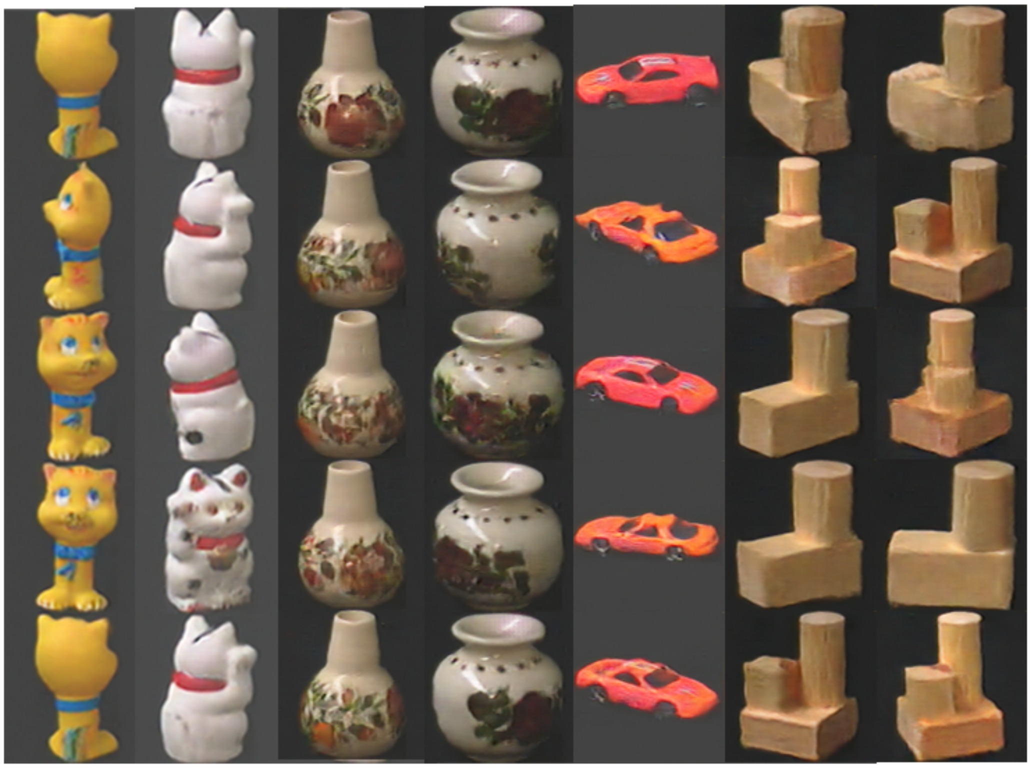


Figure 4.4: The samples generated on fixed discrete latent variables from the models trained on Coil-100 dataset. Each column corresponds to a specific cluster.

images. As shown in Table 4.9, our method also maintains the quality of image generation, which enables to achieve the superior clustering results.

4.6.7 Ablative Analysis

We perform the ablative analysis of our losses (Table 4.10). The $\mathcal{L}_{\mathrm{AE}}$ and $\mathcal{L}_{\mathrm{MMD}}$ are critical in our model. The inference network and the generator form a deterministic encoder-decoder pair. To minimize the reconstruction loss $\mathcal{L}_{\mathrm{AE}}$, the generator G_θ needs to learn to generate realistic and diverse data samples. It also indirectly forces the $\mathbf{z}_c^r$ to contain only the category information. $\mathcal{L}_{\mathrm{MMD}}$ enforces the posterior distribution $Q_\phi(\mathbf{z}_n|\mathbf{x})$ to be close to the prior distribution $P(\mathbf{z}_n)$. The clustering performance gain is also from the loss terms $\mathcal{L}_{\mathrm{CE}}$ and $\mathcal{L}_{\mathrm{n}}$.

Table 4.8: CIFAR-10 images clustering results. All baseline results are from [61]. The value marked by (*) is the best (mean) results in [61], and they also report that avg. ± STD is 0.576 ± 0.050.

Method	ACC	NMI
K-means	0.229	0.087
DCGAN (2015) [91]	0.315	0.265
JULE (2016) [120]	0.272	0.192
DEC (2016) [117]	0.301	0.257
DAC (2017) [12]	0.522	0.396
DeepCluster (2018) [9]	0.374	-
ADC (2018) [49]	0.325	-
IIC (2019) [61]	0.617 (0.576)*	0.513
GATCluster(2020) [84]	0.610	0.475
Ours	0.605	0.484

Table 4.9: FID results on the CIFAR-10 dataset (smaller is better). The results marked by (*) are from [78].

Method	FID Score
DCGANs [91]	29.7*
WGAN-GP (2017) [46]	29.3
SN-SMMDGAN (2018) [3]	25.0
MSGAN (2019) [78]	28.7*
Ours	28.5 ± 0.02

4.7 Conclusion

In this work, we present DLS-Clustering, a new clustering method that directly obtains the cluster assignments by disentangling the latent space. Unlike most existing latent space clustering algorithms, our method does not build 'clustering-friendly' latent space explicitly and does not need extra clustering operation. Therefore, our method avoids the difficulty of integrating latent feature construction and clustering. Furthermore, our method does not disentangle class relevant features from class non-relevant features. The disentanglement in our method is targeted to extract "cluster information" from data. Although our method does not depend on any explicit distance calculation in the latent space, the distance between data may be implicitly defined by the neural networks.

Table 4.10: Ablations on MNIST dataset. Each row shows the removal of a loss term. The full setting includes all loss terms.

Ablative analysis	ACC	NMI
No $\mathcal{L}_{\text{CE}}$	0.899	0.863
No $\mathcal{L}_n$	0.868	0.851
No $\mathcal{L}_{\text{MMD}}$	0.812	0.829
No $\mathcal{L}_{\text{AE}}$	0.672	0.488
Full setting	**0.976**	**0.941**

The two cycle-consistencies ($\mathbf{x} \rightarrow (\mathbf{z}_c, \mathbf{z}_n) \rightarrow \mathbf{x}$, $(\mathbf{z}_c, \mathbf{z}_n) \rightarrow \mathbf{x} \rightarrow (\mathbf{z}_c, \mathbf{z}_n)$) in DLS-Clustering can help avoid the triviality of $\mathbf{z}_n$, and then avoid the generation of low diversity images in some degree. We have used the real images to train the encoder-generation pair ($\mathbf{x} \rightarrow (\mathbf{z}_c, \mathbf{z}_n) \rightarrow \mathbf{x}$), which can help the encoder to estimate the real conditional distribution. However, due to the unsupervised fashion of clustering, the conditional distribution $Q(\mathbf{z}_c|\mathbf{x})$ specified by the generator of GAN may not match well with the true conditional distribution $P(\mathbf{z}_c|\mathbf{x})$ in real data, which is the case in both ClusterGAN and our DLS-Clustering. This may be another reason for the low diversity conditional generation [40]. Improving GAN to create more diverse images is an important task for future work.

Chapter 5

Future Work and Conclusion

5.1 Future Work

Despite the great success of unsupervised contrastive learning in representation learning, it still suffers from some limitations due to the unsupervised setting. We identify two possible future improvements to obtain better feature representation via (1) mitigating sampling bias problems, and (2) more powerful learning frameworks.

5.1.1 Mitigating sampling bias problems

Most typical contrastive learning methods have a common weakness: negative samples are drawn randomly from the training data, which leads to the sampling bias problem [21]. In other words, many negative samples from the same category are undesirably pushed apart in the representation space. Recent works [62, 60] extend the contrastive loss to a fully-supervised setting to utilize the label information. The supervised contrastive loss considers all samples from the same class as positives against the negatives from different classes of the batch. However, high-quality labels are expensive. Instead, Cl-InfoNCE [99] proposes the weakly-supervised contrastive representation by using additional auxiliary information for data, such as hashtags in Instagram images. The auxiliary information can be used to extract noisy labels for supervised contrastive learning. Although supervised contrastive learning addresses the sampling bias problem and can obtain better feature representation, the definition of positive samples becomes different, as the supervised contrastive loss distinguishes positive and negative samples according to whether they belong to the same class. Thus it ignores instance-wise discrimination, which is helpful to learn semantic or structural information. It is necessary to propose new instance-wise discrimination methods under the fully-supervised setting.

Even without labels or additional information, it's still promising to use the pretrained contrastive representation to obtain clusters and iteratively learn better representations. For example, PCL [70] utilizes the centroids of clusters in the momentum training framework for contrastive learning. The usage of cluster information is not efficient in PCL due to the following two reasons. First, PCL requires clustering the samples with different numbers of clusters multiple times, thus leading to different clusters being noisy and variable. Second, PCL performs clustering on the features from the momentum encoder, the centroids are fixed during minibatch updates, and can not contribute to the gradients as well as the other samples. In addition to using the momentum encoder, it's also possible to use the existing contrastive model as the teacher network to guide the student network learning from scratch. Unlike typical KD, the teacher's knowledge is extracted as the cluster information to help the student to pull samples closer to their centroids while pushing them away from the centroids of other classes.

5.1.2 Better Learning Framework

The current progress of representation learning relies heavily on advances in contrastive learning, and its key component is to generate different representations from the same input via data augmentation. Recently, MAE [50] proposes to use masked autoencoders instead of contrastive learning to learn representation and demonstrate excellent performance. This work also suggests that masked autoencoding can work well for vision and language. The usage of autoencoding is to ensure an approximate one-to-one mapping between the input and feature representations. This prevents the learned representation from collapsing to a single point, while similar functionality is achieved in the contrastive learning by negative samples. Although we also consider autoencoding in this dissertation, our main purpose is to introduce a new type of clustering algorithm that directly obtains the cluster information during the disentanglement of latent space. It's possible to add clustering tasks in the MAE-like learning framework to improve both representation learning and clustering results. In addition, MAE unifies the representation learning of vision and language, making it possible to learn the presentation of images and text simultaneously. We note that MAE achieves similar linear classification results as contrastive learning on ImageNet, but outperforms all existing methods when finetuned on ImageNet. This indicates that the class information is helpful to improve the quality of representation, given a sufficient number of labels. Therefore, another future research direction is to propose a framework for simultaneous clustering and representation learning.

5.2 Conclusion

Unsupervised contrastive learning has emerged as an important representation learning method by pulling positive samples closer and pushing negative samples apart. Once the low-dimensional representations are learned, K-means clustering or an additional component training are usually performed to obtain cluster assignment, which forms the widely used two-stage framework.

In this dissertation, we have shown that several solutions can be explored to improve clustering and representation learning. First, to improve feature representations on small models, we employ knowledge distillation which provides a promising solution by transferring knowledge from high-capacity teachers. We introduce the Dual-level Knowledge Distillation (DLKD) by explicitly combining knowledge alignment and correlation together instead of using one single contrastive objective. The proposed DLKD is task-agnostic and model-agnostic, and enables effective knowledge transfer from supervised or self-supervised pretrained teachers to students. Second, to improve the clustering performance, we propose Contrastive Learning based Clustering (CLC), which uses contrastive learning to directly learn cluster assignment. We decompose the representation into two parts: one encodes the categorical information under an equipartition constraint, and the other captures the instance-wise factors. We theoretically analyze the proposed contrastive loss and reveal that CLC sets different weights for the negative samples while learning cluster assignments. Experimental evaluation shows that CLC achieves overall state-of-the-art or highly competitive clustering performance on multiple benchmark datasets. In particular, we achieve 53.4% accuracy on the full ImageNet dataset and outperform existing methods by large margins (+ 10.2%). Furthermore, we also propose to achieve clustering via unsupervised conditional generation, which directly learns cluster assignments from disentangled latent space without additional clustering methods. The proposed method enforces the encoder and the generator of GAN to form an encoder-generator pair in addition to the generator-encoder pair. Experiments show that our method outperforms existing generative model-based clustering methods on multiple datasets.

Appendices

Appendix A CLC and Instance-wise Contrastive Learning

Given the similarity $s_{i,j}$, it can be written as: $s_{i,j} = \mathbf{z}_i \cdot \mathbf{z}_j = \mathbf{z}_i^c \cdot \mathbf{z}_j^c + \mathbf{z}_i^n \cdot \mathbf{z}_j^n$. Let $s_{i,j}^c = \mathbf{z}_i^c \cdot \mathbf{z}_j^c$, $s_{i,j}^n = \mathbf{z}_i^n \cdot \mathbf{z}_j^n$, then $s_{i,j} = s_{i,j}^c + s_{i,j}^n$ and $\exp(s_{i,j}/\tau) = \exp(s_{i,j}^c/\tau) \cdot \exp(s_{i,j}^n/\tau)$. We can re-write the standard contrastive loss as follows:

$$\mathcal{L}_{\text{InfoNCE}}\left(\mathbf{x}_i\right) = -\log\left[\frac{\exp\left(s_{i,i}/\tau\right)}{\sum_{k\neq i}\exp\left(s_{i,k}/\tau\right) + \exp\left(s_{i,i}/\tau\right)}\right] \tag{1}$$

$$= -\log\left[\frac{\exp(s_{i,i}^c/\tau)\cdot\exp(s_{i,i}^n/\tau)}{\sum_{k\neq i}(\exp(s_{i,k}^c/\tau)\cdot\exp(s_{i,k}^n/\tau)) + \exp(s_{i,i}^c/\tau)\cdot\exp(s_{i,i}^n/\tau)}\right] \tag{2}$$

$$= -\log\left[\frac{\exp(s_{i,i}^n/\tau)}{\sum_{k\neq i}(\exp(s_{i,k}^c/\tau)\cdot\exp(s_{i,k}^n/\tau))\cdot\exp(-s_{i,i}^c/\tau) + \exp(s_{i,i}^n/\tau)}\right] \tag{3}$$

$$= -\log\left[\frac{\exp(s_{i,i}^n/\tau)}{\sum_{k\neq i}(\exp((s_{i,k}^c - s_{i,i}^c)/\tau)\cdot\exp(s_{i,k}^n/\tau)) + \exp(s_{i,i}^n/\tau)}\right]. \tag{4}$$

Since $s_{i,j}^c$ and $s_{i,j}^n$ are symmetric in equation 2, the standard contrastive loss can also be written as:

$$\mathcal{L}_{\text{InfoNCE}}\left(\mathbf{x}_i\right) = -\log\left[\frac{\exp\left(s_{i,i}/\tau\right)}{\sum_{k\neq i}\exp\left(s_{i,k}/\tau\right) + \exp\left(s_{i,i}/\tau\right)}\right] \tag{5}$$

$$= -\log\left[\frac{\exp(s_{i,i}^c/\tau)}{\sum_{k\neq i}(\exp((s_{i,k}^n - s_{i,i}^n)/\tau)\cdot\exp(s_{i,k}^c/\tau)) + \exp(s_{i,i}^c/\tau)}\right]. \tag{6}$$

Then we observe that the coefficient $\exp((s_{i,k}^n - s_{i,i}^n)/\tau)$ can be considered as a constant because $\mathbf{z}^n$ satisfies the properties of alignment and uniformity, as analyzed in Section 3.7.3. Thus, equation 6 can be considered as a standard contrastive loss on $\mathbf{z}^c$. As shown in Figure 3.4 (a), $\mathbf{z}^c$ can retain well the categorical information, which is beneficial for learning cluster assignment via contrastive learning.

A.1 Sinkhorn-Knopp algorithm

We provide the PyTorch-like pseudo-code for Sinkhorn-Knopp algorithm, which is used in our all experiments.

Algorithm 3: PyTorch-like Pseudo-code for Sinkhorn-Knopp.

```
# eps: weight for the entropy regularization term. Defaults to 0.05.
# niters: number of times to perform row and column normalization. Defaults to 3.
# K: dimension of zc (number of clusters)
# B: batch size

def sinkhorn-knopp(logits, eps=0.05, niters=3):
    Q = exp(logits / eps).T
    K, B = Q.shape
    # make the matrix sums to 1
    Q /= sum(Q)

    for _ in range(niters):
        # normalize each row: total weight per prototype must be 1/K
        sum_of_rows = sum(Q, dim=1, keepdim=True)
        Q /= sum_of_rows
        Q /= K
        # normalize each column: total weight per sample must be 1/B
        Q /= sum(Q, dim=0, keepdim=True)
        Q /= B
    Q *= B # the colomns must sum to 1 so that Q is an assignment
    return Q.T
```

A.2 Implementation details on smaller datasets

Most of the implementation on smaller datasets (CIFAR10, CIFAR100-20 and STL10) is directly taken from the tutorial of MoCo on CIFAR10[1]. We apply the ResNet-18 as the backbone and a two-layer MLP as the projection head (512-D hidden layer and ReLU) to obtain a 128-D feature vector for contrastive learning. We adopt the same data augmentation in SimCLR [14] but disable the blur like MoCo v2 [16]. For the contrastive learning experiments, we apply the SGD optimizer with a learning rate of 6e-2, a weight decay of 5e-4 and a momentum of 0.9. The cosine scheduler is used to schedule the learning rate. We train the parameters from scratch for 1200 epochs using the batch size of 512. For the linear classification in the ablation studies, we train the linear classifier via the SGD optimizer with a learning rate of 30 and the cosine scheduler for 100 epochs.

A.3 Implementation details on ImageNet subsets

Most of the implementation on ImageNet subsets (100 classes and 200 classes) is directly taken from MoCo v2 repo[2]. We apply the ResNet-50 as the backbone and a two-layer MLP as the projection head (2048-D hidden layer and ReLU) to obtain a 256-D feature vector for contrastive learning. We follow the same data augmentation settings in MoCo v2. To speed up training, we directly initialize the backbone with the released weights (800 epochs pretrained), and only train the MLP projection head for 400 epochs using the batch size of 256. The weights are updated through an SGD optimizer with a learning rate of 0.03, a momentum of 0.9, and a weight decay of 1e-4.

[1] https://github.com/lightly-ai/lightly

[2] https://github.com/facebookresearch/moco

Although there are advanced data augmentation and training strategies, we adopt the same settings in MoCo v2 for a fair comparison.

A.4 Implementation details on full ImageNet

Most of the implementation on full ImageNet is directly taken from MoCo v3 repo[3]. We choose the ResNet-50 as the backbone and two 2-layer MLPs (4096-D hidden layer and ReLU) for the projection head and the prediction head following BYOL [44]. We obtain a 256-D feature vector for contrastive learning. The proposed contrastive loss is scaled by a constant 2τ, to make the training less sensitive to the choice of τ. The data augmentation is the same as BYOL. We freeze the backbone and initialize it using the released weights (1000 epochs). We only train two MLP heads for 200 epochs using a batch size of 2048. We use the LARS optimizer with a learning rate of 0.3, a weight decay of 1e-6 and a momentum of 0.9.

A.5 Implementation details of linear classification

We apply the same settings as MoCo v2 and train the model from scratch for 200 epochs on full ImageNet using the proposed contrastive loss. We apply the pretrained ResNet-50 backbone as a feature encoder and evaluate it for linear classification on ImageNet, VOC and Places205 datasets. The implementation code is directly taken from [4]. We continue to evaluate the pretrained ResNet-50 backbone on object detection tasks in the following Section.

[3] `https://github.com/facebookresearch/moco-v3`

[4] `https://github.com/maple-research-lab/AdCo`

Bibliography

[1] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9163–9171, 2019.

[2] Fevzi Alimoglu and Ethem Alpaydin. Methods of combining multiple classifiers based on different representations for pen-based handwritten digit recognition. In *Proceedings of the Fifth Turkish Artificial Intelligence and Artificial Neural Networks Symposium (TAINN 96)*. Citeseer, 1996.

[3] Michael Arbel, Dougal Sutherland, Mikołaj Bińkowski, and Arthur Gretton. On gradient regularizers for mmd gans. In *Advances in Neural Information Processing Systems*, pages 6700–6710, 2018.

[4] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019.

[5] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019.

[6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[7] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007.

[8] Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[9] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[10] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[12] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5879–5887, 2017.

[13] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620, 2018.

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

[15] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

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

[17] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[18] Xinlei Chen*, Saining Xie*, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

[19] Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Darkrank: Accelerating deep metric learning via cross sample similarities transfer. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[20] Xu Cheng, Zhefan Rao, Yilan Chen, and Quanshi Zhang. Explaining knowledge distillation by quantifying the knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12925–12935, 2020.

[21] Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *arXiv preprint arXiv:2007.00224*, 2020.

[22] Keh-Shih Chuang, Hong-Long Tzeng, Sharon Chen, Jay Wu, and Tzong-Jer Chen. Fuzzy c-means clustering with spatial information for image segmentation. *computerized medical imaging and graphics*, 30(1):9–15, 2006.

[23] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[24] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[25] Stanford CS231N. Tiny imagenet visual recognition challenge. *URL https://tiny-imagenet.herokuapp.com*, 2015.

[26] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

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

[28] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

[29] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[30] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[31] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *arXiv preprint arXiv:1907.02544*, 2019.

[32] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015.

[33] Emilien Dupont. Learning disentangled joint continuous and discrete representations. In *Advances in Neural Information Processing Systems*, pages 710–720, 2018.

[34] Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, Narayanaswamy Siddharth, Brooks Paige, Dana H Brooks, Jennifer Dy, and Jan-Willem van de Meent. Structured disentangled representations. *arXiv preprint arXiv:1804.02086*, 2018.

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

[36] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. Seed: Self-supervised distillation for visual representation. *arXiv preprint arXiv:2101.04731*, 2021.

[37] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5736–5745, 2017.

[38] Partha Ghosh, Mehdi SM Sajjadi, Antonio Vergari, Michael Black, and Bernhard Schölkopf. From variational to deterministic autoencoders. *arXiv preprint arXiv:1903.12436*, 2019.

[39] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[40] Mingming Gong, Yanwu Xu, Chunyuan Li, Kun Zhang, and Kayhan Batmanghelich. Twin auxilary classifiers gan. In *Advances in Neural Information Processing Systems*, pages 1328–1337, 2019.

[41] Abel Gonzalez-Garcia, Joost van de Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. In *Advances in Neural Information Processing Systems*, pages 1287–1298, 2018.

[42] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[43] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[44] 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.

[45] Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. Towards a deep and unified understanding of deep neural models in nlp. In *International Conference on Machine Learning*, pages 2454–2463, 2019.

[46] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[47] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *IJCAI*, pages 1753–1759, 2017.

[48] Naama Hadad, Lior Wolf, and Moni Shahar. A two-step disentanglement method. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 772–780, 2018.

[49] Philip Haeusser, Johannes Plapp, Vladimir Golkov, Elie Aljalbout, and Daniel Cremers. Associative deep clustering: Training a classification network with no labels. In *German Conference on Pattern Recognition*, pages 18–32. Springer, 2018.

[50] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. pages 16000–16009, 2022.

[51] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[53] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3779–3787, 2019.

[54] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[55] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017.

[56] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[57] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[58] Qianjiang Hu, Xiao Wang, Wei Hu, and Guo-Jun Qi. Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1074–1083, 2021.

[59] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1558–1567. JMLR. org, 2017.

[60] Ashraful Islam, Chun-Fu Richard Chen, Rameswar Panda, Leonid Karlinsky, Richard Radke, and Rogerio Feris. A broad study on the transferability of visual representations with contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8845–8855, 2021.

[61] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9865–9874, 2019.

[62] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

[63] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

[64] Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In *Advances in neural information processing systems*, pages 2760–2769, 2018.

[65] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[66] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 3519–3529. PMLR, 09–15 Jun 2019.

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

[68] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[69] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

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

[71] Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, pages 1718–1727, 2015.

[72] Zinan Lin, Kiran Thekumparampil, Giulia Fanti, and Sewoong Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *International Conference on Machine Learning*, pages 6127–6139. PMLR, 2020.

[73] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2604–2613, 2019.

[74] Mario Lucic, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. High-fidelity image generation with fewer labels. *arXiv preprint arXiv:1903.02271*, 2019.

[75] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[76] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[77] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[78] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1429–1437, 2019.

[79] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, pages 5040–5048, 2016.

[80] Nairouz Mrabah, Mohamed Bouguessa, and Riadh Ksantini. Adversarial deep embedded clustering: on a better trade-off between feature randomness and feature drift. *arXiv preprint arXiv:1909.11832*, 2019.

[81] Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. Clustergan: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4610–4617, 2019.

[82] T Nathan Mundhenk, Daniel Ho, and Barry Y Chen. Improvements to context based self-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9339–9348, 2018.

[83] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-20). 1996.

[84] Chuang Niu, Jun Zhang, Ge Wang, and Jimin Liang. Gatcluster: Self-supervised gaussian-attention network for image clustering. In *European Conference on Computer Vision*, pages 735–751. Springer, 2020.

[85] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[86] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9359–9367, 2018.

[87] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651. PMLR, 2017.

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

[89] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[90] Massimiliano Patacchiola, Patrick Fox-Roberts, and Edward Rosten. Y-autoencoders: disentangling latent representations via sequential-encoding. *arXiv preprint arXiv:1907.10949*, 2019.

[91] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[92] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[93] Uri Shaham, Kelly Stanton, Henry Li, Boaz Nadler, Ronen Basri, and Yuval Kluger. Spectralnet: Spectral clustering using deep neural networks. *arXiv preprint arXiv:1801.01587*, 2018.

[94] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[95] Igor Susmelj, Matthias Heller, Philipp Wirth, Jeremy Prescott, and Malte Ebner. Lightly. *GitHub. Note: https://github.com/lightly-ai/lightly*, 2020.

[96] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017.

[97] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

[98] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein autoencoders. *arXiv preprint arXiv:1711.01558*, 2017.

[99] Yao-Hung Hubert Tsai, Tianqin Li, Weixin Liu, Peiyuan Liao, Ruslan Salakhutdinov, and Louis-Philippe Morency. Learning weakly-supervised contrastive representations. *arXiv preprint arXiv:2202.06670*, 2022.

[100] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

[101] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1365–1374, 2019.

[102] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pages arXiv–1807, 2018.

[103] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

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

[105] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[106] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

[107] Chu Wang, Marcello Pelillo, and Kaleem Siddiqi. Dominant set clustering and pooling for multi-view 3d object recognition. *arXiv preprint arXiv:1906.01592*, 2019.

[108] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2021.

[109] Jianfeng Wang, Jingdong Wang, Jingkuan Song, Xin-Shun Xu, Heng Tao Shen, and Shipeng Li. Optimized cartesian k-means. *IEEE Transactions on Knowledge and Data Engineering*, 27(1):180–192, 2014.

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

[111] Yukang Wang, Wei Zhou, Tao Jiang, Xiang Bai, and Yongchao Xu. Intra-class feature variation distillation for semantic segmentation. In *European Conference on Computer Vision*, pages 346–362. Springer, 2020.

[112] Zhou Wang and Alan C Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.

[113] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[114] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 529–534, 2011.

[115] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[116] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[117] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487, 2016.

[118] Guodong Xu, Ziwei Liu, Xiaoxiao Li, and Chen Change Loy. Knowledge distillation meets self-supervision. *arXiv preprint arXiv:2006.07114*, 2020.

[119] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 3861–3870. JMLR.org, 2017.

[120] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2016.

[121] Jing Yang, Brais Marinez, Samsung AI Center, Adrian Bulat, and Georgios Tzimiropoulos. Knowledge distillation via softmax regres-sion representation learning. International Conference on Learning Representations (ICLR), 2021.

[122] Xu Yang, Cheng Deng, Feng Zheng, Junchi Yan, and Wei Liu. Deep spectral clustering using dual autoencoder network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4066–4075, 2019.

[123] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017.

[124] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[125] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[126] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

[127] Lihi Zelnik-manor and Pietro Perona. Self-tuning spectral clustering. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2005.

[128] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6688–6697, 2020.

[129] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[130] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[131] Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8:14049, 2017.

[132] Zhilin Zheng and Li Sun. Disentangling latent space for vae by label relevant/irrelevant dimensions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12192–12201, 2019.

[133] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. *Advances in neural information processing systems*, 27, 2014.