Dynamic Conceptional Contrastive Learning for Generalized Category Discovery

문제점.
GCD에서
labeled set에서 supervised contrastive learning,
total set에서 self-supervised contrastive learning을 수행하는데,
self-supervised contrastive learning 수행시 sample과 그것의 augmentation 된 sample 사이만 가까워지게 하고 나머지는 멀리 보내니까,
동일한 conception 내의 sample들끼리의 관계가 간과되어서 representation learning 성능을 저하시킨다.

직관적으로, 동일한 conception 내의 sample들은 feature space 상에서 서로 가까워지는게 맞지 않냐.
이때 이 conception 은, classes, super-classes, sub-classes.
classes : 버스, 자전거, 기차, 토끼, 고래
super-classes : 운송수단, 동물
sub-classes : 노랑색 기차, 흰색 기차

제시한 해결방안.
DCCL(Dynamic Conceptional Contrastive Learning) framework
= DCG(Dynamic Conception Generation) + DCL(Dual-level Contrastive Learning)

DCG = '*dynamically generate conceptions based on the hyper-parameter-free clustering method equipped with the proposed semi-supervised conceptional consolidation*'
DCL = '*optimize the model with conception level and instance-level contrastive learning objectives, where we maintain a dynamic memory to ensure comparing with the up-to-date conceptions*'

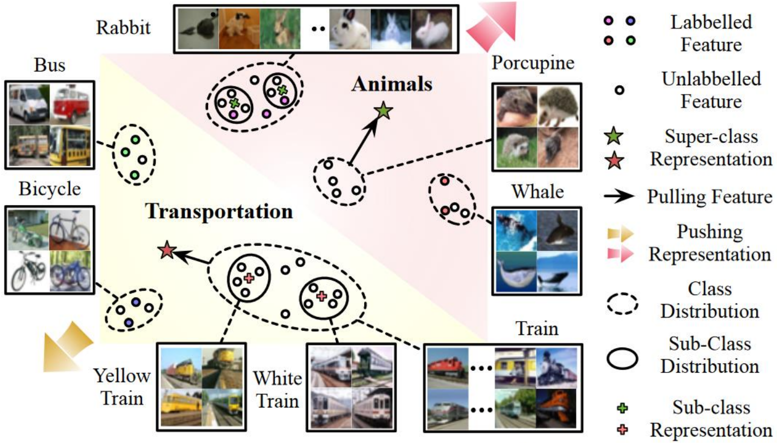


Figure 1. Diagram of the proposed Dynamic Conceptional Contrastive Learning (DCCL). Samples from the conceptions should be close to each other. For example, samples from the same classes (bus) at the class level, samples belonging to the transportation (bus and bicycle) at the super-class level, and samples from trains with different colors at the sub-class level. Our DCCL potentially learns the underlying conceptions in unlabeled data and produces more discriminative representations.

Dynamic Conceptional Contrastive Learning for Generalized Category Discovery

Contribution.

1. We propose a novel dynamic conceptional contrastive learning (DCCL) framework to effectively leverage the underlying relationships between unlabeled samples for learning discriminative representation for GCD.
2. We introduce a novel dynamic conception generation and update mechanism to ensure consistent conception learning, which encourages the model to produce more discriminative representation.
3. Our DCCL approach consistently achieves superior performance over state-of-the-art GCD algorithms on both generic and fine-grained tasks.

3. Dynamic Conceptional Contrastive Learning

3.1. Problem Formulation

$\mathcal{D} = \mathcal{D}^L \cup \mathcal{D}^U$

$\mathcal{D}^L = \{(\mathbf{x}_i^L, \mathbf{y}_i^L)\}_{i=1}^{M^L} \in \mathcal{X} \times \mathcal{Y}^L$: labeled subset

$\mathcal{D}^U = \{(\mathbf{x}_i^U, \mathbf{y}_i^U)\}_{i=1}^{M^U} \in \mathcal{X} \times \mathcal{Y}^U$: unlabeled subset $\mathbf{y}_i^U \in \mathcal{Y}^U$, $\mathcal{Y}^L \subset \mathcal{Y}^U$

N^L : labeled data class 개수 : known

N^U : unlabeled data class 개수 : unknown

f : feature extractor

h : MLP projection head

$\mathbf{v}_i = f(\mathbf{x}_i)$: extracted representation

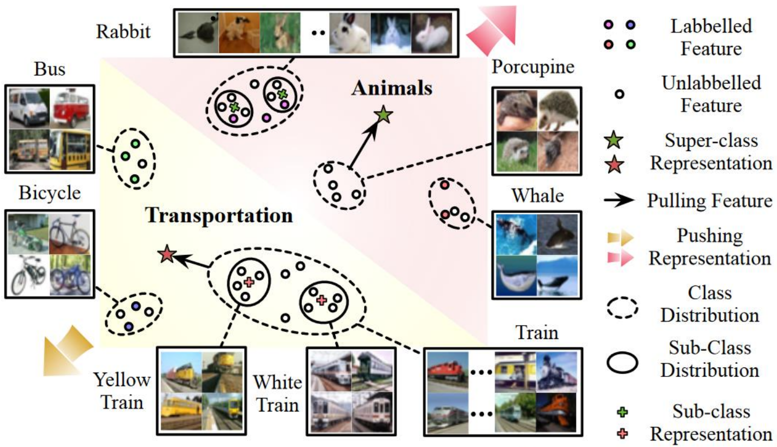


Figure 1. Diagram of the proposed Dynamic Conceptional Contrastive Learning (DCCL). Samples from the conceptions should be close to each other. For example, samples from the same classes (bus) at the class level, samples belonging to the transportation (bus and bicycle) at the super-class level, and samples from trains with different colors at the sub-class level. Our DCCL potentially learns the underlying conceptions in unlabeled data and produces more discriminative representations.

3.2. Overview

By alternately executing the dynamic conception generation (DCG) and DCL, DCL benefits from informative supervised information to generate higher-quality representations. Meanwhile, DCG gradually produces more comprehensive guidance based on a deeper understanding of conceptual relationships.

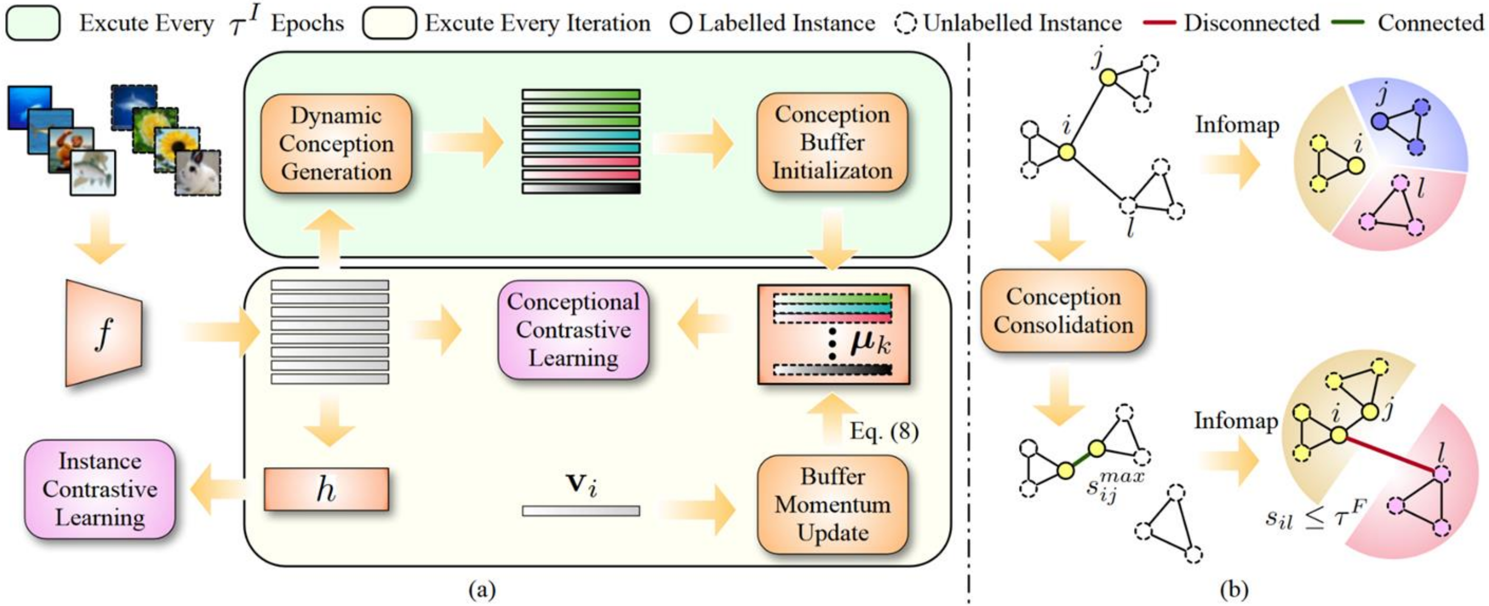


Figure 2. (a) Overview of our DCCL framework. We first extract features and cluster the features to generate conceptional labels, then initialize conception representations by our DCG, and last learn representations by joint instance-level and conception-level objectives. During the training process, the DCG and dual-level representation learning are performed alternately, in which the conception buffer is updated every iteration to keep the consistency of the changing instance features and conceptional representations. (b) Illustration of the proposed conception consolidation. Without consolidating the relationships of conceptions by label information, Infomap tends to over-cluster data and thus provides the supervision that has a high risk to over-correct affinities between neighbor instances.

3.3. Dynamic Conception Generation
DCG : hyper-parameter가 없는 Infomap 알고리즘 기반.
Specifically, we first propose a semi-supervised conceptional consolidation method to construct a similarity network, then execute the Infomap clustering algorithm on the constructed network to get conceptional label assignments, and finally calculate conception representation and initialize conceptional memory buffer.

Conception Consolidation.
In a given network, Infomap aims at partitioning semantic-similar sub-networks by the pattern of connections.
$\mathcal{A}$: adjacent matrix : represent the possible connection relationships among all instances

$$\mathcal{A}_{ij} = \begin{cases} s_i^{max}, & \text{if } \mathbf{y}_i, \mathbf{y}_j \in \mathcal{Y}^L \text{ and } \mathbf{y}_i = \mathbf{y}_j \\ s_{ij}, & \text{if } \mathbf{y}_i \text{ or } \mathbf{y}_j \in \mathcal{Y}^U \text{ and } s_{ij} > \tau^F \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

: weight of the edge of the i-th and j-th instances

$$s_i^{max} = \arg\max_j \{s_{ij} \mid j \in \mathcal{D}\}, \quad (2)$$

$$s_{ij} = [(\mathbf{v}_i / \|\mathbf{v}_i\|) \cdot (\mathbf{v}_j / \|\mathbf{v}_j\|) + 1]/2 \in [0, 1], \quad (3)$$

$\bullet$: dot product

$\|\cdot\|$: l2 normalization

τ^F : threshold to select high-confidence links

Through the conception consolidation illustrated in Fig. 2(b), we can establish a reliable relationship network with rich structural information for the subsequent clustering.

Remark.
We set the similarities of positive pairs with the maximal value of neighborhood similarities instead of 1.

Entropy Minimization Clustering.
In Infomap algorithm [26], the clustering problem is equivalent to minimizing the entropy that represents the minimum description length of the coding network.

$\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^{M^L+M^U} \in \mathcal{Y}^G$: conceptional label set for both labeled and unlabeled instances

$\mathcal{V}$: extracted feature vectors

$\mathcal{D}^G = \{(\mathbf{v}_i, \mathbf{c}_i)\}_{i=1}^{M^L+M^U} \in \mathcal{V} \times \mathcal{Y}^G$: generated feature dataset

$|\mathcal{Y}^G|$: number of estimated conceptions

Conceptional Memory Initialization.
= Initialization of the conceptional memory buffer (CMB)
Conception-level memory buffer : provides dynamic conceptional representations for dual level contrastive learning
We use the mean feature vector of the instances that share the same conceptional label to form a unique conceptional representation.

$$\mathcal{U} = \{\boldsymbol{\mu}_k\}_{k=1}^{K}, \quad \boldsymbol{\mu}_k = \frac{1}{|\mathcal{D}_k^G|} \sum_{\mathbf{v}_i \in \mathcal{D}_k^G} \mathbf{v}_i, \quad K = |\mathcal{Y}^G|, \quad (4)$$

: initial conceptional representation set

$\mathcal{D}_k^G$: k-th conception subset $\mathbf{v}_i \in \mathcal{D}_k^G, \mathbf{c}_i = k$

During whole training process, the initialization of CMB is executed every τ^I epochs on center-cropped images. Thus, the number of conceptions K is dynamically changing along with model training.

3.4. Dual-Level Contrastive Learning

Conception-Level Contrastive Learning.
Based on the generated conceptional representations in Sec. 3.3, we propose to perform conception-level contrastive learning.

1. sample N^C conception labels and a fixed number N^I of instances for each conception label
-> 그 결과 mini-batch $\mathcal{B}^C$ with $N^C \times N^I$ instances.

2. Each instance representation is compared to all the conceptional representations.
1) We pull the instance representation from its corresponding conceptional representation closer and push the instance representation away from other conceptional representations.

$$\mathcal{L}_i^C = -\log \frac{\exp\left(\mathbf{v}_i \cdot \boldsymbol{\mu}_{\mathbf{c}_i} / \tau^C\right)}{\sum_{k=1, k \neq \mathbf{c}_i}^{K} \exp(\mathbf{v}_i \cdot \boldsymbol{\mu}_k / \tau^C)}, \quad (5)$$

: conceptional contrastive loss function

τ^C : temperature hyper-parameter to control the strength of the conception-level contrastive learning.

2) In addition, in order to explicitly encourage learned representations with a large inter-conception margin, we propose a dispersion loss to further push the different conception representations away from each other.

$$\mathcal{L}(m,n) = \left[\left|\left| \frac{1}{|\mathcal{B}_m^C|} \sum_{\mathbf{v}_i \in \mathcal{B}_m^C} \mathbf{v}_i \right|\right| \cdot \left|\left| \frac{1}{|\mathcal{B}_n^C|} \sum_{\mathbf{v}_j \in \mathcal{B}_n^C} \mathbf{v}_j \right|\right| - \tau^M \right]_+, \quad (6)$$

: loss function for the m-th and the n-th conceptions in $\mathcal{B}^C$

τ^M : threshold to filter the conception pairs with high uncertainty

$$\mathcal{L}^D = \frac{1}{N^C} \sum_{m=1}^{N^C} \frac{1}{N^C} \sum_{n=1}^{N^C} \mathcal{L}(m,n). \quad (7)$$

: dispersion loss function over a mini-batch

Conceptional Memory Update.

we first adopt the resampling method as detailed in the previous section. Next, we propose to update the conception representation by each corresponding instance feature following a momentum update mechanism.

$$\boldsymbol{\mu}_{\mathbf{c}_i} \leftarrow \eta \boldsymbol{\mu}_{\mathbf{c}_i} + (1-\eta)\mathbf{v}_i, \qquad (8)$$

: momentum update mechanism

η : momentum updating factor

Remark.

DCCL updates the memory buffer and computes the losses both at the conceptional level, which consistently updates the conceptional representation to maintain the conceptional consistency during the whole training process.

Instance-Level Contrastive Learning.

Instance contrastive loss (ICL) : supervised contrastive loss and self-supervised contrastive loss : GCD와 동일하게 loss 줌.

x_i and $\hat{x}_i$: two views (random augmentations) of the same image in a randomly-sampled mini-batch $\mathcal{B}^I$

h : MLP projection head

The extracted representation $\mathbf{v}_i$ *is further projected by* h *to high-dimensional embedding space for instance-level contrastive learning.*

$$\mathcal{L}_i^I = (\lambda - 1)\log\frac{\exp\left(h\left(\mathbf{v}_i\right)\cdot h\left(\hat{\mathbf{v}}_i\right)/\tau\right)}{\sum_{j\in\mathcal{B}^I, j\neq i}\exp\left(h\left(\mathbf{v}_i\right)\cdot h\left(\mathbf{v}_j\right)/\tau^S\right)}$$
$$\frac{-\lambda}{|\mathcal{P}(i)|}\sum_{p\in\mathcal{P}(i)}\log\frac{\exp\left(h\left(\mathbf{v}_i^L\right)\cdot h\left(\mathbf{v}_p^L\right)/\tau\right)}{\sum_{j\in\mathcal{B}^L, j\neq i}\exp\left(h\left(\mathbf{v}_i^L\right)\cdot h\left(\mathbf{v}_j^L\right)/\tau^L\right)}, \qquad (9)$$

: ICL

$\mathcal{B}^L$: labeled subset within the mini-batch $\mathcal{B}^I$

$\mathcal{B}^I = \mathcal{B}^L \cup \mathcal{B}^U$

$\mathcal{P}(i)$: positive index set for the anchor image $i \in \mathcal{B}^L$

λ : trade-off factor to balance the contributions of self-supervised and supervised learning

3.5. Joint Optimization

During the whole training process, we alternately perform dynamic conception generalization and dual-level contrastive representation learning, until the maximal training epoch.

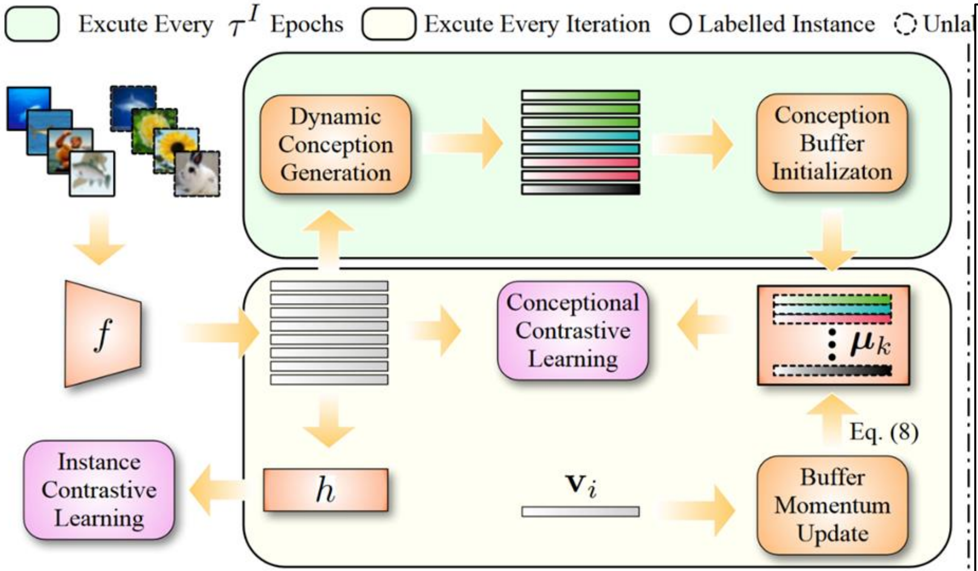
The overall objective over on mini-batch B is given by the weighted sum of each loss function:

$$\mathcal{L}_{total} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\mathcal{L}_i^I + \alpha\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\mathcal{L}_i^C + \beta\mathcal{L}^D, \qquad (10)$$

α and β : weights to adjust the strengths of two loss functions

In all experiments, we use l_2 normalized feature vector $||\mathbf{v}||_2$ for clustering evaluation.

Dynamic Conceptional Contrastive Learning for Generalized Category Discovery



(a)

Figure 2. (a) Overview of our DCCL framework. We first extract features and cluste
initialize conception representations by our DCG, and last learn representations by jo
During the training process, the DCG and dual-level representation learning are perf
is updated every iteration to keep the consistency of the changing instance features a
the proposed conception consolidation. Without consolidating the relationships of conceptions by label information, Infomap tends to over-cluster data and thus provides the supervision that has a high risk to over-correct affinities between neighbor instances.

Algorithm 1: Algorithm Pipeline of our DCCL

Input: Feature Extractor f, Projection Head h, Labeled data $\mathcal{D}^L$ and Unlabeled data $\mathcal{D}^U$.
Output: f and h.
for $n = 1$ ***in*** $[1, max\_epoch]$ **do**
  if n ***mod*** $\tau^I == 0$ **then**
    Extract features and construct adjacency matrix $\mathcal{A}$ by Eq. (1), Eq. (2) and Eq. (3);
    Perform InfoMap [27] clustering to assign conceptional labels $\mathcal{C}$;
    Initialize conceptional buffer by Eq. (4);
  end
  for $i = 1$ ***in*** $[1, max\_iteration]$ **do**
    Sample mini-batches from $\mathcal{D}^L \cup \mathcal{D}^U$;
    Calculate overall optimization objective by Eq. (10);
    Update f and h by SGD [25];
    Update conceptional buffer by Eq. (8);
  end
end

4. Experiments

4.1. Experimental Setup

Data and Evaluation Metric.

We follow [35](GCD), sample a subset of half the classes as "Old" categories. 50% of instances of each labeled class are drawn to form the labeled set, and all the remaining data constitute the unlabeled set.

For evaluation, we measure the clustering accuracy by comparing the predicted label assignment with the ground truth, following the protocol in [35](GCD).

Table 1. Statistics of the datasets and the splits for GCD. The first three are generic datasets while the last three are fine-grained datasets.

Dataset		CIFAR10 [17]	CIFAR100 [17]	ImageNet-100 [7]	CUB-200 [36]	SCars [16]	Pet [24]
Labelled	# Classes	5	80	50	100	98	19
	# Images	12,500	20,000	31,860	1,498	2,000	942
Unlabelled	# Classes	10	100	100	200	196	37
	# Images	37,500	30,000	95,255	4496	6,144	2,738

Table 2. Results on generic image recognition datasets.

Method	CIFAR10			CIFAR100			ImageNet-100		
	All	Old	New	All	Old	New	All	Old	New
k-means	83.6	85.7	82.5	52.0	52.2	50.8	72.7	75.5	71.3
RankStats+	46.8	19.2	60.5	58.2	77.6	19.3	37.1	61.6	24.8
UNO+	68.6	**98.3**	53.8	69.5	80.6	47.2	70.3	95.0	57.9
GCD	91.5	97.9	88.2	73.0	76.2	66.5	74.1	89.8	66.3
DCCL	**96.3**	96.5	**96.9**	**75.3**	**76.8**	**70.2**	**80.5**	**90.5**	**76.2**