Novel Class Discovery

Trends and Applications of Computer Vision

Noemi Canovi (230487), Kevin Depedri (229358),
Jacopo Donà (229369), Mostafa Haggag (229674)
University of Trento

1st December 2022

Outline

01
Introduction
Introduction to Novel Class Discovery (NCD) task

02
DTC
First attempt to address NCD task

03
AutoNovel
Improved approach to address NCD

04
Experiments
Results over AutoNovel runs and comparison

05
Future works
Introduce UNO, perform more experiments

01

Introduction

Introduction to Novel Class Discovery (NCD) task

Putting together related tasks

NCD

01

Semi-Supervised Learning

Tackle the problem of missing labels

02

Transfer Learning

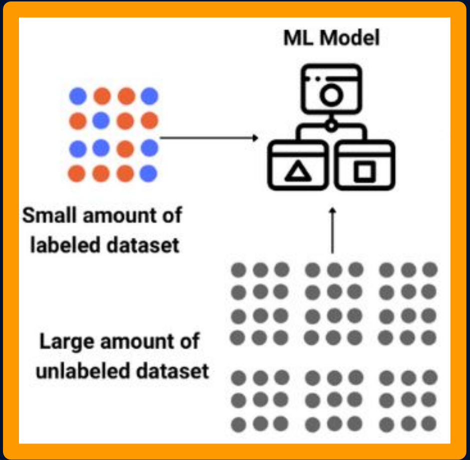
Leverage features from pre-trained models

03

Unsupervised Clustering

Find similarities between samples

Semi-Supervised Learning (SSL)

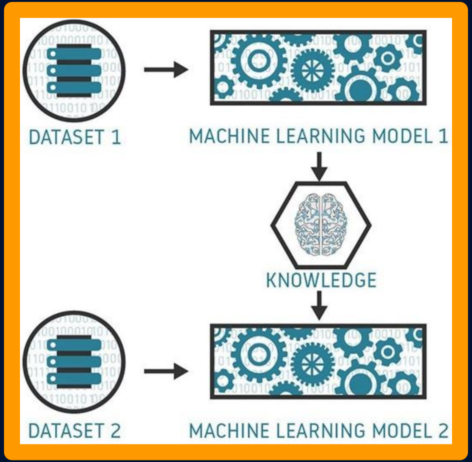


The goal is to solve a classification problem in which part of the data is labeled and the rest is not.

We assume that the classes in the labeled and unlabeled sets are the same.

In NCD this assumption does not hold. The classes between the two sets are disjoint. The unlabeled classes does not have any corresponding sample in the labeled dataset.

Transfer Learning

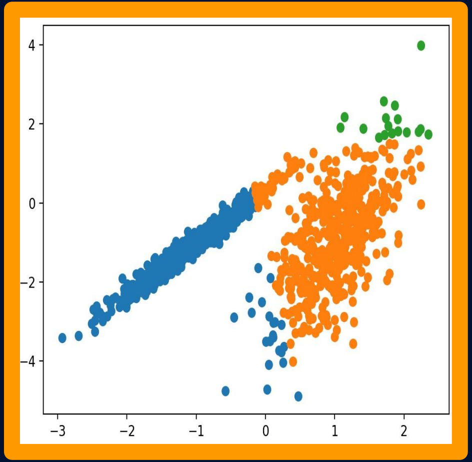


The goal is to take a model pre-trained over another (possibly bigger) dataset and leverage its features representation.

This model is then fine-tuned on task specific labeled data.

In NCD this is not fully possible. No labeled dataset is available for the new classes.

Unsupervised Clustering

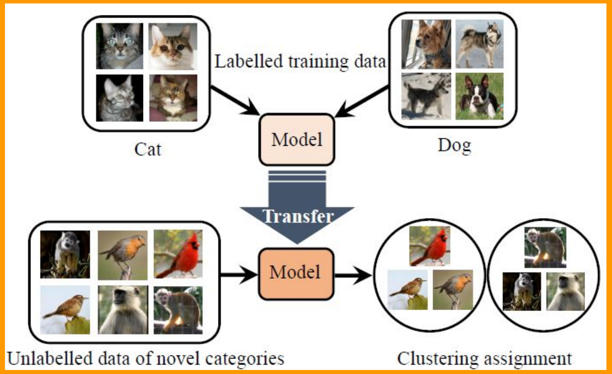


The goal is to partition unlabeled samples into different categories.

No labels are given. Therefore, samples are aggregated into clusters depending on their similarities.

In NCD this assumption is relaxed, since, labeled source data allows to learn features and pattern to distinguish between different classes.

Formal definition of NCD task



In Novel Class Discovery we assume to have two disjoint dataset:

1. Labeled dataset $D^l = \{(x^l_i, y^l_i), i = 1, ..., N\}$
2. Unlabeled dataset $D^u = \{x^u_i, i = 1, ..., M\}$

We know:

1. The $y^l_i \in \{1, ..., C^l\}$ labels of D^l
2. Number C^u of classes in the unlabeled dataset

We want to train a model that is able to classify images from both the datasets.

02

DTC

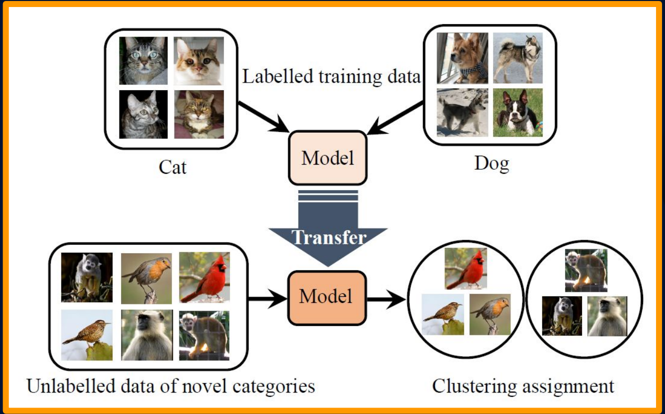
K. Han, A. Vedaldi, A. Zisserman, ICCV 2019
Learning to Discover Novel Visual Categories via Deep Transfer Clustering

DTC - Main Contributions

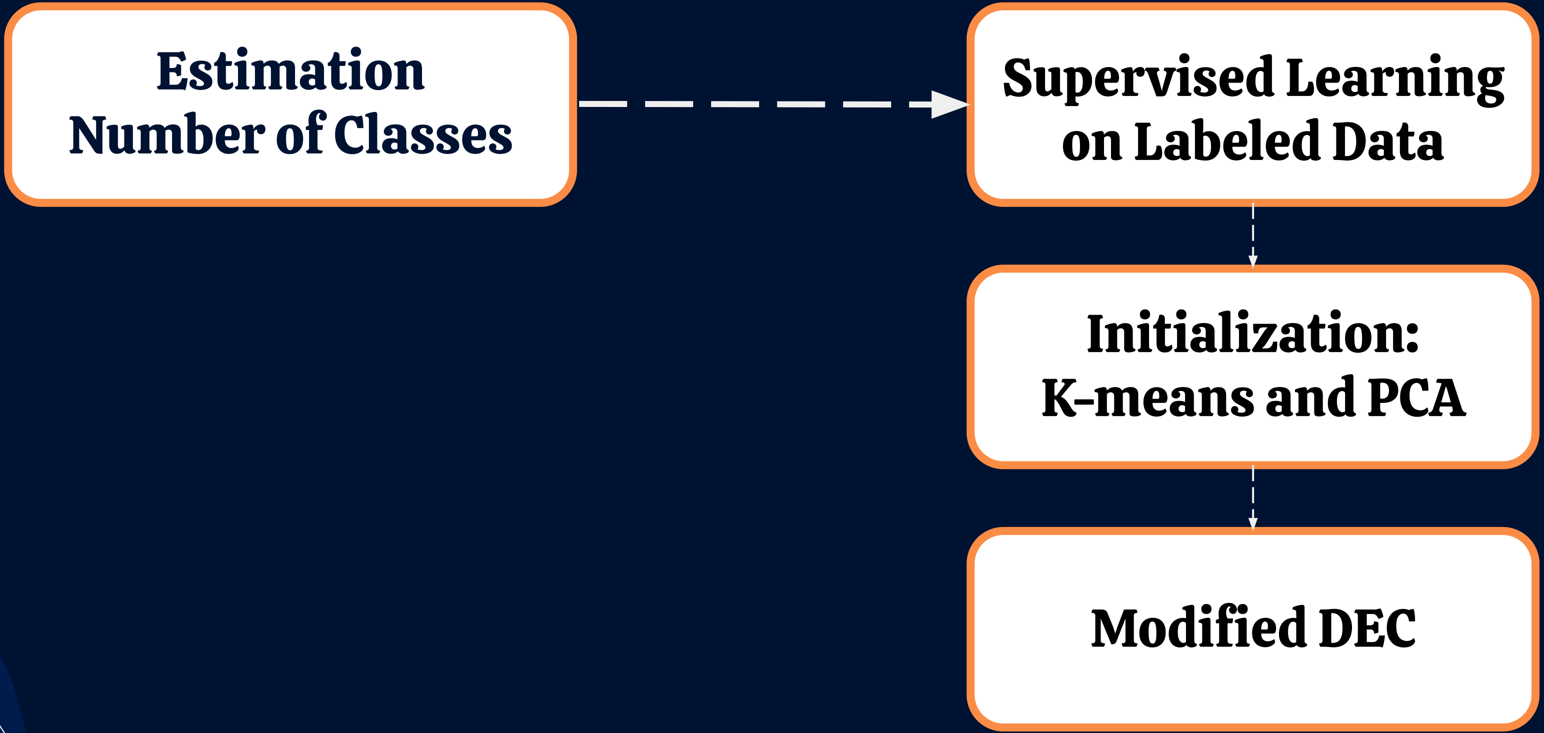
1. First method to explicitly address NCD

2. Extend Deep Embedded Clustering (DEC):
   a. representational bottleneck
   b. temporal ensembling
   c. consistency

3. Method to estimate the number of classes in the unlabeled data



DTC - Pipeline



DTC - Estimation of Number of Classes

1. Run k-means with different number of classes C

2. Take the number of classes that maximized the two indices

   average clustering accuracy (ACC)
   cluster validity index (CVI)

3. Use k-means a last time to remove the clusters that are too small

4. Output the remaining number of clusters

DTC - Supervised Learning

Goal

Transfer knowledge from known categories

Pre-train the image representation fθ:

- on the labeled dataset
- with a supervised learning approach
- using cross-entropy loss

DTC - Initialization

DEC requires an initial setting for the cluster centers U

K-means

Run k-means algorithm on the set of features extracted from the unlabeled data

PCA

This step performs better by introducing dimensionality reduction to the feature representation

DTC - DEC

Deep Clustering Algorithm that does both: Clustering and Representation Learning

Iterative process:

1. Match the model to a suitably shaped target distribution q
   a. Minimize the KL divergence between joint distributions
   b. The representation $f\theta$ is optimized using SGD
2. Compute new target distribution q:
   a. Sharpen the current distribution p to find the target distribution q

DTC – DEC

- Let p(k|i) be the probability of assigning data point i ∈ {1, . . . , N} to cluster k ∈ {1, . . . , K}
- DEC uses the following parameterization of this conditional distribution by assuming a *Student's t distribution:*

$$p(k|i) \propto \left(1 + \frac{\|z_i - \mu_k\|^2}{\alpha}\right)^{-\frac{\alpha+1}{2}}$$

DTC - DEC

- The model is matched to a suitably-shaped distribution q
- This is done by minimizing the *KL divergence* between distributions:

$$E(q) = KL(q||p) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} q(k|i)\log\frac{q(k|i)}{p(k|i)}.$$

- The target distribution q is a progressively sharper version of the current distribution p:

$$q(k|i) \propto \frac{p(k|i)^2}{\sum_{i=1}^{N} p(k|i)}.$$

DTC - Temporal ensembling and consistency

Improvements of DEC algorithm

Temporal Ensembling

The clustering models p computed at different epochs are aggregated by maintaining an exponential moving average (EMA) of the previous distributions.

Consistency

Incorporated by enforcing the predictions of a data sample p to be close to its transformed counterpart p’ or its temporal ensemble prediction p’.

03

AutoNovel

K. Han, S. Rebuffi, S. Ehrhardt, A. Vedaldi, A. Zisserman, ICLR 2020

Automatically Discovering and Learning New Visual Categories with Ranking Statistics

Three Stage Pipeline

Pre-training with self-supervision

↓

Supervised Fine-tuning

↓

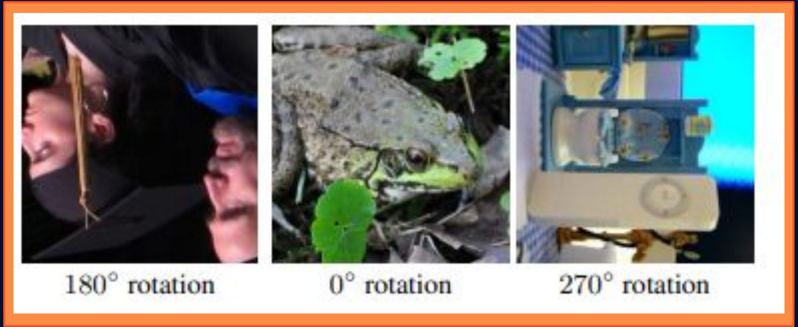
Joint Objective function minimization

Pre-training phase

Labeled data initialization introduces representational bias.

Replace supervised pre-training with self-supervision.

RotNet



$\{0°, 90°, 180°, 270°\}$

Training on Labeled Data

- Fine-tune the pre-trained model Φ on the labeled data to learn a classifier for known samples
- Compute gradients only for the last macro-block of Φ and classification head to avoid overfitting
- Uses standard Cross-Entropy Loss

Transfer Learning with Rank Statistics

Goal

Compute pseudo-labels to use as ground truth for the novel classes

- Define a similarity notion between pair of samples x_i , x_j
- Rank the feature activations of an image
- If the first k rankings of a pair i , j are the same, images are deemed similar.

$$s_{ij} = \mathbb{1}\left\{\mathrm{top}_k(\Phi(x_i^u)) = \mathrm{top}_k(\Phi(x_j^u))\right\}$$

Transfer Learning with Rank Statistics

- Compute pairwise-similarity for all unlabelled samples

- Apply a second head for the image representation of the unlabelled data:
  number of output neurons = number of novel classes

- To calculate the loss, they compute BCE on the s_{ij} pseudo-labels multiplied by the softmax activations of the unlabelled head

Joint Training

- **Model Architecture: shared backbone Φ and two heads**

- **Fine-tune last block of Φ with the two losses combining the two losses**

$$\boxed{L = L_{CE} + L_{BCE} + \omega(t) L_{MSE}}$$

- **Consistency component to limit pseudo-labels moving target problem**

- **Incremental Learning improves accuracy, single classifier**

04

Experiments

AutoNovel codepipeline

1

Self-Supervised Learning

Learning features less biased to labeled classes.

2

Supervised Learning

Learn a classifier specific to the labeled data.

3

AutoNovel

Pairing of unlabeled images using rank statistics.

4

Final Model

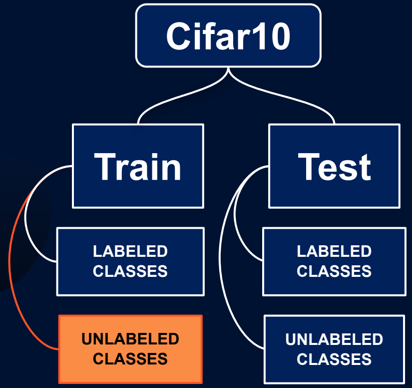
Final model after optimization.

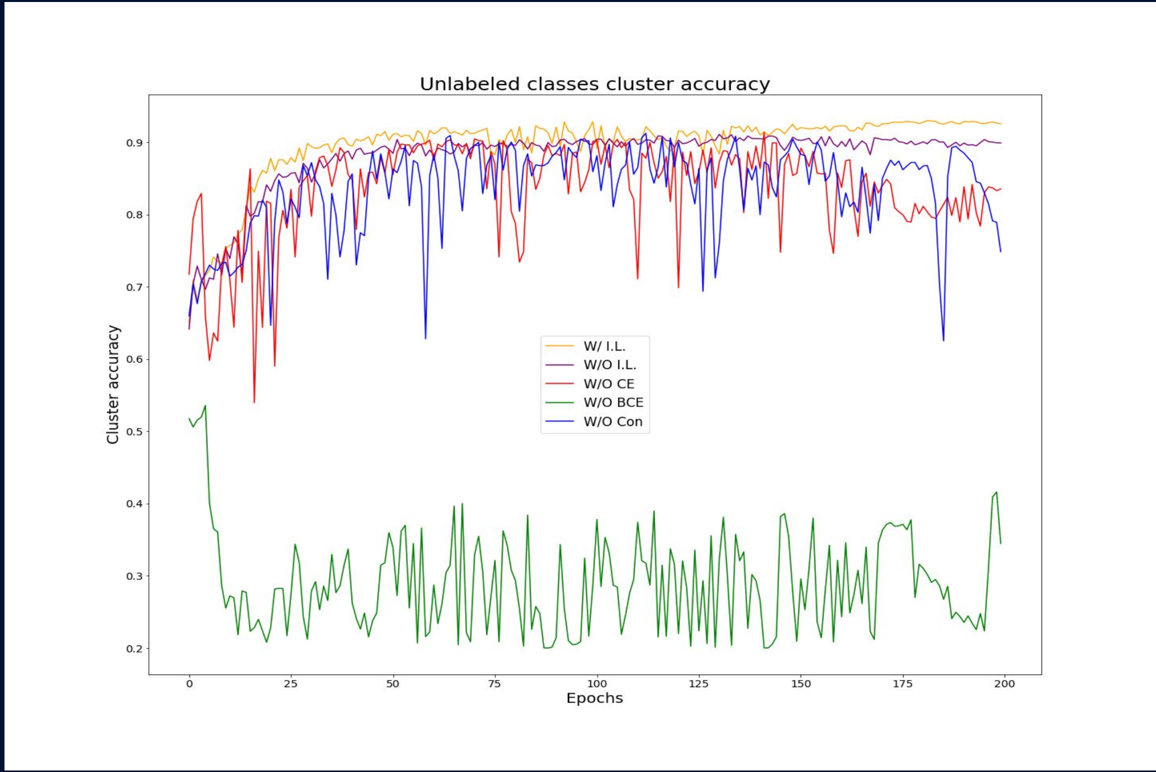
Experiment 1

Removing different loss terms

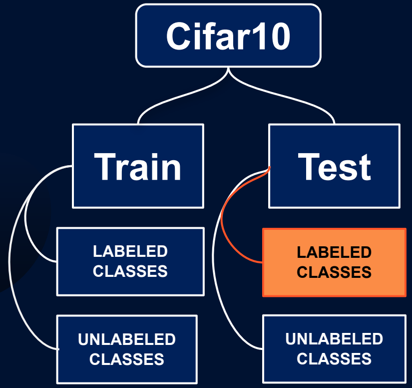
$$L_{total} = L_{BCE} + L_{MSE} + L_{CE} + L_{IL}$$

Cluster accuracy

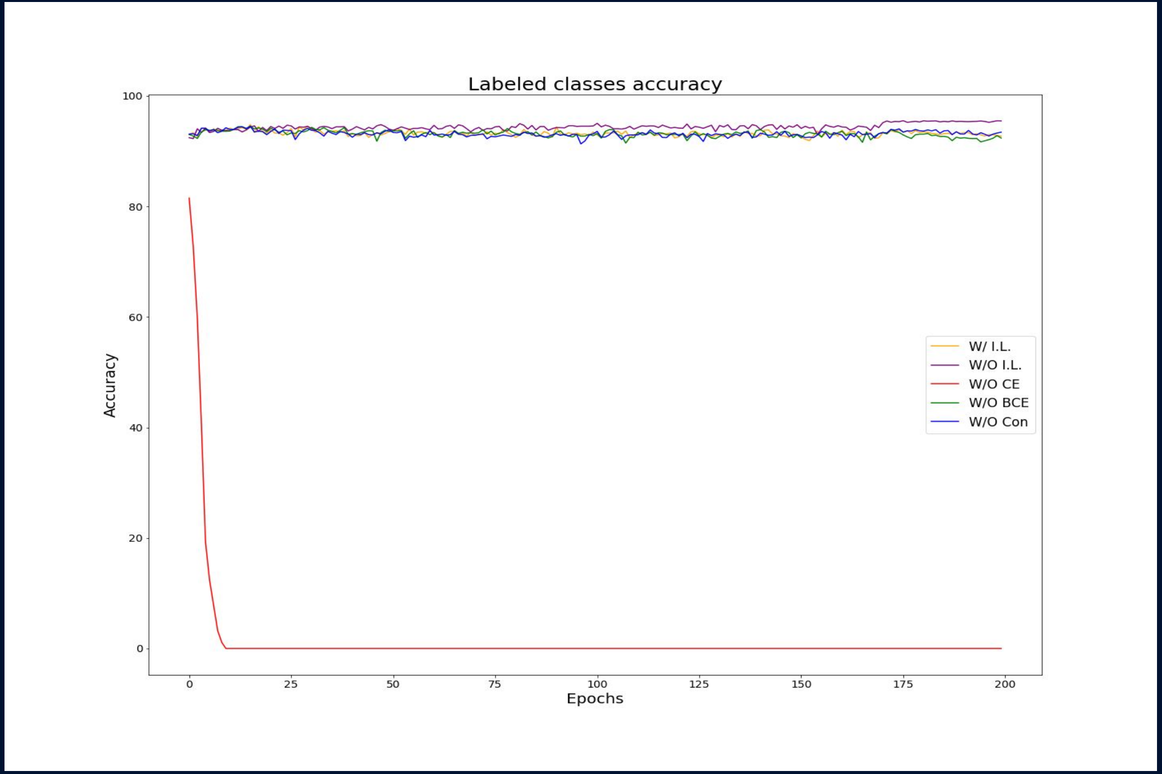




Accuracy

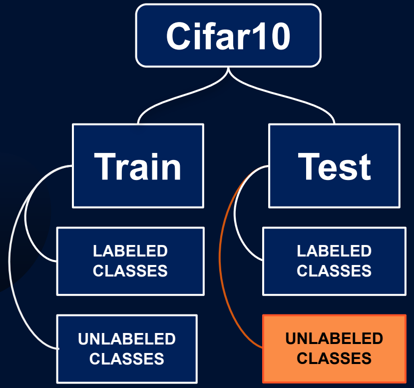


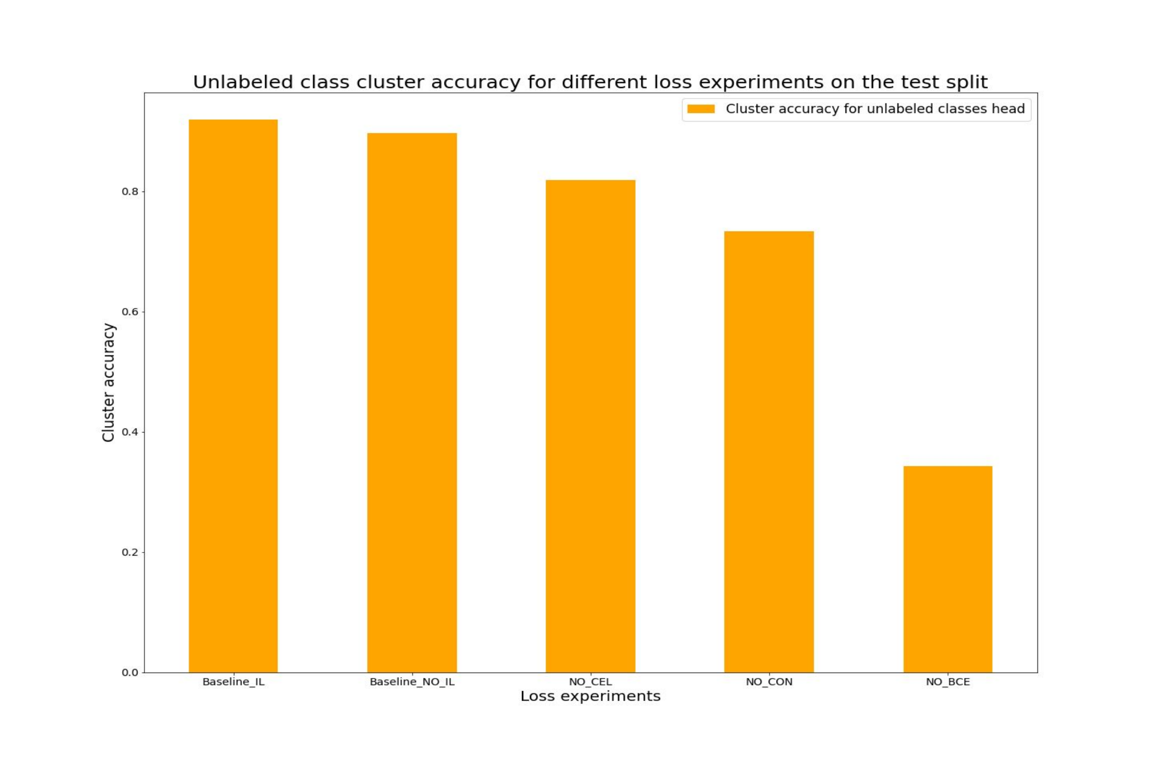
Cifar10
Train
Test
LABELED CLASSES
UNLABELED CLASSES
LABELED CLASSES
UNLABELED CLASSES



Labeled classes accuracy
Accuracy
Epochs
100
80
60
40
20
0
0
25
50
75
100
125
150
175
200
W/ I.L.
W/O I.L.
W/O CE
W/O BCE
W/O Con

Cluster accuracy

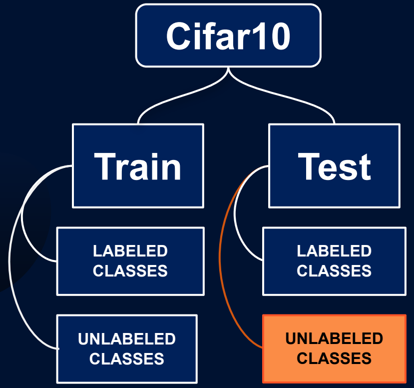


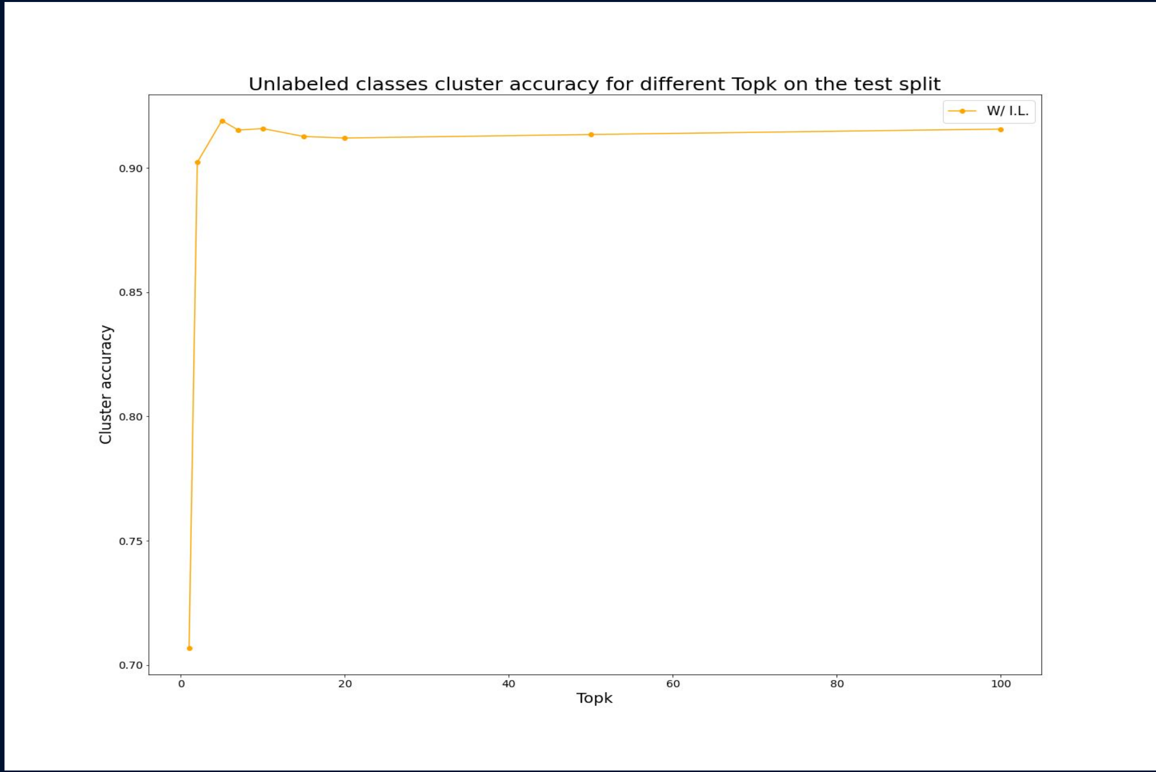


Experiment 2

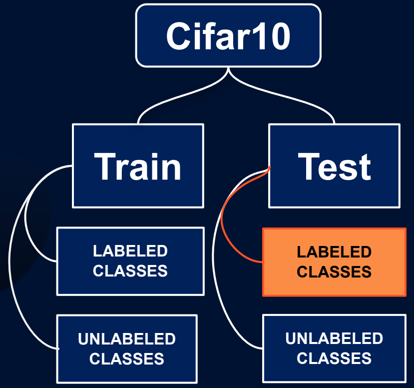
Trying different Topk parameters

Cluster accuracy

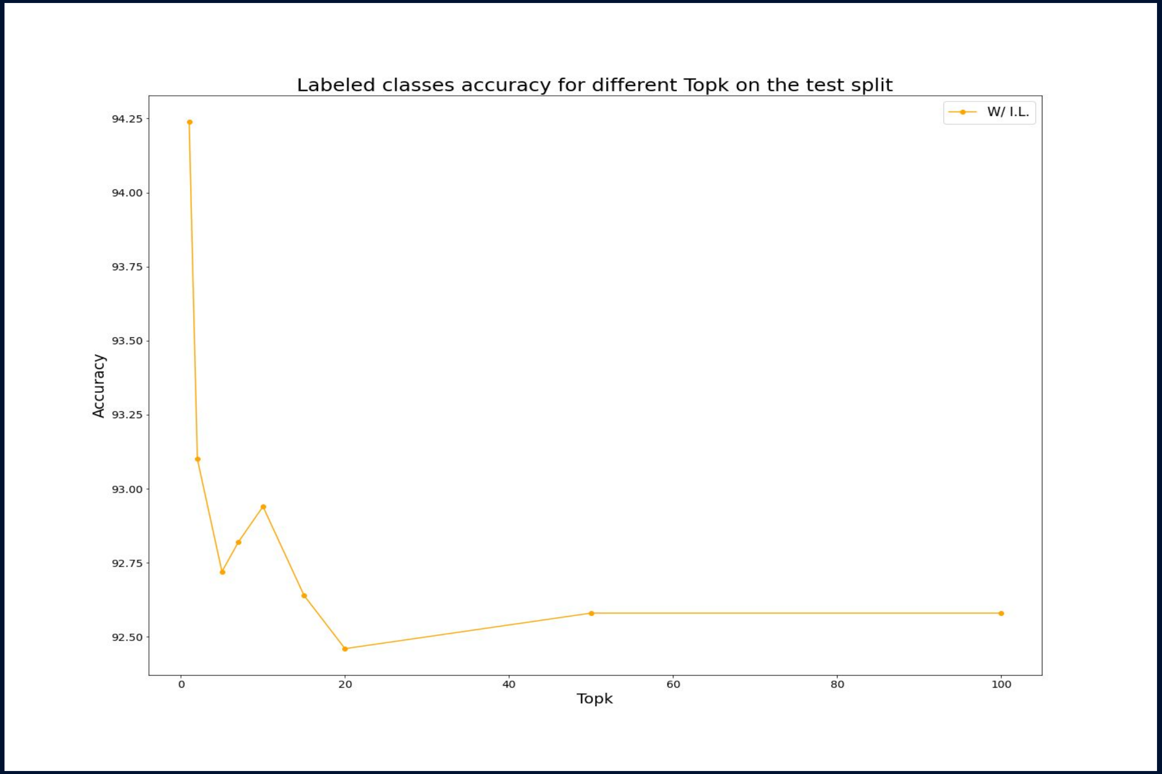




Accuracy



Cifar10
Train
Test
LABELED CLASSES
UNLABELED CLASSES
LABELED CLASSES
UNLABELED CLASSES

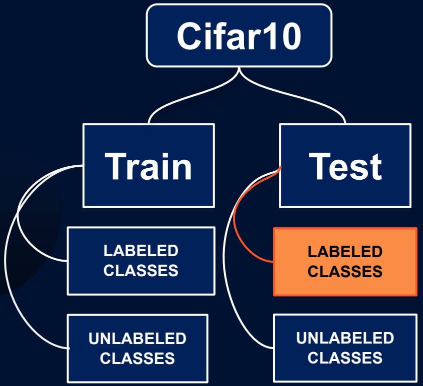


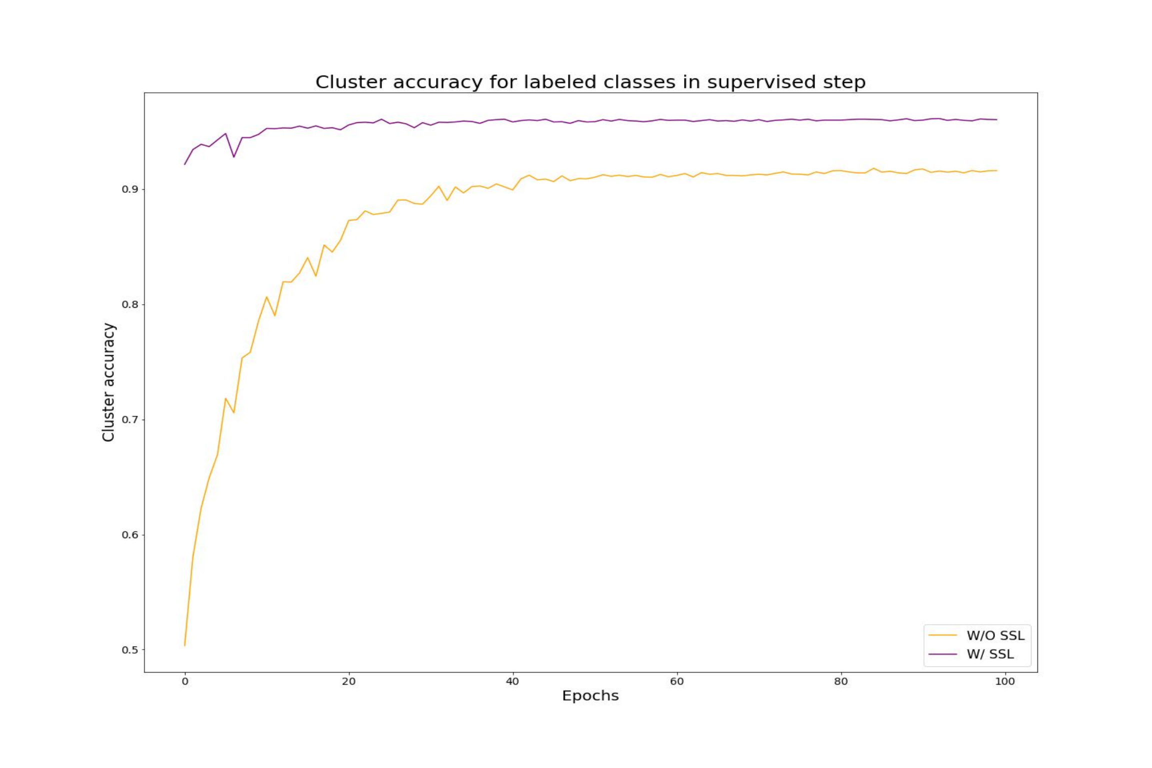
Labeled classes accuracy for different Topk on the test split
W/ I.L.
Accuracy
Topk
94.25
94.00
93.75
93.50
93.25
93.00
92.75
92.50
0
20
40
60
80
100

Experiment 3

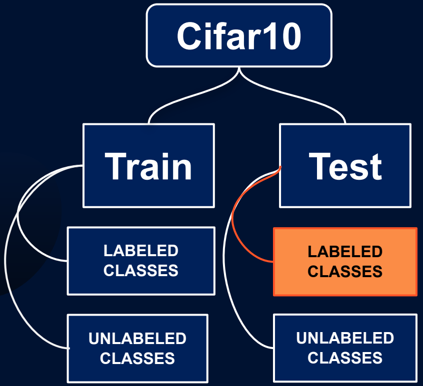
Removing Self-Supervised Learning

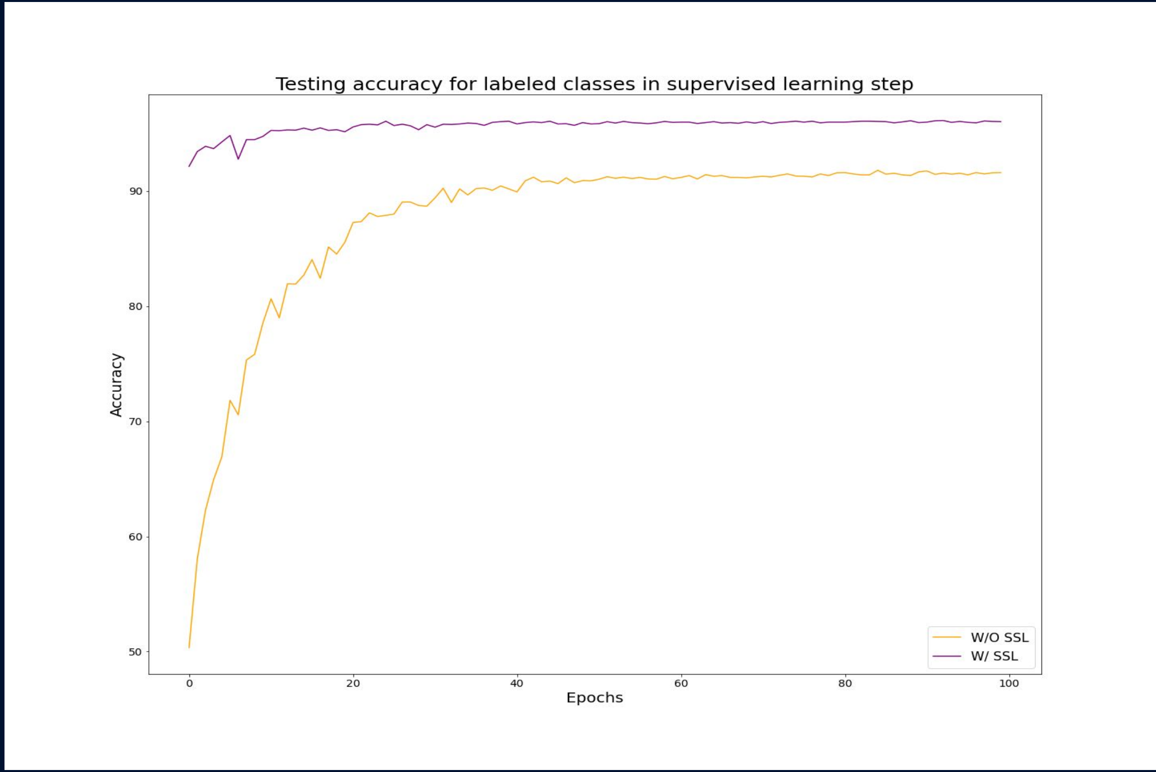
Cluster accuracy in supervised step



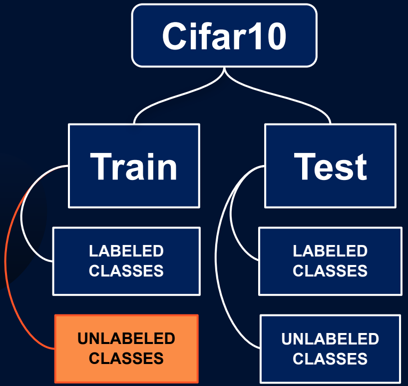


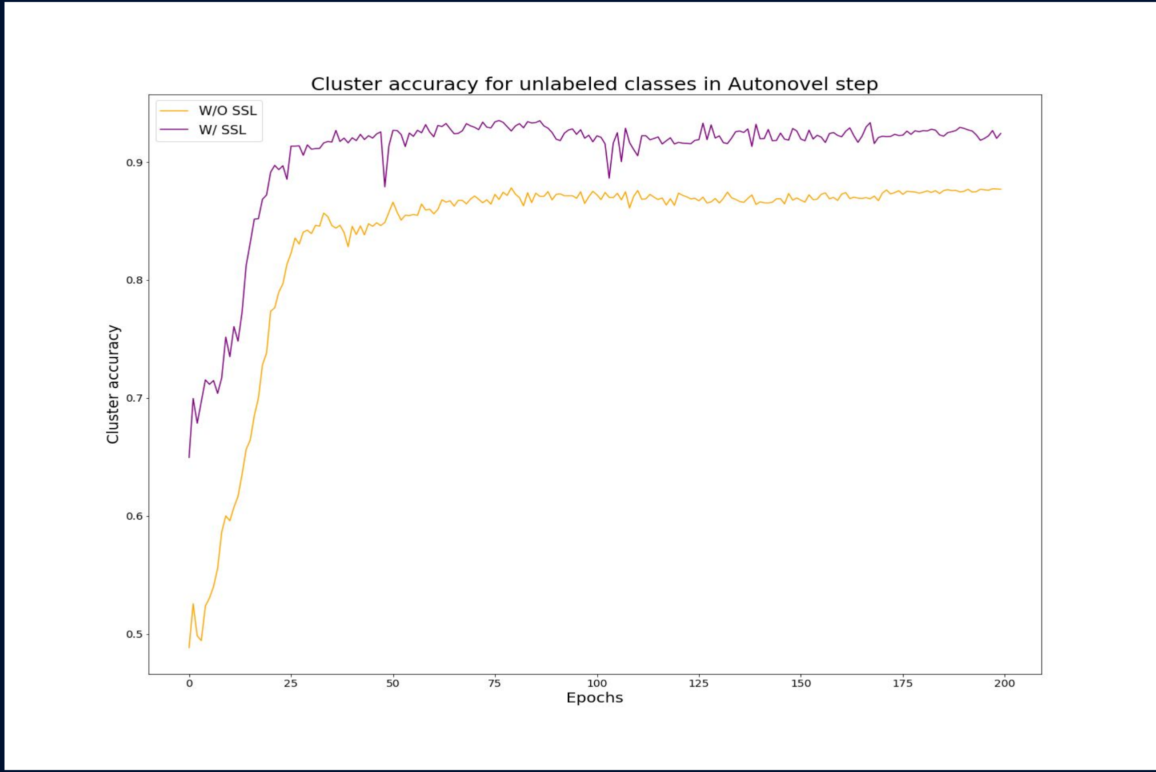
Testing accuracy in supervised step





Cluster accuracy in Autonovel step





05

Future works

Introduce UNO

Fully understand UNO theoretical formulation and its code

Run more experiments

Extend the experiments on AutoNovel and start to perform some experiments on UNO

Thank you for the attention!