

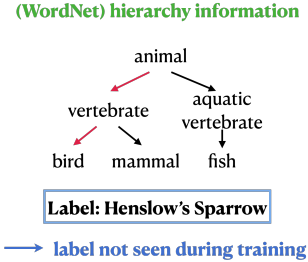
APPENDIX FOR LEARNING WEAKLY-SUPERVISED CONTRASTIVE REPRESENTATIONS

Anonymous authors

Paper under double-blind review

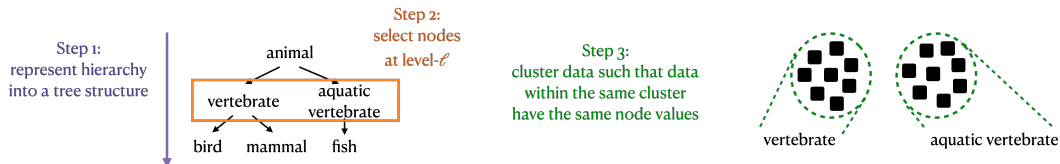
A DATA’S HIERARCHY INFORMATION AS AUXILIARY INFORMATION

In the main text, we select the discrete attributes as the auxiliary information of data, then presenting data cluster construction according to the discrete attributes. We combine the constructed clusters and the presented CI-InfoNCE objective together for learning weakly-supervised representations. In this section, we study an alternative type of the auxiliary information - data labels’ hierarchy information, more specifically, the WordNet hierarchy (Miller, 1995), illustrated in the right figure. In the example, we present the WordNet hierarchy of the label “Henslow’s Sparrow”, where only the WordNet hierarchy would be seen during training but not the label.



A.1 CLUSTER CONSTRUCTION FOR WORDNET HIERARCHY

How do we construct the data clusters according to the WordNet hierarchy? In the above example, “vertebrate” and “bird” can be seen as the coarse labels of data. We then construct the clusters such that data within each cluster will have the same coarse label. Now, we explain how we determine which coarse labels for the data. First, we represent the WordNet hierarchy into a tree structure (each children node has only one parent node). Then, we choose the coarse labels to be the nodes in the level l in the WordNet tree hierarchy (the root node is level 1). l is a hyper-parameter. We illustrate the process in the below figure.



A.2 EXPERIMENTS: DATA-HIERARCHY-DETERMINED CLUSTERS + CL-INFO NCE

The experimental setup and the comparing baselines are similar to Section 4.3 in the main text, but now we consider the WordNet (Miller, 1995) hierarchy as the auxiliary information. As discussed in prior subsection, we construct the clusters Z such that the data within a cluster have the same parent node in the level l in the data’s WordNet tree hierarchy. l is the hyper-parameter¹.

Results. Figure 1 presents our results. First, we look at the leftmost plot, and we have several similar observations when having the data attributes as the auxiliary information. One of them is that our approach consistently outperforms the auxiliary-information-determined clusters + cross-entropy loss. Another of them is that the weakly supervised representations better close the gap with the supervised representations. Second, as discussed in prior subsection, the WordNet data hierarchy clusters can be regarded as the coarse labels of the data. Hence, when increasing the hierarchy level l , we can observe the performance improvement (see the leftmost plot) and the increasing mutual information $I(Z; T)$ (see the middle plot) between the clusters Z and the labels T . Note that $H(Z|T)$ remains zero (see the rightmost plot) since the coarse labels (the intermediate nodes) can be

¹Note that we do not compare with the CMC method for fair comparisons with other method. The reason is that the CMC method will leverage the entire tree hierarchy, instead of a certain level in the tree hierarchy.

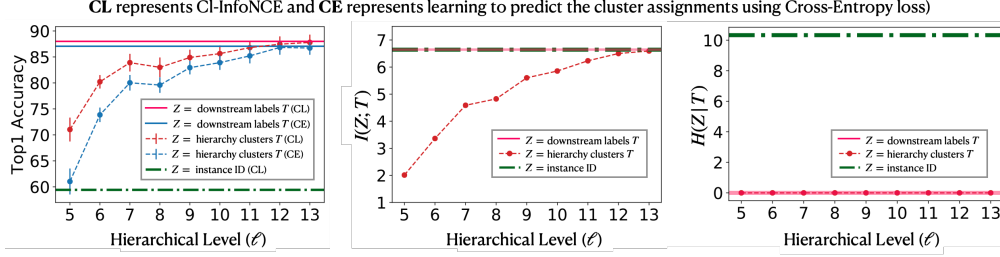


Figure 1: Experimental results on ImageNet-100 for CI-InfoNCE under supervised (clusters Z = downstream labels T), weakly supervised (Z = hierarchy clusters) and conventional self-supervised (Z = instance ID) setting. We also consider the baseline - learning to predict the clustering assignment using the cross-entropy loss. Note that we construct the clusters such that the data within a cluster have the same parent node in the level ℓ in the data’s WordNet tree hierarchy. Under this construction, the root node is of the level 1, and the downstream labels are of the level 14. $I(Z; T)$ is the mutual information, and $H(Z|T)$ is the conditional entropy.

determined by the downstream labels (the leaf nodes) under the tree hierarchy structure. Third, we discuss the conventional self-supervised setting with the special case when Z = instanced ID. Z as the instance ID has the highest $I(Z; T)$ (see the middle plot) but also the highest $H(Z|T)$ (see the rightmost plot). And we observe that the conventional self-supervised representations perform the worse (see the leftmost plot). We conclude that, when using clustering-based representation learning approaches, we shall not rely purely on the mutual information between the data clusters and the downstream labels to determine the goodness of the learned representations. We shall also take the redundant information in the clusters into account.

B THEORETICAL ANALYSIS

In this section, we provide theoretical analysis on the presented CI-InfoNCE objective. We recall the definition of CI-InfoNCE and our presented theorem:

Definition B.1 (Clustering-based InfoNCE (CI-InfoNCE), restating Definition 3.1 in the main text).

$$\text{Cl - InfoNCE} := \sup_f \mathbb{E}_{(x_i, y_i) \sim \mathbb{E}_{z \sim P_Z} [P_{X|Z} P_{Y|Z}]} \left[\frac{1}{n} \sum_{i=1}^n \log \frac{e^{f(x_i, y_i)}}{\frac{1}{n} \sum_{j=1}^n e^{f(x_i, y_j)}} \right],$$

Theorem B.2 (informal, CI-InfoNCE maximization learns to include the clustering information, restating Theorem 3.2 in the main text).

$$\text{Cl - InfoNCE} \leq D_{\text{KL}} \left(\mathbb{E}_{P_Z} [P_{X|Z} P_{Y|Z}] \parallel P_X P_Y \right) \leq H(Z)$$

and the equality holds only when $H(Z|X) = H(Z|Y) = 0$.

Our goal is to prove Theorem B.2. For a better presentation flow, we split the proof into three parts:

- Proving $\text{Cl - InfoNCE} \leq D_{\text{KL}} \left(\mathbb{E}_{P_Z} [P_{X|Z} P_{Y|Z}] \parallel P_X P_Y \right)$ in Section B.1
- Proving $D_{\text{KL}} \left(\mathbb{E}_{P_Z} [P_{X|Z} P_{Y|Z}] \parallel P_X P_Y \right) \leq H(Z)$ in Section B.2
- Proving Cl - InfoNCE maximizes at $H(Z)$ when $H(Z|X) = H(Z|Y) = 0$ in Section B.3

B.1 PART I - PROVING $\text{Cl - InfoNCE} \leq D_{\text{KL}} \left(\mathbb{E}_{P_Z} [P_{X|Z} P_{Y|Z}] \parallel P_X P_Y \right)$

The proof requires the following lemma.

Lemma B.3 (Theorem 1 by Song & Ermon (2020)). *Let \mathcal{X} and \mathcal{Y} be the sample spaces for X and Y , f be any function: $(\mathcal{X} \times \mathcal{Y}) \rightarrow \mathbb{R}$, and \mathcal{P} and \mathcal{Q} be the probability measures on $\mathcal{X} \times \mathcal{Y}$. Then,*

$$\sup_f \mathbb{E}_{(x, y_1) \sim \mathcal{P}, (x, y_{2:n}) \sim \mathcal{Q}^{\otimes(n-1)}} \left[\log \frac{e^{f(x, y_1)}}{\frac{1}{n} \sum_{j=1}^n e^{f(x, y_j)}} \right] \leq D_{\text{KL}}(\mathcal{P} \parallel \mathcal{Q}).$$

Now, we are ready to prove the following lemma:

Lemma B.4 (Proof Part I). $\text{Cl} - \text{InfoNCE} := \sup_f \mathbb{E}_{(x_i, y_i) \sim \mathbb{E}_{z \sim P_Z} [P_{X|z} P_{Y|z}]}^{\otimes n} \left[\frac{1}{n} \sum_{i=1}^n \log \frac{e^{f(x_i, y_i)}}{\frac{1}{n} \sum_{j=1}^n e^{f(x_i, y_j)}} \right] \leq D_{\text{KL}} \left(\mathbb{E}_{P_Z} [P_{X|Z} P_{Y|Z}] \parallel P_X P_Y \right).$

Proof. By defining $\mathcal{P} = \mathbb{E}_{P_Z} [P_{X|Z} P_{Y|Z}]$ and $\mathcal{Q} = P_X P_Y$, we have

$$\mathbb{E}_{(x, y_1) \sim \mathcal{P}, (x, y_{2:n}) \sim \mathcal{Q}^{\otimes (n-1)}} \left[\log \frac{e^{f(x, y_1)}}{\frac{1}{n} \sum_{j=1}^n e^{f(x, y_j)}} \right] = \mathbb{E}_{(x_i, y_i) \sim \mathbb{E}_{z \sim P_Z} [P_{X|z} P_{Y|z}]}^{\otimes n} \left[\frac{1}{n} \sum_{i=1}^n \log \frac{e^{f(x_i, y_i)}}{\frac{1}{n} \sum_{j=1}^n e^{f(x_i, y_j)}} \right].$$

Plug in this result into Lemma B.3 and we conclude the proof. \square

B.2 PART II - PROVING $D_{\text{KL}} \left(\mathbb{E}_{P_Z} [P_{X|Z} P_{Y|Z}] \parallel P_X P_Y \right) \leq H(Z)$

The proof requires the following lemma:

Lemma B.5. $D_{\text{KL}} \left(\mathbb{E}_{P_Z} [P_{X|Z} P_{Y|Z}] \parallel P_X P_Y \right) \leq \min \{ \text{MI}(Z; X), \text{MI}(Z; Y) \}.$

Proof.

$$\begin{aligned} & \text{MI}(Z; X) - D_{\text{KL}} \left(\mathbb{E}_{P_Z} [P_{X|Z} P_{Y|Z}] \parallel P_X P_Y \right) \\ &= \int_z p(z) \int_x p(x|z) \log \frac{p(x|z)}{p(x)} dx dz - \int_z p(z) \int_x p(x|z) \int_y p(y|z) \log \frac{\int_{z'} p(z'|y) p(x|z') dz'}{p(x)p(y)} dx dy dz \\ &= \int_z p(z) \int_x p(x|z) \log \frac{p(x|z)}{p(x)} dx dz - \int_z p(z) \int_x p(x|z) \int_y p(y|z) \log \frac{\int_{z'} p(z'|y) p(x|z') dz'}{p(x)} dx dy dz \\ &= \int_z p(z) \int_x p(x|z) \int_y p(y|z) \log \frac{p(x|z)}{\int_{z'} p(z'|y) p(x|z') dz'} dx dy dz \\ &= - \int_z p(z) \int_x p(x|z) \int_y p(y|z) \log \frac{\int_{z'} p(z'|y) p(x|z') dz'}{p(x|z)} dx dy dz \\ &\geq - \int_z p(z) \int_x p(x|z) \int_y p(y|z) \left(\frac{\int_{z'} p(z'|y) p(x|z') dz'}{p(x|z)} - 1 \right) dx dy dz \quad (\because \log t \leq t - 1) \\ &= 0. \end{aligned}$$

Hence, $\text{MI}(Z; X) \geq D_{\text{KL}} \left(\mathbb{E}_{P_Z} [P_{X|Z} P_{Y|Z}] \parallel P_X P_Y \right).$ Likewise, $\text{MI}(Z; Y) \geq D_{\text{KL}} \left(\mathbb{E}_{P_Z} [P_{X|Z} P_{Y|Z}] \parallel P_X P_Y \right).$ We complete the proof by combining the two results. \square

Now, we are ready to prove the following lemma:

Lemma B.6 (Proof Part II). $D_{\text{KL}} \left(\mathbb{E}_{P_Z} [P_{X|Z} P_{Y|Z}] \parallel P_X P_Y \right) \leq H(Z).$

Proof. Combining Lemma B.5 and the fact that $\min \{ \text{MI}(Z; X), \text{MI}(Z; Y) \} \leq H(Z)$, we complete the proof. Note that we consider Z as the clustering assignment, which is discrete but not continuous. And the inequality holds for the discrete Z , but may not hold for the continuous Z . \square

B.3 PART III - PROVING C1 – InfoNCE maximizes at $H(Z)$ when $H(Z|X) = H(Z|Y) = 0$

We directly provide the following lemma:

Lemma B.7 (Proof Part III). C1 – InfoNCE max. at $H(Z)$ when $H(Z|X) = H(Z|Y) = 0$.

Proof. When $H(Z|Y) = 0$, $p(Z|Y = y)$ is Dirac. The objective

$$\begin{aligned}
& D_{\text{KL}} \left(\mathbb{E}_{P_Z} [P_{X|Z} P_{Y|Z}] \parallel P_X P_Y \right) \\
&= \int_z p(z) \int_x p(x|z) \int_y p(y|z) \log \frac{\int_{z'} p(z') p(x|z') p(y|z') dz'}{p(x)p(y)} dx dy dz \\
&= \int_z p(z) \int_x p(x|z) \int_y p(y|z) \log \frac{\int_{z'} p(z'|y) p(x|z') dz'}{p(x)} dx dy dz \\
&= \int_z p(z) \int_x p(x|z) \int_y p(y|z) \log \frac{\int_{z'} p(z') p(x|z') p(y|z') dz'}{p(x)p(y)} dx dy dz \\
&= \int_z p(z) \int_x p(x|z) \int_y p(y|z) \log \frac{p(x|z)}{p(x)} dx dy dz = \text{MI}(Z; X).
\end{aligned}$$

The second-last equality comes with the fact that: when $p(Z|Y = y)$ is Dirac, $p(z'|y) = 1 \forall z' = z$ and $p(z'|y) = 0 \forall z' \neq z$. Combining with the fact that $\text{MI}(Z; X) = H(Z)$ when $H(Z|X) = 0$, we know $D_{\text{KL}} \left(\mathbb{E}_{P_Z} [P_{X|Z} P_{Y|Z}] \parallel P_X P_Y \right) = H(Z)$ when $H(Z|X) = H(Z|Y) = 0$.

Furthermore, by Lemma B.4 and Lemma B.6, we complete the proof. \square

B.4 BRINGING EVERYTHING TOGETHER

We bring Lemmas B.4, B.6, and B.7 together and complete the proof of Theorem B.2.

C ALGORITHMS

In this section, we provide algorithms for our experiments. We consider two sets of the experiments. The first one is K-means clusters + CI-InfoNCE (see Section 4.4 in the main text), where the clusters involved in CI-InfoNCE are iteratively obtained via K-means clustering on top of data representations. The second one is auxiliary-information-determined clusters + CI-InfoNCE (see Section 4.3 in the main text and Section A.2), where the clusters involved in CI-InfoNCE are pre-determined accordingly to data attributes (see Section 4.3 in the main text) or data hierarchy information (see Section A.2).

K-means clusters + CI-InfoNCE We present here the algorithm for K-means clusters + CI-InfoNCE. At each iteration in our algorithm, we perform K-means Clustering algorithm on top

of data representations for obtaining cluster assignments. The cluster assignment will then be used in our CI-InfoNCE objective.

Algorithm 1: K-means Clusters + CI-InfoNCE

Result: Pretrained Encoder $f_\theta(\cdot)$
 $f_\theta(\cdot) \leftarrow$ Base Encoder Network;
 $\text{Aug}(\cdot) \leftarrow$ Obtaining Two Variants of Augmented Data via Augmentation Functions;
 $\text{Embedding} \leftarrow$ Gathering data representations by passing data through $f_\theta(\cdot)$;
 $\text{Clusters} \leftarrow \mathbf{K\text{-}means\text{-}clustering}(\text{Embedding})$;
for epoch in $1, 2, \dots, N$ **do**
 for batch in $1, 2, \dots, M$ **do**
 $\text{data1}, \text{data2} \leftarrow \text{Aug}(\text{data_batch})$;
 $\text{feature1}, \text{feature2} \leftarrow f_\theta(\text{data1}), f_\theta(\text{data2})$;
 $L_{\text{CI-infoNCE}} \leftarrow \text{CI-InfoNCE}(\text{feature1}, \text{feature2}, \text{Clusters})$;
 $f_\theta \leftarrow f_\theta - lr * \frac{\partial}{\partial \theta} L_{\text{CI-infoNCE}}$;
 end
 $\text{Embedding} \leftarrow$ gather embeddings for all data through $f_\theta(\cdot)$;
 $\text{Clusters} \leftarrow \mathbf{K\text{-}means\text{-}clustering}(\text{Embedding})$;
end

Auxiliary information determined clusters + CI-InfoNCE We present the algorithm to combine auxiliary-information-determined clusters with CI-InfoNCE. We select data attributes or data hierarchy information as the auxiliary information, and we present their clustering determining steps in Section 3.1 in the main text for discrete attributes and Section A.1 for data hierarchy information.

Algorithm 2: Pre-Determined Clusters + CI-InfoNCE

Result: Pretrained Encoder $f_\theta(\cdot)$
 $f_\theta(\cdot) \leftarrow$ Base Encoder Network;
 $\text{Aug}(\cdot) \leftarrow$ Obtaining Two Variants of Augmented Data via Augmentation Functions;
 $\text{Clusters} \leftarrow$ Pre-determining Data Clusters from **Auxiliary Information**;
for epoch in $1, 2, \dots, N$ **do**
 for batch in $1, 2, \dots, M$ **do**
 $\text{data1}, \text{data2} \leftarrow \text{Aug}(\text{data_batch})$;
 $\text{feature1}, \text{feature2} \leftarrow f_\theta(\text{data1}), f_\theta(\text{data2})$;
 $L_{\text{CI-infoNCE}} \leftarrow \text{CI-InfoNCE}(\text{feature1}, \text{feature2}, \text{Clusters})$;
 $f_\theta \leftarrow f_\theta - lr * \frac{\partial}{\partial \theta} L_{\text{CI-infoNCE}}$;
 end
end

D EXPERIMENTAL DETAILS

The following content describes our experiments settings in details. For reference, our code is available at <https://anonymous.4open.science/r/CI-InfoNCE-02AB/README.md>.

D.1 UT-ZAPPOS50K

The following section describes the experiments we performed on UT-Zappos50K dataset in Section 4 in the main text.

Accessibility The dataset is attributed to (Yu & Grauman, 2014) and available at the link: <http://vision.cs.utexas.edu/projects/finegrained/utzap50k>. The dataset is for non-commercial use only.

Data Processing The dataset contains images of shoe from Zappos.com. We rescale the images to 32×32 . The official dataset has 4 large categories following 21 sub-categories. We utilize the

21 subcategories for all our classification tasks. The dataset comes with 7 attributes as auxiliary information. We binarize the 7 discrete attributes into 126 binary attributes. We rank the binarized attributes based on their entropy and use the top- k binary attributes to form clusters. Note that different k result in different data clusters (see Figure 4 (a) in the main text).

Training and Test Split: We randomly split train-validation images by 7 : 3 ratio, resulting in 35,017 train data and 15,008 validation dataset.

Network Design We use ResNet-50 architecture to serve as a backbone for encoder. To compensate the 32x32 image size, we change the first 7x7 2D convolution to 3x3 2D convolution and remove the first max pooling layer in the normal ResNet-50 (See code for detail). This allows finer grain of information processing. After using the modified ResNet-50 as encoder, we include a 2048-2048-128 Multi-Layer Perceptron (MLP) as the projection head (i.e., $g(\cdot)$ in $f(\cdot, \cdot)$ equation (1) in the main text) for CI-InfoNCE. During evaluation, we discard the projection head and train a linear layer on top of the encoder’s output. For both K-means clusters + CI-InfoNCE and auxiliary-information-determined clusters + CI-InfoNCE, we adopt the same network architecture, including the same encoder, the same MLP projection head and the same linear evaluation protocol. In the K-means + CI-InfoNCE settings, the number of the K-means clusters is 1,000. Kmeans clustering is performed every epoch during training. We find performing Kmeans for every epoch benefits the performance. For fair comparison, we use the same network architecture and cluster number for PCL.

Optimization We choose SGD with momentum of 0.95 for optimizer with a weight decay of 0.0001 to prevent network over-fitting. To allow stable training, we employ a linear warm-up and cosine decay scheduler for learning rate. For experiments shown in Figure 4 (a) in the main text, the learning rate is set to be 0.17 and the temperature is chosen to be 0.07 in CI-InfoNCE. And for experiments shown in Figure 5 in the main text, learning rate is set to be 0.1 and the temperature is chosen to be 0.1 in CI-InfoNCE.

Computational Resource We conduct experiments on machines with 4 NVIDIA Tesla P100. It takes about 16 hours to run 1000 epochs of training with batch size 128 for both auxiliary information aided and unsupervised CI-InfoNCE.

D.2 WIDER ATTRIBUTES

The following section describes the experiments we performed on Wider Attributes dataset in Section 4 in the main text.

Accessibility The dataset is credited to (Li et al., 2016) and can be downloaded from the link: <http://mmlab.ie.cuhk.edu.hk/projects/WIDERAttribute.html>. The dataset is for public and non-commercial usage.

Data Processing The dataset contains 13,789 images with multiple semantic bounding boxes attached to each image. Each bounding is annotated with 14 binary attributes, and different bounding boxes in an image may have different attributes. Here, we perform the OR operation among the attributes in the bounding boxes in an image. Hence, each image is linked to 14 binary attributes. We rank the 14 attributes by their entropy and use the top- k of them when performing experiments in Figure 4 (b) in the main text. We consider a classification task consisting of 30 scene categories.

Training and Test Split: The dataset comes with its training, validation, and test split. Due to a small number of data, we combine the original training and validation set as our training set and use the original test set as our validation set. The resulting training set contains 6,871 images and the validation set contains 6,918 images.

Computational Resource To speed up computation, on Wider Attribute dataset we use a batch size of 40, resulting in 16-hour computation in a single NVIDIA Tesla P100 GPU for 1,000 epochs training.

Network Design and Optimization We use ResNet-50 architecture as an encoder for Wider Attributed dataset. We choose 2048-2048-128 MLP as the projection head (i.e., $g(\cdot)$ in $f(\cdot, \cdot)$ equation (1) in the main text) for CI-InfoNCE. The MLP projection head is discarded during the linear evaluation protocol. Particularly, during the linear evaluation protocol, the encoder is frozen and a linear layer on top of the encoder is fine-tuned with downstream labels. For Kmeans + CI-InfoNCE and Auxiliary information + CI-InfoNCE, we consider the same architectures for the encoder, the MLP head and the linear evaluation classifier. For K-means + CI-InfoNCE, we consider 1,000 K-means clusters. For fair comparison, the same network architecture and cluster number is used for experiments with PCL.

For Optimization, we use SGD with momentum of 0.95. Additionally, 0.0001 weight decay is adopted in the network to prevent over-fitting. We use a learning rate of 0.1 and temperature of 0.1 in CI-InfoNCE for all experiments. A linear warm-up following a cosine decay is used for the learning rate scheduling, providing a more stable learning process.

D.3 CUB-200-2011

The following section describes the experiments we performed on CUB-200-2011 dataset in Section 4 in the main text.

Accessibility CUB-200-2011 is created by Wah et al. (2011) and is a fine-grained dataset for bird species. It can be downloaded from the link: <http://www.vision.caltech.edu/visipedia/CUB-200-2011.html>. The usage is restricted to non-commercial research and educational purposes.

Data Processing The original dataset contains 200 birds categories over 11,788 images with 312 binary attributes attached to each image. We utilize those attributes and rank them based on their entropy, excluding the last 112 of them (resulting in 200 attributes), because including these 112 attributes will not change the number of the clusters than not including them. In Figure 4 (c), we use the top- k of those attributes to construct clusters with which we perform in CI-InfoNCE. The image is rescaled to 224×224 .

Train Test Split: We follow the original train-validation split, resulting in 5,994 train images and 5,794 validation images.

Computational Resource It takes about 8 hours to train for 1000 epochs with 128 batch size on 4 NVIDIA Tesla P100 GPUs.

Network Design and Optimization We choose ResNet-50 for CUB-200-2011 as the encoder. After extracting features from the encoder, a 2048-2048-128 MLP projection head (i.e., $g(\cdot)$ in $f(\cdot, \cdot)$ equation (1) in the main text) is used for CI-InfoNCE. During the linear evaluation protocol, the MLP projection head is removed and the features extracted from the pre-trained encoder is fed into a linear classifier layer. The linear classifier layer is fine-tuned with the downstream labels. The network architectures remain the same for both K-means clusters + CI-InfoNCE and auxiliary-information-determined clusters + CI-InfoNCE settings. In the K-means clusters + CI-InfoNCE settings, we consider 1,000 K-means clusters. For fair comparison, the same network architecture and cluster number is used for experiments with PCL.

SGD with momentum of 0.95 is used during the optimization. We select a linear warm-up following a cosine decay learning rate scheduler. The peak learning rate is chosen to be 0.1 and the temperature is set to be 0.1 for both K-means + CI-InfoNCE and Auxiliary information + CI-InfoNCE settings.

D.4 IMAGENET-100

The following section describes the experiments we performed on ImageNet-100 dataset in Section 4 in the main text.

Accessibility This dataset is a subset of ImageNet-1K dataset, which comes from the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012-2017 (Russakovsky et al., 2015). ILSVRC is for non-commercial research and educational purposes and we refer to the ImageNet official site for more information: <https://www.image-net.org/download.php>.

Data Processing In the Section 4 in the main text and Section A, we select 100 classes from ImageNet-1K to conduct experiments (the selected categories can be found in https://anonymous.4open.science/r/CI-InfoNCE-02AB/data_processing/imagenet100/selected_100_classes.txt). We also conduct a slight pre-processing (via pruning a small number of edges in the WordNet graph) on the WordNet hierarchy structure to ensure it admits a tree structure. Specifically, each of the selected categories and their ancestors only have one path to the root. We refer the pruning procedure in https://anonymous.4open.science/r/CI-InfoNCE-02AB/data_processing/imagenet100/hierarchy_processing/imagenet_hierarchy.py (line 222 to 251).

We cluster data according to their common ancestor in the pruned tree structure and determine the level l of each cluster by the step needed to traverse from root to that node in the pruned tree. Therefore, the larger the l , the closer the common ancestor is to the real class labels, hence more accurate clusters will be formed. Particularly, the real class labels is at level 14.

Training and Test Split: Please refer to the following file for the training and validation split.

- training: https://anonymous.4open.science/r/CI-InfoNCE-02AB/data_processing/imagenet100/hier/meta_data_train.csv
- validation: https://anonymous.4open.science/r/CI-InfoNCE-02AB/data_processing/imagenet100/hier/meta_data_val.csv

The training split contains 128,783 images and the test split contains 5,000 images. The images are rescaled to size 224×224 .

Computational Resource It takes 48-hour training for 200 epochs with batch size 128 using 4 NVIDIA Tesla P100 machines. All the experiments on ImageNet-100 is trained with the same batch size and number of epochs.

Network Design and Optimization Hyper-parameters We use conventional ResNet-50 as the backbone for the encoder. 2048-2048-128 MLP layer and l_2 normalization layer is used after the encoder during training and discarded in the linear evaluation protocol. We maintain the same architecture for Kmeans + CI-InfoNCE and auxiliary information aided CI-InfoNCE. For Kmeans + CI-InfoNCE, we choose 2500 as the cluster number. For fair comparison, the same network architecture and cluster number is used for experiments with PCL. The Optimizer is SGD with 0.95 momentum. For K-means + CI-InfoNCE used in Figure 5 in the main text, we use the learning rate of 0.03 and the temperature of 0.2. We use the learning rate of 0.1 and temperature of 0.1 for auxiliary information + CI-InfoNCE in Figure 1. A linear warm-up and cosine decay is used for the learning rate scheduling. To stabilize the training and reduce overfitting, we adopt 0.0001 weight decay for the encoder network.

E COMPARISONS WITH SWAPPING CLUSTERING ASSIGNMENTS BETWEEN VIEWS

In this section, we provide additional comparisons between Kmeans + CI-InfoNCE and Swapping Clustering Assignments between Views (SwAV) (Caron et al., 2020). The experiment is performed on ImageNet-100 dataset. SwAV is a recent art for clustering-based self-supervised approach. In particular, SwAV adopts Sinkhorn algorithm (Cuturi, 2013) to determine the data clustering assignments for a batch of data samples, and SwAV also ensures augmented views of samples will have the same clustering assignments. We present the results in Table 1, where we see SwAV has similar performance with the Prototypical Contrastive Learning method (Li et al., 2020) and has worse performance than our method (i.e., K-means + CI-InfoNCE).

Method	Top-1 Accuracy (%)
<i>Non-clustering-based Self-supervised Approaches</i>	
SimCLR (Chen et al., 2020)	58.2±1.7
MoCo (He et al., 2020)	59.4±1.6
<i>Clustering-based Self-supervised Approaches (# of clusters = 2.5K)</i>	
SwAV (Caron et al., 2020)	68.5±1.0
PCL (Li et al., 2020)	68.9±0.7
K-means + Cl-InfoNCE (ours)	77.9±0.7

Table 1: Additional Comparision with SwAV (Caron et al., 2020) showing its similar performance as PCL on ImageNet-100 dataset.

F PRELIMINARY RESULTS ON IMAGENET-1K WITH CL-INFOANCE

We have performed experiments on ImageNet-100 dataset, which is a subset of the ImageNet-1K dataset (Russakovsky et al., 2015). We use the batch size of 1,024 for all the methods and consider 100 training epochs. We present the comparisons among Supervised Contrastive Learning (Khosla et al., 2020), our method (i.e., WordNet-hierarchy-information-determined clusters + Cl-InfoNCE), and SimCLR (Chen et al., 2020). We select the level-12 nodes in the WordNet tree hierarchy structures as our hierarchy-determined clusters for Cl-InfoNCE. We report the results in Table 2. We find that our method (i.e., hierarchy-determined clusters + Cl-InfoNCE) performs in between the supervised representations and conventional self-supervised representations.

Method	Top-1 Accuracy (%)
<i>Supervised Representation Learning ($Z = \text{downstream labels } T$)</i>	
SupCon (Khosla et al., 2020)	76.1±1.7
<i>Weakly Supervised Representation Learning ($Z = \text{level 12 WordNet hierarchy labels}$)</i>	
Hierarchy-Clusters + Cl-InfoNCE (ours)	67.9±1.5
<i>Self-supervised Representation Learning ($Z = \text{instance ID}$)</i>	
SimCLR (Chen et al., 2020)	62.9±1.2

Table 2: Preliminary results for WordNet-hierarchy-determined clusters + Cl-InfoNCE on ImageNet-1K.

G SYNTHETICALLY CONSTRUCTED CLUSTERS IN SECTION 4.2 IN THE MAIN TEXT

In Section 4.2 in the main text, on the UT-Zappos50K dataset, we synthesize clusters Z for various $I(Z;T)$ and $H(Z|T)$ with T being the downstream labels. There are 86 configurations of Z in total. Note that the configuration process has no access to data’s auxiliary information and among the 86 configurations we consider the special cases for the supervised ($Z = T$) and the unsupervised setting ($Z = \text{instance ID}$). In specific, when $Z = T$, $I(Z;T)$ reaches its maximum at $H(T)$ and $H(Z|T)$ reaches its minimum at 0; when $Z = \text{instance ID}$, both $I(Z;T)$ (to be $H(T)$) and $H(Z|T)$ (to be $H(\text{instance ID})$) reaches their maximum. The code for generating these 86 configurations can be found in lines 177-299 in https://anonymous.4open.science/r/Cl-InfoNCE-02AB/data_processing/UT-zappos50K/synthetic/generate.py.

REFERENCES

- 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.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.
- Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.
- 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*, pp. 9729–9738, 2020.
- Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.
- Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020.
- Yining Li, Chen Huang, Chen Change Loy, and Xiaoou Tang. Human attribute recognition by deep hierarchical contexts. In *European Conference on Computer Vision*, 2016.
- George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- Jiaming Song and Stefano Ermon. Multi-label contrastive predictive coding. *arXiv preprint arXiv:2007.09852*, 2020.
- Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
- Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 192–199, 2014.