Use All The Labels: A Hierarchical Multi-Label Contrastive Learning Framework

Shu Zhang Ran Xu Caiming Xiong Chetan Ramaiah
Salesforce Research
{shu.zhang,ran.xu,cxiong,cramaiah}@salesforce.com

Abstract

Current contrastive learning frameworks focus on leveraging a single supervisory signal to learn representations, which limits the efficacy on unseen data and downstream tasks. In this paper, we present a hierarchical multi-label representation learning framework that can leverage all available labels and preserve the hierarchical relationship between classes. We introduce novel hierarchy preserving losses, which jointly apply a hierarchical penalty to the contrastive loss, and enforce the hierarchy constraint. The loss function is data driven and automatically adapts to arbitrary multi-label structures. Experiments on several datasets show that our relationship-preserving embedding performs well on a variety of tasks and outperform the baseline supervised and self-supervised approaches. Code is available at https://github.com/salesforce/hierarchicalContrastiveLearning.

1. Introduction

In the real world, hierarchical multi-labels occur naturally and frequently. Biological classification of organisms is structured in a taxonomic hierarchy. In e-commerce websites, retail spaces and grocery stores, products are organized by several levels of categories. The hierarchical representation is a natural categorization of classes, and serves to efficiently represent the relationship between different classes. However, this relationship is seldom utilized in learning tasks, with traditional supervised approaches preferring to organize their classes in a flat list. In single task learning problems, where a model is learned for one objective only, a flat list of classes is a reasonable approach. However, in representation learning frameworks, where a single embedding function can be used in a variety of downstream tasks, utilizing all of the supervisory signal available is vital. In order to generalize to unknown downstream tasks and unseen data, the embedding function must represent the data concisely and accurately, which includes preserving the hierarchical categorization in the embedding space.

However, representation learning approaches that exploit this hierarchical relationship between labels have received very little attention. In recent years, several unsupervised [5, 13, 16] and supervised [17, 18, 30, 42] metric learning frameworks have been proposed. These approaches typically rely on minimizing the distance between representations of a positive pair and maximizing the distance between negative pairs. In the unsupervised (self-supervised) setting [5, 13, 33, 46], the positive pairs are different views of the same image, most typically obtained by random augmentations of the anchor image [5, 13, 33]. In the supervised setting, labels are used to construct a wider variety of positive pairs, from different images of the same class and their augmentations [17, 18]. Positive pairs constructed from augmentations of the anchor image, and pairs constructed from the anchor image and other images of the same class are considered to be equivalent, and the learning process attempts to minimize the distance between images in all of these positive pairs to the same degree. While representations learned in this paradigm may be satisfactory for a downstream task based on the supervisory label such as category prediction, other tasks such as instance prediction, retrieval, attribute prediction and clustering can suffer due to the absence of direct supervision for these tasks. Additionally, these approaches do not support multi-label learning and are unable to utilize information about the relationship between labels.

Formally, in the hierarchical multi-label setting, each data point has multiple dependent labels, and the relationship between labels is best represented in a hierarchy. See Figure 1(b) for a sample representation in a tree structure. For example, in the DeepFashion dataset [21], each data point has 3 hierarchically structured labels: category (Denim, Cardigan, Shirts etc), product (identified by product id) and variation (typically color / pattern variations). For the anchor image in Figure 1(a), which belongs to a specific product in the Denim category , the sub-category image is a different sample from the same product, and the category image is from a different product in the same cate-

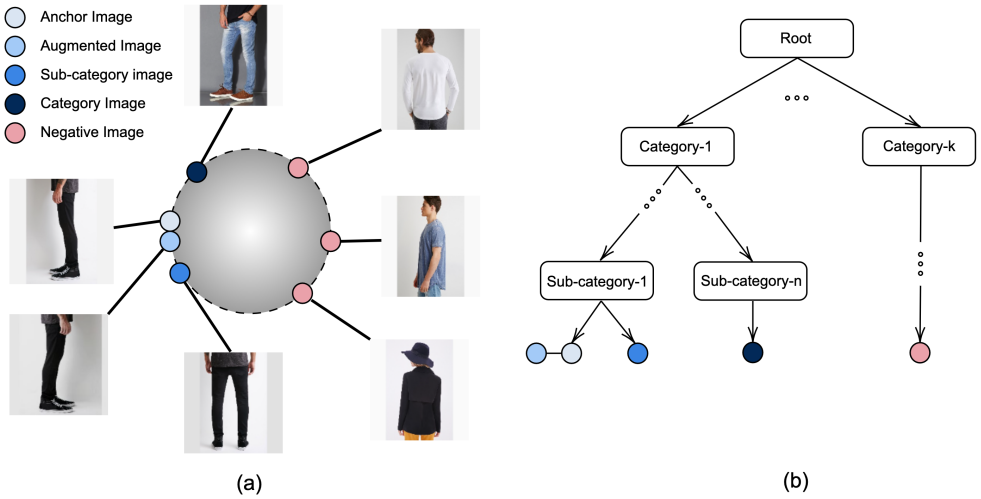

Figure 1. Hierarchical multi-label contrastive learning overview. A positive pair is constructed by pairing the anchor image with images drawn from all levels in the hierarchy. The learning objective of this work is to force positive pairs closer together, but the magnitude of the force is dependent on the common ancestry of the pair's labels. (a) The anchor image and corresponding positive pairings (blue) and negative pairings(red), visualized on a unit sphere. Different shades of blue nodes indicate their relationship to the anchor image, with darkness in shades of blue corresponding to increasing distance (both in the label space and representation space) from the anchor image. The red data points are from different categories in the dataset and hence form negative pairs with the anchor image. (b) Representation of the sample images from (a) in the label hierarchy. A tree data structure is used to to visualize the multi-labels.

gory. All the negative images are from different categories.

Our proposed approach leverages all the available labels to learn an embedding function that can preserve the label hierarchy in the embedding space. We develop a general representation learning framework that can utilize available ground truth and learn embeddings that generalize to a variety of downstream tasks. We present two novel losses (and their combination) that exploit the relationship between hierarchical multi-labels and learn representations that can retain the label relationship in the representation space. The Hierarchical Multi-label Contrastive Loss (HiMulCon) enforces a penalty that is dependent on the proximity between the anchor image and the matching image in the label space. In this setting, we define proximity in the label space as the overlap in ancestry in the tree structure. The Hierarchical Constraint Enforcing Loss (HiConE) prevents the hierarchy violation, that is, it ensures that the loss from pairs farther apart in the label space is never smaller than the loss from pairs that are closer. Models learned under this framework can be used exactly like traditional representation learning frameworks, a model is trained with our novel loss functions to learn an efficient encoder network, and embeddings generated from this approach can be used in a variety of downstream tasks.

Our framework is not limited to the hierarchical multi-label scenario. It reduces to the supervised contrastive approach [18] when only single level labels are available, and to the SimCLR [5] approach when no labels are available. We demonstrate the efficacy of our framework in comparison to Khosla *et al*. [18], Chen *et al*. [5] and a standard cross entropy based approach on downstream tasks such as category prediction, sub-category retrieval and clustering NMI [38]. These tasks also show that our approach preserves the hierarchical relationship between labels in the representation space, and generalizes to unseen data as well.

2. Related Work

Contrastive learning was first studied in the self-supervised setting [5, 13, 16, 33, 34, 46], typically relying on a pretext task to learn embeddings. The supervision for the pretext tasks would typically be generated from the data itself. Examples include denoising [37], colorization [48], image recognition [15], object detection and image segmentation [40], action recognition [26] and patch ordering [8]. The method in van den Oord *et al*. [25] used a probabilistic contrastive loss to capture mutual information between different views of data. They showed the effectiveness of their method on four domains: speech, images, text and reinforcement learning. Li *et al*. [20] performed clustering to find prototypical embedding, which is similar to an image embedding. The efficacy of learning representations in a contrastive fashion led to the development of supervised constrastive approaches. Khosla *et al*. [18] extended Chen *et al*. [5] to the supervised setting. The supervised con-

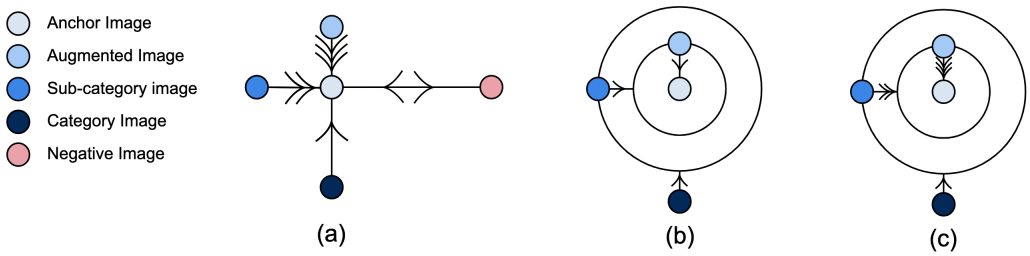

Figure 2. Conceptual visualization of the losses.(a) HiMulCon loss: A penalty, proportional to the proximity in the label space, is enforced on each positive pair. Images that are closer to the anchor image in the label space have a higher penalty (represented by the arrows), forcing them closer together. (b) HiConE loss: The hierarchy constraint is enforced by ensuring that image pairs that are farther away in the label space will never have a lower loss than image pairs that are closer. (c) Combined (HiMulConE) loss: (a) and (b) are combined, the penalty is applied in combination with the hierarchy preserving constraint. The loss term w.r.t negative samples is unchanged, and the negative images are omitted in this example.

trastive learning approach formulated positive pairs by sampling from different instances of the same class, as opposed to augmenting different views of the same image in the unsupervised setting. They show that the supervised contrastive learning is a generalization of the triplet loss [42] and the n-pair loss objectives [31]. Ho *et al*. [17] introduced an approach to utilize multiple instances of an object interchangeably to learn a viewpoint invariant representation in a self-supervised and semi-supervised setting.

Małkiński *et al*. [22] proposed a contrastive learning framework for visual reasoning in the multi-label setting, but under the assumption that any pair of images can be considered to be positive if they share even one label. Zhao *et al*. [49] proposed a normalized class-similarity-weighted sums for classification in the hierarchical setting, which is a replacement of the logistic regression. Wu *et al*. [44] learned a hierarchical classifier by calculating each hierarchy level's probability. Cho *et al*. [6] designed a fashion hierarchical classification model to classify fashion images. Bilal *et al*. [3] and Yan *et al*. [47] designed a deep network branch on side of a general network, and strategically learn super-classes at each hierarchical level. Instead of solving the label imbalance problem, [2] designed two cross entropy variations to handle top-k errors. Wehrmann *et al*. [41] introduced a multi-step pipeline learning a network which output a prediction for each layer in the hierarchy. Giunchiglia *et al*. [12] introduced the coherent hierarchical multi-label classification networks, which enforced the hierarchy constraint (described in Section 3.3). They introduced a modified version of binary cross entropy loss, where a separate module would modify the model confidence such that the confidence associated with all classes in the hierarchy will be equal or higher than the lowest class. Wang *et al*. [39] proposed a structured loss based on the similarities between query and gallery features. The method in [19] was built on Triplet loss, where levels of features are defined as degrees of similarities between pairwise points. These methods are fundamentally different from the proposed method, which directly builds the hierarchy on the natural characteristics of the data.

Ge *et al*. [11] constructed a hierarchical triplet loss, wherein the hierarchical tree representation is constructed using a single-level label structure by utilizing the intraclass distances to formulate a grouping mechanism, which is then used for hard negative mining and in the loss function. This approach differs from our approach in two important ways, first, it relies on the triplet loss instead of the contrastive loss. The triplet loss is a special case of the contrastive loss [18]. The second difference lies in the construction of the hierarchical tree. Our approach relies on the multi-level label information to construct the tree, whereas Ge *et al*. [11] constructs it on the fly from the data itself. The formulation of the tree from the data can result in propagating biases in the underlying model and the data to the representation learning framework and can be prone to noise. A recent method that was proposed by Garnot *et al*. [10] modeled the hierarchical class structure by integrating class distance into a prototypical network, where the use of a hierarchical tree is different in our approach.

Although our experiments include image classification and image retrieval, our goal is different from multi-task methods [1, 12, 41]. The proposed algorithm is a general hierarchical multi-label representation learning framework that can be applied to any downstream task. Our approach is agnostic to downstream tasks and is not directly optimized for them.

3. Methodology

In order to better explain our approach, we define some of the terminology that will be used throughout this work. A hierarchical multi-label dataset refers to a dataset where each data point has multiple related labels associated with

it, and the dependency is best described in a directed acyclic graph or a tree. Leaf nodes represent a unique image identifier, and all non-leaf nodes in the tree represent label information at various levels. The levels are analogous to depth in a tree structure. The lower levels correspond to broader categories (closer to the root of the tree), with the lowest level corresponding to the category label. For example, in Figure 1, the "DENIM" category would be the lowest label for the anchor image, and *sub-category-1* would be the highest level label. Leaf nodes would correspond to image identifiers. Positive pairs at a level $l \in L$ are formed by identifying a pair of images that have common ancestry up to level l and diverge thereafter. Referring again to Figure 1, the anchor image and the category image form a pair at the category level, as they only have the category label to be common between them. In graph terminology, a pair of images at level l implies that they will have their lowest common ancestor at level l.

Our method is constructed similar to the supervised contrastive learning [18] approach. First, an encoder network and a projection head is learned using all of the available hierarchical labels. The encoder networks weights are then frozen, and no finetuning is done on the encoder network for downstream tasks. If additional learning is necessary for downstream tasks, in classification tasks for example, a seperate classifier is trained with the embeddings generated by the encoder acting as the input to the classifier.

3.1. Contrastive Learning

The contrastive learning loss introduced in Chen *et al.* [5], was originally a self-supervised learning method. Such methods can pull an anchor and its augmented version together in the embedding space, while the anchors and negative samples are pushed apart. A set of N randomly sampled labeled pairs is defined as $\{x_k, y_k\}$, where x and y represent the samples and labels individually and $k = 1, ..., N$. Two augmentations are applied to each sample. Let i be the index of one augmented sample, and j be the index of the other, where $i \in A = \{1, ..., 2N\}$ and $j \neq i$. i is the anchor and j is the positive sample. The contrastive loss is defined as

$$L^{\text{self}} = -\sum_{i \in A} \frac{\exp(f_i \cdot f_j/\tau)}{\sum_{k \in A \setminus i} \exp(f_i \cdot f_k/\tau)} \tag{1}$$

Here, f represents the feature vector in the embedding space, and τ is a temperature parameter. Intuitively, the numerator calculates the inner dot product between an anchor i and its positive sample j. The denominator calculates all the inner dot products between i and all negative samples, where totally $2N - 1$ samples are calculated.

The supervised contrastive learning [18] extends Eq. 1 to a supervised scenario. Particularly, given the presence of labels, positive pairings for the anchor goes from one-to-many positive-negative samples in SimCLR [5], to many-to-many samples. The loss is defined as

$$L^{\text{sup}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P} \log \frac{\exp(f_i \cdot f_p/\tau)}{\sum_{a \in A \setminus i} \exp(f_i \cdot f_a/\tau)} \tag{2}$$

P represents the indices of all positives in the multi-view batches except for i. A represents all images in the batch, and $a \in A \setminus i$ is all images in the batch except the ith image. Therefore, the supervised contrastive loss consolidates information of all the positive samples in the numerator, and can essentially exploit the contrastive power between positive and negative samples.

3.2. Hierarchical Multi-label Contrastive Loss

Although the supervised contrastive learning in Eq. 2 can distinguish between multiple positive pairs, it is only designed for single labels. Define L as the set of all label levels, and $l \in L$ is a level in the multi-label. Then the loss for a pairing of the anchor image, indexed by i and a positive image at level l is defined as

$$L^{\text{pair}}(i, p_l^i) = \log \frac{\exp(f_i \cdot f_p^l/\tau)}{\sum_{a \in A \setminus i} \exp(f_i \cdot f_a/\tau)} \tag{3}$$

The hierarchical multi-label contrastive loss (HiMulCon) can then be defined as:

$$L^{\text{HMC}} = \sum_{l \in L} \frac{1}{|L|} \sum_{i \in I} \frac{-\lambda_l}{|P_l(i)|} \sum_{p_l \in P_l} L^{\text{pair}}(i, p_l^i) \tag{4}$$

where $\lambda_l = F(l)$ is a controlling parameter that applies a fixed penalty for each level in the hierarchy, P_l is the set of positive images for anchor image indexed by i. F is heuristically chosen and scales with l. See supplementary material for a study on different choices for F. Figure 2(a) provides a conceptual illustration of this loss.

The HiMulCon applies higher penalties to image pairs constructed from higher levels in the hierarchy, forcing them closer than pairs constructed from lower levels in the hierarchy. Note the construction of the loss with regards to the interaction between pairs at different levels. Pairs formed at the highest level will have all other paired images from lower levels form negative pairs at the higher levels, and the negative pairs formed by pairs with some level of lower common ancestry naturally form hard negative samples, hence becoming a form of hard negative mining. In addition, the λ_l term contributes to preserving the hierarchy explicitly.

If there is only one-level label, the HiMulCon loss reduces to the supervised contrastive loss. The supervised contrastive loss is therefore a special case of the HiMulCon.

3.3. Hierarchical Constraint Enforcing Loss

The Hierarchical Constraint Enforcing Loss, HiConE, enforces the hierarchical constraint in the representation learning setting. In the classification setting, as described in Wehrmann*et al*. [41] and Giunchiglia *et al*. [12], the hierarchical constraint ensures that if a data point belongs to a class, it should also belong to its ancestors. This can be defined in terms of confidence scores, where a class higher in the hierarchy cannot have a lower confidence score than a class lower in the ancestry sequence. Adapted to the contrastive learning scenario, we define the hierarchical constraint as the requirement that the loss between image pairs from a higher level in the hierarchy will never be higher than the loss between pairs from a lower level. This observation leads us to develop an hierarchical constraint enforcing loss (HiConE).

If we define $L^{\text{pair}}_{\text{max}}$ as the maximum loss from all positive pairs at level l:

$$L^{\text{pair}}_{\text{max}}(l) = \max_{(i,p^i_l)} L^{\text{pair}}(i, p^i_l) \tag{5}$$

Then, the HiConE loss L^{HCE} is defined as

$$\sum_{l \in L} \frac{1}{|L|} \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p_l \in P_l} \max(L^{\text{pair}}(i, p^i_l), L^{\text{pair}}_{\text{max}}(l-1)) \tag{6}$$

HiConE is computed sequentially in decreasing order of l from L to 0, which helps ensure that the pair loss at level $l-1$ can never be less than the max pair loss at l. Figure 2(b) has a conceptual visualization for this loss, where pairs formed at lower levels in the hierarchy will never have a higher loss than pairs formed at a lower level in the hierarchy.

3.4. Hierarchical Multi-label Constraint Enforcing Contrastive Loss

Intuitively Eq. 4 acts as an independent penalty defined on the level, whereas Eq. 6 is a dependent penalty that is defined in relation to the losses computed at the lower levels. We can combine the two losses to form the combined loss, the Hierarchical Multi-label Constraint Enforcing Contrastive Loss (HiMulConE), L^{HMCE}

$$\sum_{l \in L} \frac{1}{|L|} \sum_{i \in I} \frac{-\lambda_l}{|P(i)|} \sum_{p_l \in P_l} \max(L^{\text{pair}}(i, p^i_l), L^{\text{pair}}_{\text{max}}(l-1)) \tag{7}$$

Note that the combined loss is essentially adding the λ_l term to Eq. 6, giving us a loss term that has a level penalty as well as the hierarchy constraint enforcing term.

3.5. Hierarchical Batch Sampling Strategy

Wu *et al*. [43] highlighted the importance of sampling in representation learning. Khosla *et al*. [18] also showed that having a large number of hard positives/negatives in a batch leads to improved performance. In a hierarchical multi-label setting, it becomes necessary to ensure that each batch has sufficient representation from all levels of the hierarchy for each anchor image. Hence,we design a custom batch sampling strategy which ensures that each image can form a positive pair with images that share a common ancestry at all levels in the structure. The approach is straightforward: randomly sample an anchor image and get the label hierarchy. For each label in the multi-label, randomly sample an image in the sub-tree such that the anchor image and the sampled image have common ancestry up to that label. Steps are taken to ensure that each image is sampled only once in an epoch.

For example, in Figure 1 (b), the anchor image would be sampled. Positive pairings from each level need to be sampled next. First, a random image from sub-category-1 will be sampled. Next, a random image from category-1 but not sub-category-1 will be sampled. This process is repeated at all levels in the hierarchy. Finally, augmented versions of these images are also generated. Once completed, another anchor image is sampled randomly and the process repeats until $batch\_size$ number of images have been sampled.

4. Experiments

We evaluate our loss on three downstream tasks: image classification, image retrieval accuracy on sub-categories and NMI for clustering quality. We study the generalizability of our approach by evaluating the performance of our encoder network on unseen data. We also present qualititative results with t-SNE visualizations [35].

4.1. Datasets

We experiment with several popular datasets: ImageNet [29], DeepFashion In-Shop [21], iNaturalist [36] and ModelNet40 [45]. In order to showcase our results against popular benchmarks, we present results in the standard configurations of these datasets. We also split some of these datasets into seen and unseen sets, where we use the seen sets to train the encoder network, and evaluate performance of our approach and relevant baselines on the unseen dataset. We use the split to show that proposed losses are able to learn generalized representations that work well on unseen data.

DeepFashion dataset is a large-scale clothing dataset with more than 800K images. We use the In-Shop subset in our experiments as it has three-level labels: category, product ID and variation. The variation can be different colors or sub-styles for the same product. ModelNet40 is a syn-

	ImageNet [29]	DeepFashion [21]	iNaturalist [36]	ModelNet40 [45]
SimCLR [5]	69.53	70.38	54.02	79.26
Cross Entropy	77.60	72.44	56.86	81.31
SupCon [18]	78.70	72.82	57.28	81.60
Guided [10]	76.60	72.61	57.33	83.49
HiMulConE (Ours)	**79.14**	**73.21**	**59.40**	**88.46**

Table 1. Top-1 classification accuracy on the full datasets. Standard datasets and splits as described in the original papers are used here. For ImageNet and iNaturalist datasets, the task is classification at the finest sub-category level, whereas super-category level classification is performed for DeepFashion and ModelNet40. All baseline results were reproduced.

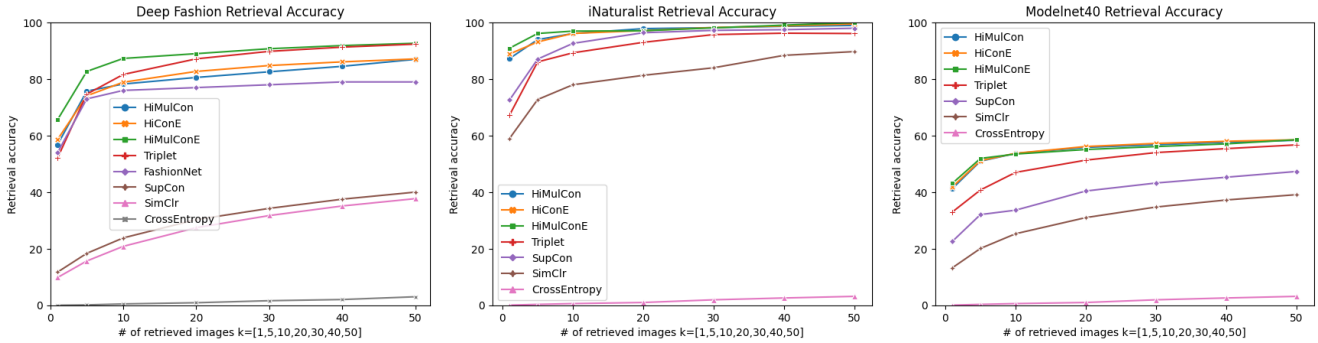

Figure 3. Retrieval results on the full datasets.

thetic dataset of 3,183 CAD models from 40 object classes. It has two-level hierarchical labels: category and CAD image ID. iNaturalist is a species classification dataset, with two levels in the hierarchy, a super-category for the genus, and species categories. ImageNet classes are hierarchicaly structured using the WordNet [23] hierarchy, we use the label hierarchy published in the *Robustness* library [9].

4.2. Implementation Details

We adopt a pre-trained ResNet-50 [14], which was trained on ImageNet [7], as the encoder network. For the ImageNet experiments, we train the model from scratch. We finetune on our datasets for 100 epochs. Specifically, we finetuned the parameters of the fourth layer of the ResNet-50 as well as a multi-layer perceptron header [18] on the seen dataset with the proposed losses. The optimizer is SGD with momentum [28]. The encoder model weights are frozen after the network is trained in this fashion. We train an additional linear classifier for 40 epochs to obtain the classification accuracy. We use the same setup for all models. Although the proposed losses can be trained together with the linear classifier, we find that the performance gap is rather small. The batch size in our experiments is 512, and we use the temperature τ as 0.1 in all experiments. We start from the learning rate as 0.1, and decrease it by 10 for every 40 epochs. The augmentations are the same as [13, 18].

Algorithm 1: Hierarchical Loss implementation

Data: Batch labels B of size N $\times$ L, Mask M of size N $\times$ L, features F of size N $\times$ d

$l \leftarrow N - 1$;
$M \leftarrow \mathbb{1}$;
$batchLoss \leftarrow 0$;
while $l \geq 0$ **do**
  $M[:, l] = 0$;
  $levelMask = M \odot B$;
  $i \leftarrow 0$;

  $levelPairings = torch.stack([torch.all($
  $torch.eq(levelMask[i], levelMask), dim =$
  $1) \quad for\ i\ in\ range(N)])$;

  $levelLoss = loss(features, levelParings)$;
  /* proposed loss functions */
  $batchLoss = batchLoss + levelLoss$;
end

The hierarchical relationship, and the batch sampling strategy described in section 3.5 requires a careful implementation of the loss calculation. Since a pairing can be a positive pair at a lower level and a negative pair at a higher level, the loss computation is aggregated by lay-

	DeepFashion		ModelNet40	
	Seen	Unseen	Seen	Unseen
SimCLR	70.26	68.12	77.09	72.26
Cross Entropy	77.81	71.94	85.17	79.77
SupCon	**81.46**	73.93	88.33	79.28
Guided	79.34	74.04	89.01	82.22
HiMulCon	80.54	74.88	89.28	84.44
HiConE	80.67	75.28	89.09	84.40
HiMulConE	80.52	**75.29**	**89.45**	**85.37**

Table 2. Top-1 classification accuracy on the seen / unseen splits of DeepFashion In-Shop and ModelNet40.

ers, with the highest layers calculated first. Restructuring the losses to be computed at each level and aggregated to form the batch loss allows for direct tensor operations and greatly speeds up the computation. See Algorithm 1 for a PyTorch [27] based pseudo code. Starting from the highest level, a $N \times N$ pairing mask is constructed at each level where $levelPairing[i,j] = 1\ if\ labels[i] == labels[j]\ at\ level\ l\ else\ 0$. In the single-level scenario, $levelPairings$ is only computed once, and all images of the same class will be set, reducing to Khosla *et al.* [18]. The $levelLoss$ is defined in Eq. 4, 6 and 7 individually. In the unlabeled scenario, $levelPairings$ will be a diagonal matrix, and this implementation reduces to SimCLR [5].

4.3. Classification Accuracy

We compare the proposed loss functions with SimCLR, an unsupervised contrastive loss [5], two supervised learning losses functions: cross entropy and supervised contrastive loss (SupCon) [18], and a state-of-the-art method (Guided) [10] that incorporates the idea of hierarchical labels. We do not compare with other metric learning approaches as Khosla *et al.* [18] showed that popular metric learning approaches like triplet loss [30, 42] are special cases of SupCon. The cross entropy uses flat list of labels and the softmax [4] function to train a classifier, and SupCon uses the labels to construct positive pairs in order to train a contrastive loss. The results are presented in Table 1, and the classification results reported here are on the standard configurations of the datasets. The supervised approaches are very competitive for this task, as the encoder is also trained with the same supervisory signal as the classification task. Although our approach has access to additional labels during the representation learning phase, these labels are not used in training the classifier. All approaches evaluated here have exactly the same classifier training mechanism.

	DeepFashion		ModelNet40	
	Category NMI	Product NMI	Category NMI	Product NMI
SimCLR	0.15	0.73	0.31	0.52
CE	0.1	0.66	0.12	0.4
SupCon	0.57	0.68	0.57	0.69
HiMulCon	0.57	0.8	**0.62**	**0.88**
HiConE	0.58	0.78	0.61	**0.88**
HiMulConE	**0.59**	**0.81**	**0.62**	**0.88**

Table 3. NMI on DeepFashion In-Shop and ModelNet40. CE represents cross entropy. Product NMI represents its mean NMI.

Table 4. Retrieval results using MAP evaluation metrics.

	DeepFashion	iNaturalist	ModelNet40
SupCon	31.5	61.5	21.6
HiMulConE	**35.6**	**66.9**	**26.0**

4.4. Image Retrieval Accuracy

This downstream task here is to retrieve images from the gallery that are the same class as the query image. The top-k accuracy is usually adopted to measure if a query image class can be found in the top-k retrieved results from the gallery. In this task, class is used to refer to the finest subcategory ID in the dataset. For DeepFashion dataset, the test set forms the query images, in ModelNet, we split the data into train, validation and test sets, and use the test set as the query images. In order to evaluate retrieval results on the iNaturalist dataset, we create a custom query and gallery set: We follow the original train/validation split, using all 579K training images to train the encoder model, and use 20 percent of the validation dataset (17K images) as the query set and the rest (78K images) as the gallery set.

In addition to the baselines from the classification experiment, we include triplet loss [30, 42] and FashionNet [21] results as well, as they are more appropriate for retrieval tasks. We were unable to find equivalent results, or an official implementation for Ge *et al.* [11] (a different encoder network is used in the paper), and hence do not include their results here. In Figure 3, we show results of the proposed three losses versus the baselines on DeepFashion In-Shop dataset, iNaturalist and Modelnet40. Our losses are clearly superior to the baseline results, with HiMulConE showing greater improvement at smaller k.

Moreover, a recent study [24] showed that top-k retrieved results have flaws, and proposed a metric, Mean Average Precision at R (MAP). We report results in Table 4.

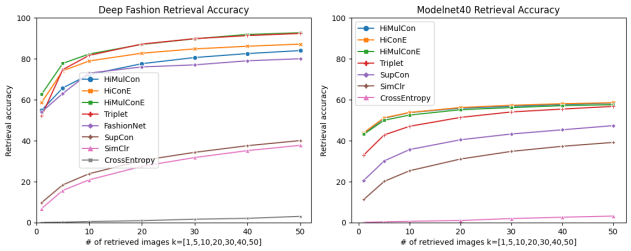

Figure 4. Retrieval results on the unseen splits of DeepFashion and ModelNet. HiMulConE performs better than the other approaches, especially at lower k.

4.5. Generalizability to Unseen Data

4.5.1 Setup

In order to evaluate the performance of our models on unseen data, we split DeepFashion and Modelnet40 datasets into a seen and unseen set. Each dataset is further split into train, validation and test sets. The seen set is used to train the encoder network, and the model is frozen. For the classification task on unseen data, a classifer is trained on the embeddings generated by the encoder network that was trained on the seen data, the encoder network is not fine-tuned on unseen data.

4.5.2 Classification

We finetuned the pretrained model on the seen dataset. To obtain results in the unseen dataset, we only train the classifier on the embeddings generated from the encoder network that was trained on the seen dataset. Table 2 shows the top-1 accuracy of classification accuracy on DeepFashion In-Shop and ModelNet40. It is seen that the proposed methods obtain better results than the baselines on the unseen part of both datasets, while obtaining comparable results to SupCon on the seen part. In theory, HiConE exploits the properties of the dataset by respecting the semantic relationships of neighboring levels, penalizing pairs by magnitudes that are correlated with the distance between the pair in the embedding space. HiMulCon on the other hand is a fixed penalty based on level differences, penalizing pairs on the distance between them in the label space. A hybrid of the two methods, HiMulConE can leverage differences from both the label and the embedding spaces. In Table 2 we see that HiConE does better on the dataset with significant semantic overlap in different levels of the tree (DeepFashion), and the gap is much smaller in the semantically well separated dataset (ModelNet).

4.5.3 Image Retrieval

The experimental setup is similar to section 4.4, with the difference being that the data used comes from the unseen set only. The results are presented in Figure 4. Once again, our losses perform better than the baselines, particularly at lower values of k. This shows the generalisability of our work in comparison to the baselines. Since our approach can incorporate the label hierarchy in the network loss, the embedding space preserves the label-space hierarchy.

4.5.4 Clustering

Clustering is another downstream task that can be used to evaluate the quality of the embeddings. As in Ho *et al.* [17], we use K-means and the NMI [38] score to evaluate clustering quality. We first generate the embeddings for all the images in the unseen test set, and perform K-means in the representation space. Clustering is done at two levels: the lowest (category) and highest (product ID) levels in the tree. At the category level, K is set to the number of categories in the dataset, and NMI measures the consistency between the category labels and *clusterId*. At the ID level, for each cateogry, we perform K-means, with K set to the number of products in that category. The mean of ID-level NMIs, across all categories, is reported in the Product NMI columns in Table 3. The significant improvement over the baseline in product NMI shows that our approach maintains separability for sub-categories within a category, and also shows that our approach preserves the hierarchical relationship between labels in the representation space.

5. Limitations and Social Impact

A limitation of our approach is the requirement of hierarchical labels for learning the encoder network, which can be expensive to acquire. In addition, our approach has been tested on datasets where labels have a tree-like structure, where each node only has one parent. However, it is relatively straightforward to extend this to general graph structures. Another common limitation arises from the underlying data used for experiments. Biases in the data [32] can be learned by the model, which can have significant societal impact. Explicit measures to debias data through re-annotation or restructuring the dataset for adequate representation is necessary.

6. Conclusion

Hierarchically categorized data is common in the real world, and our novel approach provides a general framework for utilizing all available label data, reducing to standard supervised or self-supervised approaches in the absence of sufficient data. Our approach generalizes well on a variety of downstream tasks and unseen data, and significantly outperforms the evaluated baselines. In future work, we would like to extend this work to multi-label scenarios that are not in a hierarchical framework, and to other modalities and multi-modal settings, incorporating modalities such as speech and language.

References

[1] Maxim Berman, Herve Jegou, Andrea Vedaldi, Iasonas Kokkinos, and Matthijs Douze. Multigrain: a unified image embedding for classes and instances. *arXiv preprint arXiv:1902.05509*, 2019. 3

[2] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas Lord. Making better mistakes: leveraging class hierarchies with deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12506–12515, 2020. 3

[3] Alsallakh Bilal, Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. Do convolutional neural networks learn class hierarchy? *IEEE Transactions on Visualization and Computer Graphics*, 24:152–162, 2018. 3

[4] Christopher Bishop. Pattern recognition and machine learning, 2006. Springer. 7

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607, 2020. 1, 2, 4, 6, 7

[6] Hyunsoo Cho, Chaemin Ahn, Kang Min Yoo, Jinseok Seol, and Sang-goo Lee. Leveraging class hierarchy in fashion classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision Workshop*, 2019. 3

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-fei Li. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[8] 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. 2

[9] Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. 6

[10] Vivien Sainte Fare Garnot and Loic Landrieu. Leveraging class hierarchies with metric-guided prototype learning. In *The British Machine Vision Conference*, 2021. 3, 6, 7

[11] Weifeng Ge. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–285, 2018. 3, 7

[12] Eleonora Giunchiglia and Thomas Lukasiewicz. Coherent hierarchical multi-label classification networks. *arXiv preprint arXiv:2010.10151*, 2020. 3, 5

[13] 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. 1, 2, 6

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6

[15] Olivier Henaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S.M. Ali Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, 2020. 2

[16] Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019. 1, 2

[17] Chih-Hui Ho, Bo Liu, Tz-Ying Wu, and Nuno Vasconcelos. Exploit clues from views: Self-supervised and regularized learning for multiview object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9090–9100, 2020. 1, 3, 8

[18] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 2, 3, 4, 5, 6, 7

[19] Yonghyun Kim and Wonpyo Park. Multi-level distance regularization for deep metric learning. In *AAAI*, 2021. 3

[20] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven C. H. Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020. 2

[21] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1096–1104, 2016. 1, 5, 6, 7

[22] Mikołaj Małkiński and Jacek Mańdziuk. Multi-label contrastive learning for abstract visual reasoning. *arXiv preprint arXiv:2012.01944*, 2020. 3

[23] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998. 6

[24] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pages 681–699. Springer, 2020. 7

[25] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[26] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: contrastive video representation learning with temporally adversarial examples. *arXiv preprint arXiv:2103.05905*, 2021. 2

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 7

[28] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 6

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy,

Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5, 6

[30] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 1, 7

[31] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, 2016. 3

[32] Ryan Steed and Aylin Caliskan. Image representations learned with unsupervised pre-training contain human-like biases. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 701–713, 2021. 8

[33] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 1, 2

[34] Michael Tschannen, Josip Djolonga, Paul Rubenstein, Sylvain Gelly, and Lucic Mario. On mutual information maximization for represenation learning. In *International Conference on Learning Representations*, 2020. 2

[35] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008. 5

[36] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8769–8778, 2018. 5, 6

[37] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*, pages 1096–1103, 2008. 2

[38] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:2837–2854, 2010. 2, 8

[39] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil Robertson. Ranked list loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 3

[40] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Li Lei. Dense contrastive learning for self-supervised visual pre-training. *arXiv preprint arXiv:2011.09157*, 2020. 2

[41] Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. Hierarchical multi-label classification networks. In *International Conference on Machine Learning*, pages 5075–5084. PMLR, 2018. 3, 5

[42] Kilian Weinberger and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009. 1, 3, 7

[43] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2848, 2017. 5

[44] Hui Wu, Michele Merler, Rosario Uceda-Sosa, and Smith. Learning to make better mistakes: semantics-aware visual food recognition. In *Proceedings of the ACM International Conference on Multimedia*, 2016. 3

[45] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 5, 6

[46] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 1, 2

[47] Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision*, pages 2740–2748, 2015. 3

[48] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017. 2

[49] Bin Zhao, Fei Li, and Eric Xing. Large-scale category structure aware image categorization. In *Advances in Neural Information Processing Systems*, 2011. 3