Multi-Label Supervised Contrastive Learning

Pingyue Zhang, Mengyue Wu[*]

MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{williamzhangsjtu, mengyuewu}@sjtu.edu.cn

Abstract

Multi-label classification is an arduous problem given the complication in label correlation. Whilst sharing a common goal with contrastive learning in utilizing correlations for representation learning, how to better leverage label information remains challenging. Previous endeavors include extracting label-level presentations or mapping labels to an embedding space, overlooking the correlation between multiple labels. It exhibits a great ambiguity in determining positive samples with different extent of label overlap between samples and integrating such relations in loss functions. In our work, we propose Multi-Label Supervised Contrastive learning (MulSupCon) with a novel contrastive loss function to adjust weights based on how much overlap one sample shares with the anchor. By analyzing gradients, we explain why our method performs better under multi-label circumstances. To evaluate, we conduct direct classification and transfer learning on several multi-label datasets, including widely-used image datasets such as MS-COCO and NUS-WIDE. Validation indicates that our method outperforms the traditional multi-label classification method and shows a competitive performance when comparing to other existing approaches.

Introduction

Multi-label classification (MLC) scenario commonly exists in many machine learning domains such as computer vision (Wang et al. 2017), audio signal processing (Gemmeke et al. 2017), and natural language processing (Yang et al. 2018). Unlike single-label classification, label correlation is more complex and important in MLC. Previous endeavors on MLC have included latent space learning and label correlation modeling (Wang et al. 2017; Zhang et al. 2021; Ma et al. 2021).

As a matter of fact, MLC shares a similar goal with contrastive learning paradigm, where two instances exhibit a relation based on their content, semantics, or label correlation (Zhang et al. 2022). Therefore, extending contrastive learning to MLC is a natural progression and has shown promising results in recent works (Zhang et al. 2022; Małkiński and Mańdziuk 2022; Bai, Kong, and Gomes 2022; Zhou, Kang, and Ren 2022). However, these works mainly focus on the correlation between the sample and its corresponding label set, which may limit the generalization ability of their models due to the sample-label binding and the exclusive use of fully supervised learning. In MulCon (Dao et al. 2021), several label-level embeddings are extracted for each label in the dataset for a single sample. If two samples belong to the same class, their corresponding label-level embeddings, which correspond to the class, are pulled together. Further, contrastive learning can boost multi-label prediction model based on a Gaussian mixture variational autoencoder (C-GMVAE), indicating that contrastive loss can pull together correlated label embeddings and push away unrelated label embeddings in MLC (Bai, Kong, and Gomes 2022).

Different from prior investigations into MLC contrastive learning, our approach focuses primarily on capturing the correlation between samples. In this regard, we specifically analyze the relationships between two sets of labels, instead of directly utilizing the label information itself. By decoupling samples from their associated labels, our intention is to achieve improved performance and enhance generalization capabilities. This departure from binding samples to labels is driven by the aim to enhance overall model effectiveness. Taking inspiration from successful supervised contrastive learning (SupCon) (Khosla et al. 2020) technique, which demonstrates enhanced capability in exploiting label information within single-label classification tasks, we strive to extend these concepts to the realm of MLC. SupCon leverages label information and broadens the concept of positive samples from the anchor to all samples sharing the same label. This approach consistently outperforms traditional cross-entropy loss and shows robustness against data corruptions. However, in the context of MLC, determining positive samples for an anchor proves intricate and ambiguous due to the presence of multiple labels. This complexity stands in contrast to single-label classification, where positive samples can be unambiguously identified by matching their label to that of the anchor.

An intuitive question would be, should we consider one sample as positive when its label set partially overlaps with or exactly matches the anchor's? We take both situations into consideration and properly define them as "**ANY**" for partially overlapping, and "**ALL**" for exactly matching. We then show the possible drawbacks of these two meth-

[*] Mengyue Wu is the corresponding author.

ods and propose a new multi-label supervised contrastive (**MulSupCon**) loss function to define positive samples in a finer way, based on the overlap proportion (i.e., 2 out of 3 labels are the same) between one instance and the anchor. By analyzing the gradients, we indicate (**i**) the optimizing process can be viewed as pushing the anchor's embedding to the direction calculated by averaging embeddings of each of its classes, where the class embedding is represented by averaging the embedding of samples which belong to it; (**ii**) our loss function weighs each sample according to how many labels are shared between the anchor and the designated sample. Note that our approach yields a pretrained model. We add an extra linear classifier to tune the model using BCE. Our main contributions are:

1. In this paper, we present a novel approach for incorporating correlations between samples in Multi-Label Classification (MLC) by leveraging the principles of supervised contrastive learning. Our method introduces a novel contrastive loss function, termed "MulSupCon", which effectively extends the single-label supervised contrastive learning to the multi-label context.
2. We evaluate the performance of MulSupCon by comparing it with other state-of-the-art MLC methods, the commonly adopted MLC loss BCE, and two alternative SupCon variants, on a series of datasets. Our findings reveal a notable enhancement in performance when leveraging MulSupCon.
3. Upon transferring the models trained using MulSupCon and BCE methods respectively, to downstream tasks, our approach showcases superior generalization capabilities.

Methods

For a batch of data $\mathcal{B} = [(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}), (\boldsymbol{x}^{(2)}, \boldsymbol{y}^{(2)}), \cdots, (\boldsymbol{x}^{(B)}, \boldsymbol{y}^{(B)})]$, where B is the mini-batch size and $\boldsymbol{y}^{(i)} = \{y_j^{(i)}\}_j$ is the multi-label of sample i, where $y_j^{(i)}$ means the j-th label of sample i. The main challenge of applying supervised contrastive learning to multi-label classification is that for the anchor sample i it is difficult to determine its corresponding positive samples. We will first introduce the framework of our contrastive learning and give several notations.

Framework The general framework is similar to the widely-used MoCo (He et al. 2020) framework. For sample i, we use $\boldsymbol{z}_q^{(i)}$ and $\boldsymbol{z}_k^{(i)}$ to denote the L_2-normalized output of the query model and key model, where the key model is momentum updated. We also use a queue $\mathcal{Q}$ to contain $\boldsymbol{z}_k$ from previous batches, as mentioned in MoCo.

Multi-Label Supervised Contrastive Loss As illustrated in Figure 1, two preliminary ideas to find positive samples include:

1. **ALL**: only those with exactly the same label class $\boldsymbol{y}$ are considered positive
2. **ANY**: samples with any class overlapping with the anchor $\boldsymbol{y}^{(i)}$ are positive

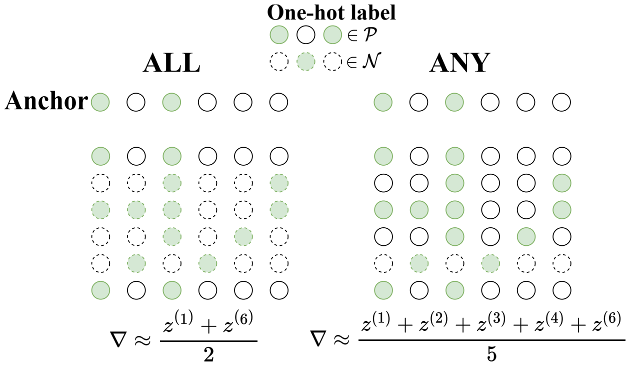
One-hot label
∈ $\mathcal{P}$
∈ $\mathcal{N}$
ALL
ANY
Anchor
$\nabla \approx \frac{z^{(1)} + z^{(6)}}{2}$
$\nabla \approx \frac{z^{(1)} + z^{(2)} + z^{(3)} + z^{(4)} + z^{(6)}}{5}$

Figure 1: Illustration for **ALL** and **ANY** and method. Each row represents one sample's label, where the first row is the anchor and the following ones are samples in $\mathcal{A}$, here $|\mathcal{A}| = 6$. Each sample's label is denoted in a one-hot form where the green circle means 1. The row with circles plotted with dotted line means that the corresponding sample is in the negative set $\mathcal{N}$, otherwise the sample is in the positive set $\mathcal{P}$.

Correspondingly, the positive sets for these two illustrations are

$$\mathcal{P}^{(i)} = \{m | \forall m, \boldsymbol{y}^{(m)} = \boldsymbol{y}^{(i)}\}$$

for **ALL** and

$$\mathcal{P}^{(i)} = \{m | \forall m, (\boldsymbol{y}^{(m)} \cap \boldsymbol{y}^{(i)}) \neq \varnothing\}$$

for **ANY**.

We use $\mathcal{A}^{(i)}$ to denote indices of all samples involved in calculating contrastive loss (from $\mathcal{B}$ and $\mathcal{Q}$), $\mathcal{A}^{(i)} = \mathcal{A}^{(j)}, \forall i, j \in \{1, 2, \cdots, B\}$, and $|\mathcal{A}^{(i)}| = |\mathcal{B}| + |\mathcal{Q}|$. Negative set $\mathcal{N}^{(i)} = \mathcal{A}^{(i)} - \mathcal{P}^{(i)}$.

Following SupCon (Khosla et al. 2020) loss:

$$\mathcal{L}_{\text{supcon}}^{(i)} = \frac{-1}{|\mathcal{P}^{(i)}|} \sum_{p \in \mathcal{P}^{(i)}} \log \frac{e^{s_p^{(i)}/\tau}}{\sum_{a \in \mathcal{A}^{(i)}} e^{s_a^{(i)}/\tau}}, \tag{1}$$

where $s_j^{(i)} = \boldsymbol{z}_q^{(i)} \cdot \boldsymbol{z}_k^{(j)}$. By taking gradient of $\mathcal{L}_{\text{supcon}}^{(i)}$ with respect to $s_p^{(i)}, s_n^{(i)}, p \in \mathcal{P}^{(i)}, n \in \mathcal{N}^{(i)}$ we get:

$$\nabla_{s_p^{(i)}} \mathcal{L}_{\text{supcon}}^{(i)} = \frac{1}{\tau}(\frac{-1}{|\mathcal{P}^{(i)}|} + \frac{e^{s_p^{(i)}/\tau}}{C}), \nabla_{s_n^{(i)}} \mathcal{L}_{\text{supcon}}^{(i)} = \frac{1}{\tau} \cdot \frac{e^{s_n^{(i)}/\tau}}{C},$$

where $C = \sum_{a \in \mathcal{A}^{(i)}} e^{s_a^{(i)}/\tau}$.

By taking gradient of $\mathcal{L}_{\text{supcon}}^{(i)}$ with respect to $z_q^{(i)}$ we get:

$$\begin{aligned} \nabla_{\boldsymbol{z}_q^{(i)}} \mathcal{L}_{\text{supcon}}^{(i)} &= \sum_{p \in \mathcal{P}^{(i)}} \nabla_{s_p^{(i)}} \mathcal{L}_{\text{supcon}}^{(i)} \cdot \boldsymbol{z}_k^{(p)} \\ &\quad + \sum_{n \in \mathcal{N}^{(i)}} \nabla_{s_n^{(i)}} \mathcal{L}_{\text{supcon}}^{(i)} \cdot \boldsymbol{z}_k^{(n)} \\ &= \sum_{p \in \mathcal{P}^{(i)}} \frac{1}{\tau}(\frac{-1}{|\mathcal{P}^{(i)}|} + \frac{e^{s_p^{(i)}/\tau}}{C}) \cdot \boldsymbol{z}_k^{(p)} \\ &\quad + \sum_{n \in \mathcal{N}^{(i)}} \frac{1}{\tau} \cdot \frac{e^{s_n^{(i)}/\tau}}{C} \cdot \boldsymbol{z}_k^{(n)}, \end{aligned} \tag{2}$$

As a result,

$$
\begin{aligned}
\nabla_{\boldsymbol{z}_q^{(i)}} \mathcal{L}^{(i)}_{\text{supcon}} &= \bar{\boldsymbol{z}} + \hat{\boldsymbol{z}} \\
\bar{\boldsymbol{z}} &= \frac{-1}{\tau} \cdot \frac{1}{|\mathcal{P}^{(i)}|} \sum_{p \in \mathcal{P}^{(i)}} \boldsymbol{z}_k^{(p)} \\
\hat{\boldsymbol{z}} &= \sum_{a \in \mathcal{A}^{(i)}} \frac{1}{\tau} \cdot \frac{e^{s_a^{(i)}/\tau}}{C} \cdot \boldsymbol{z}_k^{(a)},
\end{aligned}
\tag{3}
$$

which means the optimization strategy is pushing $\boldsymbol{z}_q^{(i)}$ towards the direction of the mean value of all $\boldsymbol{z}_k^{(p)}$ (scaled by $\frac{1}{\tau}$), where $\hat{\boldsymbol{z}}$ is the weighted average of all embeddings which prevents the collapse of representation.

As a result, for **ALL** the optimization direction is the mean representation of all samples with exactly the same label. The drawback is that $|\mathcal{P}|$ is small so the mean representation suffers from randomness. It treats samples which belong to the same class as negative ones. For **ANY**, the main drawback is that if most samples have some common classes, the averaging process will highlight the information of the common classes and give less weight to others. As illustrated in Figure 1, all positive samples belong to the third class while only three samples belong to the first class. Therefore, the mean ∇ mainly contains the information of the third class while the first class's information gains less weight.

Take one of datasets used in our work MS-COCO (Lin et al. 2014) as an example, if the anchor belongs to `person, bicycle, car` class, then there are less than 800 (1%) samples among over 80k samples which belong to exactly the same class set. Furthermore, there are about 45k, 2.3k, and 8.6k samples belong to `person, bicycle, car` respectively. The loss function we propose treats sample i as a separate sample for each class $y_j^{(i)}$ it belongs to. For each $y_j^{(i)} \in \boldsymbol{y}^{(i)}$, we construct a separate positive set:

$$\mathcal{P}_j^{(i)} = \{m | \forall m, y_j^{(i)} \in \boldsymbol{y}^{(m)}\}.$$

The proposed **MulSupCon** loss is:

$$\mathcal{L}^{(i)} = \sum_{y_j^{(i)} \in \boldsymbol{y}^{(i)}} \frac{-1}{|\mathcal{P}_j^{(i)}|} \sum_{p \in \mathcal{P}_j^{(i)}} \log \frac{e^{s_p^{(i)}/\tau}}{\sum_{a \in \mathcal{A}^{(i)}} e^{s_a^{(i)}/\tau}}, \tag{4}$$

and we illustrate it in Figure 2. The loss function of one batch is:

$$\mathcal{L} = \frac{1}{\sum_i |\boldsymbol{y}^{(i)}|} \sum_i \mathcal{L}^{(i)}. \tag{5}$$

Our loss function is a generalized version of SupCon loss in Equation (2), where it reduces to SupCon loss under the single-label circumstance, that is, $|\boldsymbol{y}^{(i)}| = 1$.

Note that we do not use $\frac{1}{|\boldsymbol{y}^{(i)}|}$ to weigh each sample as shown in Equation (6):

$$\frac{1}{|\boldsymbol{y}^{(i)}|} \sum_{y_j^{(i)} \in \boldsymbol{y}^{(i)}} \frac{-1}{|\mathcal{P}_j^{(i)}|} \sum_{p \in \mathcal{P}_j^{(i)}} \log \frac{e^{s_p^{(i)}/\tau}}{\sum_{a \in \mathcal{A}^{(i)}} e^{s_a^{(i)}/\tau}}, \tag{6}$$

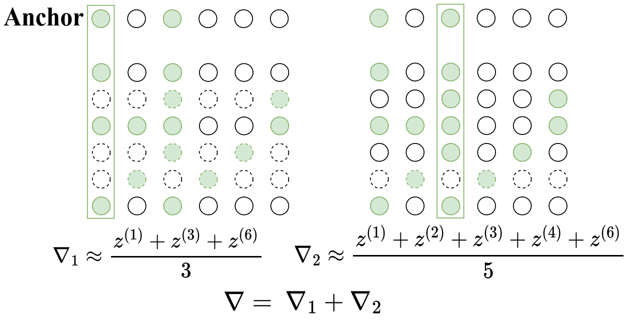
Anchor
$\nabla_1 \approx \frac{z^{(1)} + z^{(3)} + z^{(6)}}{3}$
$\nabla_2 \approx \frac{z^{(1)} + z^{(2)} + z^{(3)} + z^{(4)} + z^{(6)}}{5}$
$\nabla = \nabla_1 + \nabla_2$

Figure 2: The proposed MulSupCon loss function. We consider each label separately and form multiple positive sets for one anchor sample. The positive sets for the anchor is $\mathcal{P}_1 = \{1, 3, 6\}, \mathcal{P}_3 = \{1, 2, 3, 4, 6\}$, suppose the anchor's label is $\boldsymbol{y} = \{1, 3\}$.

and we show the effect of $\frac{1}{|\boldsymbol{y}^{(i)}|}$ in the ablation section.

By taking gradient of $\mathcal{L}^{(i)}$ with respect to $z_q^{(i)}$ we get:

$$
\begin{aligned}
\nabla_{\boldsymbol{z}_q^{(i)}} \mathcal{L}^{(i)} &= \sum_{y_j^{(i)} \in \boldsymbol{y}^{(i)}} \frac{-1}{\tau} \frac{1}{|\mathcal{P}_j^{(i)}|} \sum_{p \in \mathcal{P}_j^{(i)}} \boldsymbol{z}_k^{(p)} \\
&\quad + |\boldsymbol{y}^{(i)}| \sum_{a \in \mathcal{A}^{(i)}} \frac{1}{\tau} \cdot \frac{e^{s_a^{(i)}/\tau}}{C} \cdot \boldsymbol{z}_k^{(a)} \\
&= \sum_{y_j^{(i)} \in \boldsymbol{y}^{(i)}} \bar{\boldsymbol{z}}_j + |\boldsymbol{y}^{(i)}| \cdot \hat{\boldsymbol{z}},
\end{aligned}
\tag{7}
$$

where

$$\bar{\boldsymbol{z}}_j = \frac{-1}{\tau} \frac{1}{|\mathcal{P}_j^{(i)}|} \sum_{p \in \mathcal{P}_j^{(i)}} \boldsymbol{z}_k^{(p)}$$

is the mean value of $\boldsymbol{z}$ from samples which belong to class $y_j^{(i)}$. If we consider $\bar{\boldsymbol{z}}_j$ as the "class" representation, then the optimization direction of our loss function is the sum value of all its "class" representations with an extra mean negative vector $\hat{\boldsymbol{z}}$.

From another perspective, we pick a sample t and denote $\boldsymbol{y} = \boldsymbol{y}^{(i)} \cap \boldsymbol{y}^{(t)}$, then following Equation (7), $\boldsymbol{z}_k^{(t)}$ will be weighted by

$$\frac{-1}{\tau} \sum_{y_j \in \boldsymbol{y}} \frac{1}{|\mathcal{P}_j^{(i)}|},$$

which means more labels sample t intersects with the anchor, $\boldsymbol{z}_k^{(t)}$ will gain more weight.

Experiments

To assess our approach, we pretrain the model with the MulSupCon loss on various datasets. Subsequently, we employ the pretrained model, excluding its final non-linear projection head g. By default, we introduce a linear layer atop the pretrained model and conduct MLC using the Binary Cross Entropy (BCE) Loss.

Dataset	# Samples	# Labels	Mean L/S
MS-COCO	82081/40137	80	2.93/2.90
NUS-WIDE	150000/59347	81	2.41/2.40
Objects365	515233/80000	365	6.48/7.17
MIRFLICKR	19664/4917	24	3.78/3.76
PASCAL	5011/4952	20	1.46/1.42
Bookmarks	60000/27856	208	2.03/2.03
Mediamill	29804/12373	101	4.54/4.60
Nus-vec	125449/83898	81	2.40/2.41
Delicious	12886/3181	983	19.07/18.94
Scene	1210/1195	6	1.06/1.09
Yeast	1500/917	14	4.23/4.25

Table 1: Image and Vector dataset statistics, Mean L/S represents mean number of labels per sample. We present the statistics for the training and test sets separately, using a slash to divide them.

Data We evaluate our method by measuring the multi-label classification performance on several image datasets, including MS-COCO (Lin et al. 2014), NUS-WIDE (Chua et al. 2009), Objects365 (Shao et al. 2019), MIRFLICKR (Huiskes and Lew 2008), and PASCAL-VOC (VOC2007) (Everingham et al. 2007). The dataset statistics are shown in Table 1. Note that for Objects365 we randomly pick a subset for the whole dataset contains tremendous number of images. We also validate on several vector multi-label datasets from Mulan[1] where each sample is preprocessed and released in a vector form.

Metrics We use six commonly-used metrics to evaluate our method and compare with baseline methods. Suppose that $\boldsymbol{y}$ is the ground truth label and $\hat{\boldsymbol{y}}$ is the predicted label (all in one-hot form), we use (1) the mean value of average precision (mAP), (2) the precision of the top-1 predictions (precision@1), (3) macro-F1, (4) micro-F1, (5) Hamming Accuracy $\frac{1}{L}\sum_{i=1}^{L}\mathbb{1}[y_i = \hat{y}_i]$, L is number of class and (6) example-F1 $\frac{2\sum_{i=1}^{L} y_i\hat{y}_i}{\sum_{i=1}^{L} y_i+\sum_{i=1}^{L}\hat{y}_i}$.

Settings For **image datasets** we use a common encoder architecture: ResNet-50. During the preprocessing process before training, all images are first resized to 224 × 224 before training for easier storage and speedy training. For **vector datasets**, we use a simple multi-layer perceptron (MLP) which contains three layers of linear, ReLU activation function, and dropout. Note that we train both backbone encoders from scratch without using pretrained parameters.

For **image datasets**, during the pretraining stage, we train the model for 400 epochs using SGD optimizer with an initial learning rate of 0.1 and a cosine learning rate scheduler. A batch size of 64 is utilized for all datasets except for the Objects365 dataset, where a batch size of 128 is employed. For **vector datasets**, the model is trained for 150 epochs using the Adam optimizer with an initial learning rate of 0.0004, coupled with a cosine learning rate scheduler. A batch size of 256 is utilized, except for the yeast and scene datasets, where a batch size of 32 is used.

For two kinds of datasets, we all use data augmentation during pretraining because we find out that the performance will drop when the training period is long without a hard data augmentation. During pretraining, for image datasets, we first randomly crop the image and resize it to 224×224; then, we apply color jittering as well as randomly converting the image to greyscale. And for vector datasets, we randomly mask 50% input elements.

After the pretraining stage, for image datasets, we both linearly probe and finetune the model to evaluate the performance. To linearly probe the model, we freeze the encoder and use the Adam optimizer with an initial learning rate of 0.01. And to finetune the model, we train the whole model use the Adam optimizer with an initial learning rate of 1×10^{-4}. And for vector datasets, we finetune the model and use a learning rate of 4×10^{-4} for the linear layer and 4×10^{-5} for the backbone. For all datasets, we reduce the learning rate by a factor of 0.1 when it plateaus.

As one of the comparison methods, we train the model from scratch using BCE loss for image datasets. We employ identical settings including the optimizer, scheduler, number of epochs, augmentation strategy, and batch size as described in the pretraining process above.

Results

Following previous tradition of evaluating pretraining model like SimCLR (Chen et al. 2020) and SupCon (Khosla et al. 2020), we mainly evaluate our method on two aspects: linear probing/finetuning and transfer learning to downstream tasks, compared with existing state-of-the-art multi-label algorithms, commonly adopted BCE loss for multi-label classification, and the aforementioned **ALL** and **ANY** under supervised-contrastive learning paradigm.

Methods of comparison We present a comprehensive comparative analysis of our proposed method against several established MLC approaches, as shown in Tables 2 to 4. Specifically, the methods under consideration are as follows:

- LaMP (Lanchantin, Sekhon, and Qi 2020): LaMP employs neural message passing techniques to effectively model the joint prediction of multiple labels.
- MPVAE (Bai, Kong, and Gomes 2020): MPVAE adopts a variational autoencoder framework to learn and align probabilistic embedding spaces for both labels and features.
- ASL (Ridnik et al. 2021): ASL introduces an innovative asymmetric loss function that operates differently on positive and negative samples, contributing to improved performance.
- RBCC (Gerych et al. 2021): RBCC focuses on learning a Bayesian network of class dependencies.
- C-GMVAE (Bai, Kong, and Gomes 2022): C-GMVAE combines the strengths of Gaussian mixture variational autoencoders and contrastive learning.

Comparison with existing approaches We fine-tune the pretrained model on the same dataset. The results presented

[1] http://mulan.sourceforge.net/datasets-mlc.html

Metric	example-F1						micro-F1					
Dataset	media	yeast	scene	bkms	deli	nus	media	yeast	scene	bkms	deli	nus
LaMP	-	0.624	0.728	0.389	0.372	0.376	-	0.641	0.716	0.373	0.386	0.472
MPVAE	-	0.648	0.751	0.382	0.373	0.468	-	0.655	0.742	0.375	0.393	0.492
ASL	-	0.613	0.770	0.373	0.359	0.468	-	0.637	0.753	0.354	0.387	0.495
RBCC	-	0.605	0.758	-	-	0.466	-	0.623	0.749	-	-	0.490
C-GMVAE	$0.623^{\dagger}$	0.656	0.777	0.392	0.381	0.481	$0.626^{\dagger}$	0.665	0.762	0.377	**0.403**	0.510
Ours	**0.627**	**0.659**	**0.787**	**0.394**	**0.386**	**0.484**	**0.630**	**0.667**	**0.773**	**0.386**	0.399	**0.511**

Table 2: Comparison with existing approaches, The example-F1 and micro-F1 scores are shown, † means we run the official C-GMVAE codes with our data splits

Metric	macro-F1						Hamming Accuracy					
Dataset	media	yeast	scene	bkms	deli	nus	media	yeast	scene	bkms	deli	nus
LaMP	-	0.480	0.745	0.286	0.196	0.203	-	0.786	0.903	0.992	0.982	0.980
MPVAE	-	0.482	0.750	0.285	0.181	0.211	-	0.792	0.909	0.991	0.982	0.980
ASL	-	0.484	0.765	0.264	0.183	0.208	-	0.796	0.912	0.991	0.982	0.975
RBCC	-	0.480	0.753	-	-	0.202	-	0.793	0.904	-	-	0.975
C-GMVAE	$0.273^{\dagger}$	**0.487**	0.769	0.291	**0.197**	**0.226**	$0.970^{\dagger}$	0.796	0.915	0.992	0.983	**0.986**
Ours	**0.316**	0.475	**0.776**	**0.294**	0.185	0.220	**0.971**	**0.799**	**0.922**	0.992	0.983	0.976

Table 3: An extension to Table 2, where the macro-F1 and hamming accuracy scores are shown.

Dataset	Method	HA	example-F1	Macro-F1	Micro-F1	precision@1
MS-COCO	C-GMVAE	0.980	0.710	0.638	0.686	0.915
	Ours	0.980	**0.715**	**0.640**	**0.687**	**0.920**
PASCAL	C-GMVAE	**0.978**	0.835	**0.811**	0.832	**0.921**
	Ours	0.977	**0.841**	0.809	**0.833**	0.913
MIRFLICKR	C-GMVAE	0.943	0.744	0.680	0.764	0.925
	Ours	0.943	**0.746**	**0.691**	**0.765**	**0.928**

Table 4: Comparison of results on embeddings from image datasets extracted by Imagenet-pretrained ResNet-50

are directly obtained from C-GMVAE. It is important to note that we adhere to the same data split as stated in MPVAE (Bai, Kong, and Gomes 2021) to ensure a fair comparison. Specifically, the dataset is divided into training, validation, and testing sets. We pretrain our model on the training set and subsequently fine-tune it using the same training set. The reported results on the test set are obtained when the model achieves the best performance on the validation set. For a comprehensive and equitable comparison, we provide implemented results using our data split of the mediamill dataset, based on the original codes provided in C-GMVAE. These results are presented in Table 2 and Table 3. The findings indicate that our method can achieve competitive performance with C-GMVAE, with consistent enhancements observed in metrics such as example-F1.

Furthermore, we compare our method with C-GMVAE on several image datasets in Table 4. We use ResNet-50 pretrained on imagenet to extract embeddings from these datasets as the input features of our method and C-GMVAE. We present the results of finetuning the pretrained model. Note that on most metrics, MulSupCon oversees a better performance on three datasets. The enhancement might seem marginal yet C-GMVAE is already one of the state-of-the-art methods on MLC.

Linear Probe: ALL, ANY and MulSupCon on MS-COCO We first present the comparison among three methods of multi-label classification on MS-COCO in Table 5. Clearly, results show that our proposed method performs the best among three methods on all 6 metrics, indicating the effectiveness of our method.

Comparison with BCE on 5 image datasets We then present multi-label classification linear probing and fine-tuning performance on 5 image datasets in Table 6. The results reveal that, with the exception of the MS-COCO dataset, linear probing attains competitive or superior results. Moreover, fine-tuning consistently outperforms both linear probing and BCE across all image datasets and metrics. To interpret the reason why MulSupCon is better than BCE, we visualize representations of each class in Figure 3 based on NUS-WIDE dataset. The heatmap demonstrates that our method is better able to differentiate between different classes in the multi-label scenario. For each class, we select samples exclusively assigned with this class label and average these sample embeddings as the class embedding. Row i represents the softmax cosine similarity values between class i and the entire spectrum of classes.

Transfer learning: Comparison with BCE on downstream tasks To evaluate the transferring performance of our method, we validate our model pretrained on MS-COCO

Method	mAP	Hamming Accuracy	example-F1	Macro-F1	Micro-F1	precision@1
ALL	0.636	0.979	0.676	0.607	0.664	0.897
ANY	0.564	0.977	0.639	0.547	0.623	0.883
Ours	**0.672**	**0.980**	**0.700**	**0.636**	**0.688**	**0.916**

Table 5: Linear probe results of ALL, ANY, and MulSupCon (ours) method on MS-COCO dataset

Dataset	Method	mAP	HA	example-F1	Macro-F1	Micro-F1	precision@1
MS-COCO	BCE	0.694	0.981	0.724	0.654	0.706	0.921
	linear	0.672	0.980	0.700	0.636	0.688	0.916
	finetune	**0.708**	**0.982**	**0.736**	**0.662**	**0.714**	**0.930**
NUS-WIDE	BCE	0.524	0.984	0.707	0.505	0.722	0.821
	linear	0.552	0.984	0.701	0.553	0.721	0.822
	finetune	**0.566**	0.984	**0.714**	**0.557**	**0.728**	**0.825**
Objects365	BCE	0.180	0.983	0.361	0.133	0.432	0.716
	linear	0.297	0.984	0.441	0.288	0.479	0.784
	finetune	**0.333**	**0.984**	**0.467**	**0.311**	**0.509**	**0.802**
MIRFLICKR	BCE	0.696	0.921	0.710	0.641	0.731	0.889
	linear	0.694	0.920	0.705	0.637	0.727	0.888
	finetune	**0.711**	**0.926**	**0.729**	**0.658**	**0.754**	**0.895**
PASCAL	BCE	0.673	0.962	0.684	0.628	0.692	0.769
	linear	0.694	0.962	0.680	0.644	0.697	0.778
	finetune	**0.726**	**0.965**	**0.716**	**0.670**	**0.726**	**0.802**

Table 6: Comparison of linear probing and finetuning results with BCE on 5 image datasets

(a) Model pretrained on MS-COCO							
Dataset	Method	mAP	HA	example-F1	Macro-F1	Micro-F1	precision@1
PASCAL	BCE	**0.865**	**0.979**	**0.838**	**0.809**	**0.842**	**0.916**
	Ours	0.853	0.977	0.818	0.792	0.824	0.907
MIRFLICKR	BCE	0.656	0.911	0.673	0.613	0.700	0.873
	Ours	**0.722**	**0.923**	**0.716**	**0.668**	**0.741**	**0.899**
(b) Model pretrained on NUS-WIDE							
PASCAL	BCE	0.661	0.964	0.713	0.624	0.713	0.801
	Ours	**0.750**	**0.969**	**0.749**	**0.693**	**0.760**	**0.853**
MIRFLICKR	BCE	0.716	0.922	0.716	0.662	0.740	0.897
	Ours	**0.750**	**0.927**	**0.736**	**0.691**	**0.758**	**0.906**

Table 7: Transfer to PASCAL and MIRFLICKR datasets.

Linear probing results with and without the weight in MulSupCon							
Dataset	Weight	mAP	HA	example-F1	Macro-F1	Micro-F1	precision@1
MS-COCO	Yes	0.663	0.980	0.697	0.630	0.683	0.914
	No	**0.672**	0.980	**0.700**	**0.636**	**0.688**	**0.916**
PASCAL	Yes	0.693	0.962	0.676	0.640	0.695	0.772
	No	**0.694**	0.962	**0.680**	**0.644**	**0.697**	**0.778**
MIRFLICKR	Yes	0.689	0.919	0.703	0.634	0.725	0.884
	No	**0.694**	**0.920**	**0.705**	**0.637**	**0.727**	**0.888**

Table 8: Ablation study results

and NUS-WIDE on two downstream datasets: PASCAL and MIRFLICKR. We use an Adam optimizer with an initial learning rate of 0.04. We freeze the models and linearly probe them on other datasets. Note that in Table 6 BCE method outperform the performance of linear probing our method on MS-COCO dataset. However, the outcomes outlined in Table 7(a) establish our method's pronounced superiority over BCE on MIRFLICKR dataset. Despite the fact that directly transferring the model pretrained on the MS-COCO dataset does not yield results as favorable as those achieved by BCE on PASCAL dataset, we observe that the performance surpasses that of BCE when the pretrained model is first finetuned on MS-COCO and then transferred, achieving an mAP of 0.877. Results in Table 7(b) show that our model consistently outperforms BCE, demonstrating MulSupCon's stronger generalization ability.

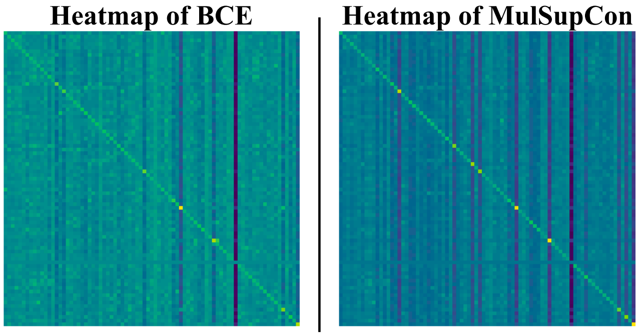

Figure 3: Heatmap of class embeddings extracted from BCE and MulSupCon

Ablation Study

To further evaluate the MulSupCon loss function, we conducted ablation study.

Loss function We explore an alternative form of the loss function, which incorporates an additional weight $\frac{1}{|\boldsymbol{y}^{(i)}|}$ for each sample, as illustrated in Equation (6). In this part, we conduct experiments on the MS-COCO, PASCAL, and MIRFLICKR datasets to compare the performance difference between using the loss function with or without the weight. The results presented in Table 8 indicate that not incorporating the weight leads to better performance.

Related Works

Multi-label Classification A simple strategy to solve MLC problem is to treat one sample's labels as several independent labels, which is also the idea of the Binary-Cross-Entropy loss. This strategy ignores the the label dependencies in an image, where (Wang et al. 2016) utilizes RNN along with CNN to address the problem. (Chen et al. 2019) uses Graph Convolutional Network (GCN) to learn label representation by treating label as the node represented by the label's word embedding. In (Wang et al. 2017), they utilize a spatial transformer layer as well as an LSTM to learn correlation between labels.

To utilize contrastive learning in the MLC problem, MulCon (Dao et al. 2021) propose to learn multiple label-level representations from each sample; that is, for each label in the dataset, they extract one separate representation. For one sample, the representation which represents the class it belongs to is treated as the anchor, while its positives are label-level representations from other samples belonging to the same class. They utilize BCE loss along with contrastive loss. C-GMVAE (Bai, Kong, and Gomes 2022) solves the problem by learning a latent embedding space shared by features and labels, which is similar to multi-modal contrastive learning. They map data and labels to the same probabilistic embedding space and conduct contrastive learning using the loss proposed in SupCon. They treat a sample's feature as the anchor and embeddings from labels it belongs to as its positives.

Contrastive Learning Recently, representation learning has achieved great success in a variety of research and gained increasing attention, including contrastive representation learning. SimCLR (Chen et al. 2020) advises hard data augmentation and more negative samples for effective representation learning. MoCo (He et al. 2020) uses a queue for contrastive learning with stored negative samples. SupCon (Khosla et al. 2020) improves single-label classification using a modified contrastive loss, allowing multiple positives for one anchor. By incorporating label information, they include all samples with the same label with the anchor sample as the positives and achieve consistent improvement on various classification tasks.

Different from the above works, we neither extract label-level representations for each sample nor align representations of labels and samples. Instead, we use one representation for each sample and mainly consider the correlation between samples, similar to the idea of SimCLR or SupCon. We use the label information in an implicit way, that is, we do not bind one sample to its labels but correlate it to the samples which share some common labels.

Conclusion

MLC is an interesting problem for applying a contrastive learning paradigm since there are a plethora of ways to determine the relations between two multi-label samples. In the current work, we proposed MulSupCon, which provides a finer way to weigh one sample's relation to the anchor based on the proportional label overlap. We compared with two other intuitive views (ALL and ANY) and the commonly-adopted multi-label BCE loss function on a series of datasets with linear probing, finetuning, and transfer learning. Robust results evaluated with six metrics suggest the effectiveness of MulSupCon. Additionally, our method demonstrates competitive performance when compared to existing approaches in the field. Our ablation study further explores the weighing mechanism in MulSupCon and its influence to the final performance.

Acknowledgements

This work has been supported by National Natural Science Foundation of China (No.92048205), the Key Research and Development Program of Jiangsu Province, China (No.BE2022059), and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102). Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

References

Bai, J.; Kong, S.; and Gomes, C. 2020. Disentangled variational autoencoder based multi-label classification with covariance-aware multivariate probit model. *arXiv preprint arXiv:2007.06126*.

Bai, J.; Kong, S.; and Gomes, C. 2021. Disentangled variational autoencoder based multi-label classification with covariance-aware multivariate probit model. In *Proceedings of the Twenty-Ninth International Conference on Inter-*

national Joint Conferences on Artificial Intelligence, 4313–4321.

Bai, J.; Kong, S.; and Gomes, C. P. 2022. Gaussian mixture variational autoencoder with contrastive learning for multi-label classification. In *International Conference on Machine Learning*, 1383–1398. PMLR.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Chen, Z.-M.; Wei, X.-S.; Wang, P.; and Guo, Y. 2019. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5177–5186.

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM international conference on image and video retrieval*, 1–9.

Dao, S. D.; Zhao, E.; Phung, D.; and Cai, J. 2021. Multi-label image classification with contrastive learning. *arXiv preprint arXiv:2107.11626*.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2007. The pascal visual object classes challenge 2007 results.

Gemmeke, J. F.; Ellis, D. P.; Freedman, D.; Jansen, A.; Lawrence, W.; Moore, R. C.; Plakal, M.; and Ritter, M. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 776–780. IEEE.

Gerych, W.; Hartvigsen, T.; Buquicchio, L.; Agu, E.; and Rundensteiner, E. A. 2021. Recurrent bayesian classifier chains for exact multi-label classification. *Advances in Neural Information Processing Systems*, 34: 15981–15992.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

Huiskes, M. J.; and Lew, M. S. 2008. The mir flickr retrieval evaluation. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, 39–43.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33: 18661–18673.

Lanchantin, J.; Sekhon, A.; and Qi, Y. 2020. Neural message passing for multi-label classification. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part II*, 138–163. Springer.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Ma, Q.; Yuan, C.; Zhou, W.; and Hu, S. 2021. Label-specific dual graph neural network for multi-label text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 3855–3864.

Małkiński, M.; and Mańdziuk, J. 2022. Multi-label contrastive learning for abstract visual reasoning. *IEEE Transactions on Neural Networks and Learning Systems*.

Ridnik, T.; Ben-Baruch, E.; Zamir, N.; Noy, A.; Friedman, I.; Protter, M.; and Zelnik-Manor, L. 2021. Asymmetric loss for multi-label classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 82–91.

Shao, S.; Li, Z.; Zhang, T.; Peng, C.; Yu, G.; Zhang, X.; Li, J.; and Sun, J. 2019. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 8430–8439.

Wang, J.; Yang, Y.; Mao, J.; Huang, Z.; Huang, C.; and Xu, W. 2016. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2285–2294.

Wang, Z.; Chen, T.; Li, G.; Xu, R.; and Lin, L. 2017. Multi-label image recognition by recurrently discovering attentional regions. In *Proceedings of the IEEE international conference on computer vision*, 464–472.

Yang, P.; Sun, X.; Li, W.; Ma, S.; Wu, W.; and Wang, H. 2018. SGM: sequence generation model for multi-label classification. *arXiv preprint arXiv:1806.04822*.

Zhang, S.; Xu, R.; Xiong, C.; and Ramaiah, C. 2022. Use all the labels: A hierarchical multi-label contrastive learning framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16660–16669.

Zhang, X.; Zhang, Q.-W.; Yan, Z.; Liu, R.; and Cao, Y. 2021. Enhancing label correlation feedback in multi-label text classification via multi-task learning. *arXiv preprint arXiv:2106.03103*.

Zhou, Y.; Kang, X.; and Ren, F. 2022. Employing Contrastive Strategies for Multi-label Textual Emotion Recognition. In *Intelligent Information Processing XI: 12th IFIP TC 12 International Conference, IIP 2022, Qingdao, China, May 27–30, 2022, Proceedings*, 299–310. Springer.