Exploiting Negative Multi-Cluster Structure in Class-Wise Embeddings for Weakly Supervised Multi-Label Learning

Bo Han[1], **Zhuoming Li**[1], **Xiaoyu Wang**[2], **Yaxin Hou**[1]
Hui Liu[4], **Junhui Hou**[5], **Yuheng Jia**[1,3,4*]
[1]School of Computer Science and Engineering, Southeast University, Nanjing, China
[2]School of Software Engineering, Southeast University, Nanjing, China
[3]Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China
[4]School of Computing Information Sciences, Saint Francis University, Hong Kong, China
[5]Department of Computer Science, City University of Hong Kong, Hong Kong, China
hanbo@seu.edu.cn, leezhuoming@seu.edu.cn, wang_xy@seu.edu.cn, yaxin@seu.edu.cn, yhjia@seu.edu.cn, hliu99-c@my.cityu.edu.hk, jh.hou@cityu.edu.hk

Abstract

Class-wise embeddings provide an effective strategy for multi-label learning (MLL) by capturing the distinct discriminative properties of each class. However, existing methods based on class-wise embeddings **typically overlook the internal distribution of negative samples**. We systematically characterize an underexplored manifestation of negative-sample heterogeneity in class-wise embedding spaces: negative instances organize into multiple semantically coherent sub-clusters, and this structure persists across modalities even under extremely sparse supervision. Leveraging this insight, we propose a Structure-Aware Pseudo-label mining framework for weakly supervised MLL (SaPu). The proposed framework mines positive pseudo-labels by aggregating cluster-level consistency across different class-wise embeddings and identifies negative pseudo-labels via neighborhood exclusion. These high-quality signals further guide class-wise contrastive learning, establishing a self-reinforcing loop that iteratively refines embeddings. Extensive experiments on image, text and audio benchmarks validate that SaPu outperforms state-of-the-art methods by an average of **4.46%** in challenging single-label supervision scenarios.

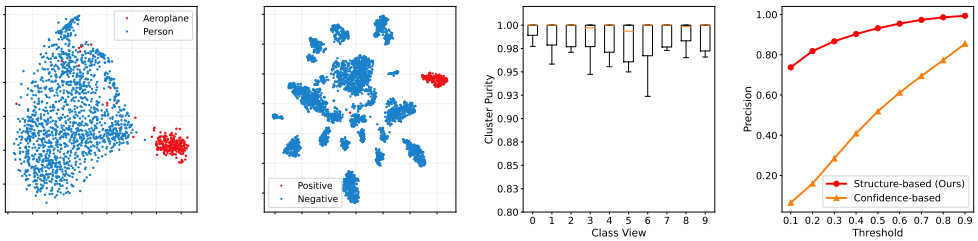
1 Introduction

Multi-label learning (MLL) addresses prediction problems where each instance can be associated with multiple labels, enabling expressive modeling in complex real-world scenarios [1]. This setting arises naturally in a wide range of applications, including image annotation [2], text classification [3], and facial expression recognition [4].

A common practice in MLL is to learn a single shared embedding for each instance and use it to predict all labels; however, this shared representation fails to capture class-specific discriminative features. To alleviate this issue, class-wise embeddings, which generates a label-conditioned embedding for each class, have become a widely adopted strategy in MLL [5, 6, 7]. For instance, LIFT [7] explored label-specific representations via clustering-based analysis of positive and negative instances and prototype-driven feature transformations. More recently, with the strong representation capacity of

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) (b) (c) (d)

Figure 1: (a) t-SNE visualization [10] of a standard binary classification problem (e.g., *aeroplane* (positive) vs. *person* (negative)). (b) t-SNE of class-wise embeddings for class 0 on VOC2007, where negatives form multiple compact sub-clusters. (c) Box plots of cluster purity on VOC2007 across multiple classes, computed as the fraction of samples in each cluster sharing the same semantic label. (d) Precision of positive pseudo-labels: structure-based vs. confidence-based baselines.

deep networks, class-wise embeddings are commonly constructed in an end-to-end manner [5, 6, 8, 9]. It is worth noting that most existing approaches perform binary classification on the corresponding class-wise embedding, where instances annotated with the label are treated as positives and the remaining ones as negatives. This practice implicitly assumes a standard binary separation in the embedding space, where samples form two clusters corresponding to positives and negatives, similar to conventional binary classification (e.g., *aeroplane* vs. *person*), as illustrated in Fig. 1(a).

However, we observe that the two-cluster assumption does not hold in practice for the class-wise embedding of MLL. Empirical results suggest that negative class-wise embeddings form multiple compact sub-clusters rather than a single aggregate cluster (Fig. 1(b)). Moreover, by clustering these embeddings and measuring cluster purity, we find that a high fraction of samples within each sub-cluster share at least one common positive label as shown in Fig. 1(c), indicating clear semantic coherence. Detailed investigation of this phenomenon are provided in Section 3.

Our experiments further indicate that the observed negative multi-cluster structure is not specific to a particular dataset or model configuration, but consistently emerges in class-wise embedding spaces under multi-label learning. Specifically, we observe it **consistently across diverse multi-label modalities**, including images, text, and audio (See Figs. 3(a)-3(b)). Meanwhile, **the phenomenon remains pronounced even when the model is trained with extremely sparse supervision**: Cluster purity is high when only one label is observed per instance (Figs. 3(c)-3(d)). This stability makes negative multi-cluster structure particularly attractive for weak supervision. We therefore consider a structure-based pseudo-labeling strategy that leverages the intrinsic geometric organization of class-wise embedding spaces. As shown in Fig. 1(d), pseudo-labels mined from this structure are more accurate than those based on prediction confidence.

Motivated by these findings, we propose a **S**tructure-**A**ware **P**se**U**do-label Mining Framework for weakly supervised MLL (**SaPu**), designed to explicitly mine the local consistency structures within the class-wise embedding space. Specifically, we first leverage naturally emerging local clusters to mine high-confidence positive pseudo-labels based on intra-cluster label consistency. Building upon this, we further utilize multi-view neighborhood exclusivity to identify reliable negative pseudo-labels. Finally, to better utilize the mined supervisory signals, we integrate class-wise supervised contrastive learning, which yields more discriminative embeddings and further strengthens the above structure-based pseudo-label mining. Extensive experiments show that our approach consistently outperforms existing state-of-the-art approaches under two prominent weakly supervised settings: multi-label learning with missing labels and semi-supervised multi-label learning.

Our main contributions are summarized as follows.

- To the best of our knowledge, we are the first to identify the negative multi-cluster structure phenomenon inherent in the class-wise embedding space of multi-label learning and to analyze the mechanism behind the phenomenon. We also empirically demonstrate that this phenomenon is universal across modalities and remains robust even under extremely sparse supervision.
- Building on this structural insight, we propose **SaPu**, a structure-aware pseudo-label mining framework for weakly supervised MLL. SaPu mines high-confidence positive pseudo-labels via cross-class cluster consistency and discovers reliable negative pseudo-labels through neighborhood exclusion.

- We further use the mined pseudo-labels to supervise class-wise contrastive learning, thereby enhancing the discriminability of class-wise embeddings, which in turn improves the quality of pseudo-label mining.
- Extensive experiments on image, text, and audio benchmarks validate the effectiveness of SaPu. When only one label is observed per instance, our method outperforms state-of-the-art approaches by an average of **4.46%**.

2 Related Work

Multi-label learning (MLL) studies prediction problems where each instance can be associated with multiple labels. Early approaches [11, 12, 13] typically learn a shared instance representation and train per-label classifiers on the same feature space, which may limit their performance by failing to capture label-specific discriminative cues.

To better capture label-specific characteristics, class-wise embedding based methods [5, 6, 9, 14, 15] that learn a discriminative embedding for each label have received growing attention in multi-label learning. Liu et al. proposed Query2Label, which treats each label embedding as a semantic query in a Transformer decoder and uses cross-attention to adaptively pool category-related features for per-label prediction [5]. Ridnik et al. introduced an efficient decoder-style attention head that removes redundant self-attention and further adopts group-decoding to scale query-based label embeddings to large label spaces [6]. Zhu et al. replaced quadratic-cost attention with a Mamba-based decoder that uses label embeddings as queries to gather label-specific visual representations for subsequent binary classification [9].

Despite their success, **the above class-wise embedding methods typically assume that negative samples for each label collapse into a single cluster in the embedding space; however, we empirically find that the negatives instead exhibit a multi-cluster structure.** In the following sections, we analyze the underlying causes of the structure and propose a novel method to exploit this structure. (More related literature and discussions about weakly supervised multi-label learning are provided in Appendix A)

3 Analysis of the Negative Multi-cluster Structure

In this section, we first review the construction of class-wise embeddings. Building on this formulation, we provide an explanation for the negative multi-cluster structure phenomenon in multi-label learning.

3.1 Recap of Class-wise Embedding Generation

Most existing approaches based on class-wise embeddings adopt a Transformer-based architecture. We recap the typical pipeline using multi-label image classification as an example (illustrated in Fig. 2(a)).

Given an input image $\mathbf{X}$, a visual backbone (e.g., ResNet) first extracts high-level semantic features, yielding a feature map:

$$\mathbf{E} = f_{\text{backbone}}(\mathbf{X}), \quad \mathbf{E} \in \mathbb{R}^{H \times W \times D}, \tag{1}$$

where H and W denote the spatial resolution and D is the channel dimension. We then flatten the feature map $\mathbf{E}$ into a set of local feature vectors:

$$\begin{aligned} \mathcal{V} &= \text{Flatten}(\mathbf{E}) \\ &= \{\mathbf{v}_i \mid \mathbf{v}_i \in \mathbb{R}^D,\ i = 1, \ldots, HW\}. \end{aligned} \tag{2}$$

For each category $c \in \{1, \ldots, C\}$, a corresponding class query vector is introduced:

$$\mathbf{q}_c \in \mathbb{R}^D, \tag{3}$$

which can be a learnable parameter [5, 6].

Next, an attention mechanism computes the relevance between the class query and each spatial feature, producing attention weights for label c:

$$\alpha_{c,i} = \frac{\exp(\mathbf{q}_c^\top \mathbf{v}_i)}{\sum_{j=1}^{HW} \exp(\mathbf{q}_c^\top \mathbf{v}_j)}, \quad i = 1, \ldots, HW. \tag{4}$$

Here, $\boldsymbol{\alpha}_c = \{\alpha_{c,i}\}_{i=1}^{HW} \in \mathbb{R}^{HW}$ characterizes the spatial importance distribution for category c, which can be reshaped into an attention map $\mathbf{W}_c \in \mathbb{R}^{H\times W}$ in Fig. 2.

Finally, the class-wise embedding $\mathbf{z}^{(c)}$ is obtained through a weighted aggregation of the local features:

$$\mathbf{z}^{(c)} = \sum_{i=1}^{HW} \alpha_{c,i}\mathbf{v}_i, \quad \mathbf{z}^{(c)} \in \mathbb{R}^D. \tag{5}$$

3.2 Mechanism Behind the Negative Multi-cluster Structure

As discussed in Section 3.1, class-wise embeddings are computed via cross-attention between a class query and the feature map. Crucially, the resulting attention patterns differ substantially between positive and negative labels, which naturally leads to the observed multi-cluster structure of negative embeddings.

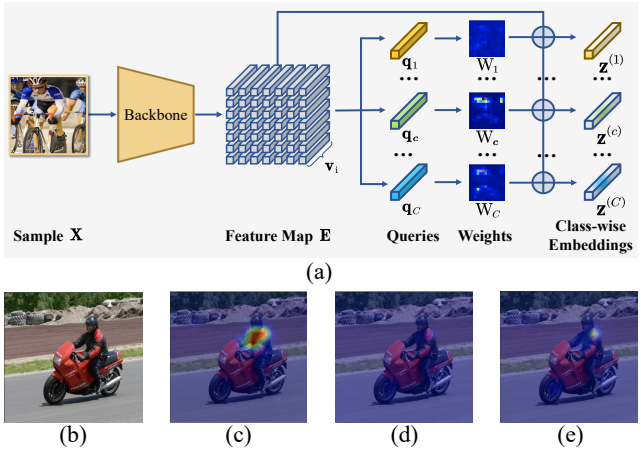


Figure 2: (a) Pipeline for constructing class-wise embeddings. (b) An example image annotated with labels *person* and *motorbike*. (c)–(e) Attention maps produced by the class queries for *person*, *bird*, and *sofa*, respectively.

Positive-label case. When an image contains category c, the query $\mathbf{q}_c$ can typically localize discriminative regions in the feature map that are semantically aligned with label c. In this case, the attention weights exhibit a pronounced peaky pattern, assigning large mass to a small number of spatial locations (Fig. 2(c)). The resulting embedding $\mathbf{z}^{(c)}$ is therefore strongly label-conditioned.

Negative-label case. In contrast, when an image does not contain category c, the query has no clear semantic match in the input feature map and cannot retrieve label-specific content. As a result, the attention weights become much flatter and less focused (Fig. 2(d)). In this case, the weighted aggregation in Eq. (5) behaves similarly to a global pooling operation. The resulting embedding is dominated by instance-level semantics rather than a unified "negative concept". Since instances can differ widely in their semantic content, two samples may both be negative for the same label while exhibiting very different instance semantics. Consequently, negative embeddings do not collapse into a single cluster; instead, they naturally organize into multiple sub-clusters aligned with latent semantic of instances.

Meanwhile, even when an image does not contain category c, the query can still induce weak but consistent preferences due to semantic similarity or visual co-occurrence with other categories (Fig. 2(e)). This query-induced bias makes negative embeddings not strictly equivalent to naive global pooling and can vary across label queries. Therefore, for the same instance, treating different labels as negative may yield different class-wise embeddings. As a result, the negative multi-cluster structure can differ across embedding spaces of different categories. Additional evidence is provided in Appendix B.

4 Proposed Method

Motivation. The analysis above indicates that class-wise embedding spaces exhibit a clear local-consistency structure: nearby instances in the representation space tend to share the same positive label. Moreover, we find that such local consistency remains evident even under weak supervision (as will be shown in Figs. 3(c)-3(d)). Motivated by these findings, we propose a method for weakly supervised multi-label learning.

4.1 Problem Formulation

We first formalize the problem and introduce the notation. Let $x \in \mathcal{X}$ denote an input instance and $y \in \mathcal{Y}$ denote its multi-label annotation. The input space is $\mathcal{X} \subseteq \mathbb{R}^d$, and the label space is $\mathcal{Y} = \{-1, 0, 1\}^C$ with C categories. For an instance x_i, $y_{ic} = 1$ indicates that label c is relevant, $y_{ic} = -1$ indicates irrelevance, and $y_{ic} = 0$ denotes an unknown label. We denote the label matrix of a dataset by $\mathbf{Y} \in \{-1, 0, 1\}^{N \times C}$.

Multi-label learning with partial labels (MLL-PL). In the MLL-PL setting, the training data contain N instances $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ with $y_i \in \{-1, 0, 1\}^C$. For each instance, labels are only partially observed: some entries are annotated as relevant or irrelevant (1 or -1), while the rest are missing and recorded as 0.

Semi-supervised multi-label learning (SSMLL). In SSMLL, the training data consist of a labeled set $\mathcal{D}_l = \{(x_i, y_i)\}_{i=1}^{N_l}$ and an unlabeled set $\mathcal{D}_u = \{x_j\}_{j=1}^{N_u}$, where $N_l \ll N_u$. Each labeled instance is fully annotated, while for each unlabeled instance $x_j \in \mathcal{D}_u$, we set its label vector to $\{0\}^C$ (i.e., $y_{jc} = 0$ for all c).

4.2 Mining Positive Pseudo-labels from Cluster Consistency

As shown in Fig. 1(c), instances within the same local cluster tend to share the same positive label sets. We exploit this property to design a simple yet effective strategy for positive pseudo-label mining.

Specifically, for a given view c (i.e., the embedding space associated with class c), we collect class-wise embeddings of all instances $\{\mathbf{z}_i^{(c)}\}_{i=1}^N$ and partition them into K mini-clusters $\{\mathcal{C}_k^{(c)}\}_{k=1}^K$, where $K = \alpha C$ and $\alpha > 1$ is a hyper-parameter. For each cluster $\mathcal{C}_k^{(c)}$ in view c, we compute a label-wise vote vector by counting observed positives within the cluster. Specifically, for any label t, the vote count is

$$v_{k,t}^{(c)} = \sum_{i \in \mathcal{C}_k^{(c)}} \mathbb{I}(y_{it} = 1), \tag{6}$$

where $v_{k,t}^{(c)}$ denotes the number of instances in the k-th cluster (under view c) that are annotated as positive for label t. Note that we only count observed positives. Intuitively, if a cluster corresponds to a stable semantic pattern, its vote distribution will concentrate on a small subset of labels. We therefore select the top-voted label

$$\hat{t} = \arg \max_{t \in \{1, \ldots, C\}} v_{k,t}^{(c)}, \tag{7}$$

and assign a positive pseudo-label to all instances in the cluster whose annotation for $\hat{t}$ is unknown, i.e., for those with $y_{i\hat{t}} = 0$ we set $\tilde{y}_{i\hat{t}}^{(c)} = 1$. This yields a view-specific pseudo-label matrix $\tilde{\mathbf{Y}}_{pos}^{(c)} \in \{0, 1\}^{N \times C}$.

Relying on a single view can be sensitive to attention bias and label similarity, potentially introducing noise. We thus repeat the above procedure for all C views and aggregate the resulting pseudo-labels. Specifically, we define an agreement matrix

$$\mathbf{M} = \frac{1}{C} \sum_{c=1}^{C} \tilde{\mathbf{Y}}_{pos}^{(c)}, \tag{8}$$

where a larger M_{it} indicates that instance i is more consistently predicted as positive for label t across views. Finally, we select high-confidence positive pseudo-labels via thresholding:

$$\tilde{y}_{it} = \begin{cases} 1, & M_{it} \geq \tau, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

where τ is a threshold. As shown in Section 5.4, even with a relatively small threshold, the mined positive pseudo-labels achieve high precision.

4.3 Negative Pseudo-labels Mining via Multi-view Neighborhood Consistency

Motivated by the preceding analysis, we further infer that if all samples in an instance's local neighborhood are negative for label t, then the instance itself is likely negative for t. Based on this idea, we propose a kNN-consistency module for mining reliable negative pseudo-labels.

Let $\mathcal{N}_k^{(c)}(i)$ denote the k nearest neighbors of instance i in view c. For any label t, if none of the neighbors of i is predicted as positive for t, i.e.,

$$\sum_{j \in \mathcal{N}_k^{(c)}(i)} \mathbb{I}\big(\tilde{y}_{jt} = 1\big) = 0, \tag{10}$$

we regard label t as absent from instance i under view c. Note that in each view we perform this check for all labels $t \in \{1, \ldots, C\}$, rather than only for the label associated with the view. Accordingly, we form a view-specific negative pseudo-label matrix $\tilde{\mathbf{Y}}_{\text{neg}}^{(c)} \in \{0, -1\}^{N \times C}$, where $\tilde{y}_{it,\text{neg}}^{(c)} = -1$ indicates that instance i is assigned a negative pseudo-label for class t under view c (and 0 otherwise).

Similar to positive pseudo-label mining, we combine the view-specific negative predictions to improve reliability. Since false negatives can be particularly harmful in multi-label learning [16, 17], we adopt a conservative rule and assign a negative pseudo-label to (i, t) only when all views agree:

$$\tilde{y}_{it} = -1, \quad \text{if} \quad \tilde{y}_{it,\text{neg}}^{(c)} = -1 \ \ \forall c \in \{1, \ldots, C\}. \tag{11}$$

Finally, we collect the mined positive pseudo-label set $\tilde{\mathbf{Y}}_{pos}$ and the mined negative pseudo-label set $\tilde{\mathbf{Y}}_{\text{neg}}$, and merge them into a pseudo-label matrix $\hat{\mathbf{Y}} \in \{-1, 0, 1\}^{N \times C}$. We analyze the effect of relaxing the all-view agreement constraint in Appendix F.7.

4.4 Class-wise Supervised Contrastive Representation Learning

Combining $\hat{\mathbf{Y}}$ with the observed labels $\mathbf{Y}$, we obtain an expanded supervision set denoted by $\mathbf{Y}^+$. To further enhance representation quality, we introduce **class-wise supervised contrastive learning**. Unlike conventional instance-level contrastive learning, we impose supervised contrastive constraints directly on class-wise embeddings.

Specifically, for each label view c, we use the expanded supervision $\mathbf{Y}^+$ and restrict contrastive learning to instances whose labels on c are available, i.e., $y_{ic}^+ \in \{1, -1\}$. Let $\mathcal{K}_c = \{i \mid y_{ic}^+ \in \{1, -1\}\}$ denote the index set under view c. We treat instances with $y_{ic}^+ = 1$ as positives and those with $y_{ic}^+ = -1$ as negatives, and use only positives as anchors. For an anchor i with $y_{ic}^+ = 1$, its positive set is

$$\mathcal{P}_c(i) = \{p \in \mathcal{K}_c \setminus \{i\} \mid y_{pc}^+ = 1\}. \tag{12}$$

Accordingly, for each positive anchor i in view c, we minimize the supervised contrastive loss:

$$\ell_i^{(c)} = -\frac{1}{|\mathcal{P}_c(i)|} \sum_{p \in \mathcal{P}_c(i)} \log \frac{\exp(s_{i,p}^c/\tau)}{\sum_{a \in \mathcal{K}_c \setminus \{i\}} \exp(s_{i,a}^c/\tau)}, \tag{13}$$

where $s_{i,a}^c = \text{sim}\Big(\mathbf{z}_i^{(c)}, \mathbf{z}_a^{(c)}\Big)$ denotes cosine similarity and τ is the temperature parameter. The overall contrastive objective is obtained by summing over label views and positive anchors:

$$\mathcal{L}_{\text{con}} = \sum_{c=1}^{C} \sum_{\substack{i \in \mathcal{K}_c \\ y_{ic}^+ = 1}} \ell_i^{(c)}. \tag{14}$$

4.5 Overall Self-enhancement Framework

Following [8, 18, 19], we adopt a two-stage training strategy. We first perform a warm-up stage using only the observed labels. The labeled loss is

$$\mathcal{L}_l = \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbb{I}(y_{ic} \neq 0)\, \ell(p_{ic}, y_{ic}), \tag{15}$$

Table 1: Comparison with state-of-the-art methods on three image benchmarks under the single-label setting. We report the mean ± standard deviation over three seeds. Best results are in bold, and second-best results are underlined.

Method	VOC07			VISPR			COCO			Average
	mAP↑	OF1↑	CF1↑	mAP↑	OF1↑	CF1↑	mAP↑	OF1↑	CF1↑	
SARB	77.97$_{\pm 1.7}$	73.73$_{\pm 0.9}$	66.80$_{\pm 2.3}$	24.87$_{\pm 1.4}$	68.87$_{\pm 1.0}$	19.93$_{\pm 1.8}$	43.00$_{\pm 3.8}$	51.17$_{\pm 1.8}$	36.03$_{\pm 3.5}$	51.37
SST	66.50$_{\pm 13.8}$	71.40$_{\pm 6.9}$	60.70$_{\pm 14.1}$	24.97$_{\pm 0.9}$	68.40$_{\pm 0.4}$	19.97$_{\pm 1.3}$	39.40$_{\pm 1.3}$	50.80$_{\pm 1.6}$	36.00$_{\pm 4.6}$	48.68
selective	63.42$_{\pm 1.2}$	34.10$_{\pm 1.7}$	24.00$_{\pm 1.7}$	24.88$_{\pm 0.8}$	2.13$_{\pm 1.0}$	0.57$_{\pm 0.3}$	45.91$_{\pm 1.4}$	3.83$_{\pm 0.9}$	5.33$_{\pm 1.8}$	22.69
LL-R*	61.06$_{\pm 0.4}$	57.97$_{\pm 1.7}$	44.90$_{\pm 1.8}$	25.03$_{\pm 0.6}$	58.50$_{\pm 3.6}$	18.63$_{\pm 1.9}$	50.49$_{\pm 0.8}$	53.33$_{\pm 1.7}$	41.73$_{\pm 1.8}$	45.74
LL-Ct*	56.03$_{\pm 2.1}$	57.10$_{\pm 1.9}$	35.60$_{\pm 3.7}$	23.66$_{\pm 1.4}$	33.80$_{\pm 7.9}$	6.13$_{\pm 2.0}$	36.96$_{\pm 1.2}$	28.10$_{\pm 0.2}$	3.43$_{\pm 0.8}$	31.20
LL-Cp*	54.89$_{\pm 2.0}$	50.77$_{\pm 0.6}$	32.13$_{\pm 2.8}$	24.31$_{\pm 0.8}$	28.77$_{\pm 3.4}$	6.73$_{\pm 0.9}$	49.11$_{\pm 1.1}$	47.60$_{\pm 0.2}$	42.73$_{\pm 0.8}$	37.45
GR	64.50$_{\pm 3.2}$	32.31$_{\pm 1.6}$	22.17$_{\pm 0.9}$	21.18$_{\pm 0.7}$	15.20$_{\pm 0.7}$	15.60$_{\pm 0.9}$	26.13$_{\pm 4.4}$	9.69$_{\pm 2.3}$	8.76$_{\pm 1.8}$	23.95
IGNORE	72.28$_{\pm 0.3}$	46.97$_{\pm 6.9}$	42.57$_{\pm 5.4}$	29.32$_{\pm 0.8}$	58.73$_{\pm 8.0}$	21.50$_{\pm 0.4}$	52.07$_{\pm 0.6}$	54.13$_{\pm 1.1}$	46.60$_{\pm 2.4}$	47.13
DSRB	77.43$_{\pm 0.8}$	74.97$_{\pm 1.4}$	67.77$_{\pm 2.3}$	24.53$_{\pm 1.0}$	67.57$_{\pm 1.3}$	22.47$_{\pm 1.0}$	43.03$_{\pm 5.4}$	51.20$_{\pm 1.7}$	37.30$_{\pm 3.4}$	51.81
HST	75.63$_{\pm 2.2}$	75.67$_{\pm 1.8}$	68.87$_{\pm 2.4}$	21.53$_{\pm 2.4}$	67.63$_{\pm 1.4}$	16.47$_{\pm 2.1}$	47.17$_{\pm 1.7}$	54.63$_{\pm 1.7}$	45.17$_{\pm 4.8}$	52.53
LC	56.71$_{\pm 0.7}$	13.97$_{\pm 1.3}$	11.43$_{\pm 1.2}$	21.39$_{\pm 0.3}$	11.60$_{\pm 1.7}$	5.07$_{\pm 0.2}$	41.31$_{\pm 0.7}$	3.00$_{\pm 0.4}$	3.43$_{\pm 0.6}$	18.66
LogicMix	78.02$_{\pm 1.2}$	69.05$_{\pm 2.4}$	66.10$_{\pm 1.6}$	28.55$_{\pm 0.3}$	54.87$_{\pm 1.6}$	27.37$_{\pm 0.7}$	59.62$_{\pm 0.3}$	40.70$_{\pm 1.4}$	36.50$_{\pm 1.0}$	51.20
DiCaP	80.07$_{\pm 1.1}$	59.47$_{\pm 5.2}$	68.37$_{\pm 2.3}$	29.72$_{\pm 0.4}$	64.10$_{\pm 2.0}$	30.23$_{\pm 0.3}$	64.61$_{\pm 0.3}$	58.53$_{\pm 0.7}$	56.00$_{\pm 0.4}$	56.79
Ours	**84.73**$_{\pm 0.5}$	**76.43**$_{\pm 1.1}$	**73.30**$_{\pm 0.4}$	**30.39**$_{\pm 0.4}$	**69.43**$_{\pm 0.7}$	**31.20**$_{\pm 0.6}$	**65.70**$_{\pm 0.2}$	**61.53**$_{\pm 0.5}$	**58.50**$_{\pm 0.5}$	**61.25**

where we use Asymmetric Loss [12] for $\ell(\cdot, \cdot)$.

After warm-up, we adopt an alternating optimization scheme, where pseudo-labels are updated every n epochs. At each cycle, we update the pseudo-label matrix $\hat{\mathbf{Y}}$ and combine it with the observed labels $\mathbf{Y}$ to form an expanded supervision set $\mathbf{Y}^+$, which is further used to supervise class-wise representation learning and improve the embedding quality. The enhanced embeddings, in turn, lead to more reliable pseudo-labels in subsequent updates, forming a closed-loop self-training process. We empirically verify this self-reinforcing effect in Appendix F.9. During this stage, the model is trained by minimizing

$$\mathcal{L} = \mathcal{L}_l + \mathcal{L}_u + \omega \mathcal{L}_{\text{con}}, \tag{16}$$

where $\mathcal{L}_u = \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbb{I}(\hat{y}_{ic} \neq 0)\, \ell(p_{ic}, \hat{y}_{ic})$ denotes the loss on pseudo-labels, ω is a trade-off parameter, and $\mathcal{L}_{\text{con}}$ is the class-wise supervised contrastive loss defined in Section 4.4. For ease of understanding, we summarize the overall procedure in Algorithm 1 in the appendix.

5 Experiments

To validate the efficacy of our proposed method, we report our experimental results under two settings: multi-label learning with partial labels (MLL-PL) (Section 5.2) and semi-supervised multi-label learning (SSMLL) (Section 5.3). Next, we present analysis on Section 5.4.

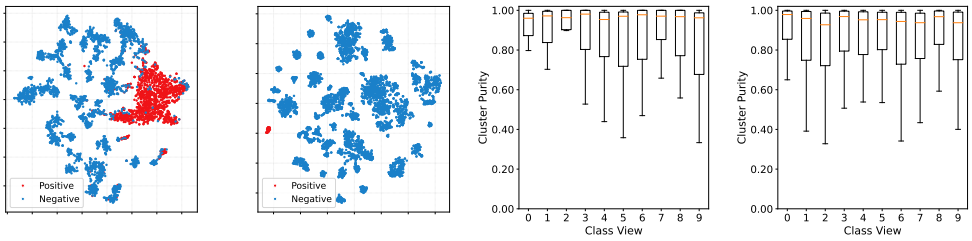
5.1 Experiment Settings

Implementation Details. In all experiments, we fixed the hyper-parameters of our method as follows. We set the number of clusters to $2C$ (twice the number of labels), use $k = 50$ for the kNN-consistency module, and set the positive threshold to $\tau = 0.6$. The weight of supervised-contrastive loss w is set to 0.01 in all experiments. We update pseudo-labels every $n = 5$ epochs. As analyzed in Appendices F.5-F.6, our method is not sensitive to these hyperparameters. More implementation details are provided in Appendix D, and runtime analysis is provided in Appendix F.6.

Evaluation metrics. Consistent with prior works [18, 20, 8], we evaluate performance using mean average precision (mAP↑), overall F1 (OF1↑), and per-class F1 (CF1↑). Results for additional metrics are reported in Appendix F.3.

5.2 Multi-label Learning with Partial Labels

Datasets. We evaluate on five widely used multi-label benchmarks, including three image datasets: MS-COCO 2014 (COCO) [21], Pascal VOC 2007 (VOC07) [22], VISPR [23], a text dataset AAPD [24], and an audio dataset FSD50K [25]. Since these datasets provide complete annotations, we simulate partial-label supervision by randomly masking a subset of label entries in the training set as unobserved. In this work, we primarily report an extreme single-label setting, where each instance has only one observed label (either $+1$ or -1) and all remaining labels are unobserved.



(a) (b) (c) (d)

Figure 3: Evidence of the negative clustering phenomenon. (a-b) t-SNE visualizations of class-wise embeddings on the text dataset AAPD and the audio dataset FSD50K. (c-d) Box plots of cluster purity on VOC07 and COCO14 under the single-label setting.

Comparison Methods. We compared our method with 13 MLL-PL methods: SARB [8], SST [26], Selective [27], LL-R* [28],LL-Ct* [28], LL-Cp* [28], GR [29], IGNORE [30], DSRB [31], HST [20], LC [32], LogicMix [33], and DiCaP [19]. Detailed descriptions of the compared methods are provided in Appendix E.1.

Results on image benchmarks under MLL-PL. Table 1 reports the mean and standard deviation over three random seeds on three image benchmarks under the single-label setting. The results show that our method consistently achieves the best performance across all datasets and metrics. In particular, on VOC07, we improve CF1 by 4.43% over the second-best approach and obtain a 4.66% gain in mAP. On COCO, we achieve a 3.00% improvement in OF1. These consistent gains suggest that exploiting the local structure of class-wise embeddings provides more reliable supervisory signals than relying solely on prediction confidence. Overall, our approach improves the average performance over the SOTA method by **4.46%**. We further report results with more observed labels per instance in Appendix F.1, where SaPu remains SOTA across different annotation levels.

Results on text and audio benchmarks. In Table 2, we further evaluate on the audio dataset FSD50K and the text dataset AAPD, where our method achieves SOTA performance across all metrics. In particular, it improves OF1 by 5.99% on FSD50K and by 7.82% on AAPD, demonstrating the robustness of our approach beyond the image modality.

Statistical significance. Our method yields statistically significant improvements in 80.34% of the cases (94/117) under a pairwise t-test at a 0.05 significance level. Detailed results and analyses are provided in Appendix F.2.

Table 2: Comparison of our model with other methods on the audio dataset FSD50K and the text dataset AAPD under the single-label setting.

Methods	FSD50K			AAPD		
	mAP↑	OF1↑	CF1↑	mAP↑	OF1↑	CF1↑
selective	50.98	32.91	36.88	33.02	29.36	29.11
LL-R*	45.15	42.87	48.42	36.34	29.52	28.92
LL-Ct*	35.20	24.43	32.74	33.26	25.13	26.82
LL-Cp*	36.14	26.02	35.89	35.35	26.79	27.94
GR	58.14	32.72	28.85	17.49	11.33	11.12
LC	47.94	23.71	35.54	32.98	28.03	32.96
DiCaP	65.15	44.13	52.82	43.45	54.71	39.98
Ours	**68.05**	**50.12**	**58.74**	**45.45**	**62.53**	**44.67**

5.3 Semi-supervised Multi-Label Learning

Datasets. We evaluate on two multi-label benchmarks, MS-COCO 2014 (COCO) [21] and Pascal VOC 2012 (VOC12) [22]. Following [34], we randomly sample a proportion ρ of training instances as labeled data and treat the remaining instances as unlabeled. In this work, we consider highly low-label settings with $\rho = 5\%$ on VOC12 and $\rho = 1\%$ on COCO.

Comparison Methods. We compare our model with 8 baseline methods, which are categorized into 2 groups. **Top-1**, **Top-k**, and **IAT** [34] are instance-based pseudo-labeling methods. **CAP** [34], **PCLP** [35], **BBAM** [36], **D2L** [18]

Table 3: Comparison with SOTA methods under the SSMLL setting ($\rho = 5\%$ labeled on VOC12, $\rho = 1\%$ on COCO).

Methods	VOC12			COCO		
	mAP↑	OF1↑	CF1↑	mAP↑	OF1↑	CF1↑
Top-1	77.10	67.87	65.61	48.83	52.09	45.91
Top-K	73.51	43.65	42.09	49.58	46.03	41.38
IAT	72.66	51.64	49.45	48.27	31.71	28.14
CAP	76.19	63.72	61.70	52.49	53.52	48.89
PCLP	77.25	60.36	59.70	53.12	57.93	53.30
BBAM	78.66	68.40	65.41	42.54	30.58	30.89
D2L	79.26	69.30	66.40	56.63	56.42	52.13
DiCaP	83.53	68.56	71.52	58.20	57.65	54.03
Ours	**85.01**	**70.74**	**73.13**	**58.87**	**58.09**	**55.01**

and **DiCaP** [19] represent the SOTA SSMLL methods. Detailed descriptions of the compared methods are provided in Appendix E.2.

Results under SSMLL. Table 3 summarizes the comparison under SSMLL on VOC12 and COCO. Despite the extremely limited labeled data (5% on VOC12 and 1% on COCO), our method consistently achieves the best performance across all metrics. For instance, on VOC12 with only 5% labeled samples, our method improves CF1 by **+1.61%** over the SOTA baseline. Together with the MLL-PL results, these improvements show that SaPu is effective under different weak-supervision regimes.

5.4 Further Analysis

Evidence of the negative multi-cluster structure phenomenon. As shown in Fig. 3, we provide additional t-SNE visualizations on the text dataset AAPD and the audio dataset FSD50K, together with box plots of cluster purity under the single-label setting. Across both datasets, we observe the negative multi-cluster structure consistently across modalities. Moreover, even under the single-label setting, the resulting clusters maintain high purity, further supporting the universality of the negative multi-cluster structure. Additional evidence for this phenomenon is provided in Appendix C.

Ablation study. In Table 4, we conduct ablation experiments on VOC07 and COCO under the single-label setting to quantify the contribution of each component. The results show that all modules consistently improve performance. On VOC07, adding positive pseudo-label mining (PLM) module yields a notable gain of +3.81%. Further incorporating negative label mining (NLM) module and class-wise supervised contrastive learning (SCL) module brings additional improvements of 0.56% and 1.29%, respectively. Notably, SCL is more effective when combined with NLM than when paired with PLM alone, since mined negatives provide more contrasting samples for representation learning. A finer-grained SCL ablation in Appendix F.8 shows that expanded pseudo-labels are crucial for effective contrastive learning. We further provide per-class results and analyses in Appendix F.4.

Table 4: Ablation study on VOC07 and COCO. ✓ indicates the result with optimal hyperparameters. Baseline corresponds to training using only the observed labels.

Methods			VOC07	COCO	Average
Baseline			78.77	63.34	71.06
PLM	NLM	SCL	mAP↑ (%)		
✓			82.58	64.75	73.67
✓	✓		83.14	64.99	74.07
✓		✓	83.87	65.15	74.51
✓	✓	✓	**84.73**	**65.70**	**75.22**

Quality of mined pseudo-labels. To assess the reliability of our pseudo-label mining strategy, Fig. 4 reports the precision of pseudo-labels with varying thresholds on COCO under the single-label setting. We compare our mined pseudo-labels against (i) *Prediction*: pseudo-labels obtained directly from prediction confidences, (ii) *Single View*: a single-view variant that uses only one class-wise embedding view, and (iii) *Backbone*: a backbone-only baseline that computes pseudo-labels from globally pooled backbone features (i.e., average pooling over the feature map). Under the same threshold, our method consistently achieves substantially higher precision than confidence-based pseudo-labeling. Moreover, it also outperforms the single-view and backbone-only baselines, highlighting the necessity of multi-view ensembling for reliable pseudo supervision.

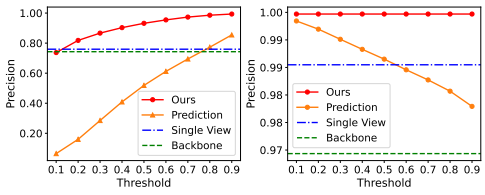


Figure 4: Precision of pseudo-labels under different thresholds on COCO (single-label setting) for (a) positive and (b) negative pseudo-labels.

6 Conclusion

In this paper, we have identified an underexplored phenomenon in class-wise embedding spaces for multi-label learning: negative embeddings naturally organize into multiple semantically coherent sub-clusters, consistently observed across modalities and remaining pronounced even under extremely sparse supervision. Motivated by this insight, we proposed **SaPu**, a Structure-aware Pseudo-label Mining framework for weakly supervised MLL. SaPu mines high-confidence positives by ensembling cluster-consistency across views and identifies reliable negatives via multi-view neighborhood exclusion. The mined pseudo-labels further supervise class-wise contrastive learning, establishing a

self-reinforcing loop that iteratively refines embeddings and pseudo-labels. Extensive experiments on image, text, and audio benchmarks show consistent gains.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Natural Science Foundation of China under Grants U24A20322, 62576094 and 62422118, in part by the Hong Kong UGC under grants UGC/FDS11/E03/25 and UGC/FDS11/E04/26, and in part by the Hong Kong Research Grants Council under Grants RFS2627-1S02, R1018-25F. This research work was also supported by the Big Data Computing Center of Southeast University.

References

[1] Weiwei Liu, Haobo Wang, Xiaobo Shen, and Ivor W Tsang. The emerging trends of multi-label learning. *IEEE TPAMI*, 44(11):7955–7974, 2021.

[2] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *CVPR*, pages 5177–5186, 2019.

[3] Kazuya Shimura, Jiyi Li, and Fumiyo Fukumoto. HFT-CNN: learning hierarchical category structure for multi-label short text categorization. In *EMNLP*, pages 811–816, 2018.

[4] Yingjian Li, Yingnan Gao, Bingzhi Chen, Zheng Zhang, Guangming Lu, and David Zhang. Self-supervised exclusive-inclusive interactive learning for multi-label facial expression recognition in the wild. *IEEE TCSVT*, 32(5):3190–3202, 2022.

[5] Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. Query2label: A simple transformer way to multi-label classification. *arXiv preprint arXiv:2107.10834*, 2021.

[6] Tal Ridnik, Gilad Sharir, Avi Ben-Cohen, Emanuel Ben Baruch, and Asaf Noy. Ml-decoder: Scalable and versatile classification head. In *WACV*, pages 32–41, 2023.

[7] Min-Ling Zhang and Lei Wu. Lift: Multi-label learning with label-specific features. *IEEE TPAMI*, 37(1):107–120, 2015.

[8] Tao Pu, Tianshui Chen, Hefeng Wu, and Liang Lin. Semantic-aware representation blending for multi-label image recognition with partial labels. In *AAAI*, 2022.

[9] Xueling Zhu, Jian Liu, Jiuxin Cao, and Bing Wang. Mambaml: Exploring state space models for multi-label image classification. In *ICCV*, 2025.

[10] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(86):2579–2605, 2008.

[11] Feng Zhu, Hongsheng Li, Wanli Ouyang, Nenghai Yu, and Xiaogang Wang. Learning spatial regularization with image-level supervisions for multi-label image classification. In *CVPR*, 2017.

[12] Tal Ridnik, Emanuel Ben Baruch, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. In *ICCV*, pages 82–91, 2021.

[13] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *CVPR*, pages 5177–5186, 2019.

[14] Jack Lanchantin, Tianlu Wang, Vicente Ordonez, and Yanjun Qi. General multi-label image classification with transformers. In *CVPR*, pages 16478–16488, 2021.

[15] Tianshui Chen, Muxin Xu, Xiaolu Hui, Hefeng Wu, and Liang Lin. Learning semantic-specific graph representation for multi-label image recognition. In *ICCV*, pages 522–531, 2019.

[16] Ming-Kun Xie, Jiahao Xiao, and Sheng-Jun Huang. Label-aware global consistency for multi-label learning with single positive labels. In *NeurIPS*, 2022.

[17] Donghao Zhou, Pengfei Chen, Qiong Wang, Guangyong Chen, and Pheng-Ann Heng. Acknowledging the unknown for multi-label learning with single positive labels. In *ECCV*, pages 423–440, 2022.

[18] Jiahao Xiao, Ming-Kun Xie, Heng-Bo Fan, Gang Niu, Masashi Sugiyama, and Sheng-Jun Huang. Dual-decoupling learning and metric-adaptive thresholding for semi-supervised multi-label learning. In *ECCV*, 2024.

[19] Bo Han, Zhuoming Li, Xiaoyu Wang, Yaxin Hou, Hui Liu, Junhui Hou, and Yuheng Jia. Dicap: Distribution-calibrated pseudo-labeling for semi-supervised multi-label learning. In *AAAI*, 2026.

[20] Tianshui Chen, Tao Pu, Lingbo Liu, Yukai Shi, Zhijing Yang, and Liang Lin. Heterogeneous semantic transfer for multi-label recognition with partial labels. *IJCV*, 132(12):6091–6106, 2024.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[22] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88:303–338, 2010.

[23] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Towards a visual privacy advisor: Understanding and predicting privacy risks in images. In *ICCV*, 2017.

[24] Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. SGM: sequence generation model for multi-label classification. In *COLING*, 2018.

[25] Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. FSD50K: an open dataset of human-labeled sound events. *IEEE TASLP*, 30:829–852, 2022.

[26] Tianshui Chen, Tao Pu, Hefeng Wu, Yuan Xie, and Liang Lin. Structured semantic transfer for multi-label recognition with partial labels. In *AAAI*, 2022.

[27] Emanuel Ben Baruch, Tal Ridnik, Itamar Friedman, Avi Ben-Cohen, Nadav Zamir, Asaf Noy, and Lihi Zelnik-Manor. Multi-label classification with partial annotations using class-aware selective loss. In *CVPR*, 2022.

[28] Youngwook Kim, Jae-Myung Kim, Jieun Jeong, Cordelia Schmid, Zeynep Akata, and Jungwoo Lee. Bridging the gap between model explanations in partially annotated multi-label classification. In *CVPR*, 2023.

[29] Yanxi Chen, Chunxiao Li, Xinyang Dai, Jinhuan Li, Weiyu Sun, Yiming Wang, Renyuan Zhang, Tinghe Zhang, and Bo Wang. Boosting single positive multi-label classification with generalized robust loss. In *IJCAI*, 2024.

[30] GyeongRyeol Song, Noo-Ri Kim, Jin-Seop Lee, and Jee-Hyong Lee. IGNORE: information gap-based false negative loss rejection for single positive multi-label learning. In *ECCV*, 2024.

[31] Tao Pu, Tianshui Chen, Hefeng Wu, Yukai Shi, Zhijing Yang, and Liang Lin. Dual-perspective semantic-aware representation blending for multi-label image recognition with partial labels. *ESWA*, 249:123526, 2024.

[32] Shayan Gharib, Pierre-Alexandre Murena, and Arto Klami. Single-positive multi-label learning with label cardinality. *TMLR*, 2025, 2025.

[33] Chak Fong Chong, Jielong Guo, Xu Yang, Wei Ke, Pedro Henriques Abreu, Yapeng Wang, and Sio Kei Im. Logicmix: Sample mixing data augmentation for multi-label image classification with partial labels. *PR*, 171:112186, 2026.

[34] Ming-Kun Xie, Jiahao Xiao, Hao-Zhe Liu, Gang Niu, Masashi Sugiyama, and Sheng-Jun Huang. Class-distribution-aware pseudo-labeling for semi-supervised multi-label learning. In *NeurIPS*, 2023.

[35] Biao Liu, Ning Xu, Xiangyu Fang, and Xin Geng. Correlation-induced label prior for semi-supervised multi-label learning. In *ICML*, 2024.

[36] Ximing Li, Silong Liang, Changchun Li, Pengfei Wang, and Fangming Gu. Semi-supervised multi-label learning with balanced binary angular margin loss. In *NeurIPS*, 2024.

[37] Zhiqiang Kou, Jing Wang, Jiawei Tang, Yuheng Jia, Boyu Shi, and Xin Geng. Exploiting multi-label correlation in label distribution learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 4326–4334, 8 2024.

[38] Zhiqiang Kou, Si Qin, Hailin Wang, Jing Wang, Mingkun Xie, Shuo Chen, Yuheng Jia, Tongliang Liu, Masashi Sugiyama, and Xin Geng. Label distribution learning with biased annotations assisted by multi-label learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 8 2025.

[39] Zhiqiang Kou, Junxiang Wu, Wenke Huang, Wenwen He, Ming-Kun Xie, Changwei Wang, Yuheng Jia, Di Jiang, Yang Liu, Xin Geng, and Qiang Yang. Fedharmony: Harmonizing heterogeneous label correlations in federated multi-label learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[40] Zhiqiang Kou, Yucheng Xie, Hailin Wang, Jing Wang, Mingkun Xie, Shuo Chen, Yuheng Jia, Tongliang Liu, and Xin Geng. Rankmatch: A novel approach to semi-supervised label distribution learning leveraging rank correlation between labels. In *Proceedings of the 39th Conference on Neural Information Processing Systems (NeurIPS 2025)*, San Diego, California, USA, December 2025. Neural Information Processing Systems Foundation.

[41] Zhuoming Li, Yuheng Jia, Mi Yu, and Zicong Miao. Calibrated disambiguation for partial multi-label learning. In *AAAI*, pages 18620–18628, 2025.

[42] Fuchao Yang, Yuheng Jia, Hui Liu, Yongqiang Dong, and Junhui Hou. Noisy label removal for partial multi-label learning. In *KDD*, pages 3724–3735, 2024.

[43] Elijah Cole, Oisin Mac Aodha, Titouan Lorieul, Pietro Perona, Dan Morris, and Nebojsa Jojic. Multi-label learning from single positive labels. In *CVPR*, pages 933–942, 2021.

[44] Ning Xu, Congyu Qiao, Jiaqi Lv, Xin Geng, and Min-Ling Zhang. One positive label is sufficient: Single-positive multi-label learning with label enhancement. In *NeurIPS*, 2022.

[45] Youngwook Kim, Jae-Myung Kim, Zeynep Akata, and Jungwoo Lee. Large loss matters in weakly supervised multi-label classification. In *CVPR*, pages 14136–14145. IEEE, 2022.

[46] Cheng Chen, Yifan Zhao, and Jia Li. Semantic contrastive bootstrapping for single-positive multi-label recognition. *IJCV*, 131(12):3289–3306, 2023.

[47] Lichen Wang, Yunyu Liu, Can Qin, Gan Sun, and Yun Fu. Dual relation semi-supervised multi-label learning. In *AAAI*, pages 6227–6234, 2020.

[48] Jun-Yi Hang and Min-Ling Zhang. Collaborative learning of label semantics and deep label-specific features for multi-label classification. *IEEE TPAMI*, 44(12):9860–9871, 2022.

[49] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[50] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[51] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019.

[52] Yuan Gong, Yu-An Chung, and James R. Glass. AST: audio spectrogram transformer. In *Interspeech*, pages 571–575. ISCA, 2021.

[53] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[54] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, pages 3008–3017, 2020.

[55] Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[56] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE TKDE*, 26(8):1819–1837, 2014.

Table of Contents for the Appendix

- A: More Detailed Related Work 14
- A.1: Multi-Label Learning with Partial Labels 14
- A.2: Semi-Supervised Multi-Label Learning 14
- B: Additional Evidence for the Origin of Negative Multi-cluster Structure 15
- C: Additional Evidence for the Negative Multi-cluster Structure 15
- D: More Implementation Details 16
- E: Details and Discussion of Compared Methods 17
- E.1: Compared Methods under the MLL-PL Setting 17
- E.2: Compared Methods under the SSMLL Setting 18
- E.3: Comparison with Existing Methods 19
- F: More Experiment Results and Analyses 19
- F.1: Results with More Observed Labels per Instance 19
- F.3: Results for Additional Metrics 20
- F.2: Statistical Significance 20
- F.4: Per-class Comparison on VOC07 20
- F.5: Hyperparameter sensitivity 21
- F.6: Influence of Pseudo-label Update Frequency 21
- F.7: Effect of Agreement Constraint in Negative Pseudo-label Mining 21
- F.8: Ablation Study on Class-wise Contrastive Learning 22
- F.9: Self-reinforcing feedback loop induced by supervised contrastive learning 22
- F.10: Controlling for the prediction head 23
- F.11: Stability under the SSMLL setting 23
- G: Compute Resources 23
- H: Limitations 23
- I: Broader Impacts 23

Algorithm 1 Algorithm of SaPu

Input: $\mathcal{D}$: dataset; T_{w}: warm-up epochs; T_{p}: pseudo-label training epochs; n: update interval.
Output: Trained multi-label classifier f_θ.

```
Warm up for T_w epochs by minimizing loss Eq. (15).
for t = 0 to T_p do
    if t mod n = 0 then
        Collect class-wise embeddings {z_i^(c)}_{i=1,c=1}^{N,C}.
        Update positive pseudo-labels (Sec. 4.2).
        Update negative pseudo-labels (Sec. 4.3).
    end if
    Update f_θ by minimizing the total loss in Eq. (16).
end for
```

A More Detailed Related Work

Due to the high cost of collecting complete and accurate multi-label annotations, recent works [20, 19, 37, 38, 39, 40, 41, 42] have increasingly studied weakly supervised variants of MLL, including multi-label learning with partial labels [26, 8, 33] and semi-supervised multi-label learning [34, 35, 18, 19].

A.1 Multi-Label Learning with Partial Labels

Multi-label learning with partial labels has attracted increasing attention as it significantly reduces annotation cost by requiring only a subset of labels per instance. A prominent line of work [43, 44, 16, 45, 46, 28, 30, 32, 29] focuses on the single-positive multi-label (SPML) setting, where only a few positive labels are annotated while the remaining labels are unknown. A common strategy treats unknown labels as negatives and mitigates the resulting false-negative bias by identifying unreliable negatives during training. Along this direction, Xie et al. introduced a label-aware global consistency regularization that combines pseudo-label consistency with contrastive learning over class-wise embeddings to recover missing positives [16]. Similarly, Song et al. identified false negative labels by measuring the logit gap between original images and CAM-masked images, and excluded them from training to alleviate false negative bias [30]. Beyond the SPML setting, another line of work [27, 26, 8, 20, 31, 33] considers partial-label multi-label learning where both positive and negative labels are partially observed, and approaches the problem from a pseudo-label generation perspective. SARB generates pseudo labels by measuring the similarity between label features and category prototypes [8]. Chen et al. proposed a heterogeneous semantic transfer framework that leverages intra-image label co-occurrence and cross-image prototype similarity to generate pseudo labels [20].

A.2 Semi-Supervised Multi-Label Learning

SSMLL tackles the challenge of learning with limited labeled data by leveraging additional unlabeled samples. Various methods [47, 34, 35, 36, 18, 19] have been proposed to generate reliable pseudo-labels or effectively utilize unlabeled data in SSMLL. For instance, Xie et al. adopted a class-specific thresholding approach, where thresholds are dynamically estimated from class priors to binarize predictions for unlabeled instances [34]. Liu et al. enhanced pseudo-label quality by introducing a causality-guided label prior, inferred via a Structural Causal Model, and incorporates this prior into a variational inference framework to guide label enhancement [35]. Li et al. addressed the variance bias between positive and negative samples by extending the binary angular margin loss with Gaussian-based feature angle transformations, and introduces a negative sampling scheme to further stabilize training [36]. Xiao et al. adopted a dual-decoupling strategy to jointly model discriminative and correlative information, and introduces a metric-adaptive thresholding mechanism to dynamically refine pseudo-label assignment throughout training [18]. Furthermore, Han et al. formulated an unsupervised class-wise contrastive framework to exploit the representational potential of uncertain samples in the semi-supervised setting [19].

Notably, several existing weakly supervised multi-label learning methods [8, 16, 26, 31, 20, 19] are also built upon class-wise embeddings. In contrast to these approaches, we reveal a previously overlooked *negative multi-cluster structure* in class-wise embedding spaces, which consistently

emerges across different modalities and persists under extremely weak supervision. Leveraging this structural insight, we propose a Structure-Aware Pseudo-label Mining strategy, which leads to substantial performance gains in highly label-scarce scenarios (Tables 1, 2, 3, and Fig. 6).

B Additional Evidence for the Origin of Negative Multi-cluster Structure

Fig. 5 provides additional qualitative evidence supporting the explanation in Section 3.2. When a queried label is present, the corresponding attention map is typically peaked and concentrates on label-relevant regions (e.g., b, f, j, n), leading to a class-wise embedding that is explicitly conditioned on the queried label. In contrast, for absent labels, the query cannot retrieve label-specific content from the instance, and the attention becomes comparatively uniform and weakly localized (e.g., c, g, k, o). In this case, the resulting embedding is dominated by the instance-level semantics, which naturally results in diverse negative embeddings across different instances. Finally, due to semantic similarity or visual co-occurrence, an absent-label query may still induce weak but consistent preferences over certain regions (e.g., d, h, l, p), which further increases the diversity of negative embeddings across label views.

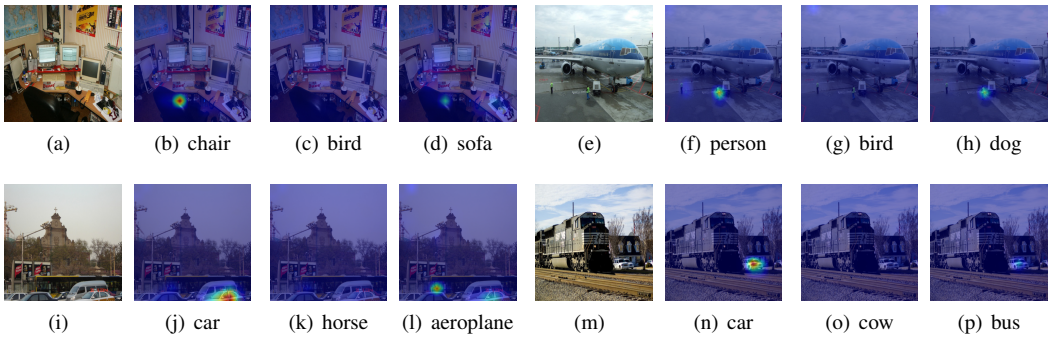


Figure 5: Additional attention visualizations for explaining the negative multi-cluster structure. In each group, the first column (a/e/i/m) shows the original image, and the remaining columns overlay the class-query attention maps for different label views (subtitle indicates the queried label). Ground-truth labels for the four images are: (a) chair, tvmonitor, (e) aeroplane, person, (i) bus, car, and (m) car, train.

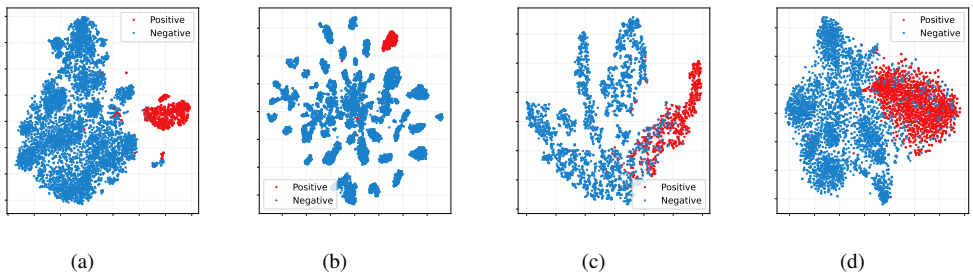
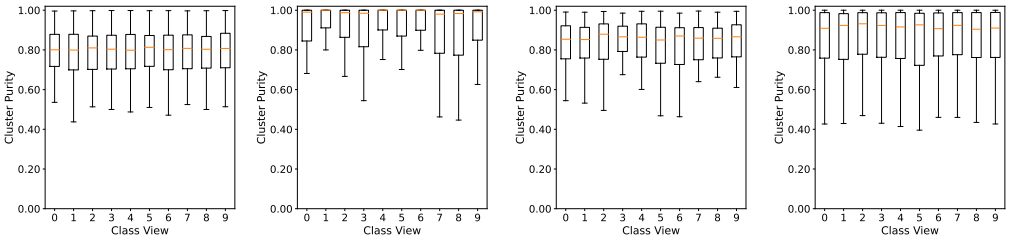


Figure 6: t-SNE visualizations of class-wise embeddings on additional multi-label benchmarks: (a) VISPR (image), (b) COCO14 (image), (c) RCV1Subset1 (text), and (d) tmc07 (text).

C Additional Evidence for the Negative Multi-cluster Structure

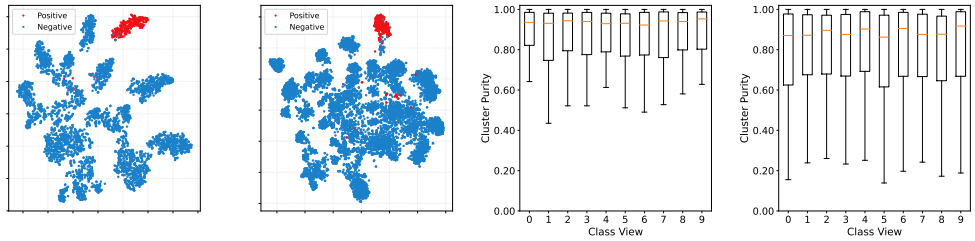
In this section, we provide further evidence for the negative multi-cluster structure of class-wise embeddings. Beyond the datasets reported in the main text (VOC07, FSD50K, and AAPD), we additionally present results on two image datasets (VISPR and COCO14) and two text datasets (RCV1Subset1 and tmc07)[2]. For RCV1Subset1 and tmc07, class-wise embeddings are extracted

[2]RCV1Subset1 and tmc07 are available at `http://mulan.sourceforge.net/datasets.html`.



(a) (b) (c) (d)

Figure 7: Cluster purity under the extreme single-label setting on: (a) AAPD (text), (b) VISPR (image), (c) tmc07 (text), and (d) FSD50K (audio).



(a) (b) (c) (d)

Figure 8: Evidence of the negative multi-cluster structure phenomenon under the SSMLL setting. (a-b) t-SNE visualizations of class-wise embeddings on VOC12 and COCO14. (c-d) Box plots of cluster purity on VOC12 and COCO14.

using CLIF [48]. As shown in Fig. 6, negative embeddings consistently exhibit clear multi-cluster structures across heterogeneous modalities. Fig. 7 further reports box plots of cluster purity under the extreme single-label setting; despite the very limited supervision, cluster purity remains high across these datasets, corroborating that the phenomenon is pervasive and remains pronounced under extremely sparse supervision.

We further investigate whether this phenomenon persists under the SSMLL setting. Specifically, Fig. 8 visualizes the class-wise embeddings (t-SNE) and reports cluster purity when the model is trained with only a small fraction of labeled data (5% on VOC and 1% on COCO). Even in this semi-supervised regime, the negative multi-cluster structure remains clearly observable, and the resulting clusters still exhibit high purity, providing additional evidence of its robustness in extreme weakly supervised scenarios.

D More Implementation Details

For all experimental settings, we adopt the same backbone architectures across our method and all baseline approaches, using ResNet-50 [49] pre-trained on ImageNet [50], BERT-base-uncased [51], and AST [52] (initialized from ast-finetuned-audioset-10-10-0.4593) for the image, text (AAPD), and audio (FSD50K) datasets, respectively. For the decoder, we employ the ML-Decoder framework [6] to produce class-wise embeddings. To stabilize training, we use an exponential moving average (EMA) of model parameters with decay 0.9997. Unless otherwise specified, we optimize the model using AdamW [53] with weight decay 10^{-4} and a one-cycle learning-rate schedule.

For image datasets, we apply RandAugment [54] and Cutout [55] for data augmentation. For audio dataset (FSD50K), we load raw waveforms, convert them to mono by channel averaging, and resample them to 16 kHz. Since audio clips have variable lengths, waveforms are zero-padded within each mini-batch to the maximum length in that batch. No additional audio augmentation is applied. For text datasets, inputs are tokenized using the BERT tokenizer with a maximum sequence length of 256, where longer sequences are truncated and shorter ones are padded to a fixed length. During training, we use back-translation as data augmentation.

For method implementation, we utilize GPU-accelerated kNN and k-means clustering provided by the `torch-clustering` library[3]. This significantly reduces the additional computational overhead. Specifically, we store the collected class-wise embeddings in CPU memory and transfer only the embeddings corresponding to a target class to the GPU when performing kNN or clustering operations, thereby reducing GPU memory consumption. A detailed discussion of the time cost is provided in Appendix F.6. All experiments are conducted on NVIDIA GeForce RTX 3090 GPUs.

Multi-label learning with partial labels. We set the maximum learning rate to 10^{-5}. Each run consists of a warm-up stage of 20 epochs (training only with observed labels) followed by a pseudo-label stage of 20 epochs. To account for randomness, we repeat each experiment using three random seeds (0, 1, 2) and report the mean and standard deviation. For image datasets, all input images are resized to 448×448. For LL-R*, LL-Ct*, and LL-Cp* [28], we use the original implementation [45] when evaluating on the text and audio datasets, due to differences in modality-specific implementations. The 2023 variants [28] extend the original method [45] by introducing an additional module that boosts the attribution scores, which further improves performance on image datasets. Since the 2023 implementation is only available for image data, we adopt the 2022 version for text and audio datasets.

Semi-supervised multi-label learning. Under SSMLL setting, all input images are resized to 224×224. We set the maximum learning rate to 10^{-4}. Following prior SSMLL protocols [34, 19], we use a warm-up stage of 12 epochs and a pseudo-label stage of 28 epochs. For reproducibility, we fix the random seed to 1 for all SSMLL experiments. To assess initialization stability, we additionally repeat SaPu and DiCaP over three random seeds in Appendix F.11.

Asymmetric thresholds for positive pseudo-labels. Our neighborhood-based negative mining relies on the positive pseudo-labels obtained in Section 4.2. To reduce false negatives, we use a more permissive rule when constructing the candidate positive set for negative mining. Specifically, we mark an instance-label pair as positive if any view predicts it as positive. In contrast, when forming the final positive pseudo-labels used for training, we apply a stricter consensus threshold to emphasize precision.

E Details and Discussion of Compared Methods

E.1 Compared Methods under the MLL-PL Setting

- **SARB** [8]: SARB complements unknown labels by blending category-specific representations across images at both the instance and prototype levels, enabling information from known labels to be transferred without relying on an additional pre-trained pseudo-labeling model.
- **SST** [26]: SST complements missing labels by jointly exploiting intra-image label co-occurrence and cross-image category-specific feature similarity, thereby transferring semantic knowledge from observed labels to unknown labels in partially annotated multi-label recognition.
- **Selective** [27]: Selective determines whether each unannotated label should be ignored or treated as negative based on its estimated label likelihood and class prior, while using a partial asymmetric loss to better balance annotated and unannotated supervision.
- **LL-R*, LL-Ct*, LL-Cp*** [28]: These methods denote the BoostLU-enhanced variants of the three Large Loss Matters strategies, where LL-R rejects large-loss unobserved labels, LL-Ct temporarily corrects them as positives during loss computation, and LL-Cp permanently converts them into positive labels, while BoostLU is incorporated as a plug-in to amplify high-attribution CAM regions and further recover positive evidence suppressed by false-negative supervision.
- **IGNORE** [30]: IGNORE identifies potential false-negative labels in single-positive multi-label learning by comparing the logit gap between the original image and a masked image that preserves only the discriminative region of the observed positive label, and rejects the corresponding losses to reduce noisy negative supervision.

[3] `https://github.com/Hzzone/torch_clustering`

- **GR** [29]: GR formulates single-positive multi-label learning under an expected-risk framework, using soft pseudo-label estimation, class- and instance-sensitive reweighting, and robust surrogate losses to jointly alleviate false negatives, noisy pseudo-labels, and positive-negative imbalance.
- **DSRB** [31]: DSRB transfers information from known labels to unknown labels by blending category-specific semantic representations across images from both instance and prototype perspectives, where instance-perspective blending provides diverse complemented samples and prototype-perspective blending generates more stable location-sensitive representations.
- **HST** [20]: HST complements unknown labels by jointly modeling image-specific label co-occurrence within each image and category-specific feature-prototype similarity across images, while employing differentiable threshold learning to automatically determine pseudo-label generation thresholds.
- **LC** [32]: LC leverages auxiliary information about the number of positive labels per instance, ranging from coarse cardinality statistics to instance-level cardinality estimates, to regularize the predictions of unobserved labels and encourage the model outputs to be consistent with the expected label cardinality.
- **LogicMix** [33]: LogicMix extends sample-mixing augmentation to partially labeled multi-label data by composing labels through logical OR equivalences, and its 2WayAug-PL framework further combines LogicMix with conventional augmentation, pseudo-labeling, and decoupled partial-asymmetric loss to improve generalization and supervision quality.

E.2 Compared Methods under the SSMLL Setting

- **Top-1** [34]: The Top-1 pseudo-labeling strategy assigns the class with the highest predicted probability as the only positive pseudo-label for each unlabeled instance, which provides a conservative prediction but may ignore other true positive labels in multi-label scenarios.
- **Top-k** [34]: The Top-k pseudo-labeling strategy assigns the k classes with the highest predicted probabilities as positive pseudo-labels, where k is typically set to the average number of positive labels per instance; however, it may introduce false positives or miss true positives when the actual label cardinality varies across instances.
- **IAT** [34]: IAT assigns positive pseudo-labels to the classes whose predicted probabilities exceed an instance-level threshold, allowing different unlabeled instances to have different numbers of pseudo-labels, while its effectiveness depends on the choice of the threshold.
- **CAP** [34]: CAP performs pseudo-label assignment from a class-wise perspective by using class-aware thresholds derived from the estimated class distribution, thereby controlling the numbers of positive and negative pseudo-labels for each class and encouraging the pseudo-label distribution to match the underlying true class distribution.
- **D2L** [18]: D2L improves pseudo-label quality from both prediction and thresholding perspectives by decoupling correlative and discriminative feature learning as well as pseudo-label generation and utilization, while estimating class-wise thresholds that optimize a target metric on labeled data.
- **BBAM** [36]: BBAM addresses the variance bias between positive and negative feature distributions in semi-supervised multi-label learning by introducing a balanced binary angular margin loss with Gaussian-based feature-angle distribution transformations and prototype-based negative sampling.
- **PCLP** [35]: PCLP constructs a correlated label prior distribution using a structural causal model and integrates it into a variational label enhancement framework, so that the generated pseudo-labels are guided to conform to inter-label correlations.
- **DiCaP** [19]: DiCaP calibrates pseudo-label training by estimating the posterior correctness likelihood of pseudo-labels as adaptive weights and further separates confident and ambiguous predictions through a dual-thresholding mechanism, using weighted pseudo-labeling for the former and contrastive learning for the latter.

Table 5: Statistical significance of performance differences assessed with pairwise t-test at a 0.05 significance level, reported as win/tie/loss counts.

method	SARB	SST	selective	LL-R	LL-Ct	LL-Cp	GR	IGNORE	DSRB	HST	LC	LogicMix	DiCaP	total
win	6	5	9	9	9	9	9	7	6	5	9	8	3	94
tie	3	4	0	0	0	0	0	2	3	4	0	1	6	23
loss	0	0	0	0	0	0	0	0	0	0	0	0	0	0

E.3 Comparison with Existing Methods

The compared methods can be broadly grouped into prediction-probability-based and feature-representation-based approaches. The former category includes loss correction, false-negative suppression, label-cardinality regularization, and threshold-based pseudo-labeling methods, such as Selective [27], LL-R / LL-Ct / LL-Cp [28], IGNORE [30], GR [29], LC [32], Top-1 [34], Top-k [34], IAT [34], CAP [34], D2L [18], PCLP [35], and DiCaP [19]. These methods mainly rely on prediction scores, losses, logits, estimated label distributions, or cardinality constraints to determine the supervision of unobserved labels. However, under limited supervision, prediction probabilities are often biased or poorly calibrated, making pseudo-label selection sensitive to unreliable model outputs.

The latter category, including SARB [8], SST [26], DSRB [31], HST [20], and BBAM [36], exploits class-wise representations, feature similarities, semantic transfer, prototype information, or feature distribution modeling to complement missing labels. These methods alleviate part of the uncertainty caused by incomplete annotations by mining the embedding space. Nevertheless, they primarily transfer information from observed positive labels or positive prototypes, or regularize feature distributions without explicitly exploiting the structure of negative labels. In particular, negative labels are commonly treated as a homogeneous set, and their potential multi-cluster structure in the class-wise embedding space is not explicitly modeled.

In contrast, our method exploits the negative multi-cluster structure for pseudo-label mining. Rather than relying mainly on prediction confidence or positive prototypes, SaPu mines candidate labels according to their relationships with structured negative representations in the class-wise embedding space. This provides a different source of supervisory information and allows SaPu to better utilize the discriminative structures hidden in negative labels under incomplete annotation.

F More Experiment Results and Analyses

F.1 Results with More Observed Labels per Instance

As shown in Fig. 9, we further test on VOC07 with more observed labels per instance. Across different annotation levels, our method consistently outperforms compared approaches. Notably, the advantage becomes larger as the supervision becomes weaker (i.e., fewer observed labels per instance), indicating that SaPu is particularly effective in highly label-scarce scenarios. This is because, when observed labels are extremely sparse, methods relying mainly on the limited annotations or prediction confidence are more prone to noisy pseudo-labels, whereas SaPu can exploit the local semantic structure in class-wise embedding spaces to mine more reliable supervisory signals. The cross-view consistency design further reduces the influence of noisy clusters, making the mined pseudo-labels more stable under severe annotation sparsity. At the same time, when more labels are observed per instance, SaPu remains competitive and achieves strong performance, demonstrating that the proposed structure-aware mining mechanism is robust across a broad range of annotation sparsity levels.

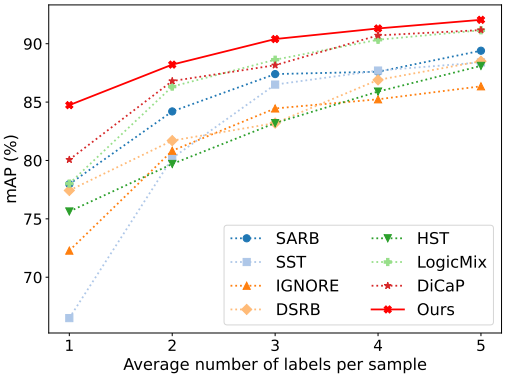


Figure 9: Performance on VOC07 with varying numbers of observed labels per instance. mAP↑(%) is adopted as metric.

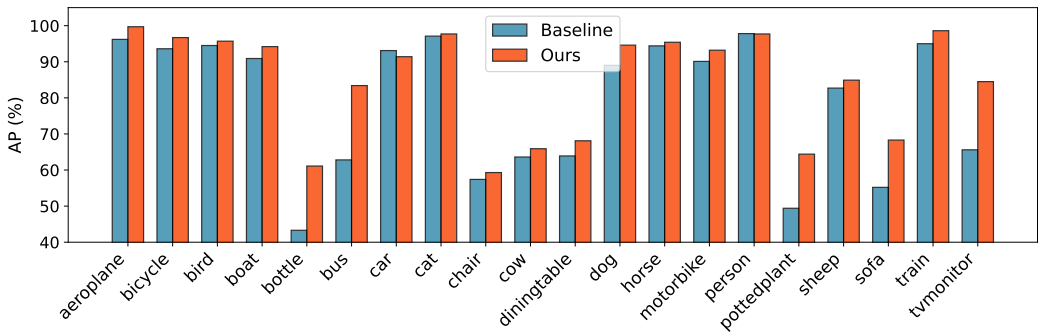


Figure 10: Per-class average precision (AP) comparison on VOC07 between our method and the baseline.

F.2 Statistical Significance

To quantify the robustness of our improvements, we conduct statistical significance tests using the three-seed runs on VOC07, COCO, and VISPR under the single-label setting. Specifically, for each dataset and metric, we compare SaPu against each of the 13 baselines using a paired t-test across seeds at a 0.05 significance level. We then summarize the outcomes as win/tie/loss counts.

Table 5 reports the aggregated results over all comparisons. SaPu **never loses** to any baseline (0 losses), achieving either a win or a tie in all 117 comparisons. Moreover, 94 out of 117 cases (80.34%) are statistically significant wins, while the remaining 23 cases are ties. These results provide strong evidence that the gains reported in the main paper are consistent across random seeds.

F.3 Results for Additional Metrics

As shown in Table 6, we further report two additional metrics, Hamming Loss (Ham-L↓) and One Loss (One-L↓) [56], on FSD50K and AAPD under the single-label setting. Ham-L measures the average label-wise prediction error, while One-L evaluates whether the top-ranked predicted label is irrelevant. These metrics therefore provide complementary evaluation beyond mAP and F1 scores.

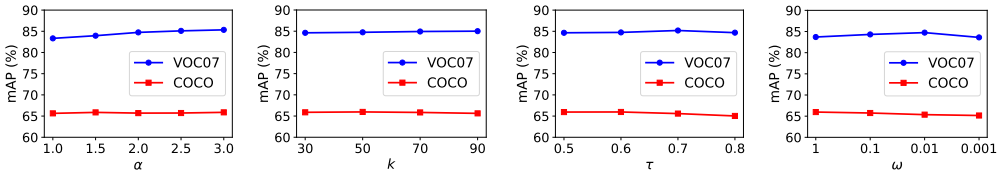
Table 6: Comparison on additional metrics (Hamming Loss↓, One Loss↓) on FSD50K and AAPD under the single-label setting.

Methods	FSD50K		AAPD	
	Ham-L↓	One-L↓	Ham-L↓	One-L↓
selective	0.033	0.422	0.039	0.345
LL-R*	0.289	0.429	0.037	0.303
LL-Ct*	0.678	0.244	0.038	0.302
LL-Cp*	0.765	0.260	0.036	0.306
GR	0.028	0.328	0.059	0.654
LC	0.044	0.548	0.043	0.355
DiCaP	<u>0.021</u>	<u>0.213</u>	<u>0.035</u>	<u>0.280</u>
Ours	**0.019**	**0.188**	**0.032**	**0.235**

Our method consistently achieves the best performance on both datasets and both metrics. For example, on FSD50K, SaPu reduces One Loss to 0.188, outperforming the second-best DiCaP (0.213). On AAPD, SaPu also decreases One Loss from 0.280 to 0.235, showing a clear improvement in ranking the relevant labels at the top. Meanwhile, the lower Hamming Loss on both datasets indicates that the proposed structure-aware pseudo-label mining does not merely improve a few highly confident predictions, but also reduces overall label-wise errors. These results further demonstrate the effectiveness of SaPu.

F.4 Per-class Comparison on VOC07

Fig. 10 reports the per-class AP comparison between our method and the baseline on VOC07. Our approach improves performance on 19 out of 20 categories, demonstrating consistent gains across classes. Notably, we observe substantial improvements on categories where the baseline underperforms (e.g., *bottle*, *bus*, and *tvmonitor*), further validating the effectiveness of exploiting the structural information in negative embeddings.



(a) (b) (c) (d)

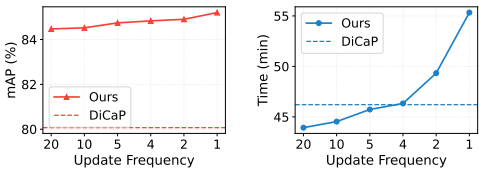
Figure 11: Hyperparameter sensitivity on VOC07 and COCO under single-label setting. We vary (a) the number αC of clusters per view, (b) the k in kNN construction, (c) the threshold τ for positive pseudo-label selection, and (d) the weight ω of the contrastive loss, while keeping other hyperparameters fixed.

F.5 Hyperparameter sensitivity

To examine the robustness of our method, we conduct a sensitivity study on VOC07 and COCO under the single-label setting. We vary one hyperparameter while fixing all others, and report the resulting mAP in Fig. 11. Overall, performance remains stable across a wide range of values, indicating that our approach is robust to all hyperparameters. In particular, we observe that setting the number of clusters to be a multiple of the number of classes (e.g., $K = \alpha C$ with $\alpha > 1$) often improves performance. This is because samples are not uniformly distributed across semantic categories and typically exhibit heterogeneous densities in the embedding space. Using finer-grained mini-clusters therefore provides a better fit to such non-uniform structure.

F.6 Influence of Pseudo-label Update Frequency

Fig. 12 studies how the pseudo-label update interval affects both runtime and performance on VOC2007 under the single-label setting. For a fair comparison, we report the total time of the *pseudo-label stage* only (20 epochs), excluding the warm-up stage. As the update becomes more frequent, performance steadily improves, but the computational cost increases sharply. We therefore choose an update interval of n=5 epochs as a practical balance between accuracy and efficiency. Moreover, under comparable runtime (using DiCaP as a reference), our method achieves noticeably higher mAP, further supporting the effectiveness of our method. Notably, our method attains superior performance even with a shorter training time.



(a) Performance (b) Runtime

Figure 12: Effect of pseudo-label update frequency on performance (a) and runtime (b) on VOC2007 in the single-label setting. DiCaP's runtime and performance are shown as references. All runtimes are measured over the pseudo-label stage (20 epochs), excluding warm-up.

F.7 Effect of Agreement Constraint in Negative Pseudo-label Mining

In the negative pseudo-label mining module, we adopt an all-view agreement rule, where an instance-label pair is assigned a negative pseudo-label only when all class-wise views consistently indicate its absence. This design is intentionally conservative, since introducing false negative labels is often more harmful than leaving some labels unassigned in multi-label learning [16, 17].

To better understand the effect of this constraint, we further evaluate relaxed agreement rules on

Table 7: Effect of relaxing the agreement constraint in negative pseudo-label mining on VOC07 and COCO14 under the single-label setting. "Constraint" denotes the minimum proportion of class-wise views that agree before assigning a negative pseudo-label.

Constraint	VOC07		COCO14	
	mAP	Recall	mAP	Recall
100%	84.73	54.61	65.70	54.01
80%	84.84	68.92	65.54	75.58
60%	84.02	75.03	65.36	81.73

VOC07 and COCO14 under the single-label setting. Specifically, instead of requiring agreement from all views, we assign a negative pseudo-label when at least a certain proportion of views agree. As shown in Table 7, relaxing the constraint increases the recall of mined negative pseudo-labels. However, the higher recall does not consistently translate into better performance. On COCO14, relaxing the agreement constraint leads to lower mAP, while on VOC07 the improvement under the 80% constraint is marginal and the 60% constraint clearly degrades performance. These results suggest that a stricter agreement rule provides a better balance for training, and support our choice of the all-view agreement strategy in the main experiments.

F.8 Ablation Study on Class-wise Contrastive Learning

To further understand the role of class-wise supervised contrastive learning, we conduct a finer-grained ablation on VOC07 and COCO under the single-label setting. In the main paper, Table 4 shows that adding the SCL module improves performance when combined with pseudo-label mining. Here, we further examine whether this improvement mainly comes from contrastive learning itself or from the expanded supervision provided by the mined pseudo-labels.

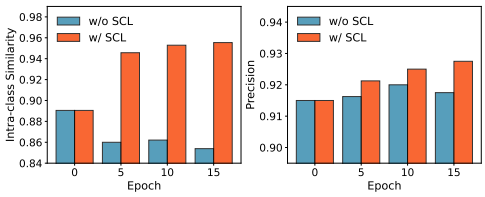
As shown in Table 8, applying class-wise contrastive learning with only the original observed labels brings limited improvement over the baseline. This is expected because the single-label setting provides very sparse supervision, making it difficult to construct sufficient positive and negative pairs for contrastive learning. In contrast, when the mined pseudo-labels are used as expanded supervision for the contrastive loss, the performance improves substantially. The full method achieves the best results, indicating that the mined pseudo-labels not only directly provide additional training targets, but also make class-wise contrastive representation learning more effective.

Table 8: Ablation study on class-wise contrastive learning under the single-label setting. "Original labels only" denotes applying contrastive learning using only observed labels, while "expanded labels only for contrastive loss" uses mined pseudo-labels only in the contrastive objective.

Method	VOC07	COCO	Average
Baseline	78.77	63.34	71.06
Baseline + CCL (original labels only)	79.56	63.88	71.72
Baseline + CCL (expanded labels only for contrastive loss)	82.76	64.63	73.70
Full method	**84.73**	**65.70**	**75.22**

F.9 Self-reinforcing feedback loop induced by supervised contrastive learning.

To reveal the interaction between representation learning and pseudo-label mining, we track (i) the precision of mined positive pseudo-labels and (ii) the average intra-class similarity among positive class-wise embeddings at several checkpoints during the pseudo-label stage (Fig. 13). At epoch 0 (immediately after warm-up), both variants start from the same embeddings and thus yield identical measurements. Once the pseudo-label stage begins, the supervised-contrastive variant consistently produces more compact positive embeddings, as evidenced by the markedly higher intra-class similarity. This improved embedding quality in turn leads to more accurate pseudo-labels at the subsequent updates, resulting in steadily higher pseudo-label precision compared to training without contrastive learning. Overall, the results support a self-reinforcing feedback loop: the mined pseudo-labels provide supervision for class-wise supervised contrastive representation learning, which improves embedding quality; the improved embeddings, in turn, enable more reliable pseudo-label mining.



(a) (b)

Figure 13: Evidence of the self-reinforcing loop on VOC: (a) average intra-class similarity among positive embeddings over training; (b) precision of mined positive pseudo-labels over training.

F.10 Controlling for the prediction head.

The main comparison follows the original prediction heads of the baselines while using the same backbone architecture. To isolate whether the gain is explained by the ML-Decoder head alone, we conduct a paired experiment on VOC07 under the MLL-PL single-label setting with seed 0. Replacing the original class-wise module of SARB with ML-Decoder improves its mAP from 77.23 to 78.12, whereas SaPu reaches 84.26 under the same setting (Table 9). Thus, the prediction head explains only part of the improvement.

Table 9: Prediction-head control on VOC07 under the MLL-PL single-label setting with seed 0. All methods use the same backbone.

Method	mAP ↑	OF1 ↑	CF1 ↑
SARB	77.23	72.86	62.84
SARB + ML-Decoder	78.12	73.31	68.65
SaPu	**84.26**	**76.78**	**72.89**

F.11 Stability under the SSMLL setting.

We repeat SaPu and the strongest baseline, DiCaP, with three seeds on VOC12 and COCO and report mean standard deviation of mAP. SaPu remains higher by 1.59 mAP on VOC12 and 0.88 mAP on COCO, showing that the improvements are consistent across random seeds rather than driven by a particular initialization.

G Compute Resources

- CPU: AMD EPYC 7642 48-Core Processor × 2
- GPU: NVIDIA GeForce RTX 3090 24G × 1
- MEM: 500G

H Limitations

SaPu introduces additional computation during pseudo-label updates, since it performs clustering and neighborhood search over class-wise embedding views. In practice, this cost is substantially reduced by updating pseudo-labels only every few epochs and by using GPU-accelerated k-means and kNN computation. Appendix F.6 further shows that, on VOC2007 under the single-label setting, SaPu achieves higher mAP than SOTA method DiCaP under comparable runtime, and even maintains superior performance with a shorter pseudo-label training time. This suggests that the additional overhead is manageable and well justified by the resulting performance gains. Nevertheless, further optimization may be needed when applying SaPu to extremely large label spaces. The main objective of this work is to reveal and validate the negative multi-cluster structure in class-wise embedding spaces, and to demonstrate that this structure can be effectively exploited for weakly supervised multi-label learning. To support this goal, we provide extensive experiments across image, text, and audio modalities, as well as both MLL-PL and SSMLL settings. Developing more efficient methods for exploiting the negative multi-cluster structure is an important direction for future work.

I Broader Impacts

This paper presents work whose goal is to advance the field of machine learning, particularly weakly supervised multi-label learning. We identify a previously overlooked phenomenon in class-wise embedding spaces: negative embeddings naturally exhibit a multi-cluster structure, and we further show that this structure can be exploited for more reliable pseudo-label mining under limited supervision. By improving the utilization of incomplete labels and unlabeled data, the proposed method may help reduce the annotation cost of multi-label learning systems and benefit applications involving image, text, and audio understanding. More broadly, our findings provide a new perspective

Table 10: Three-seed SSMLL stability analysis. We report mean $\pm$ standard deviation of mAP.

Method	VOC12	COCO
DiCaP	82.62 ± 0.80	57.87 ± 0.43
SaPu	$\mathbf{84.21 \pm 0.72}$	$\mathbf{58.75 \pm 0.31}$

on class-wise representation geometry and may inspire future structure-aware methods for pseudo-label mining and weakly supervised learning. We do not identify additional societal consequences that must be specifically highlighted beyond these general considerations.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope. Please see Abstract and Sec. 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations of the work performed by the authors. Please see Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper fully discloses all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not). Please see Sec. 5.1, Appendix D.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper provides open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results. Please see Sec. 5.1, Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments. Please see Appendix F.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments. Please see Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses both potential positive societal impacts and negative societal impacts of the work performed. Please see Appendix I.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our research does not release data or models, so the paper has no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets (e.g., code, data, models), used in the paper, are properly credited and are the license and terms of use explicitly mentioned and properly respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.