Confidence-aware Contrastive Learning for Selective Classification

Yu-Chang Wu [1 2] **Shen-Huan Lyu** [3 4 1] **Haopu Shang** [1 2] **Xiangyu Wang** [1 2] **Chao Qian** [1 2]

Abstract

Selective classification enables models to make predictions only when they are sufficiently confident, aiming to enhance safety and reliability, which is important in high-stakes scenarios. Previous methods mainly use deep neural networks and focus on modifying the architecture of classification layers to enable the model to estimate the confidence of its prediction. This work provides a generalization bound for selective classification, disclosing that optimizing feature layers helps improve the performance of selective classification. Inspired by this theory, we propose to explicitly improve the selective classification model at the feature level for the first time, leading to a novel Confidence-aware Contrastive Learning method for Selective Classification, CCL-SC, which similarizes the features of homogeneous instances and differentiates the features of heterogeneous instances, with the strength controlled by the model's confidence. The experimental results on typical datasets, i.e., CIFAR-10, CIFAR-100, CelebA, and ImageNet, show that CCL-SC achieves significantly lower selective risk than state-of-the-art methods, across almost all coverage degrees. Moreover, it can be combined with existing methods to bring further improvement.

1. Introduction

As Deep Neural Networks (DNNs) have been widely adopted across various industries, the reliability of their predictive outcomes has become increasingly critical. In many high-stakes domains such as medical diagnosis (Esteva et al., 2017), self-driving (Ghodsi et al., 2021), or security systems (Talreja et al., 2017), erroneous predictions may lead to serious repercussions (Amodei et al., 2016). The concept of selective classification for DNNs emerges in this context, providing a mechanism that allows a DNN to decide whether to make a prediction on an instance based on its prediction confidence estimation (El-Yaniv & Wiener, 2010). The goal of selective classification typically revolves around minimizing the model's selective risk while maintaining a high prediction coverage rate (Geifman & El-Yaniv, 2017).

The key issue in selective classification is how to select samples that may be predicted incorrectly and hand them over to humans for delayed prediction. A direct method is to use the maximum logit in the Softmax Layer (SR) (Hendrycks & Gimpel, 2017; Geifman & El-Yaniv, 2017) of the model as the confidence function; a higher value indicates that the model is more confident in predicting the sample. Another approach is to utilize the inferences of multiple models to estimate the prediction confidence, such as MC-dropout (Gal & Ghahramani, 2016), deep ensemble (Lakshminarayanan et al., 2017), and snapshot ensemble (Rabanser et al., 2022). Given the expensive training or prediction costs, recent works predominantly focus on individual selective classification models. SelectiveNet (SN) (Geifman & El-Yaniv, 2019) introduces an additional selection head to learn the confidence of predictions within a given coverage constraint. Deep Gamblers (DG) (Liu et al., 2019) and Self-Adaptive Training (SAT) (Huang et al., 2020; 2022) add a homogeneous logit to the output layer, serving as an "abstention head" to predict the confidence of abstaining from making predictions. Feng et al. (2023) proposed an additional Entropy-Minimization (EM) regularization loss to make the model more confident in its predictions, and applied it to the SAT method. However, their results suggest that the state-of-the-art selective classification methods with explicit selective heads actually lead to higher selective risk compared to directly using SR for confidence prediction.

In this work, we provide a generalization bound for selective classification, disclosing that optimizing feature layers to reduce variance between samples of the same category is helpful for improving the performance. In addition, the selective classification problem inherently requires models to better differentiate between correctly classified and misclassified samples, and also requires consistency between

[1]National Key Laboratory for Novel Software Technology, Nanjing University, China [2] School of Artificial Intelligence, Nanjing University, China [3]Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University, China [4]College of Computer Science and Software Engineering, Hohai University, China. Correspondence to: Chao Qian <qianc@nju.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

the predictive confidence and the reliability of the classification results. Based on these analyses, we propose to improve the performance of selective classification by explicitly optimizing the feature representation of the model for the first time, instead of focusing on modifying the classification layer as in previous works. Specifically, we propose a novel Confidence-aware Contrastive Learning method for Selective Classification named CCL-SC, which aims to pull normalized feature embeddings from the same class that are correctly classified to be closer than embeddings that are misclassified as the same class. The "pulling strength" is controlled by the model's confidence, i.e., the model pays more attention to samples with higher confidence during training, leading to a robust alignment between the model's predictive confidence and its actual accuracy.

We conduct experiments to compare our method CCL-SC with SR (Hendrycks & Gimpel, 2017; Geifman & El-Yaniv, 2017), DG (Liu et al., 2019), SAT (Huang et al., 2020; 2022), and SAT+EM (Feng et al., 2023) on four typical datasets including CIFAR-10, CIFAR-100, CelebA, and ImageNet. The results show that CCL-SC achieves significantly lower selective risk than these SOTA methods across almost all degrees of coverage. The t-SNE visualization clearly shows that CCL-SC achieves better feature representation, i.e., significant intra-class aggregation and inter-class separation in the embedded feature space. We also perform comprehensive sensitivity analyses of the hyper-parameters, demonstrating the robustness of CCL-SC, and the alignment between our our proposed method and theory. It is noteworthy that our method CCL-SC optimizes the model from a different perspective compared to previous methods, and thus it can effectively leverage techniques from existing methods to further enhance the performance of selective classification, which is empirically verified by combining with SAT (Huang et al., 2020) and EM (Feng et al., 2023).

2. Related Work

2.1. Selective Classification

Selective classification, also known as confidence-based classification, or classification with reject option (Chow, 1970), allows a model to make predictions only when it is sufficiently confident, which has been extensively studied across multiple domains in machine learning, such as support vector machines (Grandvalet et al., 2008), boosting (Cortes et al., 2016), nearest neighbours (Hellman, 1970), online learning (Cortes et al., 2018), and human assisted learning (Liu et al., 2023).

With the widespread application of deep learning, the concept of selective classification for DNNs has been receiving increasing attention, especially in situations where incorrect predictions may lead to serious consequences. Geifman & El-Yaniv (2017) proposed a method for converting trained DNNs into selective classifiers by employing two confidence functions, SR (defined as the maximal logit in the softmax layer) and MC-dropout (defined as the negative variance of aggregated predictive probabilities).

Another type of selective classification method for DNNs is to modify the classification layer and train an additional selection head (or abstention logit). SN (Geifman & El-Yaniv, 2019) is a three-headed network that includes prediction, selection, and auxiliary head, where the selection head is optimized to estimate the model's confidence in prediction for a given target coverage. DG (Liu et al., 2019) expands the original m-class problem to a $(m+1)$-class problem, and uses the extra class to estimate the confidence of the model in abstention. Similarly, SAT (Huang et al., 2020; 2022) also focuses its selection mechanism on the extra class and introduces a soft label-based training mechanism to guide the model in selecting which samples to abstain from predicting. However, recent experimental findings by (Feng et al., 2023) have shown that the methods utilizing their explicit selection heads as the confidence function are actually sub-optimal, and suggest using SR instead.

While we have been focusing on selective classification, there are two important related topics, model calibration (Guo et al., 2017) and Human-AI collaboration system (Sangalli et al., 2023). Both of them focus on the confidence of the model. Model calibration adjusts the overall confidence level of the model to align its confidence with uncertainty, which can be divided into two categories: In-process and post-hoc methods. The in-process methods involve specifically designed loss functions to optimize calibration objectives, such as Soft AvUC/ECE loss (Karandikar et al., 2021), and MMCE loss (Kumar et al., 2018). The post-hoc methods globally adjust the confidence of the model after training, such as temperature scaling (Guo et al., 2017), which, however, often do not change the ranking of confidence among samples, and thus cannot be directly used for selective classification. The definition of Human-AI collaboration (Sangalli et al., 2023) is similar to that selective classification, which uses model confidence to determine which samples are delegated to human experts. The main difference is that the goal of Human-AI collaboration is more global, that is, optimizing AUCOC (Area Under Confidence Operating Characteristics), and a loss function was proposed to directly improve the AUCOC.

2.2. Contrastive Learning

Instead of modifying the classification layer, we focus on optimizing feature representation for selective classification, which is facilitated by contrastive learning. Here, we introduce some related works on contrastive learning in both unsupervised and supervised domains.

Contrastive learning is a learning paradigm that maximizes the similarity between related samples and minimizes the similarity between unrelated samples, and has been commonly used for unsupervised representation learning. Oord et al. (2018) introduced a widely used form of contrastive loss function known as InfoNCE, which encourages the model to learn useful features by comparing each positive sample with multiple negative samples. He et al. (2020) proposed Momentum Contrast (MoCo), which uses a dynamic dictionary and a momentum encoder to solve the problem of insufficient diversity of negative samples caused by limited batch sizes in end-to-end training methods (Oord et al., 2018; Bachman et al., 2019), as well as the problem of inconsistent features caused by slow update of features in memory bank methods (Wu et al., 2018). Recently, contrastive learning has also been introduced to supervised learning, where the label information is utilized to guide the division of positive and negative samples, aiming to obtain better feature representation. Khosla et al. (2020) proposed using samples with the same label as positives and those with different labels as negatives, and extended the InfoNCE loss to scenarios with multiple positives per anchor.

While contrastive learning has been widely acknowledged as an effective approach for learning feature representations, its application in selective classification remains less explored. In this work, we leverage the strengths of contrastive learning to improve the feature representation of the model, enabling the model to better distinguish between correctly classified and incorrectly classified samples.

3. Selective Classification Problem

Let $\mathcal{X}$ and $\mathcal{Y}$ denote the feature space and the label space, respectively. Let $\mathcal{D}$ be an unknown data distribution over $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{F}$ and $\mathcal{G}$ denote two families of functions mapping $\mathcal{X}$ to $[0,1]^k$ and $[0,1]$, respectively. Our goal is to learn a selective classification model $(f, g) \in \mathcal{F} \times \mathcal{G}$:

$$(f,g)(\boldsymbol{x}, y) = \begin{cases} f(\boldsymbol{x}) & \text{if } g(\boldsymbol{x}) \geq h; \\ \text{Abstain} & \text{if } g(\boldsymbol{x}) < h. \end{cases} \tag{1}$$

Here, $f : \mathcal{X} \to [0,1]^k$ represents a conventional classifier that outputs a probability vector for k classes, with the predictive class $\hat{y}$ determined by $\hat{y} = \arg\max_j f_j(\boldsymbol{x})$, and $g : \mathcal{X} \to [0,1]$ is a selective function that estimates the confidence of $f(\boldsymbol{x})$ (also known as the confidence function), serving as a binary qualifier for f. That is, the model only predicts when $g(\boldsymbol{x})$ exceeds a predetermined threshold h.

Evaluating the performance of a selective classifier often involves two metrics: *coverage* and *selective risk* (Geifman & El-Yaniv, 2017). Coverage relies only on the selective function g, which is defined as:

$$\phi(g) \triangleq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\ I[g(\boldsymbol{x}) \geq h],$$

where the indicator function $I[\cdot]$ is 1 if the inner expression is true and 0 otherwise. Selective risk is defined as:

$$R(f,g) \triangleq \frac{\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\{L[f(\boldsymbol{x}),y] \cdot I[g(\boldsymbol{x}) \geq h]\}}{\phi(g)}, \tag{2}$$

where L is typically the $0/1$ loss for classification. Thus, coverage $\phi(g)$ measures the ratio of instances that are classified by the model, and selective risk $R(f,g)$ measures the loss of the model when making predictions. In this paper, we follow the common modeling of the selective classification problem (Geifman & El-Yaniv, 2019; Huang et al., 2020; Feng et al., 2023), i.e., to minimize the selective risk within a given target coverage c_{target}:

$$\min R(f,g) \quad \text{s.t.} \quad \phi(g) \geq c_{\text{target}}. \tag{3}$$

4. Theoretical Analysis

In this section, we analyze the generalization performance of a DNN-based selective model for selective classification. For the conventional classifier f of a selective model (f, g), we denote its feature embedding layer as c, and the final classification layer as l, i.e., $f = l \circ c : \mathcal{X} \to [0,1]^k$. For a sample $\boldsymbol{x}$ with its corresponding label y, we denote the output of the feature embedding layer c as $c(\boldsymbol{x})$, i.e., the non-normalized feature embedding of $\boldsymbol{x}$.

For analytical convenience, we add the coverage constraint in Eq. (3) to the objective function (i.e., selective risk in Eq. (2)) as a penalty term, yielding the following selective classification loss:

$$L_0(f,g,\boldsymbol{x},y) = L[f(\boldsymbol{x}),y]\cdot I[g(\boldsymbol{x}) \geq h] + \lambda \cdot I[g(\boldsymbol{x}) < h],$$

where we use the $0/1$ loss $L[f(\boldsymbol{x}),y] = I[\arg\max_j f_j(\boldsymbol{x}) \neq y]$, and $\lambda > 0$ is the penalty coefficient which regulates the trade-off between minimizing selective risk and achieving high coverage rates to satisfy the constraint. Thus, the learning problem requires utilizing a set of labeled samples $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$, which are assumed to be independently and identically distributed and drawn from the unknown data distribution $\mathcal{D}^m$, to determine a pair $(f, g) \in \mathcal{F} \times \mathcal{G}$ that achieves a small expected selective classification loss $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[L_0(f,g,\boldsymbol{x},y)]$.

Margin loss is commonly used to analyze the generalization error of models (Mohri et al., 2018; Lyu et al., 2019; 2022; Wu et al., 2022; He et al., 2024). Here, we employ the margin loss associated with Max Hinge (Cortes et al., 2023)

$$L_{\text{MH}}^{\rho,\rho'}(f,g,\boldsymbol{x},y) = \max\Big\{ \max\{1 + \frac{\alpha}{2}(\frac{g(\boldsymbol{x})}{\rho'} - \frac{\gamma(\boldsymbol{x})}{\rho}), 0\}, \max\{\lambda(1 - \beta\frac{g(\boldsymbol{x})}{\rho'}), 0\}\Big\},$$

as a surrogate loss function for the selective classification loss $L_0(f, g, \boldsymbol{x}, y)$, where ρ, ρ' are two parameters associated with the minimum margins of f and g, respectively, $\alpha, \beta > 0$, and $\gamma(\boldsymbol{x})$ is the margin of sample $\boldsymbol{x}$ on the classification layer of f, i.e., $\gamma(\boldsymbol{x}) \triangleq f_y(\boldsymbol{x}) - \max_{j \neq y} f_j(\boldsymbol{x})$. The margin loss $L_{\mathrm{MH}}^{\rho,\rho'}(f, g, \boldsymbol{x}, y)$ is actually a convex upper bound on $L_0(f, g, \boldsymbol{x}, y)$. The first term $\max\{1 + \frac{\alpha}{2}(\frac{g(\boldsymbol{x})}{\rho'} - \frac{\gamma(\boldsymbol{x})}{\rho}), 0\}$ of $L_{\mathrm{MH}}^{\rho,\rho'}$ indicates that the samples chosen for classification but classified incorrectly should be either correctly classified with a ρ-margin or modified in confidence to be rejected with a ρ'-margin (i.e., emphasizing that the model will not predict incorrectly when choosing to make predictions). The second term $\max\{\lambda(1 - \beta\frac{g(\boldsymbol{x})}{\rho'}), 0\}$ indicates that every sample subjected to rejection should have its confidence adjusted to be selected for classification with a ρ'-margin (i.e., emphasizing the model's selection of as many samples as possible to predict).

For a feed-forward DNN $f = l \circ c$ with ReLU activation function and its associated selective function g, we prove in Theorem 4.1 that the selective classification generalization error of (f, g) can be bounded by the empirical margin loss $\mathbb{E}_{(\boldsymbol{x},y)\sim S}[L_{\mathrm{MH}}^{\rho,\rho'}(f, g, \boldsymbol{x}, y)]$ at its classification layer and another term positively related with $\mathrm{Var}_{\mathrm{intra}}[c(\boldsymbol{x})]$. Note that $\mathrm{Var}_{\mathrm{intra}}[c(\boldsymbol{x})] = \mathrm{tr}[\sum_{i=1}^{k} \mathrm{Cov}[c^i(\boldsymbol{x})]/k]$ denotes the intra-class variance, where $\mathrm{Cov}[c^i(\boldsymbol{x})]$ denotes the covariance matrix of the feature embeddings of all samples with label i, k is the number of classes, and tr denotes the trace of a matrix. That is, $\mathrm{Var}_{\mathrm{intra}}[c(\boldsymbol{x})]$ denotes the variance of feature representations for samples within the same class.

Theorem 4.1. *$\forall \rho, \rho', \alpha, \beta, \lambda > 0$, and $\forall \delta > 0$, with probability at least $1 - \delta$ over a training set of size m, we have:*

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[L_0(f, g, \boldsymbol{x}, y)] \leq \mathbb{E}_{(\boldsymbol{x},y)\sim S}\left[L_{\mathrm{MH}}^{\rho,\rho'}(f, g, \boldsymbol{x}, y)\right] + 4\sqrt{\frac{\|l\|_2^2 \,\mathrm{Var}_{\mathrm{intra}}[c(\boldsymbol{x})] + 4\tilde{\rho}^2 + \tilde{\rho}^2\|l\|_2^2 \ln\frac{6m}{\delta}}{\tilde{\rho}^2 m\|l\|_2^2}},$$

where $\|l\|_2$ denotes the L2-norm of the classification layer l's parameters, and $\tilde{\rho} = \min\{\rho/(4\alpha), \rho'/(4\beta\lambda + 2\alpha)\}$.

Theorem 4.1 discloses that the generalization performance of a selective classification model is associated not only with the empirical margin loss at the classification layer but also with the feature representation at the feature layer. Specifically, a smaller intra-class variance of feature representation will enhance the generalization performance of selective classification. The proof is accomplished by bounding the Kullback–Leibler divergence term in a PAC-Bayesian lemma with the variance of perturbation, which is related to the intra-class variance $\mathrm{Var}_{\mathrm{intra}}[c(\boldsymbol{x})]$ of feature representation and the margin parameter $\tilde{\rho}$, and the proof details are provided in Appendix B due to space limitation.

5. CCL-SC Method

For selective classification, previous works focus on modifying the classification layer of a model to enable the model to better estimate its prediction confidence (Geifman & El-Yaniv, 2019; Liu et al., 2019; Huang et al., 2020; Feng et al., 2023). Inspired by Theorem 4.1, we improve the model's selective classification performance from a new perspective, that is, we optimize the feature representation of the model to aggregate the feature representations of samples in the same category, which aligns naturally with the paradigm of contrastive learning. For the loss function of the classification layer, we utilize the cross-entropy loss, which is a smooth relaxation of the margin loss and is easier to optimize for DNNs (Cao et al., 2019). SR (Hendrycks & Gimpel, 2017; Geifman & El-Yaniv, 2017), i.e., the maximum predictive class score $\max_j f_j(\boldsymbol{x})$, is used as the selective function g in our method.

Contrastive learning has been used in supervised learning (Khosla et al., 2020), which simply defines positive and negative samples as those with the same label and different labels, respectively, and optimizes the feature representation by pulling positive samples closer than negative samples. However, previous contrastive learning methods do not consider the correctness and confidence of the prediction for the samples (Wu et al., 2018; He et al., 2020; Chen et al., 2020; Khosla et al., 2020; Zhang et al., 2022), which cannot directly meet the requirement of selective classification: Correctly classified samples should be separated from misclassified samples, and the model's confidence function should reflect the reliability of its classifications, i.e., samples with higher confidence are more likely to be classified correctly. To address this issue, we redefine the positive and negative samples according to the predicted results of the current model: a sample is positive/negative if the prediction matches the anchor's label and is correct/incorrect. Then, we design a new contrastive loss function to separate the feature representations of correctly classified and misclassified samples by making the anchor's features more similar to its positive samples and less similar to its negative samples; and to pay more attention to samples with higher confidence during training by weighting the loss with the model's SR.

Figure 1 briefly illustrates our proposed method CCL-SC. In the following, we will provide detailed introduction to its key components, i.e.,

- How to define and construct positive/negative samples?
- How to design a loss function to incentivize the model to learn features conducive to selective classification, and to make it sensitive to the model's SR?
- How to use the proposed loss to train the model?

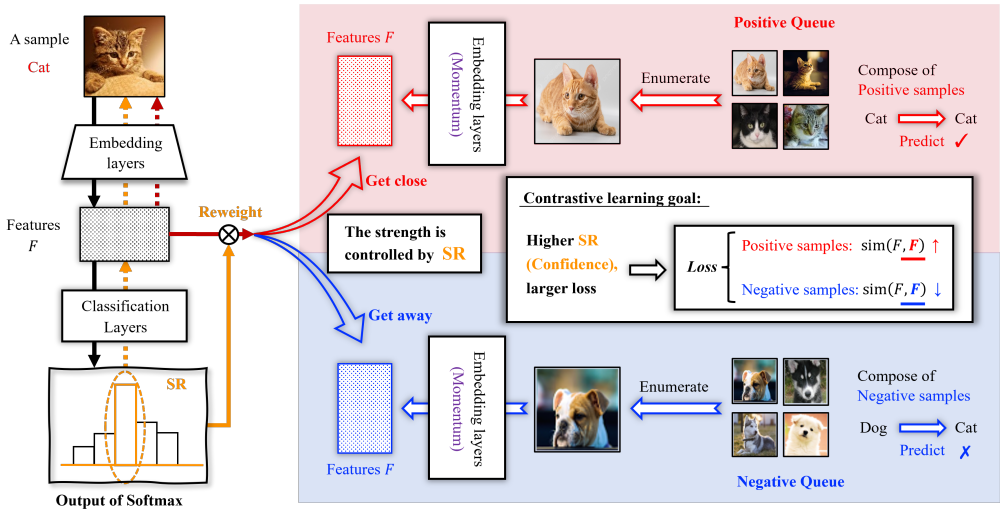


Figure 1. Illustration of the proposed CCL-SC method. The right part outlines our definition of positive/negative samples: a sample is positive/negative if the prediction matches the anchor's label and is correct/incorrect. Two independent queues store positive and negative samples, respectively. The middle part displays the characteristic of the proposed CSC loss: prompting the model to separate correctly classified and misclassified samples at the feature level and focus on samples with high prediction confidence. The black arrow on the left represents forward calculation, while the yellow and red ones represent backpropagation of the cross-entropy and CSC loss, respectively.

5.1. Constructing Positive and Negative Samples

To enhance the model's ability to discern the correctness of predictions, we define positive and negative samples based on whether the model predicts the sample correctly. A positive sample $\boldsymbol{x}_p$ for the anchor $\boldsymbol{x}$ with label y is defined as a sample that is predicted to belong to the class y and is correctly classified. A negative sample $\boldsymbol{x}_n$ is defined as a sample that is incorrectly predicted to belong to the class y. In other words, although the true label of the negative sample is different from y, the model incorrectly classifies it into the same class. The right part of Figure 1 illustrates the definition of positive and negative samples.

Due to the dependency on the model predictions, positive and negative samples have to be sampled from batches. However, sampling solely from the current mini-batch is often insufficient, particularly when dealing with a large number of classes, such as the ImageNet dataset with 1000 classes. To address this issue, we adopt the approach used in MoCo (He et al., 2020), introducing queues as dictionaries to reuse samples from different batches, while utilizing a momentum encoder (denoted as $f_{\boldsymbol{\theta}_\mathrm{m}}$) to generate sample features. The parameters $\boldsymbol{\theta}_\mathrm{m}$ of the momentum encoder are updated based on the original encoder parameters $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}_\mathrm{m} \leftarrow q \cdot \boldsymbol{\theta}_\mathrm{m} + (1-q) \cdot \boldsymbol{\theta}, \tag{4}$$

where $q \in [0, 1)$ is the momentum coefficient to control the magnitude of updates. But unlike MoCo, our method maintains two separate queues P and Q, to store positive and negative samples, respectively. Each element in the queue is a tuple composed of the normalized feature and the predicted class of a sample. We denote $P(y) \subseteq P$ and $N(y) \subseteq N$ as the positive samples and negative samples for an anchor with label y. For simplicity, we use queues of the same size s to store positive and negative samples.

5.2. Confidence-aware Supervised Contrastive Loss

Let $\boldsymbol{z}_i = c(\boldsymbol{x}_i)/\|c(\boldsymbol{x}_i)\|$ denote the normalized feature embedding of sample $\boldsymbol{x}_i$ through the embedding layer c. Then, $\boldsymbol{z}_i \cdot \boldsymbol{z}_j$ is just the cosine similarity $c(\boldsymbol{x}_i) \cdot c(\boldsymbol{x}_j)/(\|c(\boldsymbol{x}_i)\| \cdot \|c(\boldsymbol{x}_j)\|)$ between the feature representations of samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. We introduce a new loss function called Confidence-aware Supervised Contrastive (CSC) loss, by combining a contrastive learning loss in the form of infoNCE (Oord et al., 2018) with the predictive confidence SR of the model. Given an anchor $\boldsymbol{x}$ with label y, its CSC loss is defined as

$$L_\mathrm{CSC} = \frac{\max_j f_j(\boldsymbol{x})}{-|P(y)|} \sum_{\boldsymbol{x}_p \in P(y)} \log \frac{\exp(\boldsymbol{z} \cdot \boldsymbol{z}_p/\tau)}{\sum_{\boldsymbol{x}_a \in A(y)} \exp(\boldsymbol{z} \cdot \boldsymbol{z}_a/\tau)},$$

where $\max_j f_j(\boldsymbol{x})$ is SR, serving as a weight coefficient to dynamically adjust the magnitude of L_CSC, $A(y) = N(y) \cup \{\boldsymbol{x}_p\}$ denotes the set of all negative samples of the anchor $\boldsymbol{x}$ and the current positive sample $\boldsymbol{x}_p$, and τ is a temperature hyper-parameter controlling the emphasis of L_CSC on difficult samples. We set τ to a commonly used value of 0.1. L_CSC compels the anchor to be closer to the positive samples and farther from the negative samples, thereby achieving the goal of distinguishing between correctly classified and misclassified samples; and focusing on the samples with high confidence by combining SR.

Enhanced feature learning for selective classification. Unlike previous contrastive learning methods, the CSC loss utilizes both the label and prediction information of samples, providing strong supervision for each anchor. Through the contrastive learning of multiple positive and negative samples, the CSC loss encourages the model to learn features that can better distinguish between correct and incorrect predictions, enabling the model to learn more robust embedding spaces. Note that to ensure a fair comparison with previous selective classification methods, we do not acquire positive samples through additional data augmentation techniques, which was common in contrastive learning.

Mining samples with high confidence but poor features. The gradient of L_{CSC} with respect to the normalized feature embedding $\boldsymbol{z}$ of the anchor $\boldsymbol{x}$ can be calculated as

$$\frac{\max_j f_j(\boldsymbol{x})}{-\tau}\left(\sum_{\boldsymbol{x}_p\in P(y)}\left(\frac{1}{|P(y)|}-X_{\boldsymbol{z},\boldsymbol{z}_p}\right)\boldsymbol{z}_p-\sum_{\boldsymbol{x}_n\in N(y)}X_{\boldsymbol{z},\boldsymbol{z}_n}\boldsymbol{z}_n\right),$$

where $X_{\boldsymbol{z},\boldsymbol{z}_j}=\exp(\boldsymbol{z}\cdot\boldsymbol{z}_j/\tau)/\sum_{\boldsymbol{x}_a\in N(y)\cup\{\boldsymbol{x}_j\}}\exp(\boldsymbol{z}\cdot\boldsymbol{z}_a/\tau)$. Thus, the higher the value of the confidence SR (i.e., $\max_j f_j(\boldsymbol{x})$) and the poorer the feature representation (i.e., dissimilar from positive samples and similar to negative samples), the larger the gradient scale of the CSC loss for $\boldsymbol{z}$. By modulating the loss with the model's SR, the CSC loss directly prioritizes learning from instances where the model is more certain, since the loss of the samples with low prediction confidence is small. Thus, the features of samples with low prediction confidence will be distinguished from those of high-confidence samples, improving the discriminability of samples with different confidence levels. Similar to the supervised contrastive loss in (Khosla et al., 2020), the CSC loss has the ability to mine difficult samples, that is, the model will pay more attention to samples with high prediction confidence but poor feature representation.

5.3. Training with CSC Loss

We now introduce the training method with the proposed CSC loss. Different from the conventional two-stage training manner "pre-train then finetune" used in typical contrastive learning methods (He et al., 2020; Chen et al., 2020; Khosla et al., 2020), we employ a one-stage manner to optimize the classification layers and convolutional layers of the model together, because the CSC loss involves the predictive information of the model.

The model f is initially trained using the cross-entropy loss L_{CE} for E_s epochs. When $e \geq E_s$ and meanwhile both queues P and Q have been updated more than s tuples, the training gradient consists of two parts: the gradient of the CSC loss L_{CSC} returned from the last feature embedding layer c, and the gradient of the cross-entropy loss L_{CE} returned from the classification layer l. We use a weight coefficient w to balance these two loss items. The parameters $\boldsymbol{\theta}$ of the model f will be updated using an optimizer based on the combined loss. As introduced in Section 5.1, when $e \geq E_s$, a momentum encoder $f_{\boldsymbol{\theta}_{\text{m}}}$ is used to generate positive and negative samples. When the epoch e equals E_s, the parameters $\boldsymbol{\theta}_{\text{m}}$ of the momentum encoder $f_{\boldsymbol{\theta}_{\text{m}}}$ are initialized to the parameters $\boldsymbol{\theta}$ of the current model f, which ensures that the momentum encoder has a favorable initial accuracy. After that, the parameters $\boldsymbol{\theta}_{\text{m}}$ of the momentum encoder will be updated according to Eq. (4) at each training step. The samples generated by the momentum encoder will be used to update the queues P and Q, which store positive and negative samples, respectively. Note that $f_{\boldsymbol{\theta}_{\text{m}}}$ will not be optimized by the optimizer. The pseudo-code of the training method is shown in Algorithm 1 in Appendix C.

6. Experiments

In this section, we will give the experimental settings and results. Due to space limitation, some details are shown in Appendix D to I. The codes are provided in `https://github.com/lamda-bbo/CCL-SC`.

Datasets We conduct experiments on four commonly used datasets, i.e., CelebA (Liu et al., 2015), CIFAR-10/CIFAR-100 (Krizhevsky et al., 2009), and ImageNet (Deng et al., 2009). CelebA is a large-scale face attributes dataset, consisting of over 200,000 celebrity images, and the challenging label 'attractive' is used as the target for binary classification. CIFAR-10 and CIFAR-100 are two datasets containing images across 10 and 100 categories, with 5,000 and 500 images per category, respectively. ImageNet contains 1000 categories of images, with 1300 images per category. The experiments on CIFAR-10, CIFAR-100, and CelebA are run with 5 seeds, and those on ImageNet are run with 3 seeds.

Baselines We compare our method against SOTA selective classification methods, including SAT (Huang et al., 2020; 2022), SAT with Entropy-Minimization regularization (SAT+EM) (Feng et al., 2023), and DG (Liu et al., 2019). Based on (Feng et al., 2023), we include the results of using SR as the confidence function for these methods as well. We also compare with a common baseline denoted as SR (Hendrycks & Gimpel, 2017; Geifman & El-Yaniv, 2017), which is a vanilla classifier trained via the cross-entropy loss and uses SR as the confidence function. We do not compare with SN (Geifman & El-Yaniv, 2019) since previous methods such as SAT+EM have already demonstrated their superiority over SN. Additionally, SN requires retraining for different coverage rates, resulting in significantly higher training costs compared to other methods.

Hyper-parameters For each dataset, we utilize 20% of the training set as the validation set to tune hyper-parameters. We test the momentum coefficient $q \in \{0.9, 0.999, 0.999\}$,

Table 1. Selective risk (%) on CIFAR-10 for various coverage rates (%). The mean and standard deviation are calculated over 5 trials. The best entries are marked in bold. The symbol '•'/'○' indicates that CCL-SC is significantly better/worse than the corresponding method, according to the Wilcoxon rank-sum test with significance level 0.05.

Coverage	CCL-SC	SAT+EM+SR	SAT+EM	SAT+SR	SAT	DG+SR	DG	SR
100	**5.97±0.11**	6.14±0.07	6.14±0.07	6.16±0.13•	6.16±0.13•	6.34±0.16•	6.34±0.16•	6.25±0.07
99	**5.32±0.05**	5.61±0.06•	5.58±0.06•	5.63±0.09•	5.63±0.11•	5.77±0.16•	5.75±0.18•	5.69±0.07•
98	**4.87±0.04**	5.11±0.07•	5.14±0.03•	5.16±0.07•	5.14±0.13•	5.27±0.19•	5.23±0.19•	5.16±0.06
97	**4.41±0.07**	4.66±0.09•	4.69±0.06•	4.67±0.05•	4.74±0.11•	4.82±0.16•	4.70±0.20	4.67±0.04•
95	**3.56±0.06**	3.85±0.09•	3.93±0.06•	3.87±0.08•	3.97±0.11•	3.98±0.13•	3.83±0.13•	3.81±0.07•
90	**2.01±0.07**	2.20±0.07•	2.34±0.09•	2.26±0.12•	2.35±0.16•	2.37±0.13•	2.26±0.10•	2.19±0.12•
85	**1.10±0.06**	1.23±0.07•	1.31±0.07•	1.27±0.09•	1.36±0.10•	1.39±0.09•	1.32±0.11•	1.25±0.13•
80	0.69±0.08	**0.67±0.07**	0.71±0.07	0.71±0.09	0.74±0.09	0.86±0.08•	0.72±0.10	0.68±0.08

and the weight coefficient $w \in \{0.1, 0.5, 1.0\}$. For the queue size s, we set it based on the number of classes in the dataset: for datasets with fewer classes such as CelebA and CIFAR-10, $s = 300$; for datasets with more classes such as CIFAR-100 and ImageNet, we set $s = 3000$ and $s = 10000$, respectively. We train the model on the entire training set to evaluate performance. Detailed hyperparameter settings for each method on each dataset are provided in Appendix D.1.

Networks and Training Following prior work, for CIFAR-10 and CIFAR-100, we employ VGG16 (Simonyan & Zisserman, 2015) as the backbones of selective classifiers. ResNet34 and ResNet18 (He et al., 2016) are utilized for ImageNet and CelebA, respectively. For the same dataset, all compared methods utilize the same data augmentation and training parameters to ensure a fair comparison. The detailed settings of training parameters and data augmentation are provided in Appendix D.2 and D.3, respectively.

6.1. Comparison with State-of-the-art Methods

Table 1 shows the results on CIFAR-10, which has a low classification difficulty but is most widely used in selective classification. It can be observed that the selective risk of all methods on CIFAR-10 is below 1% at coverage 80%. We do not include the results for lower coverage as the selective risk approaches almost zero thereafter. Our method CCL-SC achieves the lowest selective risk when coverage is at least 85%. However, we can also see that the discrimination of different methods on CIFAR-10 is relatively low, which is consistent with previous works (Huang et al., 2020; 2022; Feng et al., 2023), mainly due to the insufficient number of misclassified samples. As a result, we focus on the other three datasets that are not saturated with accuracy.

Table 2 shows the selective risk of different methods at various coverage on CelebA and CIFAR-100. For CelebA, CCL-SC performs the best across all degrees of coverage. Compared to any other method, CCL-SC performs significantly better on at least $7/13$ coverage rates, according to the Wilcoxon rank-sum test (Wilcoxon, 1945) with significance level 0.05. For CIFAR-100, CCL-SC achieves the lowest selective risk when coverage is at least 50% or equal to 10%, and is only worse than SAT-related methods in the range of 20% to 40%. According to the significance test, CCL-SC is significantly better than SAT-related methods when the coverage is at least 60%, and only significantly worse than SAT+SR at 30% coverage.

We can also observe from Table 2 that those methods (i.e., SAT+EM, SAT, DG) based on the additional selection head will generally become better if using SR as the confidence function, as observed in (Feng et al., 2023). Furthermore, the improvement is larger on CIFAR-100 than on CelebA with only two classes, which could be attributed to the increased difficulty of learning additional logits as the number of classes increases. Note that the relative rankings of SAT+EM, SAT, and DG in our experiments are also consistent with the overall rankings reported in previous works (Huang et al., 2020; 2022; Feng et al., 2023).

Table 3 shows the comparison with the two currently best-performing methods SAT+EM+SR and SAT on ImageNet. Note that their results are directly from (Feng et al., 2023), and the comparison is fair as all training settings are same. We can observe that CCL-SC always performs the best, except at 10% coverage. However, one may argue that the performance improvement is due to the accuracy improvement, because CCL-SC achieves 1% accuracy improvement over other methods at full coverage. To mitigate this concern, we also load checkpoints when training is completed in just 100 epochs, and the results are shown in the CCL-SC* column of Table 3. Now our method achieves slightly lower accuracy than SAT+EM+SR at full coverage, but still outperforms both SAT+EM+SR and SAT when the coverage is between 20% and 80%.

6.2. Comparison with Other Related Methods

As described in Section 2.1, selective classification is closely related to model calibration and Human-AI collaboration. In this section, we introduce five methods from model calibration, including Focal loss (Lin et al., 2017), Adaptive Focal loss (Mukhoti et al., 2020), Soft AvUC loss (Karandikar

Table 2. Selective risk (%) on CelebA and CIFAR-100 for various coverage rates (%). The mean and standard deviation are calculated over 5 trials. The best entries are marked in bold. The symbol '•'/'○' indicates that CCL-SC is significantly better/worse than the corresponding method, according to the Wilcoxon rank-sum test with significance level 0.05. The w/t/l denotes the number of cases where the selective risk of CCL-SC is significantly lower, almost equivalent, or significantly higher, compared to the corresponding method.

CelebA								
Coverage	CCL-SC	SAT+EM+SR	SAT+EM	SAT+SR	SAT	DG+SR	DG	SR
100	**18.71±0.16**	19.04±0.30	19.04±0.30	19.20±0.37•	19.20±0.37•	19.26±0.23•	19.26±0.23•	19.30±0.13•
95	**17.00±0.09**	17.55±0.33•	17.74±0.31•	17.65±0.33•	17.82±0.32•	17.82±0.26•	18.06±0.29•	17.80±0.12•
90	**15.47±0.14**	16.06±0.36•	16.24±0.31•	16.17±0.31•	16.38±0.30•	16.27±0.32•	16.87±0.28•	16.30±0.12•
85	**14.03±0.17**	14.56±0.36•	14.72±0.28•	14.65±0.27•	14.92±0.21•	14.81±0.27•	15.61±0.27•	14.81±0.11•
80	**12.51±0.21**	13.12±0.39•	13.21±0.29•	13.17±0.29•	13.38±0.21•	13.43±0.34•	14.36±0.30•	13.37±0.13•
75	**11.05±0.20**	11.69±0.37•	11.72±0.36•	11.78±0.26•	11.90±0.19•	12.11±0.32•	13.01±0.33•	11.98±0.15•
70	**9.73±0.14**	10.33±0.32•	10.35±0.28•	10.41±0.27•	10.49±0.23•	10.81±0.37•	11.73±0.40•	10.62±0.14•
60	**7.16±0.09**	7.84±0.43•	7.73±0.37•	8.00±0.35•	7.94±0.28•	8.28±0.43•	8.99±0.43•	8.06±0.29•
50	**4.93±0.16**	5.51±0.44•	5.41±0.42	5.71±0.25•	5.68±0.18•	6.17±0.58•	6.27±0.37•	5.92±0.23•
40	**3.09±0.14**	3.66±0.54	3.50±0.49	3.82±0.22•	3.77±0.17•	4.35±0.67•	3.86±0.28•	4.03±0.24•
30	**1.87±0.15**	2.16±0.45	2.06±0.35	2.33±0.16•	2.15±0.21	2.81±0.68	2.16±0.26	2.49±0.24•
20	**0.92±0.12**	1.10±0.25	1.02±0.21	1.21±0.14•	1.12±0.13	1.65±0.37•	1.10±0.12	1.45±0.12•
10	**0.25±0.08**	0.41±0.12	0.37±0.12	0.54±0.12•	0.49±0.06•	0.80±0.21•	0.41±0.14	0.50±0.12•
w/t/l	/	8/5/0	7/6/0	14/0/0	12/2/0	13/1/0	11/3/0	14/0/0
Avg. Rank	**1.00**	2.54	2.77	4.46	5.00	6.85	6.62	6.15
CIFAR-100								
Coverage	CCL-SC	SAT+EM+SR	SAT+EM	SAT+SR	SAT	DG+SR	DG	SR
100	**26.55±0.26**	26.96±0.14•	26.96±0.14•	26.98±0.16•	26.98±0.16•	27.12±0.30•	27.12±0.30•	27.19±0.33•
95	**23.54±0.15**	24.14±0.12•	24.16±0.10•	24.17±0.18•	24.25±0.14•	24.28±0.26•	24.35±0.38•	24.37±0.29•
90	**20.97±0.20**	21.52±0.22•	21.56±0.10•	21.59±0.15•	21.68±0.16•	21.72±0.29•	21.84±0.40•	21.86±0.31•
85	**18.57±0.20**	19.08±0.21•	19.14±0.18•	19.09±0.22•	19.18±0.19•	19.35±0.16•	19.43±0.32•	19.43±0.40•
80	**16.07±0.15**	16.71±0.18•	16.73±0.23•	16.64±0.19•	16.84±0.23•	16.89±0.19•	17.20±0.37•	17.09±0.45•
75	**13.60±0.19**	14.30±0.19•	14.53±0.28•	14.21±0.23•	14.49±0.22•	14.44±0.29•	14.97±0.48•	14.58±0.39•
70	**11.23±0.16**	11.94±0.21•	12.07±0.20•	11.83±0.18•	12.11±0.20•	12.05±0.42•	12.81±0.64•	12.28±0.29•
60	**6.83±0.15**	7.51±0.16•	7.83±0.07•	7.54±0.09•	7.79±0.22•	7.75±0.61•	8.91±0.75•	7.71±0.33•
50	**3.95±0.22**	4.08±0.15	4.30±0.15•	4.10±0.24	4.32±0.19	4.40±0.56	5.48±0.72•	4.36±0.17•
40	2.29±0.33	2.12±0.15	2.37±0.20	**2.00±0.04**	2.38±0.10	2.45±0.50	3.16±0.49•	2.18±0.13
30	1.26±0.17	1.05±0.11	1.37±0.16	**0.96±0.10**○	1.21±0.10	1.68±0.59	2.07±0.57	1.29±0.11
20	0.71±0.12	**0.54±0.15**	0.77±0.08	**0.54±0.14**	0.67±0.12	1.14±0.50	1.65±0.49•	0.78±0.11
10	**0.36±0.08**	0.48±0.12	0.58±0.21	0.42±0.16	0.48±0.23	0.74±0.29•	1.28±0.30•	0.58±0.21
w/t/l	/	8/5/0	9/4/0	8/4/1	8/5/0	9/4/0	12/0/0/	9/4/0
Avg. Rank	**1.69**	2.23	4.54	2.46	4.69	6.00	7.62	6.23

Table 3. Selective risk (%) on ImageNet for various coverages (%). The mean and standard deviation are calculated over 3 trials. The best and runner-up entries are bolded and underlined, respectively.

Cov.	CCL-SC	SAT+EM+SR	SAT	CCL-SC*
100	**26.26±0.10**	27.27 ± 0.05	27.41 ± 0.08	27.31±0.04
90	**20.68±0.07**	21.57 ± 0.19	22.67 ± 0.24	21.71±0.03
80	**15.76±0.07**	16.83 ± 0.06	18.14 ± 0.28	16.78±0.03
70	**11.39±0.10**	12.34 ± 0.11	13.88 ± 0.14	12.25±0.04
60	**7.55±0.09**	8.45 ± 0.05	10.11 ± 0.15	8.34±0.03
50	**4.79±0.04**	5.57 ± 0.17	6.82 ± 0.07	5.33±0.06
40	**2.95±0.04**	3.77 ± 0.00	4.32 ± 0.33	3.35±0.05
30	**1.83±0.05**	2.32 ± 0.15	2.68 ± 0.14	2.03±0.04
20	**1.22±0.05**	1.35 ± 0.20	1.82 ± 0.13	1.23±0.04
10	0.72±0.05	**0.55 ± 0.05**	1.27 ± 0.34	0.68±0.07

et al., 2021), Soft ECE loss (Karandikar et al., 2021), and MMCE loss (Kumar et al., 2018), as well as AUCOC loss (Sangalli et al., 2023) from Human-AI collaboration into selective classification. Tabel 8 in Appeidx E shows that CCL-SC has the lowest selective risk at various coverage rates compared to these methods.

6.3. Alignment between Proposed Theory and Method

Theorem 4.1 discloses that a smaller intra-class variance of feature representation of a model will enhance its generalization performance of selective classification, and our method CCL-SC explicitly optimizes this aspect. Although previous experiments have confirmed the superiority of our method in the generalization performance of selective classification,

we still need to answer through experiments whether our method has really obtained lower intra-class variance and tighter bounds in Theorem 4.1 compared to other selective classification methods. We show the changes of the intra-class variance and the bound in Theorem 4.1 of different methods during the training process on CIFAR-100 in Figure 2 in Appendix F. It can be observed that CCL-SC does have the lowest intra-class variance and the lowest bound. It is worth mentioning that the relative order of the intra-class variance and the bounds of different methods is consistent with their actual order of selective risk, and the methods with similar selective risk (such as SAT+EM and SAT) also have similar intra-class variance/bound, indicating the importance of intra-class variance for selective classification performance and the usefulness of our bound.

6.4. Learned Feature Representation

Because our method CCL-SC explicitly optimizes the feature layer of the model, we compare its learned feature representations with those of SR trained only using the cross-entropy loss on CIFAR-10 at coverage 95%. The t-SNE (Van der Maaten & Hinton, 2008) visualization shown in Figure 3 in Appendix G clearly demonstrates that CCL-SC achieves more significant inter-class separation and intra-class aggregation in the feature space, which confirms that optimizing the feature layer contributes to performance improvement in selective classification.

6.5. Ablation Studies and Hyper-parameter Sensitivity

Next, we conduct ablation studies and parameter sensitivity analysis on CIFAR-100 for the proposed method CCL-SC.

SR-weighted We verify whether the SR-weighted manner in the proposed CSC loss L_{CSC} really improves the performance of selective classification. Table 9 in Appendix H shows that the original CCL-SC method using the SR-weighted CSC loss consistently achieves lower selective risk than that using the unweighted CSC loss across all coverage degrees.

The contrastive learning method of CCL-SC We first conduct ablation experiments for the construction of negative samples. For convenience, we name the ablation method CCL-SC2. For the negative samples of the samples that are correctly classified as class y, CCL-SC2 contains not only the samples misclassified as class y in the queue defined in CCL-SC, but also the samples from other classes in the queue. Table 10 in Appendix H shows that even with the addition of negative samples, the performance of CCL-SC2 will not be improved compared to the original CCL-SC. This implies that the improvement of the performance is likely to be from the negative samples we define. To confirm this conclusion, we conduct another ablation study in which we remove the negative samples defined in CCL-SC, and only use randomly sampled samples from other categories as negative samples. For simplicity, we name this ablation method CCL-SC3. Table 11 in Appendix H shows that CCL-SC3 has a significant performance decrease compared to CCL-SC, which demonstrates the effectiveness of our strategy to construct negative samples.

We also conduct ablation experiments for the whole contrastive method of CCL-SC. Specifically, we introduce the positive and negative sample definition method and loss function from (Khosla et al., 2020) into our CCL-SC method, while keeping the other components consistent. We name this ablation method CCL-SC+SupCon. The comparison results on CIFAR-100 are shown in Table 11 in Appendix H. It can be observed that the selective classification performance of the original CCL-SC is better than CCL-SC+SupCon.

Hyper-parameter Sensitivity We then analyze the influence of four important hyper-parameters: the momentum coefficient q, queue size s, weight coefficient w, and initial epochs E_s. Tables 13 to 16 in Appendix H show that the performance of CCL-SC is generally not sensitive to their settings, i.e., CCL-SC can achieve good performance in a wide range of these hyper-parameters. Detailed results can be found in Appendix H.

6.6. Combination of CCL-SC and Existing Methods

Finally, we are to verify another benefit of CCL-SC, i.e., it can be seamlessly integrated with existing methods that optimize the model at the classification layer, because CCL-SC operates on the feature representation of the model. Here, we combine CCL-SC with SAT (Huang et al., 2020; 2022) and EM (Feng et al., 2023) methods. That is, the loss function at the classification layer is modified from the cross-entropy loss to the loss of SAT with EM regularization. Table 17 and Tabel 18 in Appendix I show that such a combination outperforms the original CCL-SC significantly.

7. Conclusion

This paper proves a generalization bound for selective classification, disclosing that optimizing feature layers to reduce intra-class variance is helpful for improving the performance. Inspired by this theory, we adapt contrastive learning to explicitly optimize the model at the feature layer, resulting in the new method CCL-SC for selective classification. Extensive experiments show that CCL-SC clearly outperforms state-of-the-art methods, and can also be naturally combined with existing techniques to bring further improvement. This work supplements previous selective classification methods which focus solely on modifying the classification layer, and might encourage the exploration of new methods considering the optimization of feature layer.

Acknowledgements

The authors want to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Science and Technology Major Project (2022ZD0116600), National Science Foundation of China (62276124, 62306104), Jiangsu Science Foundation (BK20230949), China Postdoctoral Science Foundation (2023TQ0104), Jiangsu Excellent Postdoctoral Program (2023ZB140), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

Impact Statement

This paper presents work whose goal is to advance the field of selective classification. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P. F., Schulman, J., and Mané, D. Concrete problems in AI safety. *arXiv:1606.06565*, 2016.

Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pp. 15509–15519, Vancouver, Canada, 2019.

Cao, K., Wei, C., Gaidon, A., Aréchiga, N., and Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pp. 1565–1576, Vancouver, Canada, 2019.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 1597–1607, Virtual, 2020.

Chow, C. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970.

Cortes, C., DeSalvo, G., and Mohri, M. Boosting with abstention. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, pp. 1660–1668, Barcelona, Spain, 2016.

Cortes, C., DeSalvo, G., Gentile, C., Mohri, M., and Yang, S. Online learning with abstention. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 1059–1067, Stockholm, Sweden, 2018.

Cortes, C., DeSalvo, G., and Mohri, M. Theory and algorithms for learning with rejection in binary classification. *Annals of Mathematics and Artificial Intelligence*, pp. 1–39, 2023.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, Miami, FL, 2009.

El-Yaniv, R. and Wiener, Y. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.

Feng, L., Ahmed, M. O., Hajimirsadeghi, H., and Abdi, A. H. Towards better selective classification. In *The 11th International Conference on Learning Representations (ICLR)*, Kigali, Rwanda, 2023.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 1050–1059, New York, NY, 2016.

Geifman, Y. and El-Yaniv, R. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pp. 4878–4887, Long Beach, CA, 2017.

Geifman, Y. and El-Yaniv, R. Selectivenet: A deep neural network with an integrated reject option. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 2151–2159, Long Beach, CA, 2019.

Ghodsi, Z., Hari, S. K. S., Frosio, I., Tsai, T., Troccoli, A. J., Keckler, S. W., Garg, S., and Anandkumar, A. Generating and characterizing scenarios for safety testing of autonomous vehicles. *arXiv:2103.07403*, 2021.

Grandvalet, Y., Rakotomamonjy, A., Keshet, J., and Canu, S. Support vector machines with a reject option. In *Advances in Neural Information Processing Systems 21 (NeurIPS)*, pp. 537–544, Vancouver, Canada, 2008.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 1321–1330, Sydney, Australia, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the*

2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, Las Vegas, NV, 2016.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9729–9738, Seattle, WA, 2020.

He, Y., Wu, Y., Qian, C., and Zhou, Z. Margin distribution and structural diversity guided ensemble pruning. *Machine Learning*, 113(6):3545–3567, 2024.

Hellman, M. E. The nearest neighbor classification rule with a reject option. *IEEE Transactions on Systems Science and Cybernetics*, 6(3):179–185, 1970.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *The 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

Huang, L., Zhang, C., and Zhang, H. Self-adaptive training: Beyond empirical risk minimization. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 19365–19376, virtual, 2020.

Huang, L., Zhang, C., and Zhang, H. Self-adaptive training: Bridging supervised and self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–17, 2022.

Karandikar, A., Cain, N., Tran, D., Lakshminarayanan, B., Shlens, J., Mozer, M. C., and Roelofs, B. Soft calibration objectives for neural networks. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pp. 29768–29779, virtual, 2021.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, virtual, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

Kumar, A., Sarawagi, S., and Jain, U. Trainable calibration measures for neural networks from kernel mean embeddings. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 2810–2819, Stockholm, Sweden, 2018.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pp. 6402–6413, Long Beach, CA, 2017.

Lin, T., Goyal, P., Girshick, R. B., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, Venice, Italy, 2017.

Liu, D., Mu, X., and Qian, C. Human assisted learning by evolutionary multi-objective optimization. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 12453–12461, Washington, DC, 2023.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, Santiago, Chile, 2015.

Liu, Z., Wang, Z., Liang, P. P., Salakhutdinov, R., Morency, L., and Ueda, M. Deep gamblers: Learning to abstain with portfolio theory. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pp. 10622–10632, Vancouver, Canada, 2019.

Lyu, S.-H., Yang, L., and Zhou, Z.-H. A refined margin distribution analysis for forest representation learning. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pp. 5531–5541, Vancouver, Canada, 2019.

Lyu, S.-H., Wang, L., and Zhou, Z.-H. Improving generalization of deep neural networks by leveraging margin distribution. *Neural Networks*, 151:48–60, 2022.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of Machine Learning*. MIT Press, Cambridge, MA, 2018.

Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P. H. S., and Dokania, P. K. Calibrating deep neural networks using focal loss. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 15288–15299, virtual, 2020.

Neyshabur, B., Bhojanapalli, S., and Srebro, N. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. In *The 6th International Conference on Learning Representations (ICLR)*, Vancouver, Canada, 2018.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018.

Rabanser, S., Thudi, A., Hamidieh, K., Dziedzic, A., and Papernot, N. Selective classification via neural network training dynamics. *arXiv:2205.13532*, 2022.

Sangalli, S., Erdil, E., and Konukoglu, E. Expert load matters: Operating networks at high accuracy and low manual effort. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, New Orleans, LA, 2023.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *The 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, 2015.

Talreja, V., Valenti, M. C., and Nasrabadi, N. M. Multibiometric secure system based on deep learning. *arXiv:1708.02314*, 2017.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

Wilcoxon, F. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

Wu, Y., He, Y., Qian, C., and Zhou, Z. Multi-objective evolutionary ensemble pruning guided by margin distribution. In *Proceedings of the 17th International Conference on Parallel Problem Solving from Nature (PPSN)*, pp. 427–441, Dortmund, Germany, 2022.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3733–3742, Salt Lake, UT, 2018.

Zhang, L., Chen, X., Zhang, J., Dong, R., and Ma, K. Contrastive deep supervision. In *Proceedings of the 17th European Conference on Computer Vision (ECCV)*, pp. 1–19, Tel Aviv, Israel, 2022.

A. Table of Notations

Table 4. Key symbols and notations.

Sign	Description
$\mathcal{X}$	The feature space.
$\mathcal{Y}$	The label space.
k	The number of classes.
$\mathcal{D}$	The data distribution over $\mathcal{X} \times \mathcal{Y}$.
$\mathcal{F}$	The families of predictive probability functions mapping $\mathcal{X}$ to $[0,1]^k$.
$f(\boldsymbol{x})$	The vector composed of the predicted probabilities of f for sample $\boldsymbol{x}$ across k classes.
$\mathcal{G}$	The families of confidence functions mapping $\mathcal{X}$ to $[0,1]$.
$g(\boldsymbol{x})$	The confidence of $f(\boldsymbol{x})$ estimated by the selective function g.
c	The feature embedding layer of the classifier f.
l	The final classification layer of f.
$\hat{y}$	The predictive class by f.
$\phi(g)$	The coverage relies on the selective function g.
$R(f,g)$	The Selective risk of the selective model (f,g).
h	The threshold that determines whether the model chooses to classify or not.
S	The training set.
m	The number of training samples.
P	The queue of all positive samples.
Q	The queue of all negative samples.
$\boldsymbol{z}$	The the normalized feature embedding of sample $\boldsymbol{x}$ through the embedding layer c.

B. Theorem Proofs

Theorem. $\forall \rho, \rho', \alpha, \beta, \lambda > 0$, and $\forall \delta > 0$, with probability at least $1-\delta$ over a training set of size m, we have:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[L_0(f,g,\boldsymbol{x},y)] \leq \mathbb{E}_{(\boldsymbol{x},y)\sim S}\left[L_{\mathrm{MH}}^{\rho,\rho'}(f,g,\boldsymbol{x},y)\right] + 4\sqrt{\frac{\|l\|_2^2 \operatorname{Var}_{\mathrm{intra}}[c(\boldsymbol{x})] + 4\tilde{\rho}^2 + \tilde{\rho}^2\|l\|_2^2 \ln\frac{6m}{\delta}}{\tilde{\rho}^2 m\|l\|_2^2}},$$

where $\|l\|_2$ denotes the L2-norm of the classification layer l's parameters, and $\tilde{\rho} = \min\{\rho/(4\alpha), \rho'/(4\beta\lambda + 2\alpha)\}$.

Proof. To obtain the bound of the gap between the expected selective classification loss $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[L_0(f,g,\boldsymbol{x},y)]$ and the empirical margin loss $\mathbb{E}_{(\boldsymbol{x},y)\sim S}\left[L_{\mathrm{MH}}^{\rho,\rho'}(f,g,\boldsymbol{x},y)\right]$, we start to prove a PAC-Bayesian bound:

Lemma B.1. *Let $f_{\boldsymbol{w}} : \mathcal{X} \to \mathcal{Y}$ be any predictor with parameters $\boldsymbol{w}$, and P be any distribution on the parameters that is independent of the training data. Then, for any $\rho, \rho', \alpha, \beta, \delta > 0$, with probability at least $1-\delta$ over the training set of size m, for any $\boldsymbol{w}$, and any random perturbation $\boldsymbol{u}$ s.t. $\mathbb{P}_{\boldsymbol{u}}\left[\max_{\boldsymbol{x}\in S} |f_{\boldsymbol{w}+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{w}}(\boldsymbol{x})|_2 < \min\left\{\frac{\rho}{4\alpha}, \frac{\rho'}{4\beta\lambda+2\alpha}\right\}\right] \geq 1/2$, we have*

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[L_0(f,g,\boldsymbol{x},y)] \leq \mathbb{E}_{(\boldsymbol{x},y)\sim S}\left[L_{\mathrm{MH}}^{\rho,\rho'}(f,g,\boldsymbol{x},y)\right] + 4\sqrt{\frac{D_{\mathrm{KL}}(\boldsymbol{w}+\boldsymbol{u} \mid P) + \ln\frac{6m}{\delta}}{m-1}}, \tag{5}$$

where $D_{\mathrm{KL}}(P \mid Q)$ denotes the Kullback-Leibler divergence between P and Q.

Proof of Lemma B.1. Let $\boldsymbol{w}' = \boldsymbol{w} + \boldsymbol{u}$, and $\mathcal{S}_{\boldsymbol{w}}$ be the set of perturbations with the following property:

$$\mathcal{S}_{\boldsymbol{w}} \subseteq \left\{\boldsymbol{w}' \middle| \max_{\boldsymbol{x}\in S} |f_{\boldsymbol{w}'}(\boldsymbol{x}) - f_{\boldsymbol{w}}(\boldsymbol{x})|_2 < \min\{\frac{\rho}{4\alpha}, \frac{\rho'}{4\beta\lambda+2\alpha}\}\right\}.$$

Let q be the probability density function over the parameters $\boldsymbol{w}'$. We construct a new distribution $\tilde{Q}$ over predictors $f_{\tilde{\boldsymbol{w}}}$ where $\tilde{\boldsymbol{w}}$ is restricted to $\mathcal{S}_{\boldsymbol{w}}$ with the probability density function:

$$\tilde{q}(\tilde{\boldsymbol{w}}) = \begin{cases} q(\tilde{\boldsymbol{w}}) & \text{if } \tilde{\boldsymbol{w}} \in \mathcal{S}_{\boldsymbol{w}}; \\ 0 & \text{otherwise .} \end{cases}$$

According to the lemma assumption, we have $Z = \mathbb{P}\left[\boldsymbol{w}' \in \mathcal{S}_{\boldsymbol{w}}\right] \geq 1/2$. Therefore, we can bound the change of the margins and the confidence scores for any instance:

$$\max_{i,j\in[k],\boldsymbol{x}\in S} |(|f_{\tilde{\boldsymbol{w}}}(\boldsymbol{x})[i] - f_{\tilde{\boldsymbol{w}}}(\boldsymbol{x})[j]|) - (|f_{\boldsymbol{w}}(\boldsymbol{x})[i] - f_{\boldsymbol{w}}(\boldsymbol{x})[j]|)| < \frac{\rho}{2\alpha},$$
$$\max_{\boldsymbol{x}\in S} |g_{\tilde{\boldsymbol{w}}}(\boldsymbol{x}) - g_{\boldsymbol{w}}(\boldsymbol{x})| < \frac{\rho'}{4\beta\lambda + 2\alpha}.$$

Here we define a perturbed loss function as:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[L_{\mathrm{MH}}^{\rho/2,\rho'/2}(f, g, \boldsymbol{x}, y)] = \mathbb{P}_{\mathcal{D}}\left[\gamma(\boldsymbol{x}) \leq \frac{\rho}{2\alpha}\right] \cdot \mathbb{P}_{\mathcal{D}}\left[g(\boldsymbol{x}) > -\frac{\rho'}{4\beta\lambda + 2\alpha}\right] + \lambda \cdot \mathbb{P}_{\mathcal{D}}\left[g(\boldsymbol{x}) > \frac{\rho'}{4\beta\lambda + 2\alpha}\right].$$

We can get the following bounds:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[L_0\left(f_{\boldsymbol{w}}, g_{\boldsymbol{w}}, \boldsymbol{x}, y\right)] \leq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[L_{\mathrm{MH}}^{\rho/2,\rho'/2}\left(f_{\tilde{\boldsymbol{w}}}, g_{\tilde{\boldsymbol{w}}}, \boldsymbol{x}, y\right)\right],$$
$$\mathbb{E}_{(\boldsymbol{x},y)\sim S}\left[L_{\mathrm{MH}}^{\rho/2,\rho'/2}\left(f_{\tilde{\boldsymbol{w}}}, g_{\tilde{\boldsymbol{w}}}, \boldsymbol{x}, y\right)\right] \leq \mathbb{E}_{(\boldsymbol{x},y)\sim S}\left[L_{\mathrm{MH}}^{\rho,\rho'}\left(f_{\boldsymbol{w}}, g_{\boldsymbol{w}}, \boldsymbol{x}, y\right)\right].$$

Finally, using the proof of Lemma 1 in (Neyshabur et al., 2018), with probability $1-\delta$ over the training set we have:

$$\begin{aligned}
\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[L_0\left(f_{\boldsymbol{w}}, g_{\boldsymbol{w}}, \boldsymbol{x}, y\right)] &\leq \mathbb{E}_{\tilde{\boldsymbol{w}}}\left[\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[L_{\mathrm{MH}}^{\rho/2,\rho'/2}\left(f_{\tilde{\boldsymbol{w}}}, g_{\tilde{\boldsymbol{w}}}, \boldsymbol{x}, y\right)\right]\right] \\
&\leq \mathbb{E}_{\tilde{\boldsymbol{w}}}\left[\mathbb{E}_{(\boldsymbol{x},y)\sim S}\left[L_{\mathrm{MH}}^{\rho/2,\rho'/2}\left(f_{\tilde{\boldsymbol{w}}}, g_{\tilde{\boldsymbol{w}}}, \boldsymbol{x}, y\right)\right]\right] + 2\sqrt{\frac{2D_{\mathrm{KL}}(\tilde{\boldsymbol{w}}\|P) + \ln\frac{2m}{\delta}}{m-1}} \\
&\leq \mathbb{E}_{(\boldsymbol{x},y)\sim S}\left[L_{\mathrm{MH}}^{\rho,\rho'}\left(f_{\boldsymbol{w}}, g_{\boldsymbol{w}}, \boldsymbol{x}, y\right)\right] + 2\sqrt{\frac{2D_{\mathrm{KL}}(\tilde{\boldsymbol{w}}\|P) + \ln\frac{2m}{\delta}}{m-1}} \\
&\leq \mathbb{E}_{(\boldsymbol{x},y)\sim S}\left[L_{\mathrm{MH}}^{\rho,\rho'}\left(f_{\boldsymbol{w}}, g_{\boldsymbol{w}}, \boldsymbol{x}, y\right)\right] + 4\sqrt{\frac{D_{\mathrm{KL}}\left(\boldsymbol{w}'\|P\right) + \ln\frac{6m}{\delta}}{m-1}}.
\end{aligned}$$

□

Next, we will consider adding perturbation parameters $u \sim \mathcal{N}\left(0, \sigma^2\boldsymbol{I}\right)$ to the parameters of the classification layer $\boldsymbol{w}$, i.e.,

$\boldsymbol{w}' = \boldsymbol{w} + \boldsymbol{u}$, and we assume that the learned feature $c(\boldsymbol{x})$ is centered, i.e., $\mathbb{E}[c(\boldsymbol{x})] = 0$. Then, we have

$$\begin{aligned}
\mathbb{E}[\|\boldsymbol{w}'c(\boldsymbol{x}) - \boldsymbol{w}c(\boldsymbol{x})\|_2^2] &= \mathbb{E}[\|\boldsymbol{u}\|_2^2\|c(\boldsymbol{x})\|_2^2] \\
&= \sigma^2\mathbb{E}[\|c(\boldsymbol{x})\|_2^2] \\
&= \sigma^2 \operatorname{tr}(\mathbb{E}[c(\boldsymbol{x})c(\boldsymbol{x})^\top] - \mathbb{E}[c(\boldsymbol{x})]\mathbb{E}[c(\boldsymbol{x})]^\top) \\
&= \sigma^2 \operatorname{tr}[\operatorname{Cov}[c(\boldsymbol{x})]] \\
&= \sigma^2 \operatorname{tr}[\mathbb{E}[\operatorname{Cov}[c(\boldsymbol{x})|y]] + \operatorname{Cov}[\mathbb{E}[c(\boldsymbol{x})|y]]] \\
&= \sigma^2 \operatorname{tr}\left[\sum_{i\in[k]} \operatorname{Cov}[c^i(\boldsymbol{x})]/k + \sum_{i\neq j}(\mathbb{E}[c^i(\boldsymbol{x})] - \mathbb{E}[c^j(\boldsymbol{x})])(\mathbb{E}[c^i(\boldsymbol{x})] - \mathbb{E}[c^j(\boldsymbol{x})])^\top/k(k-1)\right] \\
&= \sigma^2 \operatorname{tr}\left[\sum_{i\in[k]} \operatorname{Cov}[c^i(\boldsymbol{x})]/k\right] + \sum_{i\neq j} \operatorname{tr}\left[(\mathbb{E}[c^i(\boldsymbol{x})] - \mathbb{E}[c^j(\boldsymbol{x})])(\mathbb{E}[c^i(\boldsymbol{x})] - \mathbb{E}[c^j(\boldsymbol{x})])^\top\right]/k(k-1) \\
&\leq \sigma^2 \operatorname{tr}\left[\sum_{i\in[k]} \operatorname{Cov}[c^i(\boldsymbol{x})]/k\right] + \sum_{i\neq j}(2\tilde{\rho})^2/(\|l\|_2^2 k(k-1)) \\
&= \sigma^2 \operatorname{tr}\left[\sum_{i\in[k]} \operatorname{Cov}[c^i(\boldsymbol{x})]/k\right] + 4\tilde{\rho}^2/\|l\|_2^2.
\end{aligned}$$

According to the Markov inequality, we have

$$\mathbb{P}_\beta\left[\max_{\boldsymbol{x}\in S}|f_{\boldsymbol{w}'}(\boldsymbol{x}) - f_{\boldsymbol{w}}(\boldsymbol{x})|_2^2 \geq \frac{\sigma^2}{\delta}\cdot\left(\operatorname{tr}\left[\sum_{i\in[k]}\operatorname{Cov}[c^i(\boldsymbol{x})]/k\right] + 4\tilde{\rho}^2/\|l\|_2^2\right)\right] \leq \delta.$$

We set $\delta = 1/2$, such that the inequality holds with a probability at least $1/2$:

$$\max_{\boldsymbol{x}\in S}|f_{\boldsymbol{w}'}(\boldsymbol{x}) - f_{\boldsymbol{w}}(\boldsymbol{x})|_2^2 \leq 2\sigma^2\left(\operatorname{tr}\left[\sum_{i\in[k]}\operatorname{Cov}\left[c^i(\boldsymbol{x})\right]/k\right] + 4\tilde{\rho}^2/\|l\|_2^2\right).$$

For simplicity, we use $\operatorname{Var}_{\text{intra}}[c(\boldsymbol{x})]$ to denote the intra-class variance $\operatorname{tr}\left[\sum_{i\in[k]}\operatorname{Cov}\left[c^i(\boldsymbol{x})\right]/k\right]$, then we have

$$\max_{\boldsymbol{x}\in S}|f_{\boldsymbol{w}'}(\boldsymbol{x}) - f_{\boldsymbol{w}}(\boldsymbol{x})|_2^2 \leq 2\sigma^2\left(\operatorname{Var}_{\text{intra}}[c(\boldsymbol{x})] + 4\tilde{\rho}^2/\|l\|_2^2\right).$$

Since we now prove that the perturbation caused by random vector $\boldsymbol{u}$ is bounded by a term relative to the variance σ, we can preset the value of σ to make the random perturbation satisfy the condition for Lemma B.1.

$$\begin{aligned}
\max_{x\in S}|f_{\boldsymbol{w}'}(\boldsymbol{x}) - f_{\boldsymbol{w}}(\boldsymbol{x})|_2^2 &\leq 2\sigma^2\left(\operatorname{Var}_{\text{intra}}[c(\boldsymbol{x})] + 4\tilde{\rho}^2/\|l\|_2^2\right) \\
&= \min\left\{\frac{\rho}{4\alpha}, \frac{\rho'}{4\beta\lambda + 2\alpha}\right\}^2 \\
&= \tilde{\rho}^2.
\end{aligned}$$

We can derive $\sigma = \tilde{\rho}/\sqrt{2\left(\operatorname{Var}_{\text{intra}}[c(\boldsymbol{x})] + 4\tilde{\rho}^2/\|l\|_2^2\right)}$ from the above inequality. Naturally, we can calculate the Kullback-Leibler divergence in Lemma B.1 with the chosen distributions for $P \sim \mathcal{N}\left(0, \sigma^2\boldsymbol{I}\right)$:

$$D_{\mathrm{KL}}(\boldsymbol{w} + \boldsymbol{u}\|P) \leq \frac{|\boldsymbol{w}|^2}{2|\boldsymbol{w}|^2\sigma^2} = \frac{1}{2\sigma^2} \leq \frac{\|l\|_2^2\operatorname{Var}_{\text{intra}}[c(\boldsymbol{x})] + 4\tilde{\rho}^2}{\tilde{\rho}^2\|l\|_2^2}.$$

Put it in Lemma B.1, for any $\boldsymbol{w}$, with probability of at least $1-\delta$ we have:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[L_0(f,g,\boldsymbol{x},y)] \leq \mathbb{E}_{(\boldsymbol{x},y)\sim S}\left[L_{\mathrm{MH}}^{\rho,\rho'}(f,g,\boldsymbol{x},y)\right] + 4\sqrt{\frac{\|l\|_2^2 \operatorname{Var}_{\text{intra}}[c(\boldsymbol{x})] + 4\tilde{\rho}^2 + \tilde{\rho}^2\|l\|_2^2 \ln\frac{6m}{\delta}}{\tilde{\rho}^2 m\|l\|_2^2}}.$$

□

C. Training with CSC Loss

Algorithm 1 Training with CSC loss

Input: Data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$, initial model f
Parameter: momentum coefficient q, queue size s, weight coefficient w, initial epochs E_s

for $e = 1$: maximum epochs **do**
  if $e = E_s$ **then**
    Initialize $\boldsymbol{\theta}_{\mathrm{m}} = \boldsymbol{\theta}$
  end if
  for each mini-batch in the current epoch e **do**
    if $e \geq E_s$ **then**
      if Both queues P and Q have been updated more than s tuples **then**
        $L = w \cdot L_{\mathrm{CSC}} + L_{\mathrm{CE}}$
      else
        $L = L_{\mathrm{CE}}$
        Fetch $\{(\boldsymbol{z}_i, f_{\boldsymbol{\theta}_{\mathrm{m}}}(\boldsymbol{x}_i))\}_{i=1}^n$ of mini-batch data;
        Update P by $\{(\boldsymbol{z}_i, f_{\boldsymbol{\theta}_{\mathrm{m}}}(\boldsymbol{x}_i)) \mid f_{\boldsymbol{\theta}_{\mathrm{m}}}(\boldsymbol{x}_i) = y_i\}_{i=1}^n$;
        Update Q by $\{(\boldsymbol{z}_i, f_{\boldsymbol{\theta}_{\mathrm{m}}}(\boldsymbol{x}_i)) \mid f_{\boldsymbol{\theta}_{\mathrm{m}}}(\boldsymbol{x}_i) \neq y_i\}_{i=1}^n$
      end if
      $\boldsymbol{\theta}_{\mathrm{m}} \leftarrow q \cdot \boldsymbol{\theta}_{\mathrm{m}} + (1-q) \cdot \boldsymbol{\theta}$;
    else
      $L = L_{\mathrm{CE}}$
    end if
    Update the parameters $\boldsymbol{\theta}$ of the model f using an optimizer based on L
  end for
end for

Algorithm 1 provides the pseudo-code of the training method with the proposed CSC loss. Different from the conventional two-stage training manner "pre-train then finetune" used in typical contrastive learning methods (He et al., 2020; Chen et al., 2020; Khosla et al., 2020), we employ a one-stage manner to optimize the classification layers and convolutional layers of the model together, because the CSC loss involves the predictive information of the model.

The model f is initially trained using the cross-entropy loss L_{CE} for E_s epochs. When $e \geq E_s$ and meanwhile both queues P and Q have been updated more than s tuples, the training gradient consists of two parts: the gradient of the CSC loss L_{CSC} returned from the last feature embedding layer c, and the gradient of the cross-entropy loss L_{CE} returned from the classification layer l. We use a weight coefficient w to balance these two loss items. The parameters $\boldsymbol{\theta}$ of the model f will be updated using an optimizer based on the combined loss. As introduced in Section 5.1, when $e \geq E_s$, a momentum encoder $f_{\boldsymbol{\theta}_{\mathrm{m}}}$ is used to generate positive and negative samples. When the epoch e equals E_s, the parameters $\boldsymbol{\theta}_{\mathrm{m}}$ of the momentum encoder $f_{\boldsymbol{\theta}_{\mathrm{m}}}$ are initialized to the parameters $\boldsymbol{\theta}$ of the current model f, which ensures that the momentum encoder has a favorable initial accuracy. After that, the parameters $\boldsymbol{\theta}_{\mathrm{m}}$ of the momentum encoder will be updated according to Eq. (4) at each training step. The samples generated by the momentum encoder will be used to update the queues P and Q, which store positive and negative samples, respectively. Note that $f_{\boldsymbol{\theta}_{\mathrm{m}}}$ will not be optimized by the optimizer.

D. Detailed Experimental Settings

D.1. hyper-parameters

We utilize 20% of the training set for each dataset as the validation set to tune hyper-parameters. We test the momentum coefficient $q \in \{0.90.99, 0.999\}$, and the weight coefficient $w \in \{0.1, 0.5, 1.0\}$. For the queue size s, we set it based on the number of classes in the dataset: for datasets with fewer classes such as CelebA and CIFAR-10, we set $s = 300$; for datasets with more classes such as CIFAR-100 and ImageNet, we set $s = 3000$ and $s = 10000$, respectively. For the initial epochs E_s, we test it on datasets with total training epochs of 300 using the values $\{50, 100, 150, 200\}$; for datasets with total training epochs of 150, we test the values $\{50, 100\}$; for datasets with training epochs of 50, we evaluate the values $\{0, 1\}$. Table 5 lists the hyperparameter settings of CCL-SC on each dataset. After tuning the hyper-parameters, we train the model on the entire training set to evaluate performance.

Table 5. The hyper-parameters settings of CCL-SC on various datasets.

Dataset	q	s	w	E_s
CIFAR-10	0.999	300	0.5	150
CIFAR-100	0.99	3000	1.0	150
CelebA	0.999	300	0.5	1
ImageNet	0.999	10000	0.1	50

For the hyper-parameters of the baseline methods, we follow the settings provided in their origin paper or released codes. Nevertheless, due to the absence of performance evaluation on CIFAR-100 and CelebA in prior work, we have also applied the same parameter-tuning steps outlined for our method to calibrate the parameters of the baseline methods. Table 6 lists the hyperparameter settings for each baseline on CIFAR-100 and CelebA.

Table 6. The hyper-parameters settings of the baselines on various datasets.

Dataset	Method	hyper-parameters
CIFAR-100	Deep Gambler	Initial epochs E_s = 200, Reward o = 4.6
	SAT	Initial epochs E_s = 200, Momentum term m_{SAT} = 0.9
	SAT+ER	Initial epochs E_s = 200, Momentum term m_{SAT} = 0.9, Entropy weight β = 0.001
CelebA	Deep Gambler	Initial epochs E_s = 0, Reward o = 2.0
	SAT	Initial epochs E_s = 0, Momentum term m_{SAT} = 0.9
	SAT+ER	Initial epochs E_s = 0, Momentum term m_{SAT} = 0.9, Entropy weight β = 0.01

D.2. Networks and Training

Following prior work, for CIFAR-10 and CIFAR-100, we use VGG16 (Simonyan & Zisserman, 2015) as the backbones of selective classifiers. The models are trained for 300 epochs using SGD, with an initial learning rate of 0.1, a momentum of 0.9, a weight decay of 5e-4, and a mini-batch size of 64. The learning rate was reduced by 0.5 every 25 epochs.

For ImageNet, we use ResNet34 (He et al., 2016) trained for 150 epochs using SGD, with an initial learning rate of 0.1, a momentum of 0.9, a weight decay of 5e-4, and a mini-batch size of 256. The learning rate was reduced by 0.5 every 10 epochs.

For CelebA, we use ResNet18 (He et al., 2016) trained for 50 epochs using Adam, with an initial learning rate of 1e-5, and a mini-batch size of 64. When evaluating the performance of each method on the CelebA dataset, we employ the checkpoint with the highest accuracy on the CelebA's original separate validation set to assess its performance on the test set.

D.3. Data Augmentation Methods for each Dataset

For the same dataset, all compared methods utilize the same data augmentation to ensure a fair comparison. For the commonly used selective classification benchmark datasets (including CIFAR-10, and ImageNet) in previous works (Geifman & El-Yaniv, 2019; Liu et al., 2019; Huang et al., 2020; 2022; Feng et al., 2023), we adopt the data augmentation settings that have been commonly utilized. For the newly considered challenging selective classification datasets in this work, CIFAR-100 and CelebA, we apply commonly used data augmentation methods from the field of image classification that are tailored for these datasets. Table 7 presents a summary of the data augmentation methods utilized for each of the datasets.

Table 7. Data augmentation methods utilized for each of the datasets.

Dataset	Data Augmentation
CIFAR-10	RandomCrop RandomHorizontalFlip
CIFAR-100	RandomCrop RandomHorizontalFlip
CelebA	RandomHorizontalFlip
ImageNet	RandomResizedCrop RandomHorizontalFlip ColorJitter

E. Comparison with Other Related Methods

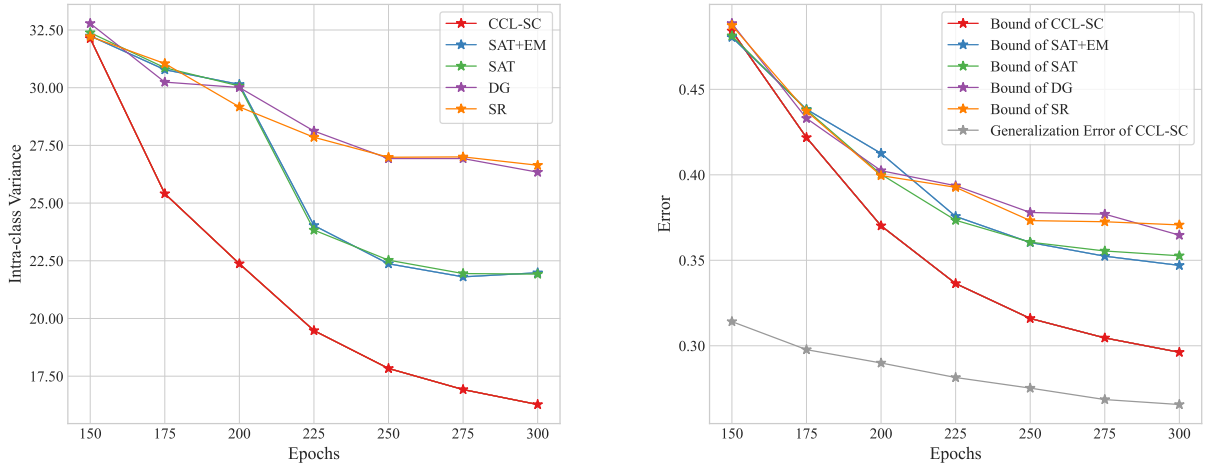
As described in Section 2.1, selective classification is closely related to model calibration and the Human-AI collaboration. In this section, we introduce five methods from model calibration, including Focal loss (Lin et al., 2017), Adaptive Focal loss (Mukhoti et al., 2020), Soft AvUC loss (Karandikar et al., 2021), Soft ECE loss (Karandikar et al., 2021), and MMCE loss (Kumar et al., 2018), as well as AUCOC loss (Sangalli et al., 2023) from the Human-AI collaboration into selective classification. Tabel 8 shows that CCL-SC has the lowest selective risk at various coverage rates compared to these methods.

Table 8. Selective risk (%) on the CIFAR-100 for various coverage rates (%). The mean and standard deviation are calculated over 5 trials. The best entries are marked in bold.

Coverage	CCL-SC	Adaptive Focal	Soft AvUC	Soft-ECE	MMCE	AUCOC Loss	Focal Loss
100	**26.55±0.26**	27.96±0.12	27.93±0.12	26.81±0.06	27.14±0.24	26.75±0.07	28.08±0.22
95	**23.54±0.15**	25.26±0.24	24.99±0.05	23.93±0.16	24.30±0.24	23.89±0.05	25.38±0.26
90	**20.97±0.20**	22.63±0.17	22.20±0.08	21.40±0.1	21.66±0.12	21.33±0.19	22.64±0.37
85	**18.57±0.20**	20.13±0.09	19.68±0.04	18.84±0.10	18.95±0.25	19.17±0.10	20.11±0.43
80	**16.07±0.15**	17.77±0.08	17.12±0.05	16.35±0.03	16.35±0.19	16.21±0.05	17.64±0.18
75	**13.60±0.19**	15.48±0.14	14.76±0.01	14.15±0.11	14.21±0.13	13.95±0.04	15.34±0.14
70	**11.23±0.16**	13.07±0.11	12.46±0.10	11.57±0.07	11.60±0.16	11.47±0.13	13.17±0.11
60	**6.83±0.15**	8.73±0.01	7.93±0.13	7.28±0.08	7.32±0.05	7.26±0.13	8.85±0.12
50	**3.95±0.22**	5.67±0.13	4.29±0.05	4.51±0.07	4.32±0.10	4.21±0.02	5.69±0.11
40	**2.29±0.33**	4.14±0.29	2.48±0.02	3.10±0.02	3.16±0.11	2.31±0.09	4.59±0.01
30	**1.26±0.17**	3.87±0.07	1.63±0.03	2.72±0.15	2.63±0.03	1.69±0.04	4.25±0.05
20	**0.71±0.12**	3.95±0.25	1.58±0.03	2.85±0.20	2.40±0.05	1.26±0.16	4.70±0.30
10	**0.36±0.08**	4.55±0.45	1.45±0.15	2.85±0.15	2.20±0.00	0.88±0.04	5.05±0.25

F. Alignment between Proposed Theory and Method

In this section, we show the changes of the intra-class variance and the bound in Theorem 4.1 of different methods during the training process on CIFAR-100 in Figure 2. It can be observed that CCL-SC does have the lowest intra-class variance and the lowest bound. It is worth mentioning that the relative order of the intra-class variance and the bounds of different methods is consistent with their actual order of selective risk, and the methods with similar selective risk (such as SAT+EM and SAT) also have similar intra-class variance/bound, indicating the importance of intra-class variance for selective classification performance and the usefulness of our bound.

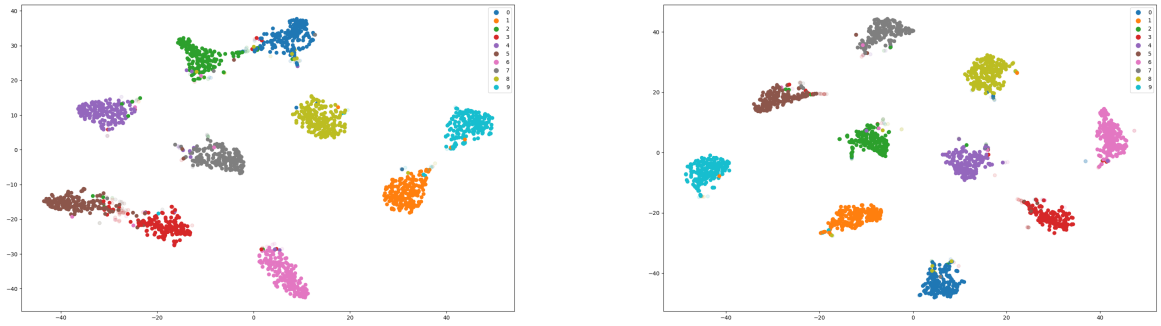


(a) Intra-variance (b) Bound in Theorem 4.1

Figure 2. The intra-class variance (a) and the bound in Theorem 4.1 (b) changes of different methods during the training process on CIFAR-100. In (b), we also include the generalization error of CCL-SC.

G. Learned Feature Representation

In this section we compare its learned feature representations with those of SR trained only using the cross-entropy loss on CIFAR-10 at coverage 95%. The t-SNE (Van der Maaten & Hinton, 2008) visualization shown in Figure 3 clearly demonstrates that compared to SR, our method CCL-SC achieves more significant inter-class separation and intra-class aggregation in the feature space for selecting samples for classification. This confirms that optimizing the feature layer contributes to performance improvement in selective classification.



(a) SR (b) CCL-SC

Figure 3. The t-SNE Visualization (Van der Maaten & Hinton, 2008) of SR and CCL-SC feature representations on the CIFAR-10 dataset at 95% coverage. Point colors indicate class categories. Light-colored points represent samples selected for abstaining from predicting.

H. Ablation Study and Hyper-parameter Sensitivity Results

SR-weighted To verify whether the weighting manner based on SR in the proposed CSC loss L_{CSC} really improves the performance of selective classification, we evaluate the model trained using CSC loss without applying weights (i.e., $L'_{\text{CSC}} = \frac{1}{-|P(y)|} \sum_{\boldsymbol{x}_p \in P(y)} \log \frac{\exp(\boldsymbol{z} \cdot \boldsymbol{z}_p / \tau)}{\sum_{\boldsymbol{x}_a \in A(y)} \exp(\boldsymbol{z} \cdot \boldsymbol{z}_a / \tau)}$), and the results on the CIFAR-100 dataset are shown in Table 9. It can be observed that SR-weighted (i.e., original CCL-SC method) consistently achieves lower selective risk than using the

unweighted CSC loss across all degrees of coverage, and performs significantly better on at $5/13$ coverage rates, according to the Wilcoxon rank-sum test (Wilcoxon, 1945) with significance level 0.05. This result demonstrates the effectiveness of CSC loss combined with model confidence for selective classification problems.

Table 9. Selective risk (%) on the CIFAR-100 dataset for various coverage rates (%). The means and standard deviations are calculated over 5 trials. The best entries are marked in bold. The symbol '•'/'○' indicates that SR-weighted (i.e., original CCL-SC) is significantly better/worse than Unweighted, according to the Wilcoxon rank-sum test with significance level 0.05.

Coverage	SR-weighted	Unweighted
100	**26.55±0.26**	26.72±0.15
95	**23.54±0.15**	23.84±0.11•
90	**20.97±0.20**	21.28±0.16•
85	**18.57±0.20**	18.69±0.21
80	**16.07±0.15**	16.15±0.17
75	**13.60±0.19**	13.67±0.16
70	**11.23±0.16**	11.40±0.20
60	**6.83±0.15**	7.31±0.14•
50	**3.95±0.22**	4.52±0.13•
40	**2.29±0.33**	2.54±0.23
30	**1.26±0.17**	1.47±0.22
20	**0.71±0.12**	0.82±0.09
10	**0.36±0.08**	0.58±0.12•

The contrastive learning method of CCL-SC We conduct experiments with different queue sizes $s = 300$ and $s = 3000$. We first conduct ablation experiments for the construction of negative samples. For convenience, we name the ablation method CCL-SC2. For the negative samples of the samples that are correctly classified as class y, CCL-SC2 contains not only the samples misclassified as class y in the queue defined in this paper, but also the samples from other classes in the queue. Table 10 shows that even with the addition of negative samples from other categories, the performance of CCL-SC2 will not be improved compared to the original CCL-SC. However, this leads to more training costs. This implies that the real improvement in model performance is likely to be from the negative samples we define.

Table 10. Selective risk (%) on the CIFAR-100 dataset for various coverage rates (%). The means and standard deviations are calculated over 5 trials. The best entries are marked in bold.

Coverage	CCL-SC $s = 300$	CCL-SC2 $s = 300$	CCL-SC $s = 3000$	CCL-SC2 $s = 3000$
100	26.73±0.21	**26.72±0.28**	26.55±0.26	**26.48±0.27**
95	23.89±0.16	**23.87±0.18**	**23.54±0.15**	23.74±0.29
90	**21.22±0.15**	**21.22±0.12**	**20.97±0.20**	21.12±0.26
85	**18.76±0.17**	18.85±0.10	**18.57±0.20**	18.67±0.24
80	**16.36±0.21**	16.37±0.08	**16.07±0.15**	16.20±0.28
75	**13.88±0.18**	14.05±0.10	**13.60±0.19**	13.79±0.27
70	**11.37±0.23**	11.58±0.19	**11.23±0.16**	11.42±0.30
60	**7.09±0.10**	7.30±0.19	**6.83±0.15**	7.50±0.28
50	**4.06±0.23**	4.36±0.18	**3.95±0.22**	4.68±0.18
40	**2.29±0.18**	2.47±0.23	**2.29±0.33**	2.62±0.02
30	**1.24±0.08**	1.32±0.31	**1.26±0.17**	1.44±0.21
20	0.78±0.14	**0.73±0.17**	**0.71±0.12**	0.97±0.12
10	0.50±0.17	**0.49±0.12**	**0.36±0.08**	0.58±0.13

To confirm this conclusion, we conduct another ablation study in which we remove the negative samples defined in CCL-SC, and only use randomly sampled samples from other categories as negative samples. We name this ablation method CCL-SC3.

The experimental results are shown in Table 11. It can be observed that CCL-SC3 has a significant performance decrease compared to CCL-SC, which demonstrates the effectiveness of our strategy to construct negative samples.

Table 11. Selective risk (%) on the CIFAR-100 dataset for various coverage rates (%). The means and standard deviations are calculated over 5 trials. The best entries are marked in bold.

Coverage	CCL-SC $s = 300$	CCL-SC3 $s = 300$	CCL-SC $s = 3000$	CCL-SC3 $s = 3000$
100	**26.73±0.21**	27.11±0.09	**26.55±0.26**	26.87±0.35
95	**23.89±0.16**	24.28±0.14	**23.54±0.15**	23.94±0.41
90	**21.22±0.15**	21.70±0.17	**20.97±0.20**	21.33±0.31
85	**18.76±0.17**	19.24±0.20	**18.57±0.20**	18.89±0.28
80	**16.36±0.21**	16.76±0.21	**16.07±0.15**	16.41±0.21
75	**13.88±0.18**	14.27±0.19	**13.60±0.19**	13.96±0.14
70	**11.37±0.23**	11.86±0.25	**11.23±0.16**	11.58±0.17
60	**7.09±0.10**	7.54±0.20	**6.83±0.15**	7.34±0.37
50	**4.06±0.23**	4.13±0.12	**3.95±0.22**	4.04±0.17
40	2.29±0.18	**2.13±0.04**	2.29±0.33	**2.19±0.11**
30	1.24±0.08	**1.14±0.14**	1.26±0.17	**1.15±0.08**
20	0.78±0.14	**0.72±0.05**	**0.71±0.12**	0.76±0.10
10	0.50±0.17	**0.44±0.08**	**0.36±0.08**	0.40±0.14

We also conduct ablation experiments for the whole contrastive method of CCL-SC. Specifically, We introduce the positive and negative sample definition method and loss function from (Khosla et al., 2020) into our CCL-SC method, while keeping the other components consistent. We name this ablation method CCL-SC+SupCon. The comparison results on CIFAR-100 are shown in Table 11. It can be observed that the selective classification performance of the original CCL-SC is better than CCL-SC+SupCon, which uses vanilla supervised CL.

Table 12. Selective risk (%) on the CIFAR-100 dataset for various coverage rates (%). The means and standard deviations are calculated over 5 trials. The best entries are marked in bold.

Coverage	CCL-SC $s = 300$	CCL-SC+SupCon $s = 300$	CCL-SC $s = 3000$	CCL-SC+SupCon $s = 3000$
100	**26.73±0.21**	26.99±0.13	**26.55±0.26**	26.77±0.17
95	**23.89±0.16**	24.14±0.06	**23.54±0.15**	23.91±0.14
90	**21.22±0.15**	21.49±0.12	**20.97±0.20**	21.31±0.15
85	**18.76±0.17**	19.10±0.13	**18.57±0.20**	18.80±0.17
80	**16.36±0.21**	16.64±0.24	**16.07±0.15**	16.36±0.22
75	**13.88±0.18**	14.19±0.19	**13.60±0.19**	13.98±0.26
70	**11.37±0.23**	11.74±0.21	**11.23±0.16**	11.63±0.09
60	**7.09±0.10**	7.26±0.16	**6.83±0.15**	7.24±0.10
50	**4.06±0.23**	4.15±0.07	**3.95±0.22**	4.13±0.03
40	2.29±0.18	**2.25±0.20**	**2.29±0.33**	2.32±0.09
30	**1.24±0.08**	1.25±0.17	**1.26±0.17**	1.27±0.11
20	**0.78±0.14**	0.82±0.13	**0.71±0.12**	0.87±0.10
10	0.50±0.17	**0.48±0.10**	**0.36±0.08**	0.50±0.09

We then conduct sensitivity analyses on the hyper-parameters in our method. Specifically, when varying one hyperparameter, we keep the other hyper-parameters fixed.

Momentum coefficient q. Tabel 13 shows the influence of momentum coefficient $q \in \{0, 0.9, 0.99, 0.999\}$ on the selective classification performance of our method. Our method achieves stable selective risk when employing q values in $\{0.9, 0.99, 0.999\}$. Specifically, our method exhibits inferior performance when q is set to 0 compared to other values. This can be attributed to the momentum encoder used for constructing positive and negative sample features losing its momentum update properties, where Eq. (4) degenerates into $\boldsymbol{\theta}_{\mathrm{m}} = \boldsymbol{\theta}$. Consequently, there is a significant reduction in the consistency

of feature representations in the queue. This observation is consistent with the findings reported in (He et al., 2020).

Table 13. Selective risk (%) of CCL-SC using various momentum coefficient m for various coverage rates (%) on the CIFAR-100. The means and standard deviations are calculated over 5 trials. The best entries are marked in bold.

Coverage	$q = 0$	$q = 0.9$	$q = 0.99$	$q = 0.999$
100	26.72±0.21	**26.41±0.22**	26.55±0.26	26.52±0.29
95	23.92±0.30	23.56±0.15	**23.54±0.15**	23.64±0.27
90	21.27±0.20	**20.91±0.21**	20.97±0.20	20.96±0.32
85	18.71±0.27	**18.47±0.17**	18.57±0.20	18.51±0.26
80	16.19±0.31	**15.96±0.22**	16.07±0.15	15.97±0.24
75	13.74±0.33	**13.53±0.19**	13.60±0.19	13.61±0.30
70	11.33±0.28	**11.19±0.18**	11.23±0.16	11.18±0.16
60	6.98±0.15	6.94±0.13	**6.83±0.15**	6.97±0.12
50	4.10±0.07	4.05±0.17	**3.95±0.22**	4.05±0.20
40	2.34±0.19	2.41±0.21	2.29±0.33	**2.26±0.23**
30	1.33±0.18	1.30±0.19	1.26±0.17	**1.23±0.20**
20	0.90±0.29	0.81±0.16	0.71±0.12	**0.67±0.14**
10	0.60±0.21	0.46±0.19	0.36±0.08	**0.34±0.05**
Avg. Rank	3.92	2.00	2.15	**1.85**

Queue size s. Tabel 14 shows the performance comparison of our method when maintaining different queue sizes $s \in \{300, 1000, 3000, 10000, 50000\}$. Surprisingly, our method demonstrates performance improvements compared to previous methods, shown in Table 2, even when the queue size is set to a remarkably small value, such as 300. Moreover, as the queue size increases, our method exhibits further improvements in performance, particularly in cases with higher coverage.

Table 14. Selective risk (%) of CCL-SC using various queue size s for various coverage rates (%) on the CIFAR-100. The means and standard deviations are calculated over 5 trials. The best entries are marked in bold.

Coverage	$s = 300$	$s = 1000$	$s = 3000$	$s = 10000$	$s = 50000$
100	26.73±0.21	26.71±0.18	26.55±0.26	26.45±0.24	**26.08±0.04**
95	23.89±0.16	23.79±0.19	23.54±0.15	23.62±0.25	**23.22±0.07**
90	21.22±0.15	21.09±0.29	20.97±0.20	21.00±0.27	**20.66±0.08**
85	18.76±0.17	18.65±0.31	18.57±0.20	18.39±0.34	**18.19±0.13**
80	16.36±0.21	16.22±0.37	16.07±0.15	15.84±0.34	**15.73±0.08**
75	13.88±0.18	13.73±0.32	13.60±0.19	**13.31±0.42**	13.42±0.11
70	11.37±0.23	11.44±0.24	11.23±0.16	**11.01±0.33**	11.09±0.17
60	7.09±0.10	7.15±0.18	**6.83±0.15**	6.88±0.24	6.94±0.18
50	4.06±0.23	4.29±0.16	**3.95±0.22**	4.13±0.21	4.15±0.14
40	2.29±0.18	2.37±0.16	2.29±0.33	**2.26±0.24**	2.44±0.09
30	1.24±0.08	1.21±0.14	1.26±0.17	**1.17±0.07**	1.42±0.21
20	0.78±0.14	0.70±0.20	0.71±0.12	**0.65±0.08**	0.75±0.20
10	0.50±0.17	0.48±0.19	**0.36±0.08**	0.40±0.09	0.48±0.12
Avg. Rank	4.23	3.85	2.38	**1.85**	2.54

Weight coefficients w. Table 15 presents a comparison of the performance of models trained with varying $w \in \{0.1, 0.5, 1.0, 2.0\}$ applied to the CSC loss. It can be observed that when a relatively larger weight coefficient is assigned, that is, when the CSC loss has a greater impact on model training, the resulting models exhibit better performance, which also confirms the effectiveness of the CSC loss.

Table 15. Selective risk (%) of CCL-SC using various weight coefficient w on the CIFAR-100. The means and standard deviations are calculated over 5 trials. The best entries are marked in bold.

Coverage	$w = 0.1$	$w = 0.5$	$w = 1.0$	$w = 2.0$
100	26.90±0.04	26.59±0.09	26.55±0.26	**26.25±0.09**
95	24.12±0.08	23.79±0.13	23.54±0.15	**23.41±0.16**
90	21.44±0.12	21.08±0.14	20.97±0.20	**20.79±0.14**
85	19.12±0.15	18.68±0.21	18.57±0.20	**18.34±0.19**
80	16.59±0.21	16.27±0.26	16.07±0.15	**15.89±0.19**
75	14.24±0.15	13.83±0.21	13.60±0.19	**13.48±0.15**
70	11.65±0.21	11.39±0.27	11.23±0.16	**11.12±0.11**
60	7.37±0.26	7.06±0.13	**6.83±0.15**	7.09±0.15
50	4.20±0.18	3.97±0.07	**3.95±0.22**	4.24±0.24
40	2.39±0.15	2.34±0.18	**2.29±0.33**	2.39±0.22
30	1.40±0.20	1.29±0.18	**1.26±0.17**	1.31±0.12
20	0.96±0.09	0.83±0.21	**0.71±0.12**	0.75±0.16
10	0.76±0.08	0.48±0.16	**0.36±0.08**	0.50±0.18
Avg. Rank	3.85	2.62	**1.54**	1.92

Initial epochs E_s. Table 16 compares the performance of our method when using different initial epochs $E_s \in \{50, 100, 150, 200, 250\}$. It can be observed that our method consistently exhibits stable and robust performance when E_s is set between 50 and 150. The performance tends to deteriorate only if the initial epochs are set too large (i.e., our training mechanism is utilized too late), which leads to insufficient convergence and fluctuations in model performance.

Table 16. Selective risk (%) of CCL-SC using various initial epochs E_s for various coverage rates (%) on the CIFAR-100. The means and standard deviations are calculated over 5 trials. The best entries are marked in bold.

Coverage	$E_s = 50$	$E_s = 100$	$E_s = 150$	$E_s = 200$	$E_s = 250$
100	26.71±0.16	26.56±0.18	**26.55±0.26**	26.81±0.24	27.13±0.34
95	23.88±0.23	23.69±0.15	**23.54±0.15**	23.97±0.23	24.25±0.35
90	21.27±0.21	21.16±0.08	**20.97±0.20**	21.34±0.17	21.65±0.34
85	18.71±0.26	18.65±0.08	**18.57±0.20**	18.86±0.23	19.20±0.28
80	16.30±0.27	16.15±0.12	**16.07±0.15**	16.37±0.21	16.77±0.27
75	13.84±0.23	13.73±0.16	**13.60±0.19**	13.91±0.15	14.40±0.32
70	11.50±0.21	11.38±0.29	**11.23±0.16**	11.43±0.27	12.02±0.24
60	7.04±0.11	7.06±0.21	**6.83±0.15**	7.16±0.24	7.55±0.17
50	4.28±0.19	4.16±0.12	**3.95±0.22**	4.10±0.17	4.59±0.25
40	2.36±0.18	**2.20±0.16**	2.29±0.33	2.48±0.36	3.13±0.34
30	1.22±0.10	**1.18±0.11**	1.26±0.17	1.51±0.24	2.22±0.29
20	**0.58±0.12**	0.79±0.08	0.71±0.12	0.94±0.20	1.57±0.14
10	0.38±0.16	0.46±0.16	**0.36±0.08**	0.62±0.21	0.94±0.41
Avg. Rank	2.77	2.15	**1.31**	3.77	5.00

I. Further improvement of CCL-SC

Since CCL-SC operates on the feature representation of the model, it can be seamlessly integrated with existing methods that optimize the model at the classification layer. In this section, we combine CCL-SC with SAT (Huang et al., 2020; 2022) and EM (Feng et al., 2023) methods. Specifically, when the current epoch is greater than E_s, for a sample $\boldsymbol{x}$ with label y,

we modify the loss function at the classification layer of the model from the cross-entropy loss to the following form:

$$L_{\text{SAT+EM}} = -t_y \log f_y(\boldsymbol{x}) - (1 - t_y) \log f_{(k+1)}(\boldsymbol{x}) + \beta \, \mathcal{H}\left(f(\boldsymbol{x})\right),$$

where $\mathcal{H}$ is the entropy function, and β controls the weight of its influence. The training target $\boldsymbol{t}$ is dynamically updated using the rule $\boldsymbol{t} \leftarrow m_{\text{SAT}} \cdot \boldsymbol{t} + (1 - m_{\text{SAT}}) \cdot f(\boldsymbol{x})$, where the momentum term $m_{\text{SAT}} \in (0, 1)$ regulates the weighting of predictions. The first term $t_y \log f_y(\boldsymbol{x})$ of $L_{\text{SAT+EM}}$ encourages the model to correctly classify the samples, while the second term $(1 - t_y) \log f_{(k+1)}(\boldsymbol{x})$ encourages the model to abstain from making predictions on samples with low confidence. Due to the combination of the SAT and EM methods, two new hyper-parameters, m_{SAT} and β, are introduced. Here we do not adjust these hyper-parameters but rather directly use the settings for m_{SAT} and β as introduced in Appendix D.

Table 17 and Tabel 18 present comparisons between the original CCL-SC and the improved CCL-SC (i.e., CCL-SC+SAT+EM) on CIFAR-100 and ImageNet, respectively. The results demonstrate that the improved CCL-SC shows superior performance in selective classification and outperforms the original CCL-SC across all degrees of coverage except 40% on ImageNet. This discovery shows that CCL-SC not only exhibits superior performance when utilized independently but also highlights high compatibility with other methods to further enhance the performance of selective classification.

Table 17. Selective risk (%) on the CIFAR-100 dataset for various coverage rates (%). The means and standard deviations are calculated over 5 trials. The best entries are marked in bold.

Coverage	CCL-SC	CCL-SC+SAT+EM
100	26.55±0.26	**26.41±0.08**
95	23.54±0.15	**23.51±0.16**
90	20.97±0.20	**20.85±0.20**
85	18.57±0.20	**18.31±0.17**
80	16.07±0.15	**15.79±0.17**
75	13.60±0.19	**13.41±0.15**
70	11.23±0.16	**11.06±0.19**
60	6.83±0.15	**6.72±0.18**
50	3.95±0.22	**3.67±0.15**
40	2.29±0.33	**1.97±0.18**
30	1.26±0.17	**1.05±0.10**
20	0.71±0.12	**0.49±0.11**
10	0.36±0.08	**0.26±0.05**

Table 18. Selective risk (%) on ImageNet dataset for various coverage rates (%). The means and standard deviations are calculated over 5 trials. The best entries are marked in bold.

Coverage	CCL-SC	CCL-SC+SAT+EM
100	26.26±0.10	**26.01±0.12**
90	20.68±0.07	**20.41±0.06**
80	15.76±0.07	**15.46±0.03**
70	11.39±0.10	**11.08±0.02**
60	7.55±0.09	**7.36±0.05**
50	4.79±0.04	**4.76±0.01**
40	**2.95±0.04**	2.99±0.03
30	**1.83±0.05**	**1.83±0.06**
20	1.22±0.05	**1.17±0.07**
10	0.72±0.05	**0.66±0.07**