We have revised the method for mining hard negative in section 3.2 to avoid a parameterization assumption of unlabeled sample's distribution , which is criticized being very similar to that used in the HCL paper. But most importantly, we believe that the over strong parametric vMF assumption degrades the theoretic foundation of BCL.

The core idea of this revision is to perform a full probability decomposition of the class conditional density of true negatives, denoted as $\phi_{\text{TN}}(\hat{x})$, to obtain desired component as desired sampling target that conditioned on hard principle.We first present the overview of densities.

Table 2. Overview of densities.

Densities	Meaning	Description
$\phi(\hat{x})$	Anchor specific density of scores $p(\hat{x}; x, f)$.	Determined by the anchor x and encoder f .
$\phi_{(1)}(\hat{x})$	Density of order statistics $X_{(1)}$	Determined by $\phi(\hat{x})$.
$\phi_{(2)}(\hat{x})$	Density of order statistics $X_{(2)}$	Determined by $\phi(\hat{x})$.
$\phi_{\text{TN}}(\hat{x})$	Class conditional density of true negative $p(\hat{x}\|x^- \in \text{TN})$	Determined by α and order statistics $\phi_{(1)}(\hat{x})$, $\phi_{(2)}(\hat{x})$
$\phi_{\text{FN}}(\hat{x})$	Class conditional density of false negative $p(\hat{x}\|x^- \in \text{FN})$	Determined by α and order statistics $\phi_{(1)}(\hat{x})$, $\phi_{(2)}(\hat{x})$
$\psi(\hat{x})$	Class conditional density of hard true negatives $p(\hat{x}\|x^- \in \text{TN}, x^- \in \text{HARD})$	A component of ϕ_{TN} conditioned on a specific hardness level β.
$\phi_{\text{UN}}(\hat{x})$	Density of unlabeled data $\phi_{\text{UN}} = \tau^- \phi_{\text{TN}} + \tau^+ \phi_{\text{FN}}$	Determined by class prior τ and class conditional densities

Revision of Section 3.1: Hard Negative Mining.

Note that the class conditional density of true negatives is

$$\phi_{\text{TN}}(\hat{x}) \quad = \quad \alpha\phi_{(1)}(\hat{x}) + (1-\alpha)\phi_{(2)}(\hat{x})$$

where the first term $\alpha\phi_{(1)}(\hat{x})$ is the *easy negative sample component* that been correctly classified by the classifier (as shown in Fig. 2) , and the second term $(1-\alpha)\phi_{(2)}(\hat{x})$ is the *hard negative sample component* that been incorrectly classified by the classifier.

To avoid parametric assumptions, we decompose $\phi_{\text{TN}}(\hat{x})$ and select the component that corresponds to the hard negative sample as the target sampling distribution. To achieve this, we introduce a parameter $\beta \in [0, 1]$ to decompose the class conditional density of true negatives $\phi_{\text{TN}}(\hat{x})$ as follows:

$$\begin{aligned} \phi_{\text{TN}}(\hat{x}) &= (1-\beta+\beta)[\alpha\phi_{(1)}(\hat{x}) + (1-\alpha)\phi_{(2)}(\hat{x})] \\ &= (1-\beta)\alpha\phi_{(1)}(\hat{x}) + \beta(1-\alpha)\phi_{(2)}(\hat{x}) \qquad (37) \\ &\quad + \beta\alpha\phi_{(1)}(\hat{x}) + (1-\beta)(1-\alpha)\phi_{(2)}(\hat{x}) \qquad (38) \end{aligned}$$

Equation (37) is a also component of $\phi_{\text{TN}}(\hat{x})$. The parameter $1-\beta$ controls the proportion of *easy sample components* $\alpha\phi_{(1)}(\hat{x})$, while β controls the proportion of *hard sample components* $(1-\alpha)\phi_{(2)}(\hat{x})$ that have been incorrectly classified by the classifier. Thus, Equation (37) can be interpreted as the density of hard true negative samples $p(\hat{x}, \text{HARD}|\text{TN})$ with a hardness level of β, a larger value of β (approaching 1) leads to $p(\hat{x}, \text{HARD}|\text{TN})$ contains higher proportion of *hard negative sample component* that have been incorrectly classified by the classifier.

Similarly, Equation (38), the mirrored counterpart of Equation (37), represents the density of easy true negative samples $p(\hat{x}, \text{EASY}|\text{TN})$ with an easiness level of β. A larger value of β (approaching 1) leads to Equation (38) contains higher proportion of *easy negative sample component* that have been correctly classified by the classifier.

The above algebraic transformation can be viewed as a complete probability decomposition of the distribution $\phi_{\text{TN}}(\hat{x})$:

$$\begin{aligned} \phi_{\text{TN}}(\hat{x}) &\triangleq p(\hat{x}|\text{TN}) \\ &= p(\hat{x}, \text{HARD} = \beta|\text{TN}) + p(\hat{x}, \text{EASY} = \beta|\text{TN}) \\ &= (1-\beta)\alpha\phi_{(1)}(\hat{x}) + \beta(1-\alpha)\phi_{(2)}(\hat{x}) + \beta\alpha\phi_{(1)}(\hat{x}) + (1-\beta)(1-\alpha)\phi_{(2)}(\hat{x}) \end{aligned}$$

We take $p(\hat{x}, \text{HARD}|\text{TN})$ given by Eq (37) as the target sampling distribution. As $p(\hat{x}, \text{HARD}|\text{TN})$ is a component of $\phi_{\text{TN}}(\hat{x})$, it is conditioned on the true principle. Furthermore, the hardness level β is conditioned on the hard principle since it controls the proportion of hard negative samples that have been incorrectly classified by the classifier.

The above full probability decomposition of the distribution $\phi_{\text{TN}}(\hat{x})$ allows us to avoid the overly strong parametric assumptions of VMF distribution. So the desired sampling distribution for drawing $\{x_i^-\}_{i=1}^N$ conditioning on both true

principle and hard principle can be derived as:

$$\begin{aligned}\psi(\hat{x};\alpha,\beta) &\triangleq p(\hat{x}|\text{TN},\text{HARD}) \\ &= p(\hat{x},\text{HARD}=\beta|\text{TN})/p(\text{HARD}=\beta|\text{TN}) \end{aligned} \tag{39}$$

$p(\text{HARD}=\beta|\text{TN})$ is the normalization constant, which can be calculated by the marginal integration of $p(\hat{x},\text{HARD}=\beta|\text{TN})$ over $\hat{x}$

$$\begin{aligned} p(\text{HARD}=\beta|\text{TN}) &= \int_{-\infty}^{\infty} p(\hat{x},\text{HARD}=\beta|\text{TN})d\hat{x} \\ &= \int_{-\infty}^{\infty} (1-\beta)\alpha\phi_{(1)}(\hat{x}) + \beta(1-\alpha)\phi_{(2)}(\hat{x})d\hat{x} \\ &= (1-\beta)\alpha + \beta(1-\alpha) \end{aligned} \tag{40}$$

Finally, the desired sampling distribution for drawing $\{x_i^-\}_{i=1}^N$ conditioning on both true principle and hard principle given by Eq (39) can be calculated as:

$$\psi(\hat{x};\alpha,\beta) = \frac{(1-\beta)\alpha\phi_{(1)}(\hat{x}) + \beta(1-\alpha)\phi_{(2)}(\hat{x})}{(1-\beta)\alpha + \beta(1-\alpha)} \tag{41}$$

As a point of transition between hardness and easiness, when $\beta = 0.5$,

$$\psi(\hat{x};\alpha,\beta) = \phi_{\text{TN}}(\hat{x}),$$

which indicates that samples are drawn from the original class conditional density of true negatives. On the other hand, when $\beta = 1$, the target of sampling distribution is *hard negative sample component* that have been misclassified by the classifier, and the function becomes $\psi(\hat{x};\alpha,\beta) = (1-\alpha)\phi_{(2)}(\hat{x})$.

It's worth noting that since we have not introduced the un-normalized von Mises-Fisher (vMF) assumption, the distribution $\psi(\hat{x};\alpha,\beta)$ is now a normalized distribution, and we can calculate its normalization constant $p(\text{HARD}=\beta|\text{TN})$ exactly using Eq (40).

Revision of Section 3.2: Monte Carlo Importance Sampling.

Now we have in batch N i.i.d unlabeled samples $\{\hat{x}_i\}_{i=1}^N \sim \phi_{\text{UN}} = \tau^-\phi_{\text{TN}} + \tau^+\phi_{\text{FN}}$. ϕ_{UN} is a function of class prior probabilities of ground truth labels τ, and the performance of encoder α, and the original distribution of unlabeled samples' scores $\phi(\hat{x})$.

In addition, we have a desired sampling distribution, denoted by ψ, of hard true negative samples, we can approximate the expectation scores over hard and true samples using classic Monte-Carlo importance sampling (Hesterberg, 1988; Bengio & Senécal, 2008):

$$\begin{aligned} \mathbb{E}_{\hat{x}\sim\psi}\hat{x} &= \int_{+\infty}^{+\infty} \hat{x}\frac{\psi(\hat{x};\alpha,\beta)}{\phi_{\text{UN}}(\hat{x})}\phi_{\text{UN}}(\hat{x})d\hat{x} \\ &= \mathbb{E}_{\hat{x}\sim\phi_{\text{UN}}}\hat{x}\frac{\psi(\hat{x};\alpha,\beta)}{\phi_{\text{UN}}(\hat{x})} \\ &\simeq \frac{1}{N}\sum_{i=1}^{N}\omega_i\hat{x}_i \end{aligned} \tag{42}$$

where ω_i is the density ratio between desired sampling distribution ψ and unlabeled data distribution ϕ, which can be calculated by:

$$\omega_i(\hat{x}_i;\alpha,\beta) = \frac{\psi(\hat{x};\alpha,\beta)}{\phi_{\text{UN}}(\hat{x})} \tag{43}$$

$\omega_i(\hat{x}_i;\alpha,\beta)$ is a function of empirical empirical cumulative distribution $\Phi_n(\hat{x})$ (sample information) and class prior probability τ (prior information). Since the desired sampling distribution $\psi(\hat{x};\alpha,\beta)$ now is normalized, the revised version

of BCL do not involves the calculation of the partition function any more. The revised BCL therefore involves only two steps: (i) calculating the empirical C.D.F, and (ii) calculating the weights by Eq (43).The implementation code of BCL is available at the anonymous repository:

https://anonymous.4open.science/r/BCL/main.py

The importance weight $\omega(\hat{x}, \alpha, \beta)$ essentially assigns customized weights to the N unlabeled samples. The parameter α corresponds to the encoder's macro-AUC metric for false negative debiasing, controlling both the desired sampling distribution $\psi(\hat{x}; \alpha, \beta)$ and the unlabeled scores distribution $\phi_{\text{UN}}(\hat{x})$, which can be empirically estimated using a validation dataset during the training process. The parameter β controls the hardness level required for specific task scenarios, by setting the proportion of misclassified *hard negative component* in the desired sampling distribution $\psi(\hat{x}; \alpha, \beta)$. Since Eq (37) is a component of the full probability decomposition of $\phi_{\text{TN}}(\hat{x})$, varying the value of β actually dividing the sample space of the original true negative samples, selecting a subset to form a new sample space as sampling target, without altering the asymptotically unbiased estimation of Eq (42) for the scores of true negatives.

Experimental details:

(1) α represents the macro AUC of the encoder, which is fixed to a function that grows at a constant rate with the training epoch. Specifically, we set α as follows: $\alpha = 0.5 + 0.35/400 * epoch$, where the training epoch is fixed at 400 epochs.

(2) β is the desired hardness level on hard negative samples, and we set it as $\beta = 1$.

The experimental setup is identical to that of DCL, HCL, and SimCLR. We fix the number of negative samples N as 510 and conduct the experiment six times to obtain six results of ACC1: [92.24, 92.26, 92.19, 92.23, 92.27, 92.25, 92.26]. At the 5% significance level, BCL outperformed HCL on the CIFAR10 dataset. To replicate the experiment and comparative methods, please use the following command.

```
python main.py --dataset_name cifar10 --batch_size 256 --estimator BCL
python main.py --dataset_name cifar10 --batch_size 256 --estimator HCL
python main.py --dataset_name cifar10 --batch_size 256 --estimator DCL
python main.py --dataset_name cifar10 --batch_size 256 --estimator SimCLR
```

Due to time constraints, we promise to report BCL's performance on ImageNet100 in the resubmitted version.

Acknowledgments: we would like to express our gratitude to the anonymous reviewers for providing valuable suggestions to enhance the quality of our paper.