Journal of Machine Learning Research 26 (2025) 1-47 Submitted 9/23; Revised 7/25; Published 8/25

Optimal Sample Selection Through Uncertainty Estimation and Its Application in Deep Learning

Yong Lin * YLINDF@CONNECT.UST.HK
Department of Computer Science
Hong Kong University of Science and Technology
Hong Kong, China

Chen Liu * CLIUDH@CONNECT.UST.HK
Department of Mathematics
Hong Kong University of Science and Technology
Hong Kong, China

Chenlu Ye * CHENLUY3@ILLINOIS.EDU
Siebel School of Computing and Data Science
University of Illinois Urbana-Champaign
Illinois, USA

Qing Lian QLIANAB@CONNECT.UST.HK
Department of Computer Science
Hong Kong University of Science and Technology
Hong Kong, China

Yuan Yao YUANY@UST.HK
Department of Mathematics
Hong Kong University of Science and Technology
Hong Kong, China

Tong Zhang TONGZHANG@TONGZHANG-ML.ORG
Siebel School of Computing and Data Science
University of Illinois Urbana-Champaign
Illinois, USA

Editor: Russell Greiner

Abstract

Modern deep learning heavily relies on large labeled datasets, which often comse with high costs in terms of both manual labeling and computational resources. To mitigate these challenges, researchers have explored the use of informative subset selection techniques. In this study, we present a theoretically optimal solution for addressing both sampling with and without labels within the context of linear softmax regression. Our proposed method, COPS (unCertainty based OPtimal Sub-sampling), is designed to minimize the expected loss of a model trained on subsampled data. Unlike existing approaches that rely

. * for equal contribution.

on explicit calculations of the inverse covariance matrix, which are not easily applicable to deep learning scenarios, COPS leverages the model's logits to estimate the sampling ratio. This sampling ratio is closely associated with model uncertainty and can be effectively applied to deep learning tasks. Furthermore, we address the challenge of model sensitivity to misspecification by incorporating a down-weighting approach for low-density samples, drawing inspiration from previous works.

To assess the effectiveness of our proposed method, we conducted extensive empirical experiments using deep neural networks on benchmark datasets. The results consistently showcase the superior performance of COPS compared to baseline methods, reaffirming its efficacy[1].

Keywords: Subset Selection, Uncertainty Estimation, Model Misspecification

1. Introduction

In recent years, deep learning has achieved remarkable success in various domains, including computer vision (CV), natural language processing (NLP), reinforcement learning (RL) and autonomous driving, among others. However, the success of deep learning often relies on a large amount of labeled data. This requirement not only incurs expensive labeling processes but also necessitates substantial computational costs. To address this challenge, an effective approach is to sub-sample from the training data. We can further learn a deep neural network to achieve comparable performance with that trained on the full dataset.

Suppose we want to learn a model that predict the label $\boldsymbol{y}$ based on the input $\boldsymbol{x}$. Suppose the whole dataset $\mathcal{S}$ contains n samples. The objective is to sub-sample from $\mathcal{S}$ and train a model based on the sub-sampled data. To improve performance the of model trained on the sub-sampled data, a popular method is to use assign non-uniform sampling probability to each sample in $\mathcal{S}$, typically more informative sample with larger weight. There are two practical situations considers: on one situation, the data in $\mathcal{S}$ already is already labeled so we can directly train a model on the sub-sampled data. This line of works are typically referred to coreset selection (Har-Peled and Kushal, 2005; Feldman and Langberg, 2011; Wang et al., 2018; Borsos et al., 2020; Zhou et al., 2022b). In another situation, the data in $\mathcal{S}$ only contains the input $\boldsymbol{x}$ and we need to query the label of the sub-sampled data $\boldsymbol{y}$. Then a model is trained on the labeled sub-samples. Methods that sample unlabeled datasets are associated with the field of active learning (Culotta and McCallum, 2005; Ash et al., 2019; Ren et al., 2021). We mainly refer these two settings as sampling with and without labels.

Following Wang et al. (2018); Wang (2019), we mainly consider sampling with replacement. We theoretically derive the optimal sub-sampling scheme for with and without labels in linear softmax regression. Our objective is to minimize the expected loss of the resulting linear classifier on the selected subset. The optimal sampling ratio is closely connected to the uncertainty of the data which has been extensively explored in reinforcement learning (Jin et al., 2020; Dean et al., 2018; Bubeck et al., 2012; Lattimore and Szepesvári, 2020;

1. Our code can be find on `https://github.com/corwinliu9669/COPS`.

Ye et al., 2023). The detailed formulation and explanation of our sampling ratio is deferred to Section 3.1. We further show that the optimal sampling ratio is equivalent to the covariance of the output logits of independently trained models with proper scaling, which can be easily estimated in deep neural networks. We name our method as unCertainty based OPtimal Sub-sampling (COPS). While prior works such as Ting and Brochu (2018); Wang et al. (2018); Imberg et al. (2020) have explored related theoretical aspects, their approaches for estimating the sampling ratio are prohibitively expensive in the context of deep learning: Ting and Brochu (2018) relies on the influence function of each data which has been recognized as computationally demanding according to existing literature (Koh and Liang, 2017). Wang et al. (2018); Imberg et al. (2020) rely on the inverse of covariance matrix of input which is also computationally expensive due to the large dimensionality of the input data. There are also vast amount of literature on coreset selection and active learning, but few of them can claim optimality, which will be briefly reviewed in Section 2.

We then conduct empirical experiments on real-world datasets with modern neural architectures. Surprisingly, we find that directly applying COPS leads to bad performance which can be even inferior to that of random sub-sampling. Upon conducting a thorough analysis of the samples selected by COPS, we observe a tendency for the method to excessively prioritize data exhibiting high uncertainty, i.e., samples from the low density region. Notably, existing literature has established that model estimation can be highly sensitive to misspecification issues encountered with low density samples (He et al., 2022; Ye et al., 2023). It is important to note that the optimality of COPS is based on a well-specified linear model. Hence, this observation has motivated us to consider modifying COPS to effectively handle potential misspecification challenges. We use the short-hand notation u_i to represent the uncertainty of ith sample, which is our original sampling ratio up to some scaling. He et al. (2022); Ye et al. (2023) show that applying the reweighting $\frac{1}{\max\{\alpha,u_i\}}$ to each sample during linear regression can make models more robust to misspecification, where α is a hyper-parameter. Thus, we simply borrow the idea and modify the sampling ratio u_i by $\frac{u_i}{\max\{\alpha,u_i\}} \propto \min\{\alpha, u_i\}$. We show the effectiveness of this modification by numerical simulations and real-world data experiments in Section 4.

In Section 6, we conduct comprehensive experiments on several benchmark datasets, including SVHN, Places, and CIFAR10, using various backbone models such as ResNet20, ResNet56, MobileNetV2, and DenseNet121. We systematically compare our COPS with existing methods in the setting of sampling with replacement. Additionally, we verify the effectiveness of our approach on the CIFAR10-N dataset, which incorporates natural label noise. Furthermore, we extend our evaluation to include a NLP benchmark, IMDB, utilizing a GRU-based neural network. Across all these scenarios, our method consistently surpasses the baselines significantly, highlighting its superior performance. The contribution of this work can be summarized as follows:

- **Theoretical derivation**: The study theoretically derives the optimal sub-sampling scheme for well-specified linear softmax regression. The objective is to minimize the expected loss of the linear classifier on the sub-sampled dataset. The optimal sampling ratio is found to be connected to the uncertainty of the data.

- **COPS method**: The proposed method, named unCertainty based OPtimal Sub-sampling (COPS), provides an efficient approach for sampling with and without labels. We show that the sampling ratio can be efficiently estimated using the covariance of the logits of independently trained models, which addresses the computational challenges faced by previous approaches Ting and Brochu (2018); Wang et al. (2018); Imberg et al. (2020).

- **Modification to handle misspecification**: We empirically identified a potential issue with COPS, which overly emphasizes high uncertainty samples in the low-density region, leading to model sensitivity to misspecification. To address this, we draw inspiration from existing theoretical works (He et al., 2022; Ye et al., 2023) that downweight low-density samples to accommodate for the misspecification. By combining their techniques, we propose a modification to COPS that involves a simple thresholding of the sampling ratio. Both numerical simulations and real-world experiments demonstrate the significant performance improvements resulting from our straightforward modification.

- **Empirical Validation**: Empirical experiments are conducted on various CV and NLP benchmark datasets, including SVHN, Places, CIFAR10, CIFAR10-N and IMDB, utilizing different neural architectures including ResNet20, ResNet56, MobileNetV2, DenseNet121 and GRU. The results demonstrate that COPS consistently outperforms baseline methods in terms of performance, showcasing its effectiveness.

2. Related Works

Statistical Subsampling Methods. A vast amount of early methods adopts the statistical leverage scores to perform subsampling which is later used for ordinary linear regression (Drineas et al., 2012, 2011; Ma et al., 2014). The leverage scores are estimated approximately (Drineas et al., 2012; Clarkson et al., 2016) or combined with random projection (Meng et al., 2014). These methods are relative computational expensive in the context of deep learning when the input dimension is large. Some recent works (Ting and Brochu, 2018; Wang et al., 2018; Imberg et al., 2020) achieves similar theoretical properties with ours. However, Ting and Brochu (2018) is based on the influence function of each sample, which is computational expensive. Ting and Brochu (2018); Imberg et al. (2020) need to compute the inverse of covariance matrix, which is also impractical for deep learning.

Active Learning. This method designs acquisition functions to determine the importance of unlabeled samples and then trains models based on the selected samples with the inquired label (Ren et al., 2021). There are mainly uncertainty-based and representative-based active learning methods. **Uncertainty-based** methods select samples with higher uncertainty, which can mostly reduces the uncertainty of the target model (Ash et al., 2019). They design metrics such as entropy (Wang and Shang, 2014; Citovsky et al., 2023), confidence (Culotta and McCallum, 2005), margin (Joshi et al., 2009; Roth and Small, 2006), predicted loss (Yoo and Kweon, 2019) and gradient (Ash et al., 2019). Some recent works leverage variational autoencoders and adversarial networks to identify samples that are poorly represented by correctly labeled data (Sinha et al., 2019; Kim et al., 2021). Some of these works provide theoretical guarantees expressed as probabilistic rates, but they do not claim to achieve optimality (Ting and Brochu, 2018). The uncertainty-related technique has also been extended to RL (He et al., 2022; Ye et al., 2023). **Representative-based** methods are also known as the diversity based methods (Citovsky et al., 2023). They try to find samples with the feature that is most representative of the unlabeled dataset (Wei et al., 2015; Sener and Savarese, 2017). Wei et al. (2015) casts the problem of finding representative samples as submodular optimization problem. Sener and Savarese (2017) tries to find the representative sampling by clustering, which is later adopted in Ash et al. (2019).

Coreset Selection. This method aims to find a subset that is highly representative of the entire labeled dataset. Some early works have focused on designing coreset selection methods for specific learning algorithms, such as SVM (Tsang et al., 2005), logistic regression (Huggins et al., 2016), and Gaussian mixture models (Lucic et al., 2017). However, these methods cannot be directly applied to deep neural networks (DNNs). To address this limitation, a solution has been proposed that leverages bi-level optimization to find a coreset specifically tailored for DNNs (Borsos et al., 2020). This approach has been further enhanced by incorporating probabilistic parameterization (Zhou et al., 2022b,a). Another line of recent research efforts have aimed to identify coreset solutions with gradients that closely match those of the full dataset (Mirzasoleiman et al., 2020; Killamsetty et al., 2021; Ash et al., 2019).

3. Theoretical Analysis

Notation. We use bold symbols $\boldsymbol{x}$ and $\boldsymbol{y}$ to denote random variables and use $\mathbf{x}$ and $\mathbf{y}$ to denote deterministic values. Consider the d-dimensional covariate $\mathbf{x}$ and the categorical label $y \in \{c_0, c_1, \ldots, c_K\}$. Denote the joint distribution $(\boldsymbol{x}, \boldsymbol{y})$ as $\mathcal{D}$. For any matrix $\mathbf{X} \in \mathbb{R}^{d_1 \times d_2}$, define $\|\mathbf{X}\|_{\mathrm{op}}$, $\|\mathbf{X}\|_N$, and $\|\mathbf{X}\|_F$ to be its l_2 operator norm, nuclear norm and Frobenius norm, respectively. Given a vector $\mathbf{v} \in \mathbb{R}^d$ and a positive-semidefinite matrix $\mathbf{M}$, we use $\|\mathbf{v}\|_{\mathbf{M}}$ to denote $\sqrt{\mathbf{v}^\top \mathbf{M} \mathbf{v}}$ for short . The vectorized version of $\mathbf{X}$ is denoted as $\mathbf{Vec}(\mathbf{X}) = (X_1^\top, X_2^\top, \ldots, X_{d_2}^\top)^\top$, where X_j is the j-th column of $\mathbf{X}$. Let $\mathcal{S}$ denote the dataset containing n labeled samples, i.e., $\mathcal{S} := \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$. We use $\mathcal{S}_X$ to denote the unlabeled dataset $\mathcal{S} := \{\mathbf{x}_i\}_{i=1}^n$. Let $\otimes$ denote the Kronecker product. For a sequence of random variables $X_1, X_2, \ldots$, we say that $X_n = o_P(1)$ if for any $\epsilon > 0$, $\lim_{n\to\infty} \mathbb{P}(|X_n| \geq \epsilon) = 0$, and $X_n = O_P(1)$ if for all $\epsilon > 0$, there exists a finite $J > 0$ and a finite $N > 0$ such that $\mathbb{P}(X_n > J) < \epsilon$ for any $n > N$.

3.1 Optimal Sampling in Linear Softmax Regression

Consider a K-class categorical response variable $\boldsymbol{y} \in \{c_0, c_1, \ldots, c_K\}$ and a d-dimensional covariate $\boldsymbol{x}$. The conditional probability of $\boldsymbol{y} = c_k$ (for $k = 0, 1, \ldots, K$) given $\boldsymbol{x}$ is

$$p_k(\beta; \boldsymbol{x}) = \frac{\exp(\boldsymbol{x}^\top \beta_k)}{\sum_{l=0}^K \exp(\boldsymbol{x}^\top \beta_l)}, \tag{1}$$

where $\beta_k,\ k = 0, 1, \ldots, K$ are unknown regression coefficients belonging to a compact subset of $\mathbb{R}^d$. Following Yao and Wang (2019), we assume $\beta_0 = 0$ for identifiability. We further denote $\beta = (\beta_1^\top, \ldots, \beta_K^\top)^\top \in \mathbb{R}^{Kd}$. We use the bold symbol $\boldsymbol{\beta}$ to denote the d-by-K matrix $(\beta_1, \ldots, \beta_K)$. In the sequel, we first derive the optimal sub-sampling schemes for sampling with and without labels in linear softmax regression which minimize the expected test loss. Suppose the model is well-specified such that there exists an true parameter $\beta^* \in \mathbb{R}^{Kd}$ with $\mathbb{P}(\boldsymbol{y} = c_k|\boldsymbol{x}) = p_k(\beta^*; \boldsymbol{x})$ for all $\boldsymbol{x}$ and k. Define $\delta_k(\boldsymbol{y}) := \mathbb{I}(y = c_k)$ where $\mathbb{I}$ is the indicator function. Let $\ell(\beta; \boldsymbol{x}, \boldsymbol{y})$ denote the cross entropy loss on the sample $(\boldsymbol{x}, \boldsymbol{y})$ as

$$\ell(\beta; \boldsymbol{x}, \boldsymbol{y}) = -\sum_{k=0}^K \delta_k(\boldsymbol{y}) \log p_k(\beta; \boldsymbol{x}) = \sum_{l=1}^K \left[-\delta_k(\boldsymbol{y})\boldsymbol{x}^\top \beta_k\right] + \log\{1 + \sum_{l=1}^K \exp(\boldsymbol{x}^\top \beta_l)\}. \tag{2}$$

We calculate the gradient and the hessian matrix of the loss function as follows:

$$\frac{\partial \ell(\beta; \boldsymbol{x}, \boldsymbol{y})}{\partial \beta} = -s(\beta; \boldsymbol{x}, \boldsymbol{y}) \otimes \boldsymbol{x}, \quad \text{and} \quad \frac{\partial^2 \ell(\beta; \boldsymbol{x}, \boldsymbol{y})}{\partial \beta^2} = \phi(\beta; \boldsymbol{x}) \otimes (\boldsymbol{x}\boldsymbol{x}^\top). \tag{3}$$

Here $s(\beta; \boldsymbol{x}, \boldsymbol{y})$ is a K-dimensional vector with each element $s_k(\beta; \boldsymbol{x}, \boldsymbol{y}) = \delta_k(y) - p_k(\beta; \boldsymbol{x})$ for $k = 1, ..., K$; and $\phi(\beta; \boldsymbol{x})$ is a $K \times K$ matrix with element

$$\phi_{kk}(\beta; \boldsymbol{x}) = p_k(\beta; \boldsymbol{x}) - p_k(\beta; \boldsymbol{x})^2, \phi_{k_1 k_2}(\beta; \boldsymbol{x}) = -p_{k_1}(\beta; \boldsymbol{x}) p_{k_2}(\beta; \boldsymbol{x}), \tag{4}$$

where $k, k_1, k_2 = 1, ..., K$ and $k_1 \neq k_2$. We further define the $K \times K$ matrix $\psi(\beta; \boldsymbol{x}, y) := s(\beta; \boldsymbol{x}, y) s(\beta; \boldsymbol{x}, y)^\top$. For $k_1, k_2 = 1, ..., K$, we have

$$\psi_{k_1 k_2}(\beta; \boldsymbol{x}, \boldsymbol{y}) = [\delta_{k_1}(\boldsymbol{y}) - p_{k_1}(\beta; \boldsymbol{x})][\delta_{k_2}(\boldsymbol{y}) - p_{k_2}(\beta; \boldsymbol{x})]. \tag{5}$$

By direct calculation, we have $\mathbb{E}_{\boldsymbol{y}}[\psi(\beta^*; \boldsymbol{x}, \boldsymbol{y})|\boldsymbol{x}] = \phi(\beta^*; \boldsymbol{x})$. We use $\mathcal{L}(\beta; \mathcal{D})$ to denote the expected cross-entropy loss on the distribution $\mathcal{D}$ as

$$\mathcal{L}(\beta; \mathcal{D}) = \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}) \sim \mathcal{D}}[\ell(\beta; \boldsymbol{x}, \boldsymbol{y})], \tag{6}$$

It is easy to know that $\beta^* = \arg\min_{\beta \in \mathbb{R}^{Kd}} \mathcal{L}(\beta; \mathcal{D})$. Given the dataset $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, we use $\mathcal{L}(\beta; \mathcal{S})$ to denote the cross entropy loss of β on $\mathcal{S}$, i.e.,

$$\mathcal{L}(\beta; \mathcal{S}) := \frac{1}{n} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{S}} \ell(\beta; \mathbf{x}, \mathbf{y}). \tag{7}$$

We further use $\mathcal{L}(\beta)$ to denote $\mathcal{L}(\beta;\mathcal{S})$ when it is clear from the context. Recall that $\mathcal{L}(\beta;\mathcal{S})$ is the negative likelihood achieved by β on $\mathcal{S}$, then the maximum log-likelihood estimation (MLE) solution of β on $\mathcal{S}$ is

$$\hat{\boldsymbol{\beta}}_{\text{MLE}} := \operatorname*{arg\,min}_{\beta\in\mathbb{R}^{Kd}} \mathcal{L}(\beta;\mathcal{S}).$$

We further define

$$\begin{aligned}\mathbf{M}_X(\beta;\mathcal{D}) &:= \frac{\partial^2\mathcal{L}(\beta;\mathcal{D})}{\partial^2\beta} = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}[\phi(\beta;\boldsymbol{x})\otimes(\boldsymbol{x}\boldsymbol{x}^\top)],\\ \mathbf{M}_X(\beta;\mathcal{S}) &:= \frac{\partial^2\mathcal{L}(\beta;\mathcal{S})}{\partial^2\beta} = \frac{1}{n}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}}[\phi(\beta;\mathbf{x})\otimes(\mathbf{x}\mathbf{x}^\top)].\end{aligned}$$

Given any $\beta\in\mathbb{R}^{Kd}$, we have

$$\begin{aligned}\Delta(\beta) =&\mathcal{L}(\beta;\mathcal{S}) - \mathcal{L}(\hat{\beta}_{\text{MLE}};\mathcal{S})\\ =&\frac{1}{n}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}}(\beta-\hat{\beta}_{\text{MLE}})^\top\left(\phi(\hat{\beta}_{\text{MLE}};\boldsymbol{x})\otimes(\boldsymbol{x}\boldsymbol{x}^\top)\right)(\beta-\hat{\beta}_{\text{MLE}}) + R(\beta),\end{aligned}\tag{8}$$

where $R(\beta)$ is the high order residual term which will be specified later. We are interested in finding a β which achieves a small $\Delta(\beta)$.

Sampling with Label. Suppose that we have access to the entire labeled dataset, i.e., $\mathcal{S}=\{\mathbf{x}_i,\mathbf{y}_i\}_{i=1}^n$. We assign an sampling $\pi(\mathbf{x},\mathbf{y})$ to each samples in $\mathcal{S}$ and then randomly select a subset of size r according to $\pi(\mathbf{x},\mathbf{y})$. Denote the selected subset as $\bar{\mathcal{S}}=\{(\bar{\mathbf{x}}_i,\bar{\mathbf{y}}_i)\}_{i=1}^r$. We then estimate the parameter $\bar{\beta}$ based on the weighted loss

$$\bar{\beta} = \operatorname*{arg\,min}_{\beta}\left(-\frac{1}{r}\sum_{(\bar{\mathbf{x}},\bar{\mathbf{y}})\in\bar{\mathcal{S}}}\frac{1}{\pi(\bar{\mathbf{x}},\bar{\mathbf{y}})}\left(\sum_{k=1}^K\delta_k(\bar{\mathbf{y}})\bar{\mathbf{x}}^\top\beta_k - \log\{1+\sum_{l=1}^K\exp(\bar{\mathbf{x}}^\top\beta_l)\}\right)\right),\tag{9}$$

Since $\bar{\beta}$ may not exist and not integrable, we define $\bar{\beta}=0$ if it does not exist. We want the $\bar{\beta}$ to achieve low expected loss $\Delta(\bar{\beta})$. Our goal is to find a sampling scheme parameterized by $\pi(\cdot)$ which minimizes the expectation of $\Delta(\bar{\beta})$, i.e.,

$$\min_{\pi}\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}\left[\Delta(\bar{\beta})\right],\tag{10}$$

where the expectation is taking over the randomness in sampling based on $\pi(\cdot)$.

Sampling without Label. For sampling without labels problem, we have the unlabeled dataset $\mathcal{S}_X=\{\mathbf{x}_i\}_{i=1}^n$. We aim to assign a sampling weight $\pi(\mathbf{x})$ for each sample $\mathbf{x}$ in $\mathcal{S}_X$. Here we use the subscript X in π_X to explicitly show that the sampling ratio in sampling without labels only depends on $\boldsymbol{x}$. When it is clear from the context, we also use $\pi(\mathbf{x})$ to denote $\pi_X(\mathbf{x})$ for simplicity. Based on the sampled subset and queried label, which is also denoted as $\bar{\mathcal{S}} = \{(\bar{\mathbf{x}}_i, \bar{\mathbf{y}}_i)\}_{i=1}^r$, we train the classifier $\bar{\beta}$ on the weighted loss as shown in Eqn (9) by replacing the weight $\pi(\mathbf{x}, \mathbf{y})$ with $\pi(\mathbf{x})$, i.e.,

$$\bar{\beta} = \arg\min_{\beta} \left(-\frac{1}{r} \sum_{(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \bar{\mathcal{S}}} \frac{1}{\pi(\bar{\mathbf{x}})} \left(\sum_{k=1}^{K} \delta_k(\bar{\mathbf{y}}) \bar{\mathbf{x}}^\top \beta_k - \log\{1 + \sum_{l=1}^{K} \exp(\bar{\mathbf{x}}^\top \beta_l)\} \right) \right). \tag{11}$$

Similar to that in sampling with labels, we try to find a sampling scheme π which optimizes the following equation:

$$\min_{\pi} \mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S}_X, \pi} \left[\Delta(\bar{\beta}) \right]. \tag{12}$$

Before presenting the main theorem, we introduce two assumptions, which are standard in the subsampling literature (Wang et al., 2018; Yao and Wang, 2019).

Assumption 1. *The covariance matrix $\boldsymbol{M}(\beta_{MLE}; \mathcal{S})$ goes to a positive definite matrix $\boldsymbol{M}(\beta_{MLE}; \mathcal{D})$ in probability; and $n^{-2} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{S}} \|\mathbf{x}\|^3 = O_P(1)$.*

Assumption 2. *For $k = 2, 4$, $n^{-2} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{S}} \pi(\mathbf{x})^{-1} \|\mathbf{x}\|^k = O_P(1)$; and there exists some $\delta > 0$ such that $n^{-(2+\delta)} \sum_{(\mathbf{x}, y) \in \mathcal{S}} \pi(\mathbf{x})^{-1-\delta} \|\mathbf{x}\|^{2+\delta} = O_P(1)$.*

Assumption 1 requires that the asymptotic matrix $\mathbf{M}(\hat{\beta}_{\text{MLE}}; \mathcal{D})$ is non-singular and $\mathbb{E}\|x\|^3$ is upper-bounded. Assumption 2 imposes conditions on both subsampling probability and covariates.

Theorem 1 (Optimal sampling in linear softmax regression)**.** *Suppose that the Assumptions 1 and 2 hold.*

(a) For sampling with labels, the optimal sampling ratio of sampling with labels that minimizes $\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S}, \pi}[\mathcal{L}(\bar{\beta}; \mathcal{S}) - \mathcal{L}(\beta^*; \mathcal{S})]$ is

$$\pi(\mathbf{x}, \mathbf{y}) = \frac{\sqrt{Tr\left(\psi(\hat{\beta}_{MLE}; \mathbf{x}, \mathbf{y}) \otimes (\mathbf{x}\mathbf{x}^\top) \boldsymbol{M}_X^{-1}(\hat{\beta}_{MLE}; \mathcal{S}) \right)}}{\sum_{(\mathbf{x}', \mathbf{y}') \in \mathcal{S}} \sqrt{Tr\left(\psi(\hat{\beta}_{MLE}; \mathbf{x}', \mathbf{y}') \otimes (\mathbf{x}'(\mathbf{x}')^\top) \boldsymbol{M}_X^{-1}(\hat{\beta}_{MLE}; \mathcal{S}) \right)}}. \tag{13}$$

(b) For sampling without labels, the optimal sampling ratio of sampling without labels that minimizes $\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S}_X, \pi}[\mathcal{L}(\bar{\beta}; \mathcal{S}) - \mathcal{L}(\beta^*; \mathcal{S})]$ is

$$\pi(\mathbf{x}) = \frac{\sqrt{Tr\left(\phi(\hat{\beta}_{MLE}; \mathbf{x}) \otimes (\mathbf{x}\mathbf{x}^\top) \boldsymbol{M}_X^{-1}(\hat{\beta}_{MLE}; \mathcal{S}) \right)}}{\sum_{\mathbf{x}' \in \mathcal{S}_X} \sqrt{Tr\left(\phi(\hat{\beta}_{MLE}; \mathbf{x}') \otimes (\mathbf{x}'(\mathbf{x}')^\top) \boldsymbol{M}_X^{-1}(\hat{\beta}_{MLE}; \mathcal{S}) \right)}}. \tag{14}$$

The interpretation of the optimal sampling ratio will become clearer as we present the results for the binary logistic regression, which we will discuss later on. In the proof, by using the asymptotic variance of $\bar{\beta}$ first derived in Wang et al. (2018); Yao and Wang (2019) and Taylor expansions, we can approximate the gap $\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}\left[\mathcal{L}(\bar{\beta};\mathcal{S}) - \mathcal{L}(\beta^*;\mathcal{S})\right]$ for sampling with labels by

$$\begin{aligned}&\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}\left[\mathcal{L}(\bar{\beta};\mathcal{S}) - \mathcal{L}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\right]\\ &= \frac{1}{rn^2}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}}\frac{1}{\pi(\mathbf{x},\mathbf{y})}\mathrm{Tr}\left(\psi(\hat{\beta}_{\mathrm{MLE}};\mathbf{x},\mathbf{y})\otimes(\mathbf{x}\mathbf{x}^\top)\mathbf{M}_X^{-1}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\right) + \mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}R(\bar{\beta}), \end{aligned}\tag{15}$$

and approximate the gap $\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S}_X,\pi}\left[\mathcal{L}(\bar{\beta};\mathcal{S}) - \mathcal{L}(\beta^*;\mathcal{S})\right]$ for sampling without label by

$$\begin{aligned}&\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S}_X,\pi}\left[\mathcal{L}(\bar{\beta};\mathcal{S}) - \mathcal{L}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\right]\\ &= \frac{1}{rn^2}\sum_{\mathbf{x}\in\mathcal{S}_X}\frac{1}{\pi(\mathbf{x})}\mathrm{Tr}\left(\phi(\hat{\beta}_{\mathrm{MLE}};\mathbf{x})\otimes(\mathbf{x}\mathbf{x}^\top)\mathbf{M}_X^{-1}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\right) + \mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}R(\bar{\beta}). \end{aligned}\tag{16}$$

The residual term $\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}R(\beta)$ is bounded above by $o(\|\bar{\beta} - \hat{\beta}_{\mathrm{MLE}}\|_2^2)$ in Lemma 1 and it will be sufficiently small with when the whole sample size and the subsample size are large enough. Then, we obtain the minimizers of the two terms above with the Cauthy-Schwarz inequality separately. The detailed proof is in Appendix A.1. Note that distinct from Wang et al. (2018); Yao and Wang (2019) that aim to reduce the variance of $\bar{\beta}$, we target on the expected generalization loss (refer to Section 5 for a more detailed discussion). However, directly computing our sampling ratio as well as those in Wang et al. (2018); Yao and Wang (2019) is computationally prohibitive in deep learning, since they rely on the inverse of the covariance matrix. Whereas, as we will show later, our sampling ratio is closely connected to sample uncertainty and can be effectively estimated by the output of DNN.

Now we illustrate the main intuition for the optimal sampling by considering the binary logistic classification problem as an example. In this case, we known that $K = 1$, $\boldsymbol{y} \in \{c_0, c_1\}$, and $\beta = \beta_1 \in \mathbb{R}^d$. Correspondingly, the binary logistic regression model is in the following form:

$$p_1(\beta;\boldsymbol{x}) = \frac{\exp(\boldsymbol{x}^\top\beta_1)}{1+\exp(\boldsymbol{x}^\top\beta_1)}.$$

The covariance matrix becomes

$$M_X = 1/n\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}}(p_1(\hat{\beta}_{\mathrm{MLE}};\mathbf{x}) - p_1(\hat{\beta}_{\mathrm{MLE}};\mathbf{x})^2)\mathbf{x}\mathbf{x}^\top.$$

Corollary 1 (Logistic regression optimal sampling)**.** *Suppose that the Assumptions 1 and 2 hold.*

(a) *For sampling with labels, the optimal sampling ratio that minimizes* $\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}[\mathcal{L}(\bar{\beta};\mathcal{S}) - \mathcal{L}(\hat{\beta}_{MLE};\mathcal{S})]$ *is*

$$\pi(\mathbf{x},\mathbf{y}) = \frac{\left|\delta_1(y) - p_1(\hat{\boldsymbol{\beta}}_{MLE};\mathbf{x})\right| \|\mathbf{x}\|_{M_X^{-1}}}{\sum_{(\mathbf{x}',\mathbf{y}')\in\mathcal{S}} \left|\delta_1(y) - p_1(\hat{\boldsymbol{\beta}}_{MLE};\mathbf{x}')\right| \|\mathbf{x}'\|_{M_X^{-1}}}. \tag{17}$$

(b) *For sampling without labels, the optimal sampling ratio that minimizes* $\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S}_X,\pi}[\mathcal{L}(\bar{\beta};\mathcal{S}) - \mathcal{L}(\hat{\beta}_{MLE};\mathcal{S})]$ *is*

$$\pi(\mathbf{x}) = \frac{\sqrt{p_1(\hat{\boldsymbol{\beta}}_{MLE};\mathbf{x}) - p_1(\hat{\boldsymbol{\beta}}_{MLE};\mathbf{x})^2}\|\mathbf{x}\|_{M_X^{-1}}}{\sum_{\mathbf{x}'\in\mathcal{S}} \sqrt{p_1(\hat{\boldsymbol{\beta}}_{MLE};\mathbf{x}) - p_1(\hat{\boldsymbol{\beta}}_{MLE};\mathbf{x})^2}\|\mathbf{x}'\|_{M_X^{-1}}}. \tag{18}$$

Here $\|\mathbf{x}\|_{M_X^{-1}}$ *is short for* $\mathbf{x}^\top M_X^{-1}\mathbf{x}$.

Intuition of the optimal sampling ratio. The optimal ratio for sampling with labels is proportional to $\left|\delta_1(y) - p_1(\hat{\boldsymbol{\beta}}_{\text{MLE}};\mathbf{x})\right| \cdot \|\mathbf{x}\|_{M_X^{-1}}$, which is decomposed of two components:

- $\left|\delta_1(y) - p_1(\hat{\boldsymbol{\beta}}_{\text{MLE}};\mathbf{x})\right|$ is related to the prediction error of $\hat{\beta}_{\text{MLE}}$.
- $\|\mathbf{x}\|_{M_X^{-1}}$ has been widely explored in RL literature which is connected to uncertainty. Specifically, $\|\mathbf{x}\|^2_{M_X^{-1}}$ represents the inverse of the effective sample number in the $\mathcal{S}$ along the $\mathbf{x}$ direction Jin et al. (2020). A larger $\|\mathbf{x}\|^2_{M_X^{-1}}$ indicates that there are less effective samples in the $\mathbf{x}$ direction. In this case, the prediction on $\mathbf{x}$ will be more uncertain. Therefore, $\|\mathbf{x}\|_{M_X^{-1}}$ is used to characterize the uncertainty along the $\mathbf{x}$ direction by .

Samples with significant uncertainty and substantial prediction errors will result in a higher sampling weight for sampling with labels. As for the sampling without labels, $\left|\delta_1(y) - p_1(\hat{\boldsymbol{\beta}}_{\text{MLE}};\mathbf{x})\right|$ is replaced by $\sqrt{p_1(\hat{\boldsymbol{\beta}}_{\text{MLE}};\mathbf{x}) - p_1(\hat{\boldsymbol{\beta}}_{\text{MLE}};\mathbf{x})^2}$ as we take conditional expectation over $\boldsymbol{y}$ since

$$p_1(\hat{\boldsymbol{\beta}}_{\text{MLE}};\mathbf{x}) - p_1(\hat{\boldsymbol{\beta}}_{\text{MLE}};\mathbf{x})^2 \approx \mathbb{E}_{\boldsymbol{y}|\boldsymbol{x}=\mathbf{x}}(\delta_1(y) - p_1(\hat{\boldsymbol{\beta}}_{\text{MLE}};\mathbf{x}))^2,$$

as $n\to\infty$. $\sqrt{p_1(\hat{\boldsymbol{\beta}}_{\text{MLE}};\mathbf{x}) - p_1(\hat{\boldsymbol{\beta}}_{\text{MLE}};\mathbf{x})^2}$ assigns large weights to those samples near the decision boundary. In summary, the optimal sampling ratios can be determined by weighting the uncertainty of samples with their corresponding prediction errors.

3.2 Efficient approximation of the optimal sampling ratio

There are some issues in estimating the optimal sampling ratio in Eqn (13) and (14):

(a) We can not obtain $\hat{\boldsymbol{\beta}}_{\text{MLE}}$ in practice since it is solved on the whole dataset;

(b) Calculating the inverse of the covariance matrix $\mathbf{M}_X(\hat{\beta}_{\mathrm{MLE}}; \mathcal{S})$ is computationally prohibitive due to the high dimentionality in deep learning.

To solve the issue (a), Wang et al. (2018) proposes to fit a β on the held out probe dataset $\mathcal{S}'$ (a small dataset independent of $\mathcal{S}$) to replace $\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}$. Whereas, the issue (b) remains to be the major obstacle for our method as well as those in Ting and Brochu (2018); Yao and Wang (2019). In the following part, we will by-pass the issue (b) by showing that $\psi(\hat{\beta}_{\mathrm{MLE}}; \mathbf{x}, y) \otimes (\mathbf{x}\mathbf{x}^\top)\mathbf{M}_X^{-1}(\hat{\beta}_{\mathrm{MLE}}; \mathcal{S})$ is related to the standard deviation of the output logits from independently trained models. To be more specific, we fit J independent MLE linear classifiers $\{\hat{\boldsymbol{\beta}}^{(j)}\}_{j=1}^J$ on J probe datasets $\{\mathcal{S}^{(j)}\}_{j=1}^J$ which is independent of $\mathcal{S}$. We then show that for each sample $(\mathbf{x}, \mathbf{y})$ in $\mathcal{S}$, we can estimate $\psi(\hat{\beta}_{\mathrm{MLE}}; \mathbf{x}, \mathbf{y}) \otimes (\mathbf{x}\mathbf{x}^\top)\mathbf{M}_X^{-1}(\hat{\beta}_{\mathrm{MLE}}; \mathcal{S})$ by the covariance of each model's logits i.e., $(\hat{\boldsymbol{\beta}}^{(j)})^\top \mathbf{x}$, as shown in Eqn (20) of Algorithm 1. Refer to Section 5 for a more detailed discussion on the difference between our methods with existing works.

Theorem 2 (Uncertainty estimation in linear models). *Supposing that Assumptions 1 and 2 hold, and $\|\mathbf{x}\| \le L$ for all $\mathbf{x} \in \mathcal{S}$, we have J probe datasets $\{\mathcal{S}^{(j)}\}_{j=1}^J$ and each $\mathcal{S}^{(j)}$ contains n' samples, we independently fit J MLE classifiers $\{\hat{\boldsymbol{\beta}}^{(j)}\}_{j=1}^J$ on $\{\mathcal{S}^{(j)}\}_{j=1}^J$. Denote $\tilde{\beta} = \frac{1}{J}\sum_{j=1}^J Vec(\hat{\boldsymbol{\beta}}^{(j)})$ and define $\Sigma_J(\mathbf{x})$ as Eqn (20) in Algorithm 1 , then, for a large range constant $B_J \ge K^4$, as $n' \to \infty$ and $n \to \infty$, for any $J \in [B_J]$ and $(\mathbf{x}, \mathbf{y}) \in \mathcal{S}$, we have*

$$n'\, Tr\Big(\psi(\tilde{\beta}; \mathbf{x}, y)\Sigma_J(\mathbf{x})\Big) - Tr\Big(\psi(\hat{\boldsymbol{\beta}}_{MLE}; \mathbf{x}, \mathbf{y}) \otimes (\mathbf{x}\mathbf{x}^\top)\boldsymbol{M}_X^{-1}(\hat{\boldsymbol{\beta}}_{MLE}; \mathcal{S})\Big) = O_P\Big(\frac{K^2}{\sqrt{J}}\Big),$$
$$n'\, Tr\Big(\phi(\tilde{\beta}; \mathbf{x})\Sigma_J(\mathbf{x})\Big) - Tr\Big(\phi(\hat{\beta}_{MLE}; \mathbf{x}) \otimes (\mathbf{x}\mathbf{x}^\top)\boldsymbol{M}_X^{-1}(\hat{\boldsymbol{\beta}}_{MLE}; \mathcal{S})\Big) = O_P\Big(\frac{K^2}{\sqrt{J}}\Big).$$

See Appendix A.2 for a proof. This theorem demonstrates that the uncertainty quantities can be approximated without explicitly calculating the inverse of covariance matrix. Instead, we only need to calculate a MLE estimator $\tilde{\beta}$ and the covariance of the output logits $\{(\hat{\boldsymbol{\beta}}^{(j)})^\top \mathbf{x}\}_{j=1}^J$ derived from J models. In other words, we only need to obtain $\{\hat{\boldsymbol{\beta}}^{(j)}\}$ on J probe sets, respectively. We then obtain the optimal sampling ratio through calculating $\Sigma_J(\mathbf{x})$, which is the covariance of $\{(\hat{\boldsymbol{\beta}}^{(j)})^\top \mathbf{x}\}_{j=1}^J$ as defined in Eqn (20). Notably, Theorem 2 provides only an upper bound on probe set number, meaning the actual error will not exceed this bound if an excessively large number of probes is used. However, in practice, we employ a much smaller probe size—typically just 3 to 5 probe sets—while still achieving state-of-the-art performance.Refer to Section 6 and Appendix E.6 for more details.

Approximations in Deep Learning. Our objective is to develop a sub-sampling method for deep learning. Let's consider a deep neural network $f_\theta(\mathbf{x})$ with parameters $\theta \in \mathbb{R}^{d'}$, where both d and d' are extremely large in the context of deep learning. There exist gaps between the theory presented in Section 3.1 and deep learning due to the nonlinearity involved in f_θ. However, we can leverage insights from learning theory, such as the Neural Tangent Kernel

```
Algorithm 1: Uncertainty estimation in linear softmax regression.
  Input: Probe datasets $\{\mathcal{S}^{(j)}\}_{j=1}^{J}$, the sampling dataset $\mathcal{S}$ for sampling with labels
         or $\mathcal{S}_X$ for sampling without labels.
  Output: The estimated uncertainty for each sample in $\mathcal{S}$ or $\mathcal{S}_X$.
1 For $j = 1, ..., J$, solve $\hat{\beta}^{(j)} = \arg\min_{\beta \in \mathbb{R}^{Kd}} \mathcal{L}(\beta; \mathcal{S}^{(j)})$. Denote
      $$\tilde{\beta} = \frac{1}{J}\sum_{j=1}^{J}\hat{\beta}^{(j)},\ \hat{\boldsymbol{\beta}}^{(j)} = [\hat{\beta}_1^{(j)}, \hat{\beta}_2^{(j)}, ..., \hat{\beta}_K^{(j)}],\ \text{and}\ \tilde{\boldsymbol{\beta}} = \frac{1}{J}\sum_{j=1}^{J}\hat{\boldsymbol{\beta}}^{(j)}. \quad (19)$$
2 For each $\mathbf{x}$, obtain $\{(\hat{\boldsymbol{\beta}}^{(j)})^\top \mathbf{x}\}_{j=1}^{J}$ and the covariance of them:
      $$\Sigma_J(\mathbf{x}) = \frac{1}{j-1}\sum_{j=1}^{j}\left(\left(\hat{\boldsymbol{\beta}}^{(j)}\right)^\top \mathbf{x} - \tilde{\boldsymbol{\beta}}^\top \mathbf{x}\right)\left(\left(\hat{\boldsymbol{\beta}}^{(j)}\right)^\top \mathbf{x} - \tilde{\boldsymbol{\beta}}^\top \mathbf{x}\right)^\top. \quad (20)$$
3 Get the predicted probability of $\mathbf{x}$, i.e., $p(\tilde{\beta}; \mathbf{x})$, as in Eqn (1). Estimate the
  uncertainty for each sample as following:
    • Case (1) sampling with labels. Obtain $\psi(\tilde{\beta}; \mathbf{x}, \mathbf{y})$ according to Eqn (5) and obtain
      the uncertainty estimation as
          $$u(\mathbf{x}, \mathbf{y}) = \mathrm{Tr}\left(\psi(\tilde{\beta}; \mathbf{x}, \mathbf{y})\Sigma_J(\mathbf{x})\right);$$
    • Case (2) sampling without labels. Obtain $\phi(\tilde{\beta}; \mathbf{x})$ according to Eqn (4) and obtain
      the uncertainty estimation as
          $$u(\mathbf{x}) = \mathrm{Tr}\left(\phi(\tilde{\beta}; \mathbf{x})\Sigma_J(\mathbf{x})\right).$$
```

```
Algorithm 2: COPS for sampling with labels on linear models
  Input: Training data $\mathcal{S}$, $J$ probe datasets $\{\mathcal{S}^{(j)}\}_{j=1}^{J}$, sub-sampling size $r$.
  Output: The selected subset $\bar{\mathcal{S}}$ and the model $\bar{\beta}$.
1 For each $(\mathbf{x}, \mathbf{y}) \in \mathcal{S}$, obtain $u(\mathbf{x}, \mathbf{y})$ by Algorithm 1 with $\{\mathcal{S}^{(j)}\}_{j=1}^{J}$;
2 Randomly draw $\bar{\mathcal{S}}$ containing $r$ samples from $\mathcal{S}$ by
    $\pi(\mathbf{x}, \mathbf{y}) = u(\mathbf{x}, \mathbf{y}) / \sum_{(\mathbf{x}', \mathbf{y}') \in \mathcal{S}} u(\mathbf{x}', \mathbf{y}')$.
3 Solve $\bar{\beta}$ on the weighted subset $\bar{\mathcal{S}}(\pi)$ according to Eqn (9).
```

Jacot et al. (2018), which demonstrates that a wide DNN can be approximated by a linear kernel with a fixed feature map $\nabla_\theta f_\theta(\cdot) \in \mathbb{R}^d \to \mathbb{R}^{d'}$. Consequently, we can approximate uncertainty by calculating the standard deviation from different linear kernels, as outlined in Theorem 2. Importantly, our method does not necessitate explicit computation of the

```
Algorithm 3: COPS for sampling without labels on linear models
   Input: Training data $\mathcal{S}_X$, $J$ probe datasets $\{\mathcal{S}^{(j)}\}_{j=1}^J$, sub-sampling size $r$.
   Output: The selected subset $\bar{\mathcal{S}}$ with inquired label and the model $\bar{\beta}$.
 1 For each $\mathbf{x} \in \mathcal{S}_X$, obtain $u(\mathbf{x})$ by Algorithm 1 with $\{\mathcal{S}^{(j)}\}_{j=1}^J$;
 2 Randomly draw $\bar{\mathcal{S}}_X$ containing $r$ samples from $\mathcal{S}$ by $\pi(\mathbf{x}) = u(\mathbf{x}) / \sum_{\mathbf{x}' \in \mathcal{S}} u(\mathbf{x}')$.
 3 Obtain the labeled data set $\bar{\mathcal{S}}$ by labeling each sample in $\bar{\mathcal{S}}_X$.
 4 Solve $\bar{\beta}$ on the weighted subset $\bar{\mathcal{S}}(\pi)$ according to Eqn (9).
```

linear kernel, as we only require the output $\beta^\top \mathbf{x}$ from Theorem 2. Thus, we can directly replace $\beta^\top \mathbf{x}$ with the output of the DNN, i.e., $f_\theta(\mathbf{x})$.

Let $f_{\theta,k}(\boldsymbol{x})$ denote the kth dimension of $f_\theta(\boldsymbol{x})$ for $k = 0, ..., K$. We denote the output probability of f_θ on sample $\boldsymbol{x}$ by

$$p(f_\theta; \boldsymbol{x}) = [p_0(f_\theta; \boldsymbol{x}), p_1(f_\theta; \boldsymbol{x}), ..., p_K(f_\theta; \boldsymbol{x})], \text{ where } p_k(f_\theta; \boldsymbol{x}) = \frac{\exp(f_{\theta,k}(\boldsymbol{x}))}{\sum_{l=0}^K \exp(f_{\theta,l}(\boldsymbol{x}))}.$$

Recall in Algorithm 1 that we train J independent linear models on J different probe sets, respectively. In practice, getting J additional probe sets can be costly. One option is to use bootstrap, where J subsets are resampled from a single probe set $\mathcal{S}'$ and the variance is estimated based on the J trained models. Theorem 2 shows that the variance estimated by bootstrap converges to the asymptotic variance, which is the uncertainty quantity. We train J neural networks, $\{f_{\theta^{(j)}}\}_{j=1}^J$, on a single probe set $\mathcal{S}'$ with different initialization and random seeds, which empirically outperforms the bootstrap method. With $\{f_{\theta^{(j)}}\}_{j=1}^J$, we then replace the linear models in Algorithm 1 by their DNN counterparts, i.e., replace $\hat{\boldsymbol{\beta}}_j^\top \mathbf{x}$ by $f_{\theta^{(j)}}(\mathbf{x})$, $\tilde{\boldsymbol{\beta}}^\top \mathbf{x}$ by $\frac{1}{J}\sum_{j=1}^J f_{\theta^{(j)}}(\mathbf{x})$, and $p(\tilde{\beta}; \mathbf{x})$ by $\frac{1}{J}\sum_{j=1}^J p(f_{\theta^{(j)}}; \mathbf{x})$. We summarize the uncertainty estimation for DNN in Algorithm 6 in Appendix C.1. Notably, our method can be further simplified by training a single model on $\mathcal{S}'$ with dropout and then can obtain $\{f_{\theta^{(j)}}\}_{j=1}^J$ by using Monte Carlo Dropout during inference. In Section 6, we also empirically compare different uncertainty estimation methods including different initialization, bootstrap, and dropout.

The detailed algorithm as summarized in Algorithm 4 and 5 in Appendix C.1.

4. Towards Effective Sampling Strategy in Real Word Applications

In this section, we enhance the theoretically motivated sampling algorithm by incorporating insights gained from empirical observations. To begin, we experiment with the optimal sampling strategy Algorithm 4 and 5 on real-world datasets with deep neural networks.

4.1 Vanilla uncertainty sampling strategy is ineffective in applications

Settings. We try out the sampling for DNN, i.e., Algorithm 4 and 5 (with uncertainty estimation in Algorithm 6) with ResNet20 He et al. (2016). We performed experiments on

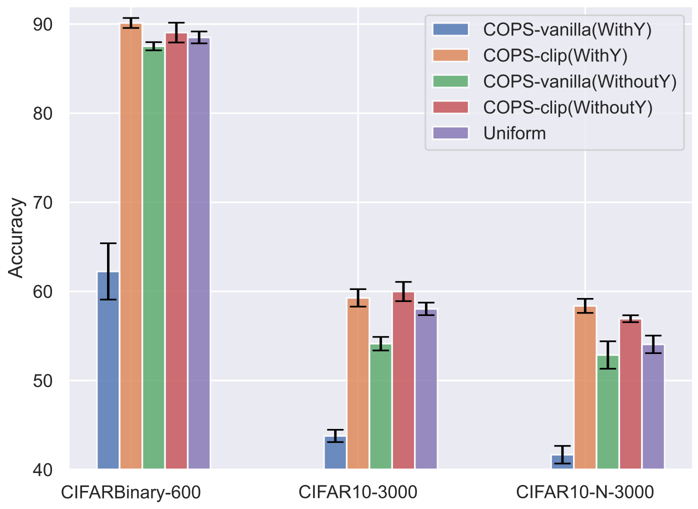
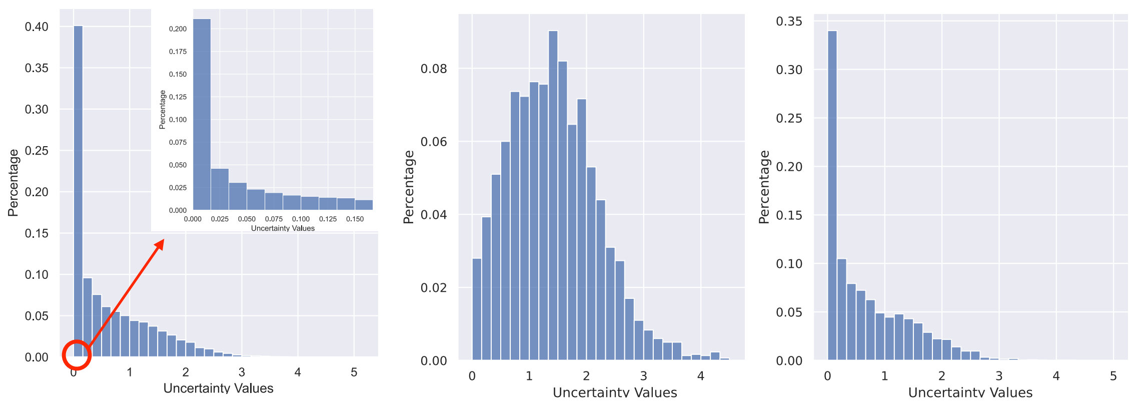


Figure 1: The vanilla implementation of the uncertainty Algorithm 4 and 5 (i.e., COPS-vanilla) displays inferior performance. Whereas, thresholding the maximum uncertainty during sample selection (i.e., COPS-clip) significantly enhances the overall performance.

three datasets: (1) CIFAR10 Krizhevsky et al. (2009), (2) CIFARBinary, and (3) CIFAR10-N Wei et al. (2021). CIFARBinary is a binary classification dataset created by selecting two classes (plane and car) from CIFAR10. CIFAR10-N is a variant of CIFAR10 with natural label noise Wei et al. (2021). For a more comprehensive description of the datasets, please refer to Section 6. For all settings, we split the training set into two subsets, i.e., the probe set ($\mathcal{S}'$ in Algorithm 4-5) and the sampling dataset set ($\mathcal{S}$ in Algorithm 4-5). We train 10 probe neural networks on $\mathcal{S}'$ and estimate the uncertainty of each sample in $\mathcal{S}$ with these networks. By sampling with replacement, we select an subset with 300 samples per class from $\mathcal{S}$ according to Algorithm 4-5, on which we train the a ResNet20 from scratch.

Since we conduct experiments on multiple datasets with different sub-sampling size, and for both sampling with labels and sampling without labels problems. We then use WithY to denote the sampling with labels and we use WithoutY for sampling without labels. We use the triple "(dataset name)-(target sub-sampling size)-(whether with Y)" to denote an experimental setting, for example: CIFAR10-3000-WithY is short for the setting to select 3,000 samples from labeled CIFAR10 dataset for sampling with labels.

Results. Surprisingly, the results in Figure 1 shows that the sampling Algorithm 4 and 5 are even inferior than uniform sampling in some settings both for sampling with and without labels. For example, in the CIFARBinary-600-WithY setting in Figure 1, uncertainty sampling leads to a testing performance of 63.2%, which is much worse than uniform sampling's performance 88.3%.



(a) The full dataset. (b) The 3000 samples selected by COPS-vanilla. (c) The 3000 samples selected by COPS-clip.

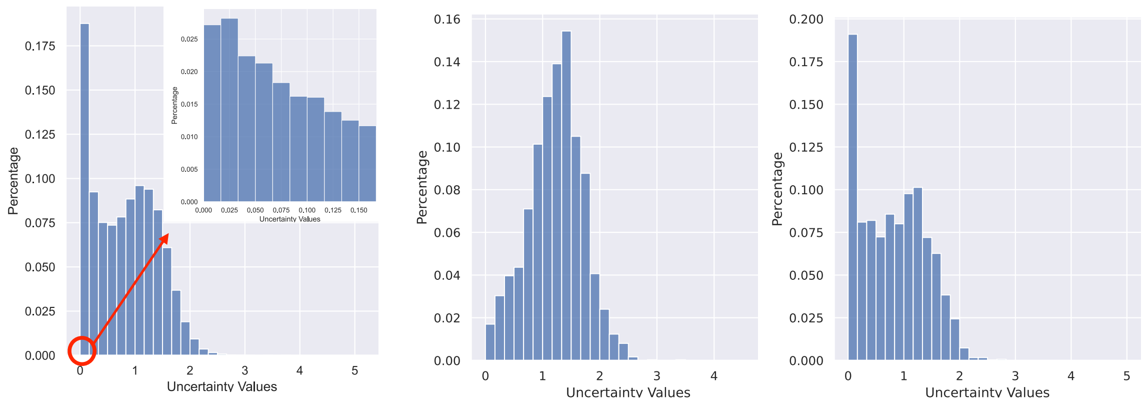
Figure 2: Histogram of estimated $u(\boldsymbol{x}, \boldsymbol{y})$ of samples on CIFAR10-3000-WithY.

A closer look at the Uncertainty sampling. Figure 2(a) visualizes the uncertainty distribution of samples in CIFAR10 estimated by Algorithm 6 . Figure 2(b) shows the uncertainty of the 3000 samples selected according to the sample selection ratio in Eqn (13), i.e., the uncertainty of 3000 samples selected by COPS in the CIFAR10-3000-WithY setting. The uncertainty distribution of the selected data in Figure 2(b) is quite different from the uncertainty distribution of the full dataset in Fig 2(a). The selected subset contains a large number of data with high uncertainty. Figure 3 shows similar trends in CIFAR10-3000-WithoutY.

Recall that the optimal sampling ratio is derived in a simplified setting where we assume that there is no model misspecification. The sampling schemes in Eqn. (13) and (14) tend to select samples from the low density region with high uncertainty. Whereas, previous studies He et al. (2022); Ye et al. (2023) demonstrate that the model estimation can be very sensitive to the model mis-specification at the low-density regions. We conjecture that the uncertainty sampling methods in Algorithms 4 and 5 suffer from this issue since they place unprecedented emphasis on the low density region (refer to Appendix E.1 for more results on this). We then illustrate this effect by a logistic linear classification example in the following section.

4.2 Simulating the effect of model mis-specification on sampling algorithms

Simulation with a linear example. The optimal sampling strategy Eqn. (13) and (14) is derived under the assumption that the model is well-specified, i.e., there exists an oracle β^* such that $\mathbb{P}(y = c_k|\mathbf{x}) = p_k(\beta^*; \mathbf{x})$ for all $\mathbf{x}$ and k. To illustrate how the uncertainty sampling can suffer from model misspecification, we conduct simulations on the following example which contains model misspecification following the setting of He et al. (2022); Ye et al. (2023); Bogunovic et al. (2021).



(a) On the full sampling dataset. (b) Subset selected by COPS-vanilla (c) Subset selected by COPS-clip

Figure 3: Histogram of estimated $u(\boldsymbol{x})$ on CIFAR10-3000-WithoutY.

$\boldsymbol{x}$	$\mathbf{x}_1 = [1,0]^\top$	$\mathbf{x}_2 = [0.1, 0.1]^\top$	$\mathbf{x}_3 = [0,1]^\top$
Sampling Set	$n_1 = 1,000$	$n_2 = 100,000$	$n_3 = 100,000$
Testing Set	$n_1 = 1,000$	$n_2 = 100,000$	$n_3 = 100,000$

Table 1: A simple example with 2-dimensional input $\boldsymbol{x} \in \mathbb{R}^2$ and binary output $y \in \{0,1\}$. There are three kinds of inputs as shown in the table. Both the training (sampling) and testing set contains 1,000 $\mathbf{x}_1$, 100,000 $\mathbf{x}_2$ and 100,000 $\mathbf{x}_3$, respectively.

Consider a binary classification problem $y \in \{0,1\}$ with 2-dimensional input $\boldsymbol{x} \in \mathbb{R}^2$. The true parameter $\beta^* = [2,2]^\top$. In this simulation, we consider adversarial corruption, a typical case of misspecification in a line of previous research He et al. (2022); Ye et al. (2023); Bogunovic et al. (2021). In this case, an adversary corrupts the classification responses $\boldsymbol{y}$ before they are revealed to the learners. Hence, if the learner still makes estimations via the linear logistic model, the misspecification occurs. Suppose that the there exists model misspecification characterized by $\zeta : \mathbb{R}^2 \to \mathbb{R}$ such that

$$P(y = 1 | \boldsymbol{x}, \beta^*, \zeta) = \frac{\exp\left(\boldsymbol{x}^\top \beta^* + \zeta(\boldsymbol{x})\right)}{1 + \exp\left(\boldsymbol{x}^\top \beta^* + \zeta(\boldsymbol{x})\right)}. \tag{21}$$

Consider a training dataset consisting of 1,000 instances of $\mathbf{x}_1$, 100,000 instances of $\mathbf{x}_2$, and 100,000 instances of $\mathbf{x}_3$, where $\mathbf{x}_1 = [1,0]$, $\mathbf{x}_2 = [0.1, 0.1]$, and $\mathbf{x}_3 = [0,1]$. It is evident that $\mathbf{x}_1$ falls within the low density region. In the following part, we will introduce non-zero corruption on $\mathbf{x}_1$. It is easy to infer that a corruption on $\mathbf{x}_1$ would induce estimation error on the first dimension of β. We incorporate $\mathbf{x}_2$ within the dataset to ensure that the estimation error on the first dimension would affect the estimation error on the second dimension.

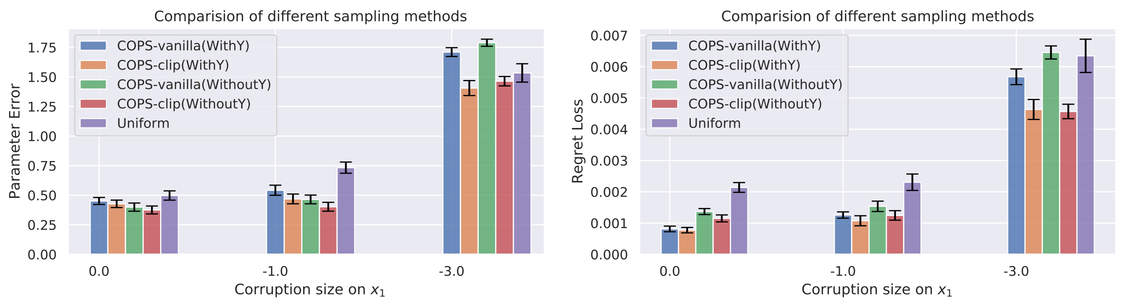


Figure 4: Comparison on of different sampling methods on simulation data. Left)the error of parameter estimation $|\bar{\beta} - \beta^*|$; Right) the regret loss $\mathcal{L}(\bar{\beta}) - \mathcal{L}(\beta^*)$ on the testing set.

We conduct simulations involving three cases of corruption in the low-density region $\mathbf{x}_1$: (a) $\zeta(\mathbf{x}_1) = 0$, (b) $\zeta(\mathbf{x}_1) = -1$, and (c) $\zeta(\mathbf{x}_1) = -3$. We select 1,000 samples from a total of 201,000 samples and obtain $\bar{\beta}$ by uniform sampling, COPS for sampling with labels (the linear Algorithm 2) and COPS for sampling without labels (the linear Algorithm 3). We also visualize parameter estimation error $|\bar{\beta} - \beta^*|$. We evaluate the regret loss $\mathcal{L}(\bar{\beta}) - \mathcal{L}(\beta^*)$ on the testing set without corruption as shown in Table 1. The results of the comparison for each method are presented in Figure 4. The simulation results demonstrate that the vanilla uncertainty sampling (i.e., COPS-vanilla) strategy performs better than uniform sampling when there is no corruption. However, as the level of corruption increases, the performance of uncertainty sampling deteriorates quickly and can be even worse than uniform sampling when $\zeta(\mathbf{x}_1) = -3$.

4.3 A simple fix

He et al. (2022); Ye et al. (2023) argues that the corruption in the low density region can make β_{MLE} deviates significantly from β^*. To alleviate this problem, He et al. (2022); Ye et al. (2023) propose to assign a smaller weight to the samples in low density regions when performing weighted linear regression, resulting in a solution closer to β^*. Specifically, they assign a weight $1/\max(\alpha, \|\mathbf{x}\|_{M_X^{-1}})$ to each sample to perform linear regression where α is a pre-defined hyper-parameter. For the samples with large uncertainty, they will have a small weight.

Recall that we select data according to the uncertainty $u(\mathbf{x}, \mathbf{y}) = |\delta_1(\mathbf{y}) - p_1(\hat{\boldsymbol{\beta}}_{\text{MLE}}; \mathbf{x})| \cdot \|\mathbf{x}\|_{M_X^{-1}}$. We can incorporate the idea of He et al. (2022); Ye et al. (2023) through modifying the uncertainty sampling ratio by multiplying $u(\mathbf{x}, \mathbf{y})$ with $1/\max(\alpha, \|\mathbf{x}\|_{M_X^{-1}})$, i.e., draw samples according to $u(\mathbf{x}, \mathbf{y})/\max(\alpha, \|\mathbf{x}\|_{M_X^{-1}})$. Furthermore, note that $u(\mathbf{x}, \mathbf{y})$ and $\|\mathbf{x}\|_{M_X^{-1}}$ only differ by an scaling term $|\delta_1(\mathbf{y}) - p_1(\hat{\boldsymbol{\beta}}_{\text{MLE}}; \mathbf{x})|$. So we use an even simpler version $u(\mathbf{x}, \mathbf{y})/\max(\alpha, u(\mathbf{x}, \mathbf{y})) \propto \min(\alpha, u(\mathbf{x}, \mathbf{y}))$, which turns out to simply threshold the maximum value of $u(\mathbf{x}, \mathbf{y})$ for sampling. Therefore, the overall sampling ratio for sampling with labels in Eqn (17) is modified as follows:

$$\pi^{\alpha}(\mathbf{x}, \mathbf{y}) = \frac{\min(\alpha, u(\mathbf{x}, \mathbf{y}))}{\sum_{(\mathbf{x}', \mathbf{y}') \in \mathcal{S}} \min(\alpha, u(\mathbf{x}', \mathbf{y}')\}}, \tag{22}$$

where $u(\mathbf{x}, \mathbf{y}) = |\delta_1(y) - p_1(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}; \mathbf{x}')| \cdot \|\mathbf{x}'\|_{M_X^{-1}}$. The full modified algorithm for sampling with labels is included in Algorithm 7 in Appendix C.2. The algorithm for sampling without labels selection is also modified accordingly as shown in Algorithm 8 in Appendix C.2. Notably, we don't modify the reweighting accordingly. Intuitively, original COPS select samples by u and the minimize the loss weighted by $1/u$. Here we select samples according to $\min\{\alpha, u\}$ but still use the original reweighting $1/u$. By this method, we can reduce the negative impact of model misspecification on the samples from the low density region i.e., samples with high uncertainty, obtaining a $\bar{\beta}$ closer to β^*.

We applied this method in the simulation experiment, testing the threshold at 3 or 10 times the minimum uncertainty. Take the threshold 3 for sampling with labels for example, we set $\alpha = 3 \cdot \min_{(\mathbf{x}, \mathbf{y}) \in \mathcal{S}} u(\mathbf{x}, \mathbf{y})$. To differentiate, we use the suffix 'COPS-clip' to represent the method with limited uncertainty from above. On the other hand, we refer to the unmodified COPS method as 'COPS-vanilla'. The outcomes displayed in Figure 4 demonstrate how this straightforward approach enhances the performance of uncertainty sampling in case of substantial corruption, achieving significant improvement over both uniform sampling and COPS-vanilla in terms of both $|\bar{\beta} - \beta^*|$ and $\mathcal{L}(\bar{\beta}) - \mathcal{L}(\bar{\beta}^*)$. The results in Figure 1 show that the 'COPS-clip' also works well in real world applications.

Figure 2(c) and Figure 3(c) illustrate the uncertainty distribution of the 3000 samples selected by COPS-clip in the CIFAR10-3000-WithY and CIFAR10-3000-WithoutY settings, respectively. We can see that compared to COPS-vanilla, COPS-clip selects samples whose uncertainty distribution is closer to the uncertainty distribution of the entire CIFAR10 dataset, with only a slight increase in samples exhibiting high uncertainty. In Appendix E.3, we provide additional results that COPS-vanilla selects a higher proportion of noisy data in CIFAR10-N compared to uniform sampling. However, COPS-clip does not exhibit an increase in the noisy ratio when compared to uniform sampling.

5. Discussion with existing works.

Difference between existing method on using multiple samples to estimate uncertainty. Previous work (Wang and Ma, 2021) also uses multiple samples to estimate uncertainty; however, our approach is fundamentally different. Additionally, the theoretical challenges we tackle are distinct from theirs. As we will show, these two methods result in different empirical performances. To illustrate this, consider the logistic regression problem, where we have a large annotated dataset containing both $\mathbf{x}$ and $\mathbf{y}$. For simplicity, we will omit the normalization term to make the annotation process easier. The optimal sampling ratio, determined by $|\delta_1(y) - p_1(\hat{\beta}_{\mathrm{MLE}}; \mathbf{x})| \|\mathbf{x}\|_{\mathbf{M}_X^{-1}}$, minimizes the expected testing loss of the model fitted on the sampled data $\bar{\mathcal{S}}$. Specifically, the expected testing loss is

Optimality	CIFAR10	CIFAR100	CIFAR10-N
A-optimality (Wang and Ma, 2021)	70.98	35.78	69.22
Ours	**72.70**	**36.84**	**70.75**

Table 2: Empirical comparison of COPS with A-optimality (Wang and Ma, 2021) sampling scheme.

$\min_\pi \mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}\left[\mathcal{L}(\bar{\beta};\mathcal{D})-\mathcal{L}(\beta^*;\mathcal{D})\right]$. Let us denote our criterion as Test-Loss-Optimality. In the existing literature Wang and Ma (2021); Wang et al. (2018), to minimize the asymptotic variance of the estimated variable $\bar{\beta}$ (referred to as A-optimality), we need to compute $\mathbf{M}_X^{-1}\mathbf{x}$ as discussed in Wang et al. (2018). Additionally, we must calculate $\mathbf{D}_N^{-1}\mathbf{x}$, where $\mathbf{D}_N$ represents the weighted design matrix for quantile regression [1]. To bypass the computational challenges in estimating $\mathbf{M}_X^{-1}$ and $\mathbf{D}_N^{-1}$, Wang and Ma (2021); Wang et al. (2018) propose modifying the objective to minimize the asymptotic variance of $\mathbf{M}_X\bar{\beta}$ (Wang et al., 2018) or $\mathbf{D}_N\bar{\beta}$ (Wang and Ma, 2021), which corresponds to a linearly transformed parameter (referred to as L-optimality in the literature). However, **there are key differences between our method and theirs.**

- The A-optimality and L-optimality discussed in [1, 2] do not imply Test-Loss-Optimality, and vice versa. Test-Loss-Optimality is a natural criterion, as a substantial body of literature focuses on reducing expected testing loss. As we will demonstrate later, a Test-Loss-Optimal sampling strategy can lead to better testing accuracy compared to methods developed using A-optimality.

- The optimal sampling ratio for Test-Loss-Optimality is computationally intractable in its vanilla version. The main theoretical contribution of our work is to demonstrate that Test-Loss-Optimality can be efficiently estimated based on the output logits of models, without changing the objective. Specifically, we are the first to show that we can use the variance of the outputs from multiple models to address the computational challenge of calculating $\mathbf{M}_X^{-1}$ for estimating the optimal sampling ratio, without changing the objective as done in Wang and Ma (2021); Wang et al. (2018).

Theoretically, we cannot definitively say which criterion—A-optimality, L-optimality, or Test-Loss-Optimality—is superior. Since our primary focus is on the performance of trained DNNs in classification tasks, we compare A-optimality and Test-Loss-Optimality in terms of empirical performance. We conduct experiments on CIFAR-10, CIFAR-100, and CIFAR-10-N for sampling with labels. For each dataset, we train a ResNet-20 using the 10,000 samples selected by our method and the A-optimality method (Wang and Ma, 2021), respectively. The results are presented in Table 2. Our findings indicate that our method consistently outperforms the A-optimality (Wang and Ma, 2021).

Difference with existing works on the sampling ratio threshold. A key contribution of our work is to highlight that the optimal sampling strategy derived in a well-specified setting can be adversely affected by model misspecification, particularly when it overly emphasizes low-density samples. This issue can be mitigated through the use of thresholding. We want to clarify that the threshold discussed in our paper is fundamentally different from that used in Poisson sampling, as described by Wang et al. (2022). While Wang et al. (2022) highlight a sampling ratio threshold for Poisson sampling, their study suggests that when sampling with replacement, a threshold is unnecessary. However, our findings and simulations in Section 4 demonstrate that the need of a sampling threshold for sampling with replacement due to model misspecification in applications. It is also important to note that the approach taken by Fithian and Hastie (2012), which involves thresholding for presence-only data modeling, is not directly relevant to the sub-sampling task we investigate in this paper. The challenges we address are fundamentally different. Our primary contribution lies in identifying the sensitivity of sub-sampling techniques to model misspecification and proposing a sampling ratio threshold to enhance the performance of the optimal sampling strategy. To the best of our knowledge, no existing work has recognized this issue within sub-sampling tasks or employed our method to address it.

Difference with existing works on re-weight threshold. To simplify the discussion, let u denote the $u(\mathbf{x}, \mathbf{y})$ for sampling with labels, and $u(\mathbf{x})$ for sampling without label. In the vanilla COPS method, two stages are performed: (Stage 1): Data subsampling according to u. (Stage 2): Weighted learning, where each selected sample is assigned a weight of $1/u$ to get an unbiased estimator. Since the sample weighting in Stage 2 involves calculating the inverse of u, it can result in high variance if u approaches zero. To address this, previous work has implemented a threshold of $1/\max\{\beta, u\}$ Ionides (2008); Swaminathan and Joachims (2015); Citovsky et al. (2023), which limits the minimum value of u. Both COPS-vanilla and COPS-clip adopt this strategy by default in the second stage to limit the variance and β is set to 0.1 for all real-world dataset experiments (including the experiments in Figure 1). Appendix D.1 shows the full details on this part. However, our empirical analysis reveals the importance of also limiting the maximum of u by $\min\{\alpha, u\}$ in the first stage, which can alleviate the negative impact of potential model misspecification on COPS. To the best of our knowledge, this hasn't been discussed in existing works Wang et al. (2018); Ting and Brochu (2018); Yao and Wang (2019); Swaminathan and Joachims (2015); Citovsky et al. (2023). Appendix E.4 presents empirical results to compare the impact of threshold on the first and second stages.

6. Experiments and results

Settings. In this section, we conduct extensive experiments to verify COPS. Here the COPS method refers to COPS-clip in Section 4 by default and the detailed algorithms are in Algorithm 9-10. We compare COPS with various baseline methods, validate COPS on various datasets including both CV and NLP task and also datasets with natural label noise. For all the methods studied in this section, we use the same setting as described in

Section 4 that we train probe networks on one probe dataset and performing sampling at once on the sampling dataset. The datasets used in our experiments are as follows:

- CIFAR10 Krizhevsky et al. (2009): We utilize the original CIFAR10 dataset Krizhevsky et al. (2009). To construct the probe set, we randomly select 1000 samples from each class, while the remaining training samples are used for the sampling set. For our experiments, we employ ResNet20, ResNet56 He et al. (2016), MobileNetV2 Sandler et al. (2018), and DenseNet121 Huang et al. (2017) as our backbone models.
- CIFARBinary: We choose two classes, plane and car, from the CIFAR10 dataset for binary classification. Similar to CIFAR10, we assign 1000 samples from the training images for the probe set of each class, and the remaining training samples form the sampling set. In this case, we employ ResNet20 as our backbone model.
- CIFAR100: From the CIFAR100 dataset Krizhevsky et al. (2009), we randomly select 200 samples for each class and assign them to the probe set. The remaining training samples are used in the sampling set. For this dataset, ResNet20 is utilized as the backbone model.
- CIFAR10-N: We use CIFAR10-N, a corrupted version of CIFAR10 introduced by Wei et al. Wei et al. (2021). The training set of CIFAR10-N contains human-annotated real-world noisy labels collected from Amazon Mechanical Turk and the testing set of CIFAR10-N is the same with CIFAR10. Similar to CIFAR10, we split 1000 samples from each class for the probe set, while the rest are included in the sampling set. We employ ResNet20 as our backbone model.
- IMDB: The IMDB dataset Maas et al. (2011) consists of positive and negative movie comments, comprising 25000 training samples and 25000 test samples. We split 5000 samples from the training set for uncertainty estimation and conduct our scheme on the remaining 20000 samples. For this dataset, we use a GRU-based structure Cho et al. (2014), and further details can be found in Appendix D.3.
- SVHN: The SVHN dataset contains images of house numbers. We split 1000 samples from the train set for each class to estimate uncertainty, while the remaining train samples are used for the sampling schemes. ResNet20 serves as our backbone model in this case.
- Place365 (subset): We select ten classes from the Place365 dataset Zhou et al. (2017), each consisting of 5000 training samples and 100 testing samples. The chosen classes are department_store, lighthouse, discotheque, museum-indoor, rock_arch, tower, hunting_lodge-outdoor, hayfield, arena-rodeo, and movie_theater-indoor. We split the training set, assigning 1000 instances for each class to the probe set, and the remaining samples form the sampling set. ResNet18 is employed as the backbone model for this dataset.

We summarize the datasets in Table 3:

Dataset	Class Number	Probe Set	Sampling Set	Target Size of Sub-sampling	Test Set
CIFARBinary	2	2,000	8,000	600/2,000/6,000	2,000
CIFAR10	10	10,000	40,000	3,000/10,000/20,000	10,000
CIFAR10-N	10	10,000	40,000	3,000/10,000/20,000	10,000
CIFAR100	100	20,000	30,000	3,000/10,000/20,000	10,000
SVHN	10	10,000	63,257	3,000/10,000/20,000	26,032
Places365	10	10,000	40,000	3,000/10,000/20,000	1,000
IMDB	2	5,000	20,000	2,000/4,000/10,000	25,000

Table 3: The table provides descriptions of the datasets used in our study. The "Probe Set/ Sampling Set" column indicates the number of samples included in the Probe Set and Sampling Set for each dataset. The "Target Size of Sub-sampling" column represents the number of samples selected from the Sampling Set for sub-sampling. For example, if the value is shown as "600", it indicates that we choose 600 instances from the Sampling Set for sub-sampling.

Comparison with Baselines. In this part, we compare our method COPS with existing sample selection methods. We adopts sampling with replacement for our COPS and baselines. We adopt competitive baselines for sampling with and without labels, respectively. Notably, the sampling ratios presented below are normalized to form a sampling distribution. For clarity, we omit the normalization term to simplify the notation. The baselines for sampling with labels are as follows:

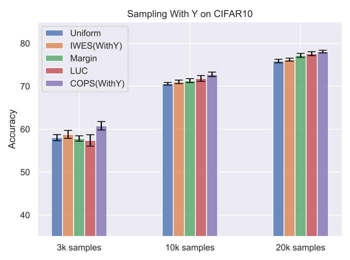
- **Uniform sampling**.
- **IWeS(WithY)**. Citovsky et al. (2023) first fit two functions $f_{\theta^{(1)}}$ and $f_{\theta^{(2)}}$ on the probe set and then use the disagreement of the two functions with respect to entropy to calculate the sampling ratio for a sample $(\mathbf{x}, \mathbf{y})$:

$$\pi(\mathbf{x}, \mathbf{y}) = \sum_{k=0}^{K} \delta_k(\mathbf{y}) \big| p_k(f_{\theta^{(1)}}; \mathbf{x}) \log_2(p_k(f_{\theta^{(1)}}; \mathbf{x})) - p_k(f_{\theta^{(2)}}; \mathbf{x}) \log_2(p_k(f_{\theta^{(2)}}; \mathbf{x})) \big| \quad (23)$$

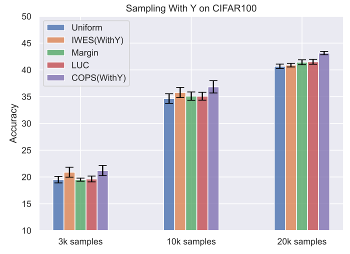
- **LUC**. Han et al. (2020) proposes to assign probabilities to each point and sample subset according to the assigned probabilities. They first obtain a roughly estimated probability $\tilde{p}_i$ for each sample by fitting a pilot model with a smaller independent data set. Then the final sampling probabilities $a(x_i, y_i)$ is determined as follows,

$$a(x_i, y_i) = \begin{cases} \frac{1-\tilde{q}_i}{\gamma - \max(\tilde{q}_i, 0.5\gamma)}, & \text{if } \tilde{p}_{i,y_i} = \tilde{q}_i \geq 0.5 \\ \min(1, 2\tilde{q}_i/\gamma), & \text{otherwise} \end{cases}, \quad (24)$$

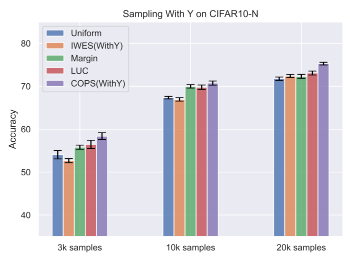
Here $\tilde{q}_i = max(0.5, \tilde{p}_{i,1}, \cdots, \tilde{p}_{i,K})$ and γ is a hyperparameter.



(a) CIFAR10.

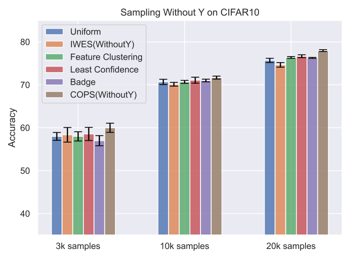


(c)CIFAR100

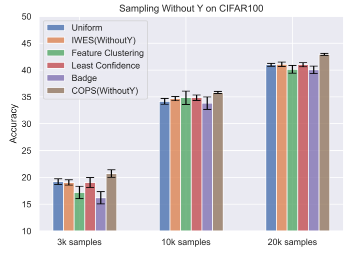


(b)CIFAR10-N

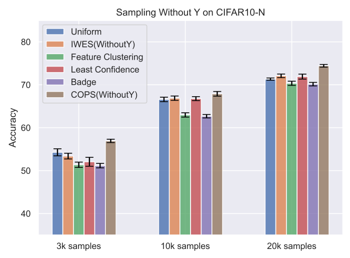
Figure 5: Results for sampling with labels. For Badge with 3000 samples, the performance is lower than 50, so the bar is clipped in our figures.



(a) CIFAR10.



(b)CIFAR100



(c) CIFAR10-N

Figure 6: Results for sampling without labels (WithoutY). For Badge with 3000 samples, the performance is lower than 50, so the bar is clipped in our figures.

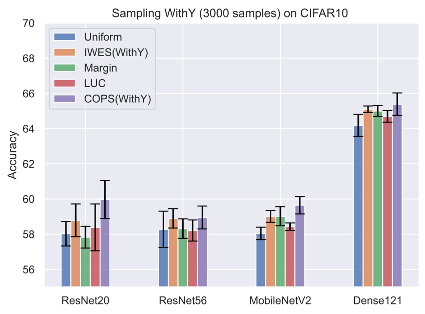
- **BADGE(WithY)**. Ash et al. (2019) calculates the gradient of the last layer and use kmeans++ to cluster the gradient. They then select the samples closest to cluster centers.
- **Margin** Scheffer et al. (2001). The margin is computed by subtracting the predicted probability of the true class from 1.

$$\pi(\mathbf{x}, \mathbf{y}) = 1 - \sum_{k=1}^{K} \delta_k(\mathbf{y}) p_k(f_\theta; \mathbf{x}) \tag{25}$$

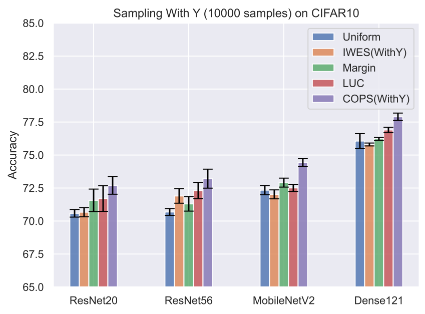
The baselines for sampling without labels are as follows:

- **Uniform sampling**.
- **IWeS (WithoutY)** Citovsky et al. (2023) uses a normalized version of entropy is as follows,

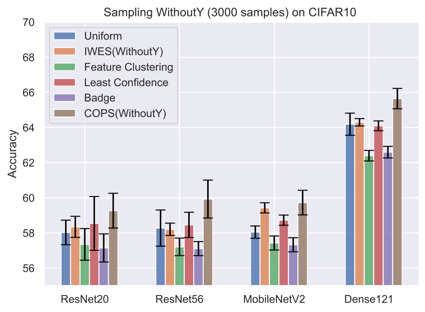
$$\pi(\mathbf{x}) = -\sum_{k=0}^{K} p_k(f_\theta; \mathbf{x}) \log_2(p_k(f_\theta; \mathbf{x})) / \log_2(K) \tag{26}$$



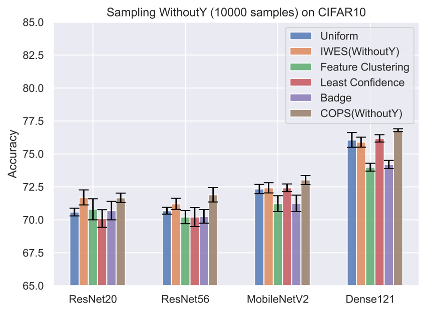
(a) CIFAR10-3000-WithY



(b)CIFAR10-10000-WithY



(c) CIFAR10-3000-WithoutY



(d)CIFAR10-10000-WithoutY

Figure 7: Results for Cifar10 with different architectures.

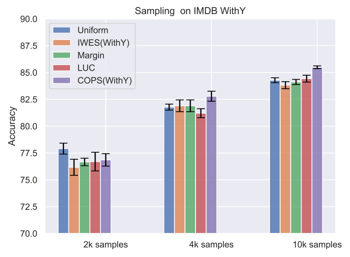
- **BADGE (WithoutY)**Ash et al. (2019) first obtain the pseudo label $\hat{\mathbf{y}} = \arg\max_k p_k(f_\theta; \mathbf{x})$ and the calculates the gradient of the last layer with the pseudo label $\hat{\mathbf{y}}$. Then they use K-means++ to cluster samples and select the samples closest to cluster centers.
- **Least confidence** The paper Scheffer et al. (2001) is determined by calculating the difference between 1 and the highest probability assigned to a class:

$$\pi(\mathbf{x}) = 1 - \max_k p_k(f_\theta; \mathbf{x}) \tag{27}$$

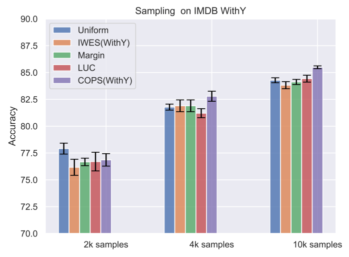
- **Feature Clustering** Sener and Savarese (2017)[2] first latent feature of the model and then uses K-means cluster the samples by its feature. They further select the samples closest to cluster.

We first compare COPS with the above baselines on both sampling with and without labels on three datasets, CIFAR10, CIFAR100 and CIFAR10-N. Since some deep learning empirical methods, such as IWeS and Badge, were initially validated only on sampling without

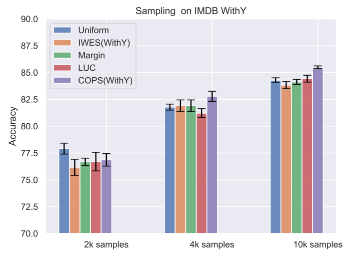
2. Sener and Savarese (2017) named their method as coreset, whereas, we refer to their method as feature clustering in order to avoid confusion with the coreset task.



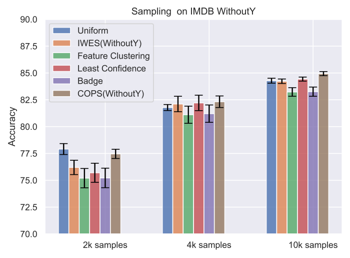
(a)IMDB.



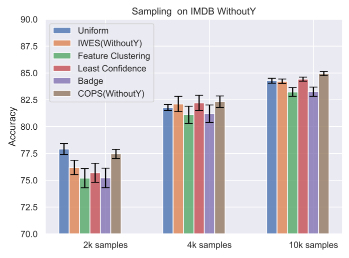
(b) Places365.



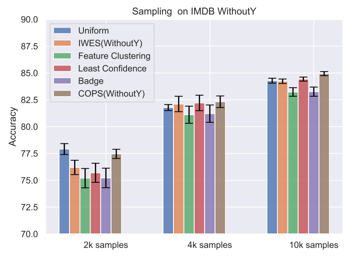
(c) SVHN.



(a)IMDB.

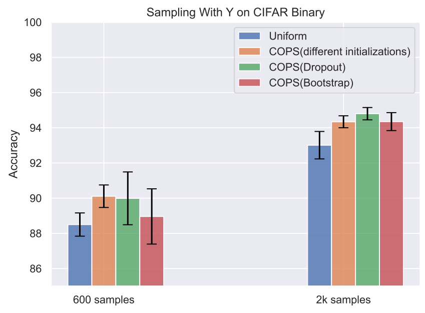


(b) Places365.

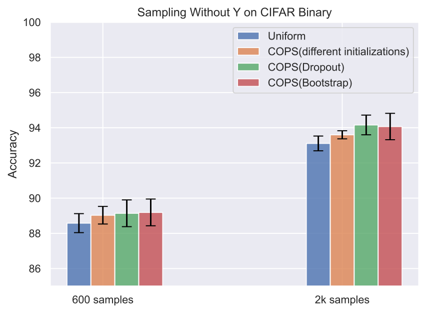


(c) SVHN.

Figure 8: Results of COPS on IMDB, PLACE365 and SVHN.



(a) WithY.



(b)WithoutY.

Figure 9: Results for different kinds of uncertainty on CIFAR10-8,000 .

replacement, we also tested COPS under the same conditions in Appendix E.2. Although COPS is not specifically designed for sampling without replacement, our naive application of COPS still achieves state-of-the-art performance, as demonstrated in Appendix E.2.

The results in Figure 5 and 6 show that COPS can consistently outperform the baselines in these settings. The improvement is even more significant on CIFAR10-N, which contains nature label noise.

Multiple Architectures. To verify the effectiveness of COPS, we conduct experiments on CIFAR10 with different neural network structures. Specifically, we choose several widely-used structures, including ResNet56 He et al. (2016), MobileNetV2 Sandler et al. (2018) and DenseNet121 Huang et al. (2017). The results are shown in Figure. 7. Our method COPS can stably improve over random sampling for both WithY and WithoutY on different DNN architectures.

Additional Datasets. Furthermore, we evaluate the effectiveness of COPS on three additional datasets: SVHN, Places365 (subset), and IMDB (an NLP dataset). The results in Fig. 8 consistently demonstrate that our method consistently outperforms random sampling on these datasets.

Different methods for uncertainty estimation. In Algorithm 6, we obtain J models $\{f_{\theta^{(j)}}\}_{j=1}^J$ on the probe dataset $\mathcal{S}'$ by training DNNs independently with different initializations and random seeds. This method is referred to as the **different initialization** method. In this section, we compare this method with two alternative approaches to obtain $\{f_{\theta^{(j)}}\}_{j=1}^J$ given $\mathcal{S}'$:

(a) **Bootstrap**: Each $f_{\theta^{(j)}}$ is obtained by training a DNN on a randomly drawn subset from $\mathcal{S}'$.

(b) **Dropout** Gal and Ghahramani (2016): A single DNN is trained on $\mathcal{S}'$ with dropout. Then, $\{f_{\theta^{(j)}}\}_{j=1}^J$ are obtained by performing Monte Carlo Dropout during inference for J iterations.

The comparison of these three methods on CIFAR10 is depicted in Figure 9. It is evident that the different initialization method achieves the best performance, while the bootstrap method performs the worst among the three. The dropout method shows similar performance to the different initialization method for sampling with labels.

7. Conclusion

This study presents the COPS method, which offers a theoretically optimal solution for sampling with and without labels in linear softmax regression. By leveraging the output of the models, the sampling ratio of COPS can be effectively estimated even in deep learning contexts. To address the challenge of model sensitivity to misspecification, we introduce a downweighting approach for low-density samples. By incorporating this strategy, we modify the sampling ratio of COPS through thresholding the sampling ratio. Empirical experiments conducted on benchmark datasets, utilizing deep neural networks, further demonstrate the effectiveness of COPS in comparison to baseline methods. The results highlight the superiority of COPS in achieving optimal subsampling and performance improvement.

References

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

Ilija Bogunovic, Arpan Losalka, Andreas Krause, and Jonathan Scarlett. Stochastic linear bandits robust to adversarial attacks. In *International Conference on Artificial Intelligence and Statistics*, pages 991–999. PMLR, 2021.

Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in neural information processing systems*, 33: 14879–14890, 2020.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1): 1–122, 2012.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

Gui Citovsky, Giulia DeSalvo, Sanjiv Kumar, Srikumar Ramalingam, Afshin Rostamizadeh, and Yunjuan Wang. Leveraging importance weights in subset selection. *arXiv preprint arXiv:2301.12052*, 2023.

Kenneth L Clarkson, Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, Xiangrui Meng, and David P Woodruff. The fast cauchy transform and faster robust linear regression. *SIAM Journal on Computing*, 45(3):763–810, 2016.

Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *AAAI*, volume 5, pages 746–751, 2005.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. Regret bounds for robust adaptive control of the linear quadratic regulator. *Advances in Neural Information Processing Systems*, 31, 2018.

Petros Drineas, Michael W Mahoney, Shan Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. *Numerische mathematik*, 117(2):219–249, 2011.

Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. *The Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 569–578, 2011.

William Fithian and Trevor Hastie. Finite-sample equivalence in statistical models for presence-only data. *The annals of applied statistics*, 7(4):1917, 2012.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

Lei Han, Kean Ming Tan, Ting Yang, and Tong Zhang. Local uncertainty sampling for large-scale multiclass logistic regression. 2020.

Sariel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pages 126–134, 2005.

Jiafan He, Dongruo Zhou, Tong Zhang, and Quanquan Gu. Nearly optimal algorithms for linear contextual bandits with adversarial corruptions. *arXiv preprint arXiv:2205.06811*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

Jonathan Huggins, Trevor Campbell, and Tamara Broderick. Coresets for scalable bayesian logistic regression. *Advances in neural information processing systems*, 29, 2016.

Henrik Imberg, Johan Jonasson, and Marina Axelson-Fisk. Optimal sampling in unbiased active learning. In *International Conference on Artificial Intelligence and Statistics*, pages 559–569. PMLR, 2020.

Edward L Ionides. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311, 2008.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: convergence and generalization in neural networks (invited paper). *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 2018.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.

Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *2009 ieee conference on computer vision and pattern recognition*, pages 2372–2379. IEEE, 2009.

Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pages 5464–5474. PMLR, 2021.

Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. Task-aware variational adversarial active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8166–8175, 2021.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

Mario Lucic, Matthew Faulkner, Andreas Krause, and Dan Feldman. Training gaussian mixture models at scale via coresets. *The Journal of Machine Learning Research*, 18(1): 5885–5909, 2017.

Ping Ma, Michael Mahoney, and Bin Yu. A statistical perspective on algorithmic leveraging. In *International conference on machine learning*, pages 91–99. PMLR, 2014.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P11-1015`.

Xiangrui Meng, Michael A Saunders, and Michael W Mahoney. Lsrn: A parallel iterative solver for strongly over-or underdetermined systems. *SIAM Journal on Scientific Computing*, 36(2):C95–C118, 2014.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR, 2020.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021.

Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *Machine Learning: ECML 2006: 17th European Conference on Machine Learning Berlin, Germany, September 18-22, 2006 Proceedings 17*, pages 413–424. Springer, 2006.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction. In *International symposium on intelligent data analysis*, pages 309–318. Springer, 2001.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.

Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*, pages 814–823. PMLR, 2015.

Daniel Ting and Eric Brochu. Optimal subsampling with influence functions. *Advances in neural information processing systems*, 31, 2018.

Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12:389–434, 2012.

Ivor W Tsang, James T Kwok, Pak-Ming Cheung, and Nello Cristianini. Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research*, 6(4), 2005.

Dan Wang and Yi Shang. A new active labeling method for deep learning. In *2014 International joint conference on neural networks (IJCNN)*, pages 112–119. IEEE, 2014.

HaiYing Wang. More efficient estimation for logistic regression with optimal subsamples. *Journal of machine learning research*, 20(132):1–59, 2019.

Haiying Wang and Yanyuan Ma. Optimal subsampling for quantile regression in big data. *Biometrika*, 108(1):99–112, 2021.

HaiYing Wang, Rong Zhu, and Ping Ma. Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association*, 113(522):829–844, 2018.

Jing Wang, Jiahui Zou, and HaiYing Wang. Sampling with replacement vs poisson sampling: a comparative study in optimal subsampling. *IEEE Transactions on Information Theory*, 68(10):6605–6630, 2022.

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. *arXiv preprint arXiv:2110.12088*, 2021.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International conference on machine learning*, pages 1954–1963. PMLR, 2015.

Yaqiong Yao and HaiYing Wang. Optimal subsampling for softmax regression. *Statistical Papers*, 60:585–599, 2019.

Chenlu Ye, Wei Xiong, Quanquan Gu, and Tong Zhang. Corruption-robust algorithms with uncertainty weighting for nonlinear contextual bandits and markov decision processes. In *International Conference on Machine Learning*, pages 39834–39863. PMLR, 2023.

Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 93–102, 2019.

Tong Zhang. *Mathematical analysis of machine learning algorithms.* Cambridge University Press, 2023.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Xiao Zhou, Yong Lin, Renjie Pi, Weizhong Zhang, Renzhe Xu, Peng Cui, and Tong Zhang. Model agnostic sample reweighting for out-of-distribution learning. In *International Conference on Machine Learning*, pages 27203–27221. PMLR, 2022a.

Xiao Zhou, Renjie Pi, Weizhong Zhang, Yong Lin, Zonghao Chen, and Tong Zhang. Probabilistic bilevel coreset selection. In *International Conference on Machine Learning*, pages 27287–27302. PMLR, 2022b.

Appendix A. Proofs of main results

A.1 Proof of Theorem 1

Proof. **Part 1**. We first derive the optimal sampling ratio for **sampling with labels** problem. By Lemma 1, we have

$$\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}[\mathcal{L}(\bar{\beta};\mathcal{S})-\mathcal{L}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})]=\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}\left[(\bar{\beta}-\hat{\beta}_{\mathrm{MLE}})^{\top}\mathbf{M}_X(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})(\bar{\beta}-\hat{\beta}_{\mathrm{MLE}})+R(\bar{\beta})\right],$$

where the residual term is

$$R(\bar{\beta}) = O_{P|\mathcal{S}}(\|\bar{\beta} - \hat{\beta}_{\mathrm{MLE}}\|_2^2) = O_{P|\mathcal{S}}(r^{-1}),$$

the last inequality holds since according to Lemma 1 of Wang et al. (2018).

Therefore, we only need to find the sampling scheme which minimizes the following:

$$\begin{aligned}
&\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}\left[(\bar{\beta} - \hat{\beta}_{\mathrm{MLE}})^\top \mathbf{M}_X(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})(\bar{\beta} - \hat{\beta}_{\mathrm{MLE}})\right] \\
=&\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}\left[\mathrm{Tr}\left(\mathbf{M}_X(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})(\bar{\beta} - \hat{\beta}_{\mathrm{MLE}})(\bar{\beta} - \hat{\beta}_{\mathrm{MLE}})^\top\right)\right] \\
=&\mathrm{Tr}\left(\mathbf{M}_X(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}\left[(\bar{\beta} - \hat{\beta}_{\mathrm{MLE}})(\bar{\beta} - \hat{\beta}_{\mathrm{MLE}})^\top\right]\right).
\end{aligned}$$

By invoking Lemma 2, we further have as $n, r \to \infty$ in probability

$$\begin{aligned}
&\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}\left[(\bar{\beta} - \hat{\beta}_{\mathrm{MLE}})^\top \mathbf{M}_X(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})(\bar{\beta} - \hat{\beta}_{\mathrm{MLE}})\right] \\
\to&\mathrm{Tr}\left(\mathbf{M}_X(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\mathbf{M}_X^{-1}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\mathbf{V}_c(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\mathbf{M}_X^{-1}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\right) \\
=&\frac{1}{rn^2}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}}\frac{1}{\pi(\mathbf{x},\mathbf{y})}\mathrm{Tr}\left(\psi(\hat{\beta}_{\mathrm{MLE}};\mathbf{x},\mathbf{y})\otimes(\mathbf{x}\mathbf{x}^\top)\mathbf{M}_X^{-1}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\right),
\end{aligned}$$

where the last equality is due to the definition of $\mathbf{V}_c(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})$. Since $\sum_{(\mathbf{x},\mathbf{y})}\pi(\mathbf{x},\mathbf{y}) = 1$, we have

$$\begin{aligned}
&\frac{1}{rn^2}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}}\pi(\mathbf{x},\mathbf{y})\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}}\frac{1}{\pi(\mathbf{x},\mathbf{y})}\mathrm{Tr}\left(\psi(\hat{\beta}_{\mathrm{MLE}};\mathbf{x},\mathbf{y})\otimes(\mathbf{x}\mathbf{x}^\top)\mathbf{M}_X^{-1}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\right) \\
\geq&\frac{1}{rn^2}\left(\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}}\sqrt{\mathrm{Tr}\left(\psi(\hat{\beta}_{\mathrm{MLE}};\mathbf{x},\mathbf{y})\otimes(\mathbf{x}\mathbf{x}^\top)\mathbf{M}_X^{-1}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\right)}\right)^2,
\end{aligned}$$

where the inequality is due to the Cauchy-Schwarz inequality and the equality holds when

$$\pi(\mathbf{x},\mathbf{y}) \propto \sqrt{\mathrm{Tr}\left(\psi(\hat{\beta}_{\mathrm{MLE}};\mathbf{x},\mathbf{y})\otimes(\mathbf{x}\mathbf{x}^\top)\mathbf{M}_X^{-1}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\right)}.$$

Part 2. We then derive the optimal sampling ratio for **sampling without labels** problem. We first note that

$$\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S}_X,\pi}\left[\mathcal{L}(\bar{\beta};\mathcal{S}) - \mathcal{L}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\right] = \mathbb{E}_{\mathcal{S}|\mathcal{S}_X,\pi}\left[\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}\left[\mathcal{L}(\bar{\beta};\mathcal{S}) - \mathcal{L}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\right]\right],$$

By lemma 1, we only need to find π which minimizes the following equation

$$\begin{aligned}
&\mathbb{E}_{\mathcal{S}|\mathcal{S}_X,\pi}\left[\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}\left[(\bar{\beta} - \hat{\beta}_{\mathrm{MLE}})^\top \mathbf{M}_X(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})(\bar{\beta} - \hat{\beta}_{\mathrm{MLE}})\right]\right] \\
=&\mathbb{E}_{\mathcal{S}|\mathcal{S}_X,\pi}\left[\frac{1}{rn^2}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}}\frac{1}{\pi(\mathbf{x})}\mathrm{Tr}\left(\psi(\hat{\beta}_{\mathrm{MLE}};\mathbf{x},\mathbf{y})\otimes(\mathbf{x}\mathbf{x}^\top)\mathbf{M}_X^{-1}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\right)\right] \\
\to&\frac{1}{rn^2}\sum_{(\mathbf{x},y)\in\mathcal{S}}\frac{1}{\pi(\mathbf{x})}\mathrm{Tr}\left(\mathbb{E}_{\mathcal{S}|\mathcal{S}_X,\pi}[\psi(\hat{\beta}_{\mathrm{MLE}};\mathbf{x},\mathbf{y})]\otimes(\mathbf{x}\mathbf{x}^\top)\mathbf{M}_X^{-1}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\right) \\
=&\frac{1}{rn^2}\sum_{(\mathbf{x},y)\in\mathcal{S}}\frac{1}{\pi(\mathbf{x})}\mathrm{Tr}\left(\phi(\hat{\beta}_{\mathrm{MLE}};\mathbf{x})\otimes(\mathbf{x}\mathbf{x}^\top)\mathbf{M}_X^{-1}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})\right),
\end{aligned}$$

in probability, where the last equality is due to $\mathbb{E}_{\boldsymbol{y}}[\psi(\hat{\beta}_{\text{MLE}};\boldsymbol{x},\boldsymbol{y})|\boldsymbol{x}] = \phi(\hat{\beta}_{\text{MLE}};\boldsymbol{x})$. Similar to the derivation for sampling with labels, the optimal sampling ratio for sampling without labels is

$$\pi(\mathbf{x}) \propto \sqrt{\operatorname{Tr}\left(\phi(\hat{\beta}_{\text{MLE}};\mathbf{x}) \otimes (\mathbf{x}\mathbf{x}^\top)\mathbf{M}_X^{-1}(\hat{\beta}_{\text{MLE}};\mathcal{S})\right)}.$$

We thus finish the proof. □

A.2 Proof of Theorem 2

Proof of Theorem 2. Revision: For a matrix A, let $A_{i,j}$ denote its (i,j)-th element. For a fixed number of subsets J, the term $\operatorname{Tr}\left(\psi(\tilde{\beta};\mathbf{x},y)\Sigma_J(\mathbf{x})\right)$ can be written as

$$\begin{aligned}\operatorname{Tr}\left(\psi(\tilde{\beta};\mathbf{x},y)\Sigma_J(\mathbf{x})\right) &= \sum_{i,j=1}^{K} (\psi(\tilde{\beta};\mathbf{x},y))_{i,j}(\Sigma_J(\mathbf{x}))_{i,j} \\ &= \sum_{i,j=1}^{K} (\psi(\tilde{\beta};\mathbf{x},y))_{i,j} \cdot \frac{1}{J-1}\sum_{j=1}^{J}\big(\hat{\beta}_i^{(j)} - \tilde{\beta}_i\big)^\top \mathbf{x}\mathbf{x}^\top \big(\hat{\beta}_j^{(j)} - \tilde{\beta}_j\big).\end{aligned}$$

For each $(\Sigma_J(\mathbf{x}))_{i,j}$ for $i,j \in [K]$, we can decompose it as

$$\begin{aligned}&(\Sigma_J(\mathbf{x}))_{i,j} \\ =&\frac{1}{J-1}\sum_{j=1}^{J}\left(\left(\hat{\beta}_i^{(j)} - \hat{\beta}_{\text{MLE},j}\right)^\top \mathbf{x} - \left(\tilde{\beta}_i - \hat{\beta}_{\text{MLE},i}\right)^\top \mathbf{x}\right)\left(\left(\hat{\beta}_j^{(j)} - \hat{\beta}_{\text{MLE},j}\right)^\top \mathbf{x} - \left(\tilde{\beta} - \hat{\beta}_{\text{MLE}}\right)^\top \mathbf{x}\right)^\top \\ =&\frac{J}{J-1}\underbrace{\frac{1}{J}\sum_{j=1}^{J}\left(\hat{\beta}_i^{(j)} - \hat{\beta}_{\text{MLE},i}\right)^\top \mathbf{x}\mathbf{x}^\top\left(\hat{\beta}_j^{(j)} - \hat{\beta}_{\text{MLE},j}\right)}_{q_1} - \frac{J}{J-1}\underbrace{\left(\tilde{\beta}_i^\top - \hat{\beta}_{\text{MLE},i}\right)^\top \mathbf{x}\mathbf{x}^\top\left(\tilde{\beta}_j - \hat{\beta}_{\text{MLE},j}\right)}_{q_2}\end{aligned} \tag{28}$$

Let $n = |\mathcal{S}|$, $n' = |\mathcal{S}^{(j)}|$, and the covariance matrix of the whole dataset be $\Sigma_{\mathcal{S}}$. For each estimator $\hat{\beta}^{(j)}$, we know that $\mathbb{E}[\hat{\boldsymbol{\beta}}^{(j)} - \hat{\boldsymbol{\beta}}_{\text{MLE}} \mid \mathcal{S}] = 0$. By invoking Theorem 1 from Wang et al. (2018), we have given $\delta_0 = \delta/J$, there exists a finite $\Delta_{\delta,J,1}$ and $n_{\delta,J,1}$ such that for all $n' > n_{\delta,J,1}$

$$\mathbb{P}\left(\|\hat{\beta}^{(j)} - \hat{\beta}_{\text{MLE}}\|_2 \geq (n')^{-1/2}\Delta_{\delta_0} \,|\, \mathcal{S}\right) < \delta/J. \tag{29}$$

Define event that $\mathcal{E}_0 = \{\|\hat{\beta}^{(j)} - \hat{\beta}_{\text{MLE}}\|_2 \leq (n')^{-1/2}\Delta_{\delta_0},\ \forall\, j \in [J]\}$.

Besides, for each estimator $\hat{\beta}^{(j)}$, by invoking Theorem 2 from Wang et al. (2018), we have as $n \to \infty$ and $n' \to \infty$, conditional on $\mathcal{S}$ in probability

$$\sqrt{n'}\mathbf{M}_X^{-1/2}(\hat{\beta}_{\text{MLE}};\mathcal{S})(\hat{\beta}^{(j)} - \hat{\beta}_{\text{MLE}}) \to N(0,I). \tag{30}$$

This further implies that given $\delta_0 = \delta/J$, there exists a small $\epsilon_{\delta,J,2}$ and $n_{\delta,J,2}$ such that for all $n' > n_{\delta,J,2}$

$$\mathbb{P}\Big(\big|\mathbb{E}(\hat{\beta}^{(j)} - \hat{\beta}_{\text{MLE}})^\top \mathbf{x}\mathbf{x}^\top (\hat{\beta}^{(j)} - \hat{\beta}_{\text{MLE}}) - \frac{1}{n'}\mathbf{M}_X(\hat{\beta}_{\text{MLE}}; \mathcal{S})\big| \geq \epsilon_{\delta,2}\Big) \leq \delta/J. \tag{31}$$

We also define the event $\mathcal{E}_1 = \{\big|(\hat{\beta}^{(j)} - \hat{\beta}_{\text{MLE}})^\top \mathbf{x}\mathbf{x}^\top (\hat{\beta}^{(j)} - \hat{\beta}_{\text{MLE}}) - \frac{1}{n'}\mathbf{M}_X(\hat{\beta}_{\text{MLE}}; \mathcal{S})\big| \leq \epsilon_{\delta,2},\ \forall j \in [J]\}$. Hence, we have when $n' \geq \max\{n_{\delta,J,1}, n_{\delta,J,2}\}$,

$$\mathbb{P}(\mathcal{E}_0 \cap \mathcal{E}_1) \geq 1 - 2\delta.$$

Term q_1: First, we deal with the term q_1. Conditioning on the events $\mathcal{E}_0, \mathcal{E}_1$, we have

$$\Big|\frac{1}{J}\sum_{j=1}^{J}\left(\hat{\beta}_i^{(j)} - \hat{\beta}_{\text{MLE},i}\right)^\top \mathbf{x}\mathbf{x}^\top \left(\hat{\beta}_j^{(j)} - \hat{\beta}_{\text{MLE},j}\right) - \frac{1}{n'}\text{Tr}\left((\mathbf{x}\mathbf{x}^\top)(\mathbf{M}_X(\hat{\beta}_{\text{MLE}}; \mathcal{S})^{-1})_{i,j}\right)\Big| \leq \epsilon_{\delta,2},$$

$$\Big|\left(\hat{\beta}_i^{(j)} - \hat{\beta}_{\text{MLE},i}\right)^\top \mathbf{x}\mathbf{x}^\top \left(\hat{\beta}_j^{(j)} - \hat{\beta}_{\text{MLE},j}\right)\Big| \leq \frac{L^2}{n'},$$

where $(\mathbf{M}_X(\hat{\beta}_{\text{MLE}}; \mathcal{S})^{-1})_{i,j}$ means the (i,j)-th block of $Kd \times Kd$ matrix $\mathbf{M}_X$. By using Hoeffding's inequality (Zhang, 2023), we can prove that with probability at least $1 - 2\delta$,

$$\big|q_1 - \frac{1}{n'}\text{Tr}\left((\mathbf{x}\mathbf{x}^\top)(\mathbf{M}_X(\hat{\beta}_{\text{MLE}}; \mathcal{S})^{-1})_{i,j}\right)\big| \leq O(\frac{1}{n'\sqrt{J}}). \tag{32}$$

Term q_2: Then, we deal with the term q_2. Conditioning on $\mathcal{E}_1$ and based on the independence of the J estimators given $\mathcal{S}$, we have

$$\begin{aligned}\mathbb{E}\left(\tilde{\beta}_i^\top - \hat{\beta}_{\text{MLE},i}\right)^\top \mathbf{x}\mathbf{x}^\top \left(\tilde{\beta}_j - \hat{\beta}_{\text{MLE},j}\right) =& \text{Cov}\big((\mathbf{x}\mathbf{x}^\top)^{1/2}\tilde{\beta}_i, (\mathbf{x}\mathbf{x}^\top)^{1/2}\tilde{\beta}_j\big)\\ =& \frac{1}{J}\text{Cov}\big((\mathbf{x}\mathbf{x}^\top)^{1/2}\hat{\beta}_i^{(j)}, (\mathbf{x}\mathbf{x}^\top)^{1/2}\hat{\beta}_j^{(j)}\big)\\ =& \frac{1}{J}\mathbb{E}(\hat{\beta}_i^{(j)} - \hat{\beta}_{\text{MLE},i})^\top \mathbf{x}\mathbf{x}^\top (\hat{\beta}_j^{(j)} - \hat{\beta}_{\text{MLE},j}),\end{aligned}$$

which implies that conditioning on event $\mathcal{E}_1$,

$$\begin{aligned}&\big|\mathbb{E}(\tilde{\beta}_i^{(j)} - \hat{\beta}_{\text{MLE}})_i^\top \mathbf{x}\mathbf{x}^\top (\hat{\beta}_j^{(j)} - \hat{\beta}_{\text{MLE},j}) - \frac{1}{Mn'}\text{Tr}\left((\mathbf{x}\mathbf{x}^\top)(\mathbf{M}_X(\hat{\beta}_{\text{MLE}}; \mathcal{S})^{-1})_{i,j}\right)\big|\\ =& \frac{1}{J}\big|\mathbb{E}(\hat{\beta}_i^{(j)} - \hat{\beta}_{\text{MLE},i})^\top \mathbf{x}\mathbf{x}^\top (\hat{\beta}_j^{(j)} - \hat{\beta}_{\text{MLE},j}) - \frac{1}{n'}\text{Tr}\left((\mathbf{x}\mathbf{x}^\top)(\mathbf{M}_X(\hat{\beta}_{\text{MLE}}; \mathcal{S})^{-1})_{i,j}\right)\big|\\ \leq& \frac{\epsilon_{\delta,2}}{J}.\end{aligned}$$

Additionally, conditioning on $\mathcal{E}_0$, by using the matrix Hoeffding's inequality, we have for all $\epsilon > 0$,

$$\mathbb{P}\left(\left\|\tilde{\beta} - \hat{\beta}_{\text{MLE}}\right\|_2 \geq \epsilon\right) \leq 2\exp\left(-J\frac{2\epsilon^2 n'}{\Delta_{\delta_0}}\right) = \delta.$$

By taking $\epsilon = \sqrt{\frac{\Delta_{\delta_0}\log(2/\delta)}{2Mn'}}$, with probability at least $1-\delta-J\delta_0$,

$$\|\tilde{\beta}-\hat{\beta}_{\text{MLE}}\|_2 \leq \sqrt{\frac{\Delta_{\delta_0}\log(2/\delta)}{2Mn'}},$$

which means that

$$\left|(\tilde{\beta}_i^{(j)}-\hat{\beta}_{\text{MLE},i})^\top \mathbf{x}\mathbf{x}^\top(\tilde{\beta}_j^{(j)}-\hat{\beta}_{\text{MLE},j})\right| \leq O(\frac{L^2}{Mn'}).$$

By using Hoeffding's inequality, we can prove that with probability at least $1-2\delta$,

$$\left|q_2 - \frac{1}{Mn'}\text{Tr}\left((\mathbf{x}\mathbf{x}^\top)(\mathbf{M}_X(\hat{\beta}_{\text{MLE}};\mathcal{S})^{-1})_{i,j}\right)\right| \leq O(\frac{1}{Mn'}). \tag{33}$$

By combining (32) and (33) with (28), we have with a high probability,

$$\left|(\Sigma_J(\mathbf{x}))_{i,j} - \frac{1}{n'}\text{Tr}\left((\mathbf{x}\mathbf{x}^\top)(\mathbf{M}_X(\hat{\beta}_{\text{MLE}};\mathcal{S})^{-1})_{i,j}\right)\right| \leq O\Big(\frac{1}{n'\sqrt{J}}\Big). \tag{34}$$

Therefore, we have with a high probability

$$\begin{aligned}
& n'\text{Tr}\left(\psi(\tilde{\beta};\mathbf{x},y)\Sigma_J(\mathbf{x})\right) - \text{Tr}\left(\psi(\hat{\beta}_{\text{MLE}};\mathbf{x},\mathbf{y})\otimes(\mathbf{x}\mathbf{x}^\top)\mathbf{M}_X^{-1}(\hat{\beta}_{\text{MLE}};\mathcal{S})\right) \\
\leq & \frac{1}{\sqrt{n}} + \Bigg| n'\sum_{i,j=1}^{K}(\psi(\hat{\beta}_{\text{MLE}};\mathbf{x},y))_{i,j}\cdot(\Sigma_J(\mathbf{x}))_{i,j} \\
& \quad - \sum_{i,j=1}^{K}(\psi(\hat{\beta}_{\text{MLE}};\mathbf{x},y))_{i,j}\text{Tr}\left((\mathbf{x}\mathbf{x}^\top)(\mathbf{M}_X(\hat{\beta}_{\text{MLE}};\mathcal{S})^{-1})_{i,j}\right)\Bigg| \\
\leq & \frac{1}{\sqrt{n}} + \frac{K^2}{\sqrt{J}},
\end{aligned}$$

where the first inequality uses $|\psi(\tilde{\beta};\mathbf{x},y)-\psi(\hat{\beta}_{\text{MLE}};\mathbf{x},y)| \leq O(1/\sqrt{n})$ and $|(\Sigma_J(\mathbf{x}))_{i,j}| \leq O(1/n')$, and the last inequality follows from (34). Hence, by taking a uniform bound for all $J \in B_J$, we finish the proof. □

Appendix B. Supporting Lemmas

Lemma 1. *For any subset $\bar{\mathcal{S}}$ and subsampling estimator $\bar{\beta}$ yielded by some subsampling probability π, we have*

$$\mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}[\mathcal{L}(\bar{\beta};\mathcal{S})-\mathcal{L}(\hat{\beta}_{MLE};\mathcal{S})] = \mathbb{E}_{\bar{\mathcal{S}}|\mathcal{S},\pi}\left[(\bar{\beta}-\hat{\beta}_{MLE})^\top \boldsymbol{M}_X(\hat{\beta}_{MLE};\mathcal{S})(\bar{\beta}-\hat{\beta}_{MLE}) + O_{P|\mathcal{S}}(\|\bar{\beta}-\hat{\beta}_{MLE}\|_2^2)\right]$$

Proof. By following the analysis of Lemma 1 in Wang et al. (2018), we obtain

$$\begin{aligned}
\mathcal{L}(\bar{\beta};\mathcal{S})-\mathcal{L}(\hat{\beta}_{\text{MLE}};\mathcal{S}) =& (\bar{\beta}-\hat{\beta}_{\text{MLE}})^\top\frac{\partial\mathcal{L}(\beta;\mathcal{S})}{\partial\beta}\Big|_{\beta=\hat{\beta}_{\text{MLE}}} + (\bar{\beta}-\hat{\beta}_{\text{MLE}})^\top\left(\frac{\partial^2\mathcal{L}(\beta;\mathcal{S})}{\partial\beta^2}\Big|_{\beta=\hat{\beta}_{\text{MLE}}}\right)(\bar{\beta}-\hat{\beta}_{\text{MLE}}) \\
& + O_{P|\mathcal{S}}(\|\bar{\beta}-\hat{\beta}_{\text{MLE}}\|_2^2)
\end{aligned}$$

Since $\frac{\partial \mathcal{L}(\beta;\mathcal{S})}{\partial \beta}\Big|_{\beta=\hat{\beta}_{\mathrm{MLE}}} = 0$, and $\frac{\partial^2 \mathcal{L}(\beta;\mathcal{S})}{\partial \beta^2}\Big|_{\beta=\hat{\beta}_{\mathrm{MLE}}} = \mathbf{M}_X(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})$, we have

$$\mathcal{L}(\bar{\beta};\mathcal{S}) - \mathcal{L}(\hat{\beta}_{\mathrm{MLE}};\mathcal{S}) = (\bar{\beta} - \hat{\beta}_{\mathrm{MLE}})^\top \mathbf{M}_X(\hat{\beta}_{\mathrm{MLE}};\mathcal{S})(\bar{\beta} - \hat{\beta}_{\mathrm{MLE}}) + O_{P|\mathcal{S}}(\|\bar{\beta} - \hat{\beta}_{\mathrm{MLE}}\|_2^2).$$

□

Lemma 2 (Variance of $\bar{\beta}$, Theorem 1 of Yao and Wang (2019)). *If Assumptions 1 and 2 hold, as $n \to \infty$ and $r \to \infty$, condition on $\mathcal{S}_{X,2}$ in probability,*

$$\boldsymbol{V}^{-1/2}(\bar{\beta} - \hat{\beta}_{MLE}) \to N(0, \boldsymbol{I}),$$

where

$$\begin{aligned}
\boldsymbol{V} &= \boldsymbol{M}_X^{-1}(\hat{\beta}_{MLE};\mathcal{S})\,\boldsymbol{V}_c(\hat{\beta}_{MLE};\mathcal{S})\boldsymbol{M}_X^{-1}(\hat{\beta}_{MLE};\mathcal{S}),\\
\boldsymbol{M}_X(\hat{\beta}_{MLE};\mathcal{S}) &= \frac{1}{n}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}} \phi(\hat{\beta}_{MLE};\mathbf{x}) \otimes (\mathbf{x}\mathbf{x}^\top),\\
\boldsymbol{V}_c(\hat{\beta}_{MLE};\mathcal{S}) &= \frac{1}{rn^2}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}} \frac{\psi(\hat{\beta}_{MLE};\mathbf{x}) \otimes (\mathbf{x}\mathbf{x}^\top)}{\pi(\mathbf{x},\mathbf{y})}.
\end{aligned}$$

Lemma 3. *Consider a finite sequence $\{\boldsymbol{X}_i\}_{i=1}^n$ of independent, $D \times D$ random matrices with the same expectation $\mathbb{E}\boldsymbol{X}_i := \bar{M}_X$. Let $M_X = \frac{1}{n}\sum_{i=1}^n \boldsymbol{X}_i$. If we assume that $\lambda_{\min}(\bar{M}_X) \wedge \lambda_{\min}(M_X) \geq \kappa_{\min}$ and $\lambda_{\max}(\bar{M}_X) \vee \lambda_{\max}(M_X) \leq \kappa_{\max}$ for some constant $\kappa_{\min}, \kappa_{\max} > 0$, we have with probability at least $1 - \delta$,*

$$\|M_X^{-1} - \bar{M}_X^{-1}\|_{op} \leq \kappa_{\min}^{-2}\kappa_{\max}\sqrt{\frac{8\log(D/\delta)}{n}}.$$

Proof. We deduce that

$$\begin{aligned}
\|M_X^{-1} - \bar{M}_X^{-1}\|_{\mathrm{op}} &= \left\|M_X^{-1}(\bar{M}_X - M_X)\bar{M}_X^{-1}\right\|_{\mathrm{op}}\\
&\leq \left\|M_X^{-1}\right\|_{\mathrm{op}} \cdot \left\|\bar{M}_X - M_X\right\|_{\mathrm{op}} \cdot \left\|\bar{M}_X^{-1}\right\|_{\mathrm{op}}\\
&\leq \kappa_{\min}^{-2} \cdot \left\|\bar{M}_X - M_X\right\|_{\mathrm{op}},
\end{aligned}$$

where the last inequality is due to $\lambda_{\min}(M_X) \geq \kappa_{\min}$ as $n \to \infty$. Then, it remains to show that $\|\bar{M}_X - M_X\|_{\mathrm{op}}$ goes to zero.

Since $M_X^2 \preceq \kappa_{\max}^2$, and

$$\mathbb{E}\left[M_X - \bar{M}_X\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \mathbf{X}_i - \mathbb{E}\mathbf{X}\right] = 0,$$

we apply matrix Hoeffding inequality from Theorem 1.3 in Tropp (2012) to obtain that for all $t \geq 0$,

$$\mathbb{P}\left(\left\|M_X - \bar{M}_X\right\|_{\mathrm{op}} \geq t\right) \leq D \cdot e^{-nt^2/8\kappa_{\max}^2}.$$

By taking $t = \kappa_{\max}\sqrt{8\log(D/\delta)/n}$ for any $\delta > 0$, we get with probability at least $1-\delta$

$$\left\|M_X - \bar{M}_X\right\|_{\text{op}} \leq \kappa_{\max}\sqrt{\frac{8\log(D/\delta)}{n}},$$

which completes the proof. □

Appendix C. Algorithms

C.1 Uncertainty Sampling Algorithm for DNNs

The uncertainty sampling algorithms for DNNs are shown in Algorithm 4 and 5.

```
Algorithm 4: COPS for sampling with labels on DNNs
  Input: Training data $\mathcal{S}$, one probe datasets $\mathcal{S}'$, sub-sampling size $r$.
  Output: The selected subset $\bar{\mathcal{S}}$ and the model $\bar{\theta}$.
1 For each $(\mathbf{x}, \mathbf{y}) \in \mathcal{S}$, obtain $u(\mathbf{x}, \mathbf{y})$ by Algorithm 6 with the probe set $\mathcal{S}'$;
2 Randomly draw $\bar{\mathcal{S}}$ containing $r$ samples from $\mathcal{S}$ by $u(\mathbf{x}, \mathbf{y}) / \sum_{(\mathbf{x}', \mathbf{y}') \in \mathcal{S}} u(\mathbf{x}', \mathbf{y}')$.
3 Solve $\bar{\beta}$ on the weighted subset $\bar{\mathcal{S}}(\pi)$ as follows.
4 $$\bar{\theta} = \arg\min_{\theta} \left( -\frac{1}{r} \sum_{(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \bar{\mathcal{S}}} \frac{1}{\pi(\bar{\mathbf{x}}, \bar{\mathbf{y}})} \left( \sum_{k=1}^{K} \delta_k(\bar{\mathbf{y}}) \bar{f}_k(\theta; \bar{\mathbf{x}}) - \log\{1 + \sum_{l=1}^{K} \exp(f_l(\theta; \bar{\mathbf{x}}))\} \right) \right). \tag{35}$$
```

```
Algorithm 5: COPS for sampling without labels on DNNs
  Input: Training data $\mathcal{S}_X$, one probe datasets $\mathcal{S}'$, sub-sampling size $r$.
  Output: The selected subset $\bar{\mathcal{S}}$ with inquired label and the model $\bar{\theta}$.
1 For each $\mathbf{x} \in \mathcal{S}_X$, obtain $u(\mathbf{x})$ by Algorithm 6 with the probe set $\mathcal{S}'$;
2 Randomly draw $\bar{\mathcal{S}}_X$ containing $r$ samples from $\pi(\mathbf{x}) = \mathcal{S}$ by $u(\mathbf{x}) / \sum_{\mathbf{x}' \in \mathcal{S}} u(\mathbf{x}')$.
3 Obtain the labeled data set $\bar{\mathcal{S}}$ by labeling each sample in $\bar{\mathcal{S}}_X$.
4 Solve $\bar{\beta}$ on the weighted subset $\bar{\mathcal{S}}(\pi)$ as follows
  $$\bar{\theta} = \arg\min_{\theta} \left( -\frac{1}{r} \sum_{\bar{\mathbf{x}} \in \bar{\mathcal{S}}_X} \frac{1}{\pi(\bar{\mathbf{x}})} \left( \sum_{k=1}^{K} \delta_k(\bar{\mathbf{y}}) \bar{f}_k(\theta; \bar{\mathbf{x}}) - \log\{1 + \sum_{l=1}^{K} \exp(f_l(\theta; \bar{\mathbf{x}}))\} \right) \right). \tag{36}$$
```

C.2 COPS-clip Algorithm for DNNs

The algorithm of COPS-clip for DNNs is shown in Algorithm 8.

```
Algorithm 6: Uncertainty estimation for DNNs.
  Input: Probe datasets $\mathcal{S}'$, the sampling dataset $\mathcal{S}$ for sampling with labels or $\mathcal{S}_X$
         for sampling without labels.
  Output: The estimated uncertainty for each sample in $\mathcal{S}$ or $\mathcal{S}_X$.
1 For $j = 1, ..., J$, randomly initialize $f_{\theta^{(j)}}$ with different seeds and then minimize the
   loss of $f_{\theta^{(j)}}$ on $\mathcal{S}'$ by SGD independently.
2 For each $\mathbf{x}$, obtain the output logits of each model $\{f_{\theta^{(j)}}(\mathbf{x})\}_{j=1}^J$ and the covariance
   of the logits:
     $$\Sigma_J(\mathbf{x}) = \frac{1}{J-1}\sum_{j=1}^J \left(f_{\theta^{(j)}}(\mathbf{x}) - \frac{1}{J}\sum_{j=1}^J f_{\theta^{(l)}}(\mathbf{x})\right)\left(f_{\theta^{(j)}}(\mathbf{x}) - \frac{1}{J}\sum_{j=1}^J f_{\theta^{(l)}}(\mathbf{x})\right)^\top. \quad (37)$$
3 Get the predicted probability of sample $\mathbf{x}$ by $\frac{1}{J}\sum_{j=1}^J p(f_{\theta^{(j)}}; \mathbf{x})$. Estimate the
   uncertainty for each sample
     • Case (1) sampling with labels. Obtain $\psi(\tilde{\beta}; \mathbf{x}, \mathbf{y})$ according to Eqn (5) and obtain
       the uncertainty the estimation
         $$u(\mathbf{x}, \mathbf{y}) = \mathrm{Tr}\left(\psi(\tilde{\beta}; \mathbf{x}, \mathbf{y})\Sigma_J(\mathbf{x})\right);$$
     • Case (2) sampling without labels. Obtain $\phi(\tilde{\beta}; \mathbf{x})$ according to Eqn (4) and obtain
       the uncertainty the estimation as
         $$u(\mathbf{x}) = \mathrm{Tr}\left(\phi(\tilde{\beta}; \mathbf{x})\Sigma_J(\mathbf{x})\right).$$
```

Appendix D. Experimental Details

D.1 Details of the experiment in Section 6

We evaluate the performance of the model on the original test set. We use AdamW Optimizer Loshchilov and Hutter (2019) with cosine lr decay for 150 epochs, the batch size is 256. We put a limit on the maximum weight when solving Eqn (9) to avoid large variance. Specifically, let u_i denote the uncertainty of ith sample. In Eqn (9), we use $\frac{1}{u_i}$ to reweight the selected data. To avoid large variance, we use $\frac{1}{\max\{\beta, u_i\}}$ as the reweighting to replace $\frac{1}{u_i}$. We simply set $\beta = 0.1$ for all experiments following Citovsky et al. (2023). So the sampling with labels algorithm with full details are shown in Algorithm 9 and 10. Comparing Algorithm 9-10 with Algorithm 7-8, we can see the only difference is that we use π_β instead of π to re-weight the selected data, which is consistent with Citovsky et al. (2023). We use Algorithm 9 and 10 in the main experiment part by default.

Algorithm 7: COPS with uncertainty clipping for sampling with labels on DNNs

```
Input: Training data $\mathcal{S}$, one probe datasets $\mathcal{S}'$, sub-sampling size $r$,
       hyper-parameter $\alpha$.
Output: The selected subset $\bar{\mathcal{S}}$ and the model $\bar{\theta}$.
1 For each $(\mathbf{x}, \mathbf{y}) \in \mathcal{S}$, obtain $u(\mathbf{x}, \mathbf{y})$ by Algorithm 6 with the probe set $\mathcal{S}'$;
2 Randomly draw $\bar{\mathcal{S}}$ containing $r$ samples from $\mathcal{S}$ by
      $\pi^{\alpha}(\mathbf{x}, \mathbf{y}) = \min\{\alpha, u(\mathbf{x}, \mathbf{y})\} / \sum_{(\mathbf{x}', \mathbf{y}') \in \mathcal{S}} \min\{\alpha, u(\mathbf{x}', \mathbf{y}')\}.$
3 Calculate the reweighting of each selected sample $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ as $\frac{1}{\pi(\bar{\mathbf{x}}, \bar{\mathbf{y}})}$, where
      $\pi(\mathbf{x}, \mathbf{y}) = u(\mathbf{x}, \mathbf{y}) / \sum_{(\mathbf{x}', \mathbf{y}') \in \mathcal{S}} u(\mathbf{x}', \mathbf{y}').$
4 Solve $\bar{\theta}$ on the weighted subset as:
      $\bar{\theta} = \arg\min_{\theta} \left( -\frac{1}{r} \sum_{(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \bar{\mathcal{S}}} \frac{1}{\pi(\bar{\mathbf{x}}, \bar{\mathbf{y}})} \left( \sum_{k=1}^{K} \delta_k(\bar{\mathbf{y}}) f_{\theta,k}(\bar{\mathbf{x}}) - \log\{1 + \sum_{l=1}^{K} \exp(f_{\theta,l}(\bar{\mathbf{x}}))\} \right) \right)$
```

D.2 Hyper-parameters of the experiments in Section 6

Dataset	CIFARBinary	CIFAR10/CIFAR10-N	CIFAR100	SVHN	Places365	IMDB
Class Number	2	10	100	10	10	2
Size of the probe set	2,000	10,000	20,000	10,000	10,000	5,000
Start learning rate 1	0.1	0.1	0.1	0.1	0.1	0.1
Learning rate schedule 1	schedule 1	schedule 1	schedule 1	schedule 1	schedule 1	no schedule
Optimizer 1	SGD	SGD	SGD	SGD	SGD	AdamW
Epoch 1	100	100	100	100	100	20
Size of the sampling set	8,000	40,000	30,000	63,257	40,000	20,000
Start learning rate 2	0.1	0.1	0.1	0.1	0.1	0.1
Learning rate schedule 2	schedule 2	schedule 2	schedule 2	schedule 2	schedule 1	no schedule
Optimizer 2	AdamW	AdamW	AdamW	AdamW	SGD	AdamW
Epoch 2	150	150	150	150	100	20
Size of the testing set	2,000	10,000	10,000	26,032	1,000	25,000

Table 4: This table illustrates the training details. Here we set weight decay as 5e-4 for all the experiments. Here no schedule means using the start learning rate without modification during training. Schedule 1 stands for the decaying of the learning rate by 0.1 every 30 epochs. Schedule 2 means using the cosine learning schedule with $T_{max} = 50$ and $eta_{min} = 0$

Algorithm 8: COPS with uncertainty clipping for sampling without labels on DNNs

```
Input: Training data $\mathcal{S}_X$, one probe datasets $\mathcal{S}'$, sub-sampling size $r$,
       hyper-parameter $\alpha$.
Output: The selected subset $\bar{\mathcal{S}}$ with inquired label and the model $\bar{\theta}$.
1 For each $\mathbf{x} \in \mathcal{S}_X$, obtain $u(\mathbf{x})$ by Algorithm 6 with the probe set $\mathcal{S}'$;
2 Randomly draw $\bar{\mathcal{S}}_X$ containing $r$ samples from $\mathcal{S}$ by
      $$\pi^{\alpha}(\mathbf{x}) = \min\{\alpha, u(\mathbf{x})\} / \sum_{\mathbf{x}' \in \mathcal{S}_X} \min\{\alpha, u(\mathbf{x}')\}.$$
3 Calculate the reweighting of each selected sample $\bar{\mathbf{x}}$ as $\frac{1}{\pi(\bar{\mathbf{x}})}$, where
      $$\pi(\mathbf{x}) = u(\mathbf{x}) / \sum_{(\mathbf{x}') \in \mathcal{S}_X} u(\mathbf{x}').$$
4 Obtain the labeled data set $\bar{\mathcal{S}}$ by labeling each sample in $\bar{\mathcal{S}}_X$
5 Solve $\bar{\theta}$ on the weighted subset as:
      $$\bar{\theta} = \arg\min_{\theta} \left( -\frac{1}{r} \sum_{(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \bar{\mathcal{S}}} \frac{1}{\pi(\bar{\mathbf{x}})} \left( \sum_{k=1}^{K} \delta_k(\bar{\mathbf{y}}) f_{\theta,k}(\bar{\mathbf{x}}) - \log\{1 + \sum_{l=1}^{K} \exp(f_{\theta,l}(\bar{\mathbf{x}}))\} \right) \right).$$
```

D.3 Structure of models for IMDB

We adopt GRU for IMDB, whose structure is shown as follows:

layer	GRU Model
1	Embedding(2000, 200)
2	Dropout(p=0.3)
3	GRU(hidden_size= 24, num_layers=2, dropout=0.3, bidirectional=True)
4	Maxpool() & Avgpool()
5	Concat(last,max, avg)
6	Linear(in_features=72, out_features=1, bias=True)

Table 5: Model structure for GRU

Appendix E. More experimental results

E.1 The Sampling Weight on Low Density Region

The optimal sampling ratio based on uncertainty assumes no model misspecification. Sampling schemes from low-density regions show high uncertainty, but prior studies indicate

```
Algorithm 9: COPS with full details for sampling with labels on DNNs
    Input: Training data $\mathcal{S}$, one probe datasets $\mathcal{S}'$, sub-sampling size $r$,
           hyper-parameter $\alpha$.
    Output: The selected subset $\bar{\mathcal{S}}$ and the model $\bar{\theta}$.
  1 For each $(\mathbf{x}, \mathbf{y}) \in \mathcal{S}$, obtain $u(\mathbf{x}, \mathbf{y})$ by Algorithm 6 with the probe set $\mathcal{S}'$;
  2 Randomly draw $\bar{\mathcal{S}}$ containing $r$ samples from $\mathcal{S}$ by
        $$\pi^{\alpha}(\mathbf{x}, \mathbf{y}) = \min\{\alpha, u(\mathbf{x}, \mathbf{y})\} / \sum_{(\mathbf{x}', \mathbf{y}') \in \mathcal{S}} \min\{\alpha, u(\mathbf{x}', \mathbf{y}')\}.$$
  3 Calculate the reweighting of each selected sample $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ as $\frac{1}{\pi_\beta(\bar{\mathbf{x}}, \bar{\mathbf{y}})}$, where
        $$\pi_{\beta}(\mathbf{x}, \mathbf{y}) = \max\{\beta, u(\mathbf{x}, \mathbf{y})\} / \sum_{(\mathbf{x}', \mathbf{y}') \in \mathcal{S}} \max\{\beta, u(\mathbf{x}', \mathbf{y}')\}.$$
  4 Solve $\bar{\theta}$ on the weighted subset as:
        $$\bar{\theta} = \arg\min_{\theta} \left( -\frac{1}{r} \sum_{(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \bar{\mathcal{S}}} \frac{1}{\pi_\beta(\bar{\mathbf{x}}, \bar{\mathbf{y}})} \left( \sum_{k=1}^{K} \delta_k(\bar{\mathbf{y}}) f_{\theta,k}(\bar{\mathbf{x}}) - \log\{1 + \sum_{l=1}^{K} \exp(f_{\theta,l}(\bar{\mathbf{x}}))\} \right) \right)$$
```

that significant misspecification on the low density region can greatly affect model estimation. The proposed uncertainty sampling methods may also face this issue due to their focus on low-density areas. To illustrate the focus of the COPS on the low density region, we showcase the weight of COPS put on x_1, which is the low density region of the simulation example in Section 4.2.

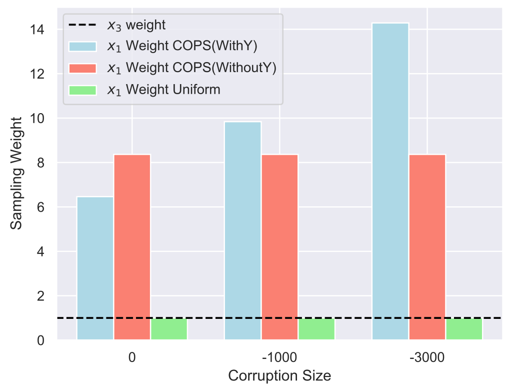


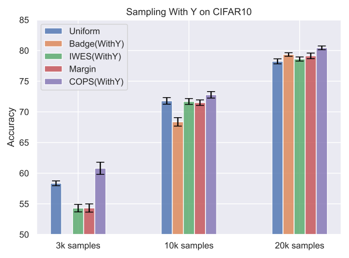
Figure 10: Illustration of the sampling weight on the low density region given by different sampling methods.

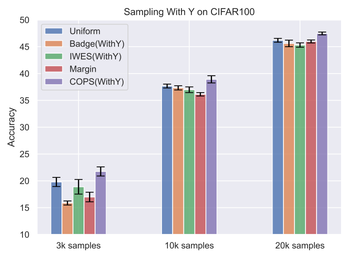
```
Algorithm 10: COPS with full details for sampling without labels on DNNs
  Input: Training data $\mathcal{S}_X$, one probe datasets $\mathcal{S}'$, sub-sampling size $r$,
         hyper-parameter $\alpha$.
  Output: The selected subset $\bar{\mathcal{S}}$ with inquired label and the model $\bar{\theta}$.
1 For each $\mathbf{x} \in \mathcal{S}_X$, obtain $u(\mathbf{x})$ by Algorithm 6 with the probe set $\mathcal{S}'$;
2 Randomly draw $\bar{\mathcal{S}}_X$ containing $r$ samples from $\mathcal{S}$ by
      $$\pi^{\alpha}(\mathbf{x}) = \min\left\{\alpha, u(\mathbf{x})\right\} / \sum_{\mathbf{x}' \in \mathcal{S}_X} \min\left\{\alpha, u(\mathbf{x}')\right\}.$$
3 Calculate the reweighting of each selected sample $\bar{\mathbf{x}}$ as $\frac{1}{\pi_\beta(\bar{\mathbf{x}})}$, where
      $$\pi_\beta(\mathbf{x}) = \max\{\beta, u(\mathbf{x})\} / \sum_{(\mathbf{x}') \in \mathcal{S}_X} \max\{\beta, u(\mathbf{x}')\}.$$
4 Obtain the labeled data set $\bar{\mathcal{S}}$ by labeling each sample in $\bar{\mathcal{S}}_X$
5 Solve $\bar{\theta}$ on the weighted subset as:
      $$\bar{\theta} = \arg\min_{\theta} \left( -\frac{1}{r} \sum_{(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \bar{\mathcal{S}}} \frac{1}{\pi_\beta(\bar{\mathbf{x}})} \left( \sum_{k=1}^{K} \delta_k(\bar{\mathbf{y}}) f_{\theta,k}(\bar{\mathbf{x}}) - \log\{1 + \sum_{l=1}^{K} \exp(f_{\theta,l}(\bar{\mathbf{x}}))\} \right) \right).$$
```

Specifically, we standardized the weight of x_3 from the high-density region to 1, allowing us to calculate the weight of x_1 relative to x_3. For instance, a weight of 10 for x_1 indicates that x_1 is 10 times heavier than x_3. We can see the sampling weight of x_1 is much larger than that of x_3, indicating that COPS put a large weight on the low-density region.

E.2 Additional Experiment Without Replacement

Since some deep learning empirical methods such as IWeS and Badge were originally only verified on sampling without replacement, we also naively try COPS in the sampling without replacement in this section. We found that even though COPS is not designed for sampling without replacement, a naive application of COPS still achieves the state-of-the-art performance in this setting.


(a) CIFAR10.


(c)CIFAR100

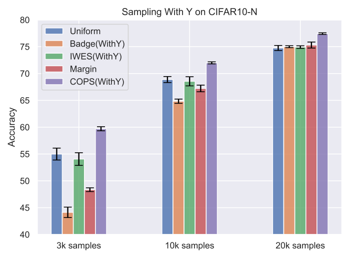
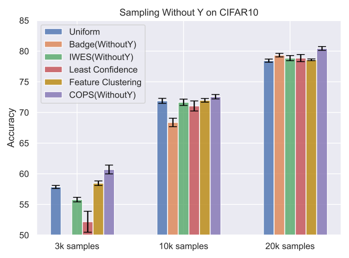
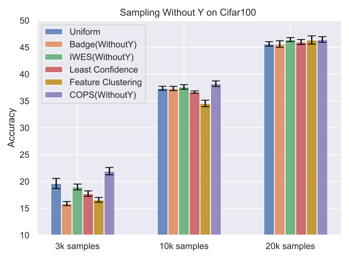

(b)CIFAR10-N

Figure 11: Results for sampling with labels and without replacement. For Badge with 3000 samples, the performance is lower than 50, so the bar is clipped in our figures.


(a) CIFAR10.


(b)CIFAR100

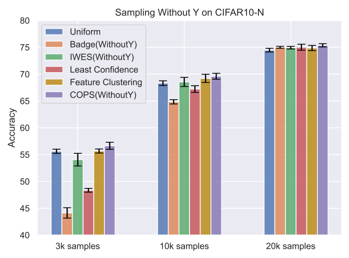

(c) CIFAR10-N

Figure 12: Results for sampling without labels and without replacement. For Badge with 3000 samples, the performance is lower than 50, so the bar is clipped in our figures.

E.3 Label noise

We found that COPS-vanilla selects large number of samples with large uncertainty, which is later shown to exacerbate label noise when sub-sampling a dataset with natural label noise, CIFAR10-N.

Dataset	Sampling Method	WithY	WithoutY
CIFAR10-N	Uniform	0.0872	0.0912
	COPS-vanilla	0.1314	0.0934
	COPS-clip	0.0941	0.0906

Table 6: Noise ratio comparison for different sampling methods. Here we sample 1000 instances from each class and use the uncertainty with the label known. The noise ratio of uniform sampling for sampling with and without labels is slightly different. This is because we sample within each class in sampling with labels. For example, in the CIFAR10-N-1000-WithY, we uniformly select 100 samples in each class. However, in the CIFAR10-N-1000-WithoutY, we uniformly select 1000 samples in the whole dataset.

E.4 Comparison of threshold in the sampling and reweighting stages

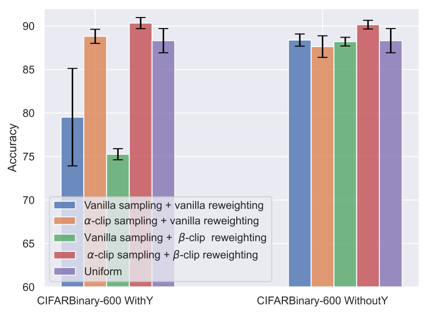
Let u represent $u(\mathbf{x}, \mathbf{y})$ for sampling with labels and $u(\mathbf{x})$ for sampling without labels. The COPS method consists of two stages:

- Stage 1: Data sampling based on u. To prevent COPS from oversampling low-density data, we propose limiting the maximum value of u by $\min\{\alpha, u\}$ in this stage. (Section 4)
- Stage 2: Weighted learning, where each selected sample is assigned a weight of $1/u$ to obtain an unbiased estimator. To reduce variance, Ionides (2008); Swaminathan and Joachims (2015); Citovsky et al. (2023) propose adding a threshold of $1/\max\{\beta, u\}$ in this stage.

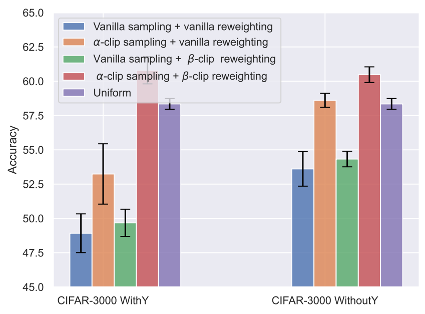
We refer to the thresholding method in the sampling stage as "α-clip" and the thresholding method in the reweighting stage as "β-clip". We investigate the impact of "α-clip" and "β-clip" on the final performance. We compare four methods based on whether thresholding is applied in the first and second stages:

- "Vanilla sampling + vanilla reweighting": No thresholding is used in either stage.
- "α-clip sampling + vanilla reweighting": Thresholding of $\min\{\alpha, u\}$ is applied in the sampling stage, while reweighting remains unchanged.
- "Vanilla sampling + β-clip reweighting": Sampling stage remains unchanged, but a threshold of $1/\max\{\beta, u\}$ is used in the reweighting stage.
- "α-clip sampling + β-clip reweighting": Both sampling and reweighting stages utilize thresholding methods.

The comprehensive results are displayed in Figure 13. The results clearly demonstrate that both the utilization of α-clip in sampling and β-clip in reweighting lead to performance improvement. Importantly, it is observed that the performance gain of each method cannot be solely attributed to the other. The optimal performance is achieved by effectively combining the benefits of both techniques.



(a) CIFARBinary-600.

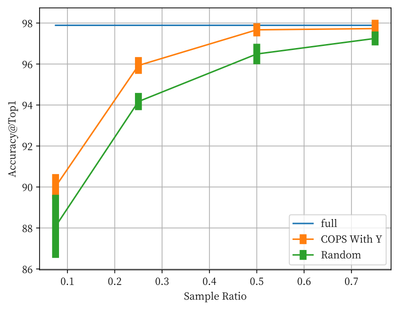


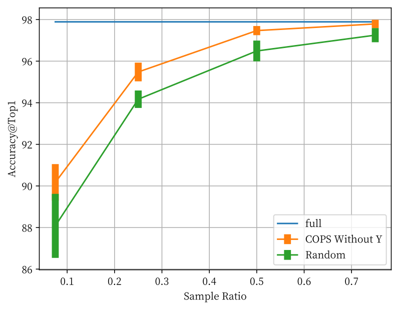
(b)CIFAR10-3000.

Figure 13: Results of comparing the α-clip in the first stage (sampling stage) and β-clip in the second stage (reweighting stage)

E.5 Comparison with full data

CIFAR10Binary. The figure for CIFAR10Binary is shown in Figure.14.

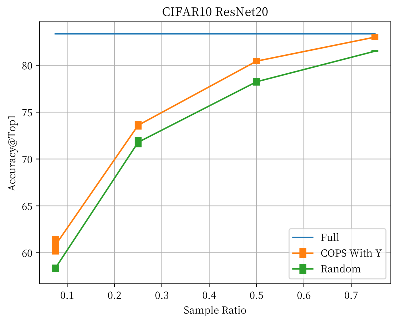




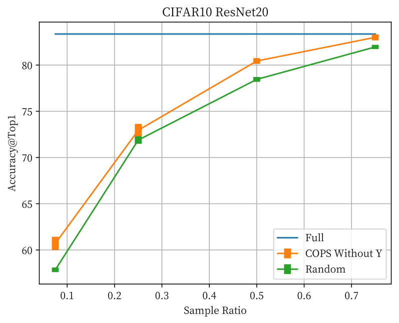
(a)Comparison on CIFAR10Binary with Y. (b)Comparison on CIFAR10Binary without Y.

Figure 14: Comparison with full dataset.

CIFAR10. The figure for CIFAR10 is shown in Figure.15.



(a)Comparison on CIFAR10 with Y.



(b)Comparison on CIFAR10 without Y.

Figure 15: Comparison with full dataset.

E.6 Computational Cost

To further demonstrate efficiency, we conducted experiments on CIFAR-10 Binary to compare the computational cost of uncertainty generation versus retraining. As shown in Table 7, uncertainty generation requires significantly less computation than full retraining. Moreover, with larger datasets and sufficient per-class samples, the relative cost of uncertainty generation can be further reduced.

Stage	GPU hours
Uncertainty Generation	0.013
Retraining	0.041

Table 7: A comparison on the GPU hour of different stages