DIRECT OPTIMIZATION OF SELECTIVE CLASSIFIER FOR AUTO-LABELING VIA EXACT REFORMULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

With the increasing demand for large-scale labeled data, auto-labeling has emerged as an effective technique for annotating unlabeled data with minimal human intervention. The goal of auto-labeling is to achieve a coverage as high as possible on the unlabeled data while keeping the error rate below a user-specified value. Threshold-based auto-labeling (TBAL) is a multi-round framework that leverages a classifier to selectively label samples based on whether the confidence scores exceed an estimated threshold. However, standard classifier training does not take into account the error-coverage tradeoff in auto-labeling and may yield confidence scores that poorly separate correct from incorrect predictions. Furthermore, directly optimizing this objective is challenging because both coverage and error involve indicator functions with zero gradients almost everywhere, while replacing them with smooth surrogates introduces an approximation gap that undermines optimality. In this paper, we propose *AL-ERO* (Auto-Labeling via Exact Reformulation and Optimization), which trains a selective classifier by jointly optimizing classification and selection. AL-ERO uses an exact reformulation of the selection indicator, thereby eliminating the approximation gap introduced by smooth surrogates. Experiments on four datasets covering image and text modalities show that AL-ERO achieves improved auto-labeling performance.

1 INTRODUCTION

Large-scale, high-quality labeled data are essential for developing supervised predictive models, while the acquisition of labels is expensive and time-consuming. Auto-labeling offers a practical way to reduce this burden by assigning labels to unlabeled data using predictive models, e.g., task-specific classifiers or zero- and few-shot foundation models (Lee et al., 2013; Xie et al., 2020; Arazo et al., 2020; Ratner et al., 2017; 2016; Bach et al., 2019; Ding et al., 2023; Gilardi et al., 2023). Since typical predictive models can make mistakes, after label predictions, these auto-labeling methods use heuristic acceptance rules to determine which predictions to accept—i.e., selection. But, they generally do not provide explicit control over the post-selection error rate.

Threshold-Based Auto-Labeling (TBAL) (Vishwakarma et al., 2023) provides a principled framework for controlling the post-selection error rate, aiming to achieve high coverage—the fraction of selected samples—while keeping the error rate below a user-specified value. In each round, TBAL learns a classifier and **designs** a confidence-threshold-based selector. The resulting error-coverage tradeoff depends on both prediction and selection quality; to improve the tradeoff, one needs a more accurate classifier, and a sharper selector—that tells apart correct vs incorrect predictions. Colander (Vishwakarma et al., 2024a) sharpens the selector by **learning**: they target the error-coverage tradeoff directly and try to learn a selector in each round through error-constrained coverage maximization. However, their post-classifier selector optimization might still be suboptimal, as the selection performance is jointly determined by both the classifier and the selector.

In this paper, we propose to optimize the error-coverage tradeoff by optimizing the classifier and selector **jointly**—vs. Colander's **separate** classifier training and selector learning. Based on the same error-constrained coverage maximization framework as Colander but with more involved functional structures, our formulation shares common computational challenges with Colander: (1) **Zero gradient of indicator function.** Both the coverage objective and error constraint involve indicator function, which has zero gradient almost everywhere, preventing gradient-based optimization methods; (2)

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Constrained nonconvex optimization. Due to the nonconvex constraint, it might be hard to find a feasible point in the first place, never mind a global solution.

Typical heuristics to handle them, e.g., smooth surrogates (Qin et al., 2010; Chapelle & Wu, 2010; Bénédict et al., 2021; Patel et al., 2022; Eban et al., 2017) to address (1) and penalty methods (Cotter et al., 2019; Kumar et al., 2021) to address (2)—as exactly done in Colander, may corrupt both error and coverage computation and lead to solutions that fail to meet the error constraint. To tackle both challenges in a principled way, we adopt the exact reformulation and optimization (ERO) idea of Peng et al. (2025), which concerns direct metric optimization in imbalanced classification. We derive an exact constrained reformulation of our joint classification–selection formulation by converting indictor functions into simple, equivalent constraints with reasonable, non-vanishing gradients, enabling gradient-based methods. Moreover, we apply the classic exact penalty method to handle the error and reformulated indicator constraints. This leads to our *AL-ERO* (Auto-Labeling via Exact Reformulation and Optimization) framework. Our contributions are summarized as follows:

- We propose *AL-ERO*, a joint training formulation for classification and selection for auto-labeling task, targeting high coverage under a prescribed error constraint.
- We derive an exact constrained reformulation of the classification–selection training problem. The reformulation preserves hard selection decisions while providing gradient information for optimization, avoiding the approximation gap introduced by smooth selection surrogates.
- Extensive experiments on four datasets spanning image and text modalities demonstrate that AL-ERO consistently improves auto-labeling performance over both classifier-optimization baselines and post-classifier selector optimization methods.

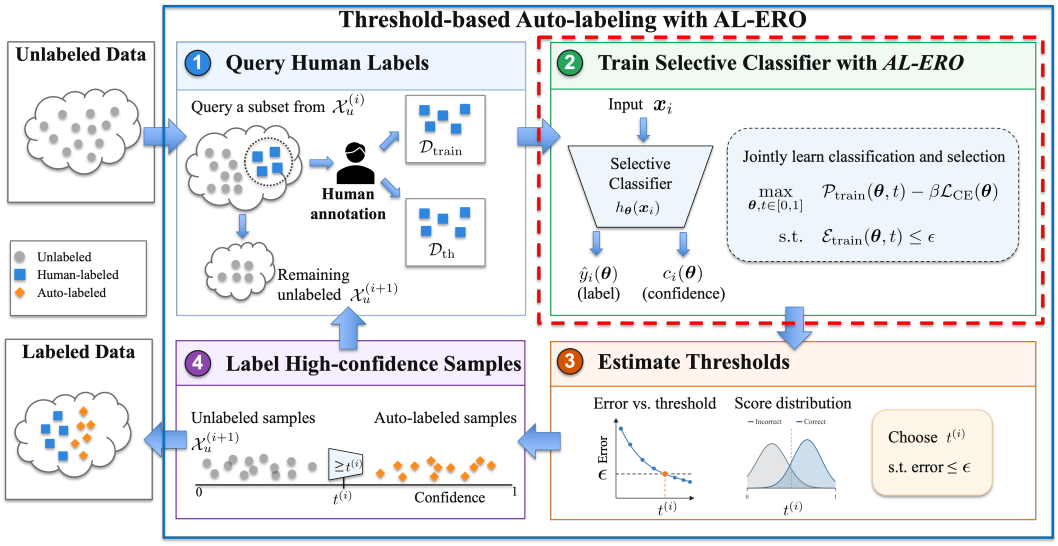


Figure 1: The pipeline for threshold-based auto-labeling with AL-ERO. (1) Query human labels from the unlabeled pool to form the training and threshold-estimation sets. (2) Train the selective classifier jointly for classification and selection with the AL-ERO method, producing both a predicted label and a confidence score. (3) Estimate a deployment threshold from the error-coverage tradeoff on the threshold-estimation set. (4) Auto-label samples whose confidence exceeds the threshold, and repeat on the remaining pool.

2 Background and Related Work

TBAL and Colander The Threshold-Based Auto-Labeling (TBAL) framework (Vishwakarma et al., 2023) formulates the auto-labeling problem as a multi-round selective classification task that aims to achieve high coverage—the fraction of samples that are automatically labeled—while keeping the auto-labeling error rate below the error rate ϵ. TBAL proceeds iteratively to assign class labels to samples in the unlabeled data pool $\mathcal{X}_u$. In each round, a subset of unlabeled samples is queried for human annotation to obtain the ground-truth label y_i, and a classifier $h_{\boldsymbol{\theta}}$ is trained on the current

human-labeled training set $\mathcal{D}_{\text{train}}$ to provide a confidence score in $[0,1]^K$ over K classes and give prediction $\hat{y}_i = \text{argmax}_k\, h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)[k]$. Then a threshold t is estimated from a held-out annotated dataset D_{th}. Each sample $\boldsymbol{x}_i$ with $h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)[\hat{y}_i] > t$ is automatically assigned the classifier's predicted label $\hat{y}_i$, while the remaining samples proceed to the next round.

The effectiveness of TBAL depends critically on how well confidence scores distinguish correct from incorrect classifier predictions. Colander (Vishwakarma et al., 2024a) learns a separate selector tailored to the selection objective. In each TBAL iteration, the classifier $h_{\boldsymbol{\theta}}$ is held fixed after the training stage, and a separate selector $g_{\boldsymbol{\phi}}$ and a class-specific threshold $\boldsymbol{t}$ is trained on a labeled selector-training set $\mathcal{D}_{\text{se}}$ to refine the confidence scores for better selection performance.

Specifically, let $\mathcal{I}_{\text{se}} = \{1, \ldots, |\mathcal{D}_{\text{se}}|\}$ denote the index set of $\mathcal{D}_{\text{se}}$ and $\mathcal{I}_{\text{se}}^N = \{i \in \mathcal{I}_{\text{se}} \mid \hat{y}_i \neq y_i\}$ be the index set of incorrect predictions. Let $g_{\boldsymbol{\phi}}(h_{\boldsymbol{\theta}}(\boldsymbol{x}_i))$ denote the selector that takes extracted features as inputs. The empirical TBAL objective in Colander can be written as:

$$\max_{\boldsymbol{\phi},\, \boldsymbol{t}\in[0,1]^K} \mathcal{P}_{\text{se}}(g_{\boldsymbol{\phi}}, \boldsymbol{t} \mid h_{\boldsymbol{\theta}}) \quad \text{s.t.} \quad \mathcal{E}_{\text{se}}(g_{\boldsymbol{\phi}}, \boldsymbol{t} \mid h_{\boldsymbol{\theta}}) \leq \epsilon, \tag{2.1}$$

where the coverage and auto-labeling error are

$$\begin{aligned} \mathcal{P}_{\text{se}}(g_{\boldsymbol{\phi}}, \boldsymbol{t} \mid h_{\boldsymbol{\theta}}) &:= \tfrac{1}{|\mathcal{D}_{\text{se}}|} \sum\nolimits_{i\in\mathcal{I}_{\text{se}}} \mathbb{1}\{g_{\boldsymbol{\phi}}(h_{\boldsymbol{\theta}}(\boldsymbol{x}_i))[\hat{y}_i] > \boldsymbol{t}[\hat{y}_i]\}\,, \\ \mathcal{E}_{\text{se}}(g_{\boldsymbol{\phi}}, \boldsymbol{t} \mid h_{\boldsymbol{\theta}}) &:= \frac{\sum_{i\in\mathcal{I}_{\text{se}}^N} \mathbb{1}\{g_{\boldsymbol{\phi}}(h_{\boldsymbol{\theta}}(\boldsymbol{x}_i))[\hat{y}_i] > \boldsymbol{t}[\hat{y}_i]\}}{\sum_{i\in\mathcal{I}_{\text{se}}} \mathbb{1}\{g_{\boldsymbol{\phi}}(h_{\boldsymbol{\theta}}(\boldsymbol{x}_i))[\hat{y}_i] > \boldsymbol{t}[\hat{y}_i]\}}. \end{aligned} \tag{2.2}$$

The limitations of Colander have two folds. **Surrogate formulation:** In order to enable gradient-based optimization, Colander replaces the indicator functions in Eq. (2.1) with sigmoid surrogates $\sigma(\alpha, z) := 1/(1+\exp(-\alpha z))$. This relaxation by Colander introduces an approximation gap to the original indicator-based coverage and error, whose magnitude depends on the sharpness parameter α. As illustrated in Fig. 2, a small α yields a smooth but loose approximation, potentially distorting the original selection objective and leading to suboptimal selection result. Increasing α reduces this approximation gap, but also makes the sigmoid increasingly steep, causing its gradients to vanish away from a narrow region around the decision boundary and thereby making optimization more difficult. Colander also converts the constraint into a penalty term added to the objective, which creates even more significant gap between the relaxed formulation and the original problem. **Separated classification and selection:** In Colander, the classifier is trained independently of the selection objective, so its confidence scores are not explicitly optimized for the desired error-coverage tradeoff. The classifier's predictions and original confidence scores are then held fixed, restricting subsequent selector optimization separate from classifier training and potentially limiting the coverage achievable under the prescribed error constraint. Moreover, under a fixed human-label budget, reserving a separate selector-training set reduces the labeled data available for both training of the classifier and selector, leading to compromised classification and selection performance and also preventing classification and selection from being optimized jointly, even though the two objectives are closely coupled in TBAL.

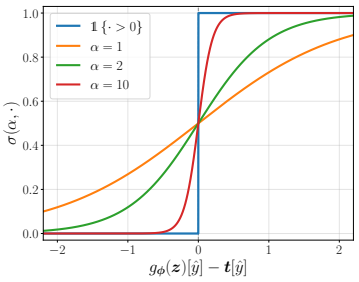


Figure 2: Sigmoid surrogates approximate the indicator but assign fractional weights near the decision boundary, inducing a mismatch with the indicator-based objective.

Selective Classification The limitations of the separate selector motivate jointly optimizing the classification and selection. Some of the existing methods improve confidence ranking for better selection by modifying the training objective or optimization procedure (Pugnana et al., 2024; Luo et al., 2026; Liang et al., 2024). Correctness Ranking Loss (CRL) (Moon et al., 2020) introduces a ranking regularizer that encourages confidence scores to follow the relative correctness of predictions. FMFP (Zhu et al., 2022) improves failure prediction by seeking flat minima via sharpness-aware minimization. AURC optimization (Zhou et al., 2024) instead directly targets the error-coverage tradeoff used in selective prediction. None of these methods, however, directly optimize the objective in auto-labeling of maximizing coverage under an explicit error constraint.

Exact Reformulation of Indicator Functions. The approximation gap introduced by smooth surrogates is a common challenge in optimization problems involving indicator functions. Existing work typically replaces indicator-based terms with surrogate approximations to enable gradient-based optimization (Qin et al., 2010; Chapelle & Wu, 2010; Bénédict et al., 2021; Patel et al., 2022; Eban et al., 2017; Vishwakarma et al., 2025; 2024b; Yamada et al., 2026). For constrained problems, related approaches use proxy constraints or implicit formulations to handle indicator-based objectives and constraints (Cotter et al., 2019; Kumar et al., 2021). Although these methods make the optimization tractable, they still rely on surrogate or proxy quantities. ERO (Peng et al., 2025) instead introduces an exact reformulation of the indicator functions that preserves optimality and solves the resulting constrained problem using an exact penalty method (Han & Mangasarian, 1979; Di Pillo, 1994).

3 METHOD

3.1 DIRECT CLASSIFIER OPTIMIZATION FOR AUTO-LABELING

As discussed in Sec. 2, Colander's separation of classification and selection prevents the two components from being optimized jointly, while also splitting the limited labeled data budget between classifier training and separate selector training. These limitations motivate training a selective classifier jointly for classification and confidence-based selection: learning accurate predictions while optimizing their confidence scores to maximize coverage under the prescribed error constraint.

To instantiate this idea, we consider a foundation-model setting in which a pretrained model provides strong transferable representations that are kept fixed, while only a lightweight task-specific classifier is trained on top (Radford et al., 2021). With such representations, it is natural to keep the feature extractor fixed and focus downstream training on the classifier head. In our TBAL formulation, selection is performed by thresholding the confidence scores produced by this classifier. Consequently, updating the classifier parameters directly affects both its predictions and the confidence scores that determine selection. We therefore optimize the linear classifier jointly for classification and confidence-based selection, while keeping the pretrained representation fixed.

Specifically, let $\mathcal{I} = \{1, \dots, |\mathcal{D}_{\text{train}}|\}$ denote the index set of $\mathcal{D}_{\text{train}}$, and let $h_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \in [0,1]^K$ denote the classifier output over K classes. For simplicity, we define the predicted label $\hat{y}_i(\boldsymbol{\theta}) = \arg\max_k h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)[k]$ and confidence score given by the classifier $c_i(\boldsymbol{\theta}) = \max_k h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)[k]$, and let $q_i(\boldsymbol{\theta}) = \mathbb{1}\{\hat{y}_i(\boldsymbol{\theta}) \neq y_i\}$ denote the indicator of incorrect prediction. Since both the selection and classification performance matter for auto-labeling objective, we jointly optimize the error-coverage tradeoff with the standard classification objective $\mathcal{L}_{\text{CE}}(\boldsymbol{\theta}) = -\frac{1}{|\mathcal{D}_{\text{train}}|}\sum_{i\in\mathcal{I}} \log h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)[y_i]$ in each round. The cross-entropy term provides direct supervision for class prediction, while the error-coverage tradeoff shapes the confidence scores toward the desired error-coverage tradeoff. We then jointly optimize:

$$\max_{\boldsymbol{\theta},\, t\in[0,1]} \quad \mathcal{P}_{\text{train}}(\boldsymbol{\theta}, t) - \beta\, \mathcal{L}_{\text{CE}}(\boldsymbol{\theta}) \quad \text{s.t.} \quad \mathcal{E}_{\text{train}}(\boldsymbol{\theta}, t) \leq \epsilon \tag{3.1}$$

where $\beta > 0$ controls the strength of the classification objective relative to coverage, and the empirical coverage and auto-labeling error on the training set are

$$\mathcal{P}_{\text{train}}(\boldsymbol{\theta}, t) := \frac{1}{|\mathcal{D}_{\text{train}}|}\sum_{i\in\mathcal{I}} \mathbb{1}\{c_i(\boldsymbol{\theta}) > t\}\,, \quad \mathcal{E}_{\text{train}}(\boldsymbol{\theta}, t) := \frac{\sum_{i\in\mathcal{I}} q_i(\boldsymbol{\theta})\mathbb{1}\{c_i(\boldsymbol{\theta}) > t\}}{\sum_{i\in\mathcal{I}} \mathbb{1}\{c_i(\boldsymbol{\theta}) > t\}}. \tag{3.2}$$

Unlike Colander's separation strategy in Eq. (2.1), Eq. (3.1) optimizes the classifier $h_{\boldsymbol{\theta}}$ jointly for classification and selection, without requiring an additional selector-training set $\mathcal{D}_{\text{se}}$ or selector. Since $\hat{y}_i(\boldsymbol{\theta})$, $c_i(\boldsymbol{\theta})$, and $q_i(\boldsymbol{\theta})$ all depend on $\boldsymbol{\theta}$, optimizing Eq. (3.1) jointly shapes the classifier's predictions and confidence toward better auto-labeling performance. We also used a shared threshold t across all classes instead of a different threshold for each class. This choice is consistent with the population-level error constraint considered here and avoids estimating separate error rates from potentially small class-specific subsets.

Directly solving Eq. (3.1) remains challenging because both $q_i(\boldsymbol{\theta})$ and $\mathbb{1}\{c_i(\boldsymbol{\theta}) > t\}$ are piecewise constant and have zero gradients almost everywhere. We derive different treatments for the two indicators. The correctness indicator is retained $q_i(\boldsymbol{\theta})$ using a lagged optimiation technique: its value is evaluated from the current classifier and treated as fixed during the subsequent parameter

update, then refreshed as the classifier changes. This is reasonable because the predicted class, and hence $q_i(\boldsymbol{\theta})$, remains locally unchanged away from decision boundaries, so the lagged value is exact within the current prediction region; similar iterate-dependent approximations are common in successive approximation and iteratively reweighted optimization (Razaviyayn et al., 2013; Ochs et al., 2015). In contrast, $\mathbb{1}\{c_i(\boldsymbol{\theta}) > t\}$ directly determines the selection decision through which coverage is optimized, so freezing it would prevent the optimization from changing which samples are selected. We therefore keep the correctness indicator in this lagged form and apply an exact reformulation to the confidence-selection indicator, which is stated in the next section.

3.2 Exact Reformulation of the Selection Indicators

To remove the discrepancy gap induced by surrogate function, Peng et al. (2025) introduces an *exact reformulation* for direct metric optimization involved with indicator function. Following this idea, an auxiliary variable $s_i \in [0, 1]$ is used to replace the indicator function $\mathbb{1}\{c_i(\boldsymbol{\theta}) > t\}$ so that

$$s_i = \mathbb{1}\{c_i(\boldsymbol{\theta}) > t\} \;\; \forall i \in \mathcal{I} \quad \Longleftrightarrow \quad s_i - \mathbb{1}\{c_i(\boldsymbol{\theta}) > t\} = 0 \;\; \forall i \in \mathcal{I}, \tag{3.3}$$

resulting in the following equivalent problem,

$$\begin{aligned} \max_{\boldsymbol{\theta}, t \in [0,1], s_i \in [0,1]} \quad & \frac{1}{|\,\mathcal{D}_{\text{train}}\,|} \sum\nolimits_{i \in \mathcal{I}} s_i - \beta\, \mathcal{L}_{\text{CE}}(\boldsymbol{\theta}) \\ \text{s.t.} \quad & \frac{\sum_{i \in \mathcal{I}} q_i(\boldsymbol{\theta}) s_i}{\sum_{i \in \mathcal{I}} s_i} \le \epsilon; \quad s_i - \mathbb{1}\{c_i(\boldsymbol{\theta}) > t\} = 0 \;\; \forall i \in \mathcal{I}. \end{aligned} \tag{3.4}$$

The two problems Eq. (3.1) and Eq. (3.4) are equivalent in the sense that any $(\boldsymbol{\theta}^*, t^*, \boldsymbol{s}^*)$ is a global solution to Eq. (3.4) if and only if $(\boldsymbol{\theta}^*, t^*)$ is a global solution to Eq. (3.1). This is stated as Lem. A.1 and is proven in Sec. A.2.

To further remove the indicator function in Eq. (3.4), we introduce the following lemma that removes the indicator function while preserving equivalence to the original optimization problem.

Lemma 3.1. *For any fixed $t \in \mathbb{R}$ and any $a \in \mathbb{R}$ with $a \neq t$,*

$$s - \mathbb{1}\{a > t\} = 0 \quad \Longleftrightarrow \quad s + [s + a - t - 1]_+ - [s + a - t]_+ = 0, \tag{3.5}$$

where $[\,\cdot\,]_+ := \max(\,\cdot\,, 0)$. Whenever either side holds, $s \in \{0, 1\} \subset [0, 1]$.

The proof is shown in Sec. A.1. Lem. 3.1 replaces the zero-gradient indicator equality with a piecewise-linear equality. Three points are worth highlighting. First, the equivalence is *exact* on the non-singular regime $a \neq t$, so no approximation gap is introduced. Second, although the auxiliary variable s is relaxed to the continuous interval $[0, 1]$, the reformulated equality constraint automatically forces $s \in \{0, 1\}$ at any feasible point, recovering the original binary semantics of $\mathbb{1}\{a > t\}$. Third, in contrast to $\mathbb{1}\{a > t\}$ whose gradient with respect to a and t vanishes almost everywhere, the reformulated constraint has non-zero gradients in both a and t, allowing the score $c_i(\boldsymbol{\theta})$ and the threshold t to be updated jointly with s_i via gradient-based methods. These facts are shown in Fig. 3. Note that we present the value and gradient of the absolute value of the functions, as that will be used in our exact penalty updates. From the gradient field of the reformulated version we can see that s is pushed to either end when the state is confident enough to accept or reject the sample (when $s + a < t$ or $s + a > 1 + t$) and a is pushed to t when the state is not confident enough. (when $t < s + a < 1 + t$).

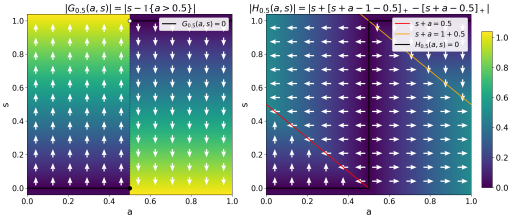


Figure 3: $G_t(a, s)$ is the indicator function and $H_t(a, s)$ is the reformulated version. The reformulated version does not change the solution and has none-zero gradient of a in the middle region.

Applying Lem. 3.1 to each i with $a = c_i(\boldsymbol{\theta})$ turns Eq. (3.4) into a fully differentiable reformulation of the original auto-labeling problem. The optimization problem can be reformulated as

$$\begin{aligned} \max_{\boldsymbol{\theta}, t \in [0,1], s_i \in [0,1]} \quad & \tfrac{1}{|\mathcal{D}_{\text{train}}|} \sum\nolimits_{i \in \mathcal{I}} s_i - \beta\, \mathcal{L}_{\text{CE}}(\boldsymbol{\theta}) \\ \text{s.t.} \quad & \sum\nolimits_{i \in \mathcal{I}} q_i(\boldsymbol{\theta}) s_i - \epsilon \sum\nolimits_{i \in \mathcal{I}} s_i \le 0, \\ & s_i + [s_i + c_i(\boldsymbol{\theta}) - t - 1]_+ - [s_i + c_i(\boldsymbol{\theta}) - t]_+ = 0, \; \forall i \in \mathcal{I}. \end{aligned} \tag{3.6}$$

Note that the reformulation provided by Lem. 3.1 requires the removal of $c_i(\boldsymbol{\theta}) = t$, leading to the following definition:

Definition 3.2 (Non-singular $(\boldsymbol{\theta}, t)$)**.** *A pair $(\boldsymbol{\theta}, t)$ is said to be **non-singular** over the training set $\{(\boldsymbol{x}_i, y_i)\}_{i\in\mathcal{I}}$ if $c_i(\boldsymbol{\theta}) \neq t, \forall i \in \mathcal{I}$. A triad $(\boldsymbol{\theta}, t, \boldsymbol{s})$ is said to be **non-singular** if $(\boldsymbol{\theta}, t)$ is non-singular.*

Now to justify the legitimacy of this reformulation, we fully establish the equivalence between Eq. (3.1) and Eq. (3.6) as Thm. 3.3 stated below:

Theorem 3.3. *Any non-singular $(\boldsymbol{\theta}^*, t^*, \boldsymbol{s}^*)$ is a global solution to Eq.* (3.6) *if and only if $(\boldsymbol{\theta}^*, t^*)$ is a global solution to Eq.* (3.1)

where the non-singular condition simply rules out the points where some sample lies exactly on the threshold. The proof is shown in Sec. A.3. Thm. 3.3 shows that solving the reformulated problem does not alter the global optimum and allows us to recover a global solution to the original problem and thus we can directly deal with Eq. (3.6) which allows gradient based optimization.

3.3 Regularization and Optimization

The exact reformulation in Eq. (3.6) relies on the non-singular condition $c_i(\boldsymbol{\theta}) \neq t$, but as we have discussed in Sec. 3.2, the indicator constraint term does not automatically avoid singularity. To discourage the optimization from approaching singular points, following Peng et al. (2025), we introduce the regularization term

$$\psi(\boldsymbol{\theta}, \boldsymbol{s}) = \tfrac{1}{|\mathcal{D}_{\text{train}}|} \sum\nolimits_{i\in\mathcal{I}} \left(s_i \log c_i(\boldsymbol{\theta}) + (1 - s_i) \log\left(1 - c_i(\boldsymbol{\theta})\right)\right). \tag{3.7}$$

The term ψ can be viewed as the negative binary cross-entropy between the auxiliary selection variable s_i and the confidence score $c_i(\boldsymbol{\theta})$. Maximizing ψ encourages the two quantities to align: as s_i approaches its binary semantics, selected samples ($s_i \approx 1$) are encouraged to have higher confidence, whereas unselected samples ($s_i \approx 0$) are encouraged to have lower confidence, consistent with the selection relation $s_i = \mathbb{1}\{c_i(\boldsymbol{\theta}) > t\}$ and effectively pushes the confidence scores across the threshold to either end of [0,1].

Adding this regularization to Eq. (3.6) gives

$$\begin{aligned} \max_{\boldsymbol{\theta},\, t\in[0,1],\, s_i\in[0,1]} \quad & \phi_{\text{obj}}(\boldsymbol{s}) - \beta\, \mathcal{L}_{\text{CE}}(\boldsymbol{\theta}) + \gamma\, \psi(\boldsymbol{\theta}, \boldsymbol{s}) \\ \text{s.t.} \quad & \phi_{\text{con}}(\boldsymbol{\theta}, \boldsymbol{s}) \leq 0, \quad \eta(\boldsymbol{\theta}, t, \boldsymbol{s}) = \mathbf{0}, \end{aligned} \tag{3.8}$$

where

$$\begin{aligned} &\phi_{\text{obj}}(\boldsymbol{s}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum\nolimits_{i\in\mathcal{I}} s_i, \quad \phi_{\text{con}}(\boldsymbol{\theta}, \boldsymbol{s}) = \sum\nolimits_{i\in\mathcal{I}} q_i(\boldsymbol{\theta}) s_i - \epsilon \sum\nolimits_{i\in\mathcal{I}} s_i, \\ &\big(\eta(\boldsymbol{\theta}, t, \boldsymbol{s})\big)_i = s_i + \big[s_i + c_i(\boldsymbol{\theta}) - t - 1\big]_+ - \big[s_i + c_i(\boldsymbol{\theta}) - t\big]_+, \end{aligned} \tag{3.9}$$

and $\gamma > 0$ controls the strength of the non-singularity regularization.

We solve Eq. (3.8) using an exact penalty method (Han & Mangasarian, 1979), assigning the error constraint $\phi_{\text{con}} \leq 0$ and each of the $|\mathcal{I}|$ indicator constraints $\eta_i = 0$ its own penalty weight. The resulting penalty objective is the AL-ERO training objective:

$$\begin{aligned} \mathcal{F}_{\text{AL-ERO}}(\boldsymbol{\theta}, t, \boldsymbol{s}; \mu, \boldsymbol{\lambda}) = & -\phi_{\text{obj}}(\boldsymbol{s}) + \beta\, \mathcal{L}_{\text{CE}}(\boldsymbol{\theta}) - \gamma\, \psi(\boldsymbol{\theta}, \boldsymbol{s}) \\ & + \mu\big[\phi_{\text{con}}(\boldsymbol{\theta}, \boldsymbol{s})\big]_+ + \sum\nolimits_{i\in\mathcal{I}} \lambda_i \big|\big(\eta(\boldsymbol{\theta}, t, \boldsymbol{s})\big)_i\big|, \end{aligned} \tag{3.10}$$

where $\mu > 0$ penalizes the error constraint, $\boldsymbol{\lambda} = (\lambda_i)_{i\in\mathcal{I}}$, with $\lambda_i > 0$ for all $i \in \mathcal{I}$, penalizes the indicator constraints, one weight per sample, and $\beta > 0$ is the classification weight carried over from Eq. (3.1). For sufficiently large finite penalty parameters $\mu, \boldsymbol{\lambda}$, the exact penalty formulation promotes satisfaction of the constraints in Eq. (3.8) while allowing the problem to be optimized with standard gradient-based methods.

At the k-th outer iteration, given the current penalty parameters $\mu^{(k)}$ and $\boldsymbol{\lambda}^{(k)}$, we approximately minimize the penalized objective with respect to the classifier, threshold, and auxiliary selection variables:

Algorithm 1 Solving AL-ERO with Exact Penalty Method

Input: Training Dataset $\mathcal{D}_{\text{train}}$, target error ϵ, initial penalty parameters $\mu^{(0)}$ and $\boldsymbol{\lambda}^{(0)}$, initial point $(\boldsymbol{\theta}^{(0)}, t^{(0)}, \boldsymbol{s}^{(0)})$, penalty multiplier ρ, maximum number of outer iterations M, regularization parameter γ, classification weight β.

1: $k := 0$
2: **while** $k \leq M$ **do**
3: Apply Adam, initialized at $(\boldsymbol{\theta}^{(k)}, t^{(k)}, s^{(k)})$, to solve for

$$(\boldsymbol{\theta}^{(k+1)}, t^{(k+1)}, s^{(k+1)}) = \underset{\boldsymbol{\theta}, t\in[0,1], s_i\in[0,1]}{\operatorname{argmin}} \mathcal{F}_{\text{AL-ERO}}(\boldsymbol{\theta}, t, \boldsymbol{s}, \mu^{(k)}, \boldsymbol{\lambda}^{(k)})$$

4: $\mu^{(k+1)} \leftarrow \mu^{(k)} \cdot \rho$
5: **for** $i \in \mathcal{I}$ **do**
6: $\lambda_i^{(k+1)} \leftarrow \lambda_i^{(k)} \cdot \rho$
7: **end for**
8: $k \leftarrow k + 1$
9: **end while**

$$(\boldsymbol{\theta}^{(k+1)}, t^{(k+1)}, \boldsymbol{s}^{(k+1)}) = \underset{\boldsymbol{\theta},\, t\in[0,1],\, s_i\in[0,1]}{\operatorname{argmin}} \mathcal{F}_{\text{AL-ERO}}(\boldsymbol{\theta}, t, \boldsymbol{s}; \mu^{(k)}, \boldsymbol{\lambda}^{(k)}). \tag{3.11}$$

After each outer iteration, the penalty parameters are updated according to the corresponding constraint violations. The complete optimization procedure is summarized in Algorithm 1. The full TBAL loop, combining AL-ERO with threshold estimation and pool updates, is described in Algorithm 2 in Appendix B.

4 Experiments

We begin by describing the datasets and experimental setting. **Datasets:** We evaluate AL-ERO on four datasets spanning image and text classification. For image classification, we use OrganACSMNIST and PathMNIST (Yang et al., 2023). For text classification, we use 20 Newsgroups (Mitchell, 1997) and PubMed 200k RCT (Dernoncourt & Lee, 2017). We extract 512-dimensional image representations using CLIP ViT-B/32 (Radford et al., 2021) and 1024-dimensional text representations using Qwen3-Embedding-0.6B (Zhang et al., 2025b). More dataset details are in Sec. C.1.

Competing Methods. We compare *AL-ERO* against several classifier optimization baselines, including Cross-Entropy (CE), Squentropy (Hui et al., 2023), CRL (Moon et al., 2020), FMFP (Zhu et al., 2022), MDCA (Hebbalaguppe et al., 2022) and AURC (Zhou et al., 2024). We additionally compare with the post-classifier method Colander (Vishwakarma et al., 2024a). For Colander, we parameterize the selector with a linear head and divide the same human-labeled budget equally between classifier and selector training, so that Colander and the direct classifier optimization methods use the same total number of human labels. To isolate the effect of exact reformulation, we further introduce *AL-Surrogate*, a direct classifier optimization baseline that targets the same classification–selection objective as AL-ERO but replaces the selection indicators with the sigmoid approximation used by Colander. Further details for all competing methods are provided in Sec. C.2.

Implementation Details. For all datasets, we keep the pretrained feature extractor fixed and train a linear classifier on the extracted representations. All direct classifier optimization methods use the same classifier architecture, data split, and human-label budget. Every method except the cross-entropy baseline itself is initialized from a shared CE warm-up on $\mathcal{D}_{\text{train}}$ and then optimized using their respective objectives. For AL-Surrogate and AL-ERO, we fix threshold $t = 0.5$ without optimizing it during training time, as the bias of the linear layer can absorb the corresponding threshold shift. Hyperparameters are selected from predefined search grids using a separate labeled set $\mathcal{D}_{\text{hyp}}$, disjoint from $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{th}}$; the search spaces are reported in Sec. C.2. After the classifier training, the deployment threshold is estimated on D_{th} such that the error constraint is met on D_{th}. We evaluate all methods at target auto-labeling error rates $\epsilon = 5\%, 10\%$.

4.1 Results and Discussion

Table 1: Auto-labeling results under error constraints $\epsilon = 5\%$ (top) and $\epsilon = 10\%$ (bottom). Best feasible candidate is **bold** and second best is underlined.

Method	OrganACSMNIST		PathMNIST		PubMed 200k RCT		20 Newsgroups	
	Cov	Err	Cov	Err	Cov	Err	Cov	Err
CE	48.05 ± 0.16	3.56 ± 0.02	67.71 ± 0.18	4.03 ± 0.05	15.70 ± 1.16	3.32 ± 0.10	19.34 ± 0.05	4.08 ± 0.03
Squentropy	50.54 ± 0.09	3.77 ± 0.04	70.90 ± 0.09	4.22 ± 0.06	28.44 ± 0.68	4.28 ± 0.08	19.40 ± 0.02	4.17 ± 0.04
CRL	51.46 ± 0.35	3.52 ± 0.10	71.22 ± 0.06	4.03 ± 0.02	21.83 ± 0.74	3.42 ± 0.07	20.57 ± 0.06	4.27 ± 0.07
FMFP	50.72 ± 0.29	3.60 ± 0.09	70.64 ± 0.08	4.08 ± 0.02	21.94 ± 0.78	3.44 ± 0.05	19.55 ± 0.08	4.20 ± 0.03
MDCA	50.78 ± 0.38	3.60 ± 0.12	71.03 ± 0.12	4.09 ± 0.04	19.37 ± 0.21	3.32 ± 0.02	20.10 ± 0.19	4.27 ± 0.07
AURC	52.62 ± 0.58	3.69 ± 0.07	72.13 ± 0.13	4.15 ± 0.06	31.61 ± 2.18	4.01 ± 0.20	21.65 ± 0.08	3.97 ± 0.07
Colander	38.31 ± 1.94	3.62 ± 0.32	60.00 ± 0.76	4.18 ± 0.17	15.32 ± 0.78	3.97 ± 0.12	16.02 ± 0.82	4.35 ± 0.32
AL-Surrogate	52.81 ± 0.13	3.83 ± 0.04	71.51 ± 0.13	4.06 ± 0.06	24.07 ± 0.43	3.59 ± 0.04	22.04 ± 0.06	4.34 ± 0.02
AL-ERO	$\mathbf{53.59 \pm 0.18}$	3.62 ± 0.05	$\mathbf{73.45 \pm 0.09}$	3.94 ± 0.04	$\mathbf{37.22 \pm 2.16}$	4.39 ± 0.25	$\mathbf{23.13 \pm 0.42}$	4.89 ± 0.20

Method	OrganACSMNIST		PathMNIST		PubMed 200k RCT		20 Newsgroups	
	Cov	Err	Cov	Err	Cov	Err	Cov	Err
CE	61.15 ± 0.17	7.85 ± 0.10	79.46 ± 0.05	8.61 ± 0.04	55.60 ± 0.33	8.59 ± 0.05	29.10 ± 0.08	9.16 ± 0.06
Squentropy	63.77 ± 0.07	8.16 ± 0.04	80.96 ± 0.11	8.37 ± 0.06	59.48 ± 0.20	8.59 ± 0.03	29.94 ± 0.05	9.22 ± 0.03
CRL	64.47 ± 0.18	7.92 ± 0.12	81.92 ± 0.18	8.59 ± 0.12	59.55 ± 0.13	8.58 ± 0.14	30.57 ± 0.05	9.52 ± 0.10
FMFP	64.20 ± 0.17	8.02 ± 0.05	81.26 ± 0.15	8.53 ± 0.11	59.59 ± 0.32	8.55 ± 0.08	30.40 ± 0.03	9.42 ± 0.02
MDCA	64.33 ± 0.23	7.99 ± 0.08	81.48 ± 0.14	8.54 ± 0.08	59.35 ± 0.35	8.66 ± 0.04	30.39 ± 0.02	9.43 ± 0.05
AURC	65.26 ± 0.12	7.96 ± 0.08	82.33 ± 0.04	8.59 ± 0.05	60.96 ± 0.34	8.52 ± 0.05	30.82 ± 0.11	9.86 ± 0.08
Colander	52.26 ± 0.72	8.09 ± 0.33	68.16 ± 0.16	8.74 ± 0.10	38.80 ± 0.20	8.35 ± 0.05	23.17 ± 0.31	8.67 ± 0.20
AL-Surrogate	65.52 ± 0.23	8.26 ± 0.11	82.24 ± 0.08	8.74 ± 0.07	60.28 ± 0.35	8.55 ± 0.07	30.73 ± 0.09	9.80 ± 0.07
AL-ERO	$\mathbf{68.15 \pm 0.34}$	8.52 ± 0.12	$\mathbf{83.19 \pm 0.01}$	8.31 ± 0.05	$\mathbf{62.95 \pm 0.71}$	8.48 ± 0.08	$\mathbf{31.11 \pm 0.09}$	9.72 ± 0.10

We evaluate auto-labeling performance using coverage and error: a method is feasible if its auto-labeling error does not exceed the prescribed value ϵ, and among feasible methods, higher coverage indicates better performance. The results are reported in Tab. 1.

Overall auto-labeling performance. As shown in Tab. 1, AL-ERO achieves the highest coverage among all classifier optimization methods across the four datasets and both target error levels, while maintaining auto-labeling error below the prescribed target ϵ in every setting. The error-coverage curves in Fig. 4 further show that AL-ERO provides consistently better confidence ranking in the low-error region. Compared with AL-ERO, the competing methods improve classifier confidence from different perspectives but do not directly optimize the selection objective. CE focuses on predictive accuracy; Squentropy and MDCA primarily target calibration; CRL and FMFP encourage confidence ranking on failure prediction; and AURC optimizes performance over the overall error-coverage curve rather than at the operating point specified by ϵ. These objectives can improve confidence quality, but their optimization signals are not explicitly aligned with maximizing coverage under the prescribed error constraint. In contrast, AL-ERO directly optimizes this objective, aligning classifier training with the selection task in auto-labeling.

Comparison with surrogate-based direct optimization. AL-Surrogate, like AL-ERO, directly optimizes the classifier for both prediction and confidence-based selection, but replaces the indicator functions with sigmoid surrogates. Across all four datasets and both target error levels, AL-ERO consistently achieves higher coverage. This suggests that directly aligning classifier training with the auto-labeling objective is beneficial but not sufficient: the surrogate formulation still introduces a mismatch with the indicator-based selection objective, which can lead to suboptimal coverage.

Comparison with post-classifier method. Under the same human-label budget, AL-ERO achieves higher coverage than Colander across all datasets. Unlike Colander, AL-ERO unifies the classification and selection optimization and without dividing the training set for an additional selector. The resulting advantage is consistent with our motivation for integrating confidence optimization directly into classifier training.

4.2 Ablation Study

To better understand the effectiveness of AL-ERO, we conduct two experiments in this subsection: an ablation isolating the contribution of the classification term and further comparison with post-classifier methods. Additional experiments are provided in Secs. D.1 and D.2.

Effect of the cross-entropy term. The AL-ERO objective includes the cross-entropy term βL_{CE} to preserve the classifier's predictive performance while optimizing for the auto-labeling objective.

To evaluate its contribution, we set $\beta = 0$ while keeping all other settings unchanged, and compare the resulting coverage and error on OrganACSMNIST. We additionally report classifier accuracy at the first TBAL round to examine how removing the classification objective affects predictive performance. The results are reported in Tab. 2.

Table 2: Effect of removing the cross-entropy term ($\beta = 0$) from the AL-ERO objective on OrganACSMNIST with a fixed CLIP ViT-B/32 encoder. Coverage (Cov) and error (Err) are reported over the complete TBAL loop; classifier accuracy (Acc) is reported at the first TBAL round.

Target Error	**Method**	Cov	Err	Acc
$\epsilon = 5\%$	AL-ERO	$\mathbf{53.59 \pm 0.18}$	3.62 ± 0.05	83.86 ± 0.04
	AL-ERO ($\beta = 0$)	51.30 ± 0.35	3.91 ± 0.13	81.72 ± 0.18
$\epsilon = 10\%$	AL-ERO	$\mathbf{68.15 \pm 0.34}$	8.52 ± 0.12	84.04 ± 0.17
	AL-ERO ($\beta = 0$)	63.70 ± 0.69	8.43 ± 0.21	81.69 ± 0.06

Removing the cross-entropy term reduces coverage by 2.29 and 4.45 percentage points at $\epsilon = 5\%$ and $\epsilon = 10\%$, respectively, while error changes only slightly ($+0.29$ points at 5%; -0.09 points at 10%). At the first TBAL round, removing the cross-entropy term also reduces classifier accuracy by 2.14 and 2.35 percentage points at $\epsilon = 5\%$ and $\epsilon = 10\%$, respectively. These results indicate that the cross-entropy term helps preserve classification performance during optimization, while also yielding higher feasible coverage under the target error constraint.

Comparison with post-classifier methods. To further examine the benefit of integrating confidence optimization into classifier training, we compare AL-ERO with Colander and four additional post-classifier baselines—Dirichlet (Kull et al., 2019), Temperature scaling (Guo et al., 2017), Scaling-binning (Kumar et al., 2019), and top-label histogram binning (Gupta & Ramdas, 2021)—on OrganACSMNIST under the same human-label budget. In contrast to these post-classifier methods, which adjust confidence after the classifier has been trained, AL-ERO incorporates the coverage-error objective directly into classifier optimization. As shown in Tab. 3, AL-ERO achieves higher coverage than all post-classifier baselines while keeping the auto-labeling error below the target. These results support the benefit of integrating confidence optimization into classifier training, enabling higher auto-labeling coverage under the prescribed error constraint and human-label budget.

Table 3: Comparison of AL-ERO with post-classifier baselines on OrganACSMNIST at 5% target error.

Method	**OrganACSMNIST**	
	Cov	Err
Dirichlet	$\underline{42.78 \pm 0.32}$	3.45 ± 0.05
Temperature scaling	41.92 ± 0.75	3.40 ± 0.13
Histogram binning	40.43 ± 0.85	3.51 ± 0.08
Scaling-binning	39.13 ± 0.14	2.89 ± 0.02
Colander	38.31 ± 1.94	3.62 ± 0.32
AL-ERO	$\mathbf{53.59 \pm 0.18}$	3.62 ± 0.05

5 Conclusion and Limitations

We propose AL-ERO, a direct classifier optimization method for threshold-based auto-labeling. By incorporating coverage maximization under a prescribed error constraint into classifier training, AL-ERO jointly optimizes prediction and confidence-based selection without requiring a separate selection model. To handle the selection indicators, we introduce an exact reformulation that avoids surrogate approximations and optimize the resulting constrained problem using an exact penalty method. Experiments on four datasets spanning image and text classification show that AL-ERO consistently achieves better auto-labeling performance.

Our current study has two main limitations. First, our current optimization procedure is deterministic and operates on the full training set, which may limit its scalability to larger datasets. Developing a stochastic version of AL-ERO that can optimize the reformulated constrained objective using mini-batches is therefore an important direction for future work. Second, satisfying the training-set error constraint does not by itself guarantee error control on unseen data. Our auto-labeling procedure therefore still relies on a labeled threshold-estimation set, which consumes part of the human-label budget. Reducing or eliminating the need for an additional threshold-estimation set while retaining reliable error control is also an important direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools primarily to assist with drafting and editing the manuscript, improving clarity and readability, writing and editing part of research code, implementing parts of the experimental pipeline, and preparing and submitting experiment jobs. We also used generative AI tools for feedback on interpretation of experimental results, literature search and summarization, translation, and assistance with the presentation of some mathematical arguments. We did not use generative AI tools to generate synthetic datasets, clean or reformat datasets.

All AI-assisted work was reviewed by the authors. AI-assisted code and experimental procedures were reviewed and tested for correctness by two authors, and the resulting experimental outputs and reported results were independently checked by two authors. References and literature claims were verified against the original sources, and mathematical claims and derivations were checked by the authors. AI-generated text and suggestions were revised as needed before inclusion in the manuscript. We take responsibility for the final content of this work, including text, claims, code, experimental results, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide detailed experimental settings in the appendix, including dataset statistics and preprocessing, implementation details of the competing methods, training procedures, hyperparameter search spaces and selection protocols, and the complete threshold-based auto-labeling pipeline used for evaluation. We also report the data splits, human-label budgets, threshold-estimation procedure, and evaluation protocol used across all experiments. Anonymous source code, experiment configurations, and instructions for reproducing the reported results are also provided as supplementary material.

REFERENCES

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE, 2020.

Stephen H Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alex Ratner, Braden Hancock, Houman Alborzi, et al. Snorkel drybell: A case study in deploying weak supervision at industrial scale. In *Proceedings of the 2019 International Conference on Management of Data*, pp. 362–375, 2019.

Gabriel Bénédict, Vincent Koops, Daan Odijk, and Maarten de Rijke. Sigmoidf1: A smooth f1 score surrogate loss for multilabel classification. *arXiv preprint arXiv:2108.10566*, 2021.

Olivier Chapelle and Mingrui Wu. Gradient descent optimization of smoothed information retrieval metrics. *Information retrieval*, 13(3):216–235, 2010.

Andrew Cotter, Heinrich Jiang, and Karthik Sridharan. Two-player games for efficient non-convex constrained optimization. In *Algorithmic Learning Theory*, pp. 300–332. PMLR, 2019.

Franck Dernoncourt and Ji-Young Lee. Pubmed 200k rct: a dataset for sequential sentence classification in medical abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 308–313, 2017.

G. Di Pillo. *Exact Penalty Methods*, pp. 209–253. Springer Netherlands, Dordrecht, 1994. ISBN 978-94-009-0369-2. doi: 10.1007/978-94-009-0369-2_8. URL https://doi.org/10.1007/978-94-009-0369-2_8.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. Is gpt-3 a good data annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11173–11195, 2023.

Elad Eban, Mariano Schain, Alan Mackey, Ariel Gordon, Ryan Rifkin, and Gal Elidan. Scalable learning of non-decomposable objectives. In *Artificial Intelligence and Statistics*, pp. 832–840. PMLR, 2017.

Sedigheh Eslami, Christoph Meinel, and Gerard De Melo. Pubmedclip: How much does clip benefit visual question answering in the medical domain? In *Findings of the association for computational linguistics: EACL 2023*, pp. 1181–1193, 2023.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Chirag Gupta and Aaditya Ramdas. Top-label calibration and multiclass-to-binary reductions. *arXiv preprint arXiv:2107.08353*, 2021.

S.-P. Han and O. L. Mangasarian. Exact penalty functions in nonlinear programming. *Mathematical Programming*, 17:251–269, 1979. doi: 10.1007/BF01588250.

Ramya Hebbalaguppe, Jatin Prakash, Neelabh Madan, and Chetan Arora. A stitch in time saves nine: A train-time regularizing loss for improved neural network calibration. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16060–16069. IEEE, 2022.

Like Hui, Mikhail Belkin, and Stephen Wright. Cut your losses with squentropy. In *International Conference on Machine Learning*, pp. 14114–14131. PMLR, 2023.

Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32, 2019.

Abhishek Kumar, Harikrishna Narasimhan, and Andrew Cotter. Implicit rate-constrained optimization of non-decomposable objectives. In *International Conference on Machine Learning*, pp. 5861–5871. PMLR, 2021.

Ananya Kumar, Percy S Liang, and Tengyu Ma. Verified uncertainty calibration. *Advances in neural information processing systems*, 32, 2019.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Int. Conf. Mach. Learn.*, pp. 896. Atlanta, 2013.

Hengyue Liang, Le Peng, and Ju Sun. Selective classification under distribution shifts. *Transactions on machine learning research*, 2024:dmxMGW6J7N, 2024.

Gaoxiang Luo, Yifan Wu, Sinian Zhang, Aryan Deshwal, and Ju Sun. Aligning language models with selective prediction. *arXiv preprint arXiv:2607.03528*, 2026.

Tom Mitchell. Twenty Newsgroups. UCI Machine Learning Repository, 1997. DOI: https://doi.org/10.24432/C5C323.

Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. Confidence-aware learning for deep neural networks. In *international conference on machine learning*, pp. 7034–7044. PMLR, 2020.

Peter Ochs, Alexey Dosovitskiy, Thomas Brox, and Thomas Pock. On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision. *SIAM Journal on Imaging Sciences*, 8(1):331–372, 2015. doi: 10.1137/140971518. URL https://doi.org/10.1137/140971518.

Yash Patel, Giorgos Tolias, and Jiří Matas. Recall@ k surrogate loss with large batches and similarity mixup. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7502–7511, 2022.

Le Peng, Yash Travadi, Chuan He, Ying Cui, and Ju Sun. Exact reformulation and optimization for direct metric optimization in binary imbalanced classification. *arXiv preprint arXiv:2507.15240*, 2025.

Andrea Pugnana, Lorenzo Perini, Jesse Davis, and Salvatore Ruggieri. Deep neural network benchmarks for selective classification. *J. Data-Centric Mach. Learn. Res.*, 1:17–1, 2024.

Tao Qin, Tie-Yan Liu, and Hang Li. A general approximation framework for direct optimization of information retrieval measures. *Information retrieval*, 13(4):375–397, 2010.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB endowment. International conference on very large data bases*, pp. 269, 2017.

Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. *Advances in neural information processing systems*, 29, 2016.

Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013. doi: 10.1137/120891009. URL https://doi.org/10.1137/120891009.

Harit Vishwakarma, Heguang Lin, Frederic Sala, and Ramya Korlakai Vinayak. Promises and pitfalls of threshold-based auto-labeling. *Advances in Neural Information Processing Systems*, 36: 51955–51990, 2023.

Harit Vishwakarma, Yi Chen, Sui Jiet Tay, Satya Sai Srinath Namburi, Frederic Sala, and Ramya Korlakai Vinayak. Pearls from pebbles: Improved confidence functions for auto-labeling. *Advances in Neural Information Processing Systems*, 37:15983–16015, 2024a.

Harit Vishwakarma, Heguang Lin, and Ramya Korlakai Vinayak. Taming false positives in out-of-distribution detection with human feedback. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2024b.

Harit Vishwakarma, Yi Chen, Satya Sai Srinath Namburi Gnvv, Sui Jiet Tay, Ramya Korlakai Vinayak, and Frederic Sala. Rethinking confidence scores and thresholds in pseudolabeling-based ssl. In *Forty-second International Conference on Machine Learning*, 2025.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10687–10698, 2020.

Daisuke Yamada, Harit Vishwakarma, and Ramya Korlakai Vinayak. ASAT: Adaptive scoring and thresholding with human feedback for robust out-of-distribution detection. *Transactions on Machine Learning Research (TMLR)*, 2026.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific data*, 10(1):41, 2023.

Sheng Zhang, Yanbo Xu, Naoto Usuyama, Hanwen Xu, Jaspreet Bagga, Robert Tinn, Sam Preston, Rajesh Rao, Mu Wei, Naveen Valluri, et al. A multimodal biomedical foundation model trained from fifteen million image–text pairs. *Nejm Ai*, 2(1):AIoa2400640, 2025a.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025b.

Han Zhou, Jordy Van Landeghem, Teodora Popordanoska, and Matthew B Blaschko. A novel characterization of the population area under the risk coverage curve (aurc) and rates of finite sample estimators. *arXiv preprint arXiv:2410.15361*, 2024.

Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. Rethinking confidence calibration for failure prediction. In *European conference on computer vision*, pp. 518–536. Springer, 2022.

A THEOREMS AND PROOFS

A.1 PROOF OF LEM. 3.1

We first provide proof for Lem. 3.1

Proof.

1° If $s + a - t > 1$

$$\begin{aligned} & s + [s + a - t - 1]_+ - [s + a - t]_+ = 0 \\ \Rightarrow & s + s + a - t - 1 - (s + a - t) = 0 \\ \Rightarrow & s = 1 \\ \Rightarrow & a - t > 0 && (s + a - t > 1) \\ \Rightarrow & s - \mathbb{1}\{a > t\} = 0 \end{aligned}$$

Similarly,

$$\begin{aligned} & s - \mathbb{1}\{a > t\} = 0 \\ \Rightarrow & \mathbb{1}\{a > t\} + a - t > 0 && (s + a - t > 1) \\ \Rightarrow & a > t \wedge s = 1 \\ \Rightarrow & a - t \geq 0 \\ \Rightarrow & s + [s + a - t - 1]_+ - [s + a - t]_+ = 0 \end{aligned}$$

2° If $s + a - t < 0$

$$\begin{aligned} & s + [s + a - t - 1]_+ - [s + a - t]_+ = 0 \\ \Rightarrow & s = 0 && (s + a - t < 0) \\ \Rightarrow & a - t < 0 && (s + a - t < 0) \\ \Rightarrow & s - \mathbb{1}\{a > t\} = 0 \end{aligned}$$

Similarly,

$$\begin{aligned} & s - \mathbb{1}\{a > t\} = 0 \\ \Rightarrow & \mathbb{1}\{a > t\} + a - t < 0 && (s + a - t < 0) \\ \Rightarrow & a < t \wedge s = 0 \\ \Rightarrow & s + [s + a - t - 1]_+ - [s + a - t]_+ = 0 \end{aligned}$$

3° $0 \leq s + a - t \leq 1$
It's easy to verify that when either side holds and $a \neq t$, this situation doesn't exist:
If $s - \mathbb{1}\{a > t\} = 0$, then $s + a - t = \mathbb{1}\{a > t\} + a - t > 1$ or < 0 given $a \neq t$; if $0 \leq s + a - t \leq 1$ is true, then $s + [s + a - t - 1]_+ - [s + a - t]_+ = s + 0 - (s + a - t) = -(a - t) \neq 0$.

We've also shown that $s \in \{0, 1\}$ in each case. □

A.2 PROOF OF LEM. A.1

Next we state and prove the equivalence between Eq. (3.4) and Eq. (3.1).

Lemma A.1. *Any $(\boldsymbol{\theta}^*, t^*, \boldsymbol{s}^*)$ is a global solution to Eq.* (3.4) *if and only if $(\boldsymbol{\theta}^*, t^*)$ is a global solution to Eq.* (3.1)*.*

Proof. "⇒": Let $(\boldsymbol{\theta}^*, t^*, \boldsymbol{s}^*)$ be a global solution to Eq. (3.4). From the indicator constraint in Eq. (3.4) we know:

$$s_i^* = \mathbb{1}\{c_i(\boldsymbol{\theta}^*) > t^*\} \quad \forall i \in \mathcal{I}.$$

Replacing this identity back into the error constraint of Eq. (3.4) and we have:

$$\frac{\sum_{i\in\mathcal{I}} q_i(\boldsymbol{\theta}^*)\mathbb{1}\{c_i(\boldsymbol{\theta}^*) > t^*\}}{\sum_{i\in\mathcal{I}} \mathbb{1}\{c_i(\boldsymbol{\theta}^*) > t^*\}} \leq \epsilon$$

Suggesting that $(\boldsymbol{\theta}^*, t^*)$ is a feasible solution for Eq. (3.1).
To prove the optimality, let $(\boldsymbol{\theta}, t)$ be a feasible solution of Eq. (3.1). $(\boldsymbol{\theta}, t, \{\mathbb{1}\{c_i(\boldsymbol{\theta}) > t\}\}_{i\in\mathcal{I}})$ is then a feasible solution of Eq. (3.4). If $\mathcal{P}_{\text{train}}(\boldsymbol{\theta}, t) - \beta\mathcal{L}_{\text{CE}}(\boldsymbol{\theta}) > \mathcal{P}_{\text{train}}(\boldsymbol{\theta}^*, t^*) - \beta\mathcal{L}_{\text{CE}}(\boldsymbol{\theta}^*)$, then $(\boldsymbol{\theta}, t, \{\mathbb{1}\{c_i(\boldsymbol{\theta}) > t\}\}_{i\in\mathcal{I}})$ as a solution of Eq. (3.4) conflicts the optimality of $(\boldsymbol{\theta}^*, t^*, \boldsymbol{s}^*)$. Therefore $\mathcal{P}_{\text{train}}(\boldsymbol{\theta}, t) - \beta\mathcal{L}_{\text{CE}}(\boldsymbol{\theta}) \leq \mathcal{P}_{\text{train}}(\boldsymbol{\theta}^*, t^*) - \beta\mathcal{L}_{\text{CE}}(\boldsymbol{\theta}^*)$ and thus $(\boldsymbol{\theta}^*, t^*)$ is a global solution to Eq. (3.1).

"$\Leftarrow$": Let $(\boldsymbol{\theta}^*, t^*)$ be a global solution to Eq. (3.1). Similarly, $(\boldsymbol{\theta}^*, t^*, \{\mathbb{1}\{c_i(\boldsymbol{\theta}^*) > t^*\}\}_{i\in\mathcal{I}})$ is a feasible solution to Eq. (3.4). For any feasible solution $(\boldsymbol{\theta}, t, \boldsymbol{s})$ to Eq. (3.4), we've shown that $(\boldsymbol{\theta}, t)$ is feasible to Eq. (3.1). If $\mathcal{P}_{\text{train}}(\boldsymbol{\theta}, t) - \beta\mathcal{L}_{\text{CE}}(\boldsymbol{\theta}) > \mathcal{P}_{\text{train}}(\boldsymbol{\theta}^*, t^*) - \beta\mathcal{L}_{\text{CE}}(\boldsymbol{\theta}^*)$, then $(\boldsymbol{\theta}, t)$ as a solution to Eq. (3.1) conflicts the optimality assumption of $(\boldsymbol{\theta}^*, t^*)$ to Eq. (3.1). Therefore $\mathcal{P}_{\text{train}}(\boldsymbol{\theta}, t) - \beta\mathcal{L}_{\text{CE}}(\boldsymbol{\theta}) \leq \mathcal{P}_{\text{train}}(\boldsymbol{\theta}^*, t^*) - \beta\mathcal{L}_{\text{CE}}(\boldsymbol{\theta}^*)$ and thus $(\boldsymbol{\theta}^*, t^*, \{\mathbb{1}\{c_i(\boldsymbol{\theta}^*) > t^*\}\}_{i\in\mathcal{I}})$ is a global solution to Eq. (3.4). □

A.3 Proof of Thm. 3.3

Now we are ready to prove our main theorem Thm. 3.3:

Proof. We only need to show the equivalence between Eq. (3.4) and Eq. (3.6). By applying Lem. 3.1 we have:

$$\begin{aligned}
&(\boldsymbol{\theta}, t, \boldsymbol{s}) \text{ is a non-singular feasible solution to Eq. (3.4)} \\
\Leftrightarrow &s_i - \mathbb{1}\{c_i(\boldsymbol{\theta}) > t\} = 0,\ \forall i \in \mathcal{I} \ \wedge\ \frac{\sum_{i\in\mathcal{I}} q_i(\boldsymbol{\theta}) s_i}{\sum_{i\in\mathcal{I}} s_i} \leq \epsilon \ \wedge\ (\boldsymbol{\theta}, t, \boldsymbol{s}) \text{ non-singular} \\
\Leftrightarrow &s_i + [s_i + c_i(\boldsymbol{\theta}) - t - 1]_+ - [s_i + c_i(\boldsymbol{\theta}) - t]_+ = 0,\ \forall i \in \mathcal{I}. \\
&\wedge\ \frac{\sum_{i\in\mathcal{I}} q_i(\boldsymbol{\theta}) s_i}{\sum_{i\in\mathcal{I}} s_i} \leq \epsilon \ \wedge\ (\boldsymbol{\theta}, t, \boldsymbol{s}) \text{ non-singular} \\
\Leftrightarrow &(\boldsymbol{\theta}, t, \boldsymbol{s}) \text{ is a non-singular feasible solution to Eq. (3.6)}
\end{aligned}$$

Thus the non-singular feasible set of Eq. (3.4) and Eq. (3.6) are identical. Note that the objective of Eq. (3.4) and Eq. (3.6) are also identical, thus a non-singular $(\boldsymbol{\theta}^*, t^*, \boldsymbol{s}^*)$ is a global solution to Eq. (3.4) if and only if it is also a global solution to Eq. (3.6). Then by applying Lem. A.1, we've proven the conclusion desired. □

B Full TBAL Loop with AL-ERO

Given an unlabeled pool $\mathcal{X}_u$, a target error rate ϵ, and a human-label budget N_t, AL-ERO is applied within each round of TBAL as follows. **(1) Initialization.** At $i = 1$, randomly query human labels from $\mathcal{X}_u$ to form a training set $\mathcal{D}_{\text{train}}^{(1)}$, a disjoint threshold-estimation set $\mathcal{D}_{\text{th}}^{(1)}$, and a hyperparameter set $\mathcal{D}_{\text{hyp}}$ of prescribed sizes; denote the remainder as $\mathcal{X}_u^{(1)}$. **(2) AL-ERO.** Warm-start $\boldsymbol{\theta}^{(i)}$ with cross-entropy on $\mathcal{D}_{\text{train}}^{(i)}$, then train $\boldsymbol{\theta}^{(i)}$ with the AL-ERO objective Eq. (3.10) on the same set using Algorithm 1; the error constraint therein is enforced against the empirical target error $\hat{\epsilon}$ in place of the nominal target ϵ; $\hat{\epsilon}$ is computed once before training via $\hat{\epsilon} + C_1\sqrt{\hat{\epsilon}(1-\hat{\epsilon})} = \epsilon$. Algorithm 1 gives the exact-penalty procedure used to train the classifier under the AL-ERO objective Eq. (3.10). **(3) Threshold estimation.** After training in step (2) completes, we re-estimate a deployment threshold $t^{(i)}$ on $\mathcal{D}_{\text{th}}^{(i)}$, used to auto-label points in the unlabeled pool. Specifically, choose the smallest threshold $t^{(i)} \in [0, 1]$ such that the resulting error on $\mathcal{D}_{\text{th}}^{(i)}$ does not exceed $\hat{\epsilon}$. **(4) Auto-label, prune, and replenish.** For every unlabeled point $\boldsymbol{x}_j \in \mathcal{X}_u^{(i)}$, let the class prediction of each point be $\hat{y}_j = \arg\max_k h_{\boldsymbol{\theta}}^{(i)}(\boldsymbol{x}_j)[k]$. Every $\boldsymbol{x}_j \in \mathcal{X}_u^{(i)}$ with $h_{\boldsymbol{\theta}}^{(i)}(\boldsymbol{x_j})[\hat{y}_j] > t^{(i)}$ is auto-labeled by $\hat{y}_j$ and removed from $\mathcal{X}_u^{(i)}$. Accepted points are likewise removed from $\mathcal{D}_{\text{train}}^{(i)}$ and $\mathcal{D}_{\text{th}}^{(i)}$ so that their

Algorithm 2 TBAL Loop with AL-ERO

Input: Unlabeled dataset $\mathcal{X}_u$, human-label quota N_t, set sizes $(N_{\text{train}}, N_{\text{th}}, N_{\text{hyp}})$, target error rate ϵ, initial penalty parameters $\mu^{(0)}$ and $\boldsymbol{\lambda}^{(0)}$, regularization parameter γ, penalty multiplier ρ, maximum number of outer iterations M for Algorithm 1, classification weight β, maximum number of rounds $I_{\max}$.

▷ /*** Compute the empirical target error ***/
$\hat{\epsilon} \leftarrow$ the value solving $\hat{\epsilon} + C_1\sqrt{\hat{\epsilon}(1-\hat{\epsilon})} = \epsilon$
▷ /*** Query initial training, threshold-estimation, and hyperparameter sets ***/
$(\mathcal{D}_{\text{train}}^{(1)}, \mathcal{D}_{\text{th}}^{(1)}, \mathcal{D}_{\text{hyp}}) \leftarrow \text{RandomQuery}(\mathcal{X}_u, N_{\text{train}}, N_{\text{th}}, N_{\text{hyp}})$
$\mathcal{X}_u^{(1)} \leftarrow \mathcal{X}_u \backslash (\mathcal{D}_{\text{train}}^{(1)} \cup \mathcal{D}_{\text{th}}^{(1)} \cup \mathcal{D}_{\text{hyp}})$
$\mathcal{D}_{\text{AL}}^{(1)} \leftarrow \{\}$ ▷ Set of points auto-labeled
$N_{\text{budget}}^{(1)} \leftarrow N_t - N_{\text{train}} - N_{\text{th}} - N_{\text{hyp}}$
$i \leftarrow 1$
while $i \le I_{\max}$ and $N_{\text{budget}}^{(i)} > 0$ and $|\mathcal{X}_u^{(i)}| > 0$ **do**
  ▷ /*** Cross entropy warm-up ***/
  Initialize $\boldsymbol{\theta}^{(i)}$, $t^{(0)}$, $\boldsymbol{s}^{(0)}$
  Warm up the classifier on $\mathcal{D}_{\text{train}}^{(i)}$:

$$\boldsymbol{\theta}^{(i)} \leftarrow \operatorname*{argmin}_{\boldsymbol{\theta}} \mathcal{L}_{\text{CE}}(\boldsymbol{\theta})$$

  Train $\boldsymbol{\theta}^{(i)}$ with the AL-ERO objective: ▷ Optimization shown in Algorithm 1

$$\boldsymbol{\theta}^{(i)} \leftarrow \text{AL-ERO}\big(\mathcal{D}_{\text{train}}^{(i)}, \hat{\epsilon}, \mu^{(0)}, \boldsymbol{\lambda}^{(0)}, (\boldsymbol{\theta}^{(i)}, t^{(0)}, \boldsymbol{s}^{(0)}), \rho, M, \gamma, \beta\big)$$

  $\hat{y}_j^{(i)} \leftarrow \arg\max_k h_{\boldsymbol{\theta}^{(i)}}(\boldsymbol{x}_j)[k]$; $c_j^{(i)} \leftarrow \max_k h_{\boldsymbol{\theta}^{(i)}}(\boldsymbol{x}_j)[k]$ ▷ Class prediction and confidence
  ▷ /*** Threshold tuning ***/
  $t^{(i)} \leftarrow \min\left\{t \in [0,1] \mid \mathcal{E}_{\text{th}}(\boldsymbol{\theta}^{(i)}, t) \le \hat{\epsilon}\right\}$
  ▷ /*** Label and remove points above threshold ***/
  $\mathcal{D}_{\text{new}} \leftarrow \left\{\boldsymbol{x}_j \in \mathcal{X}_u^{(i)} \;\middle|\; c_j^{(i)} > t^{(i)}\right\}$ ▷ New auto-labeled points
  $\mathcal{D}_{\text{AL}}^{(i+1)} \leftarrow \mathcal{D}_{\text{AL}}^{(i)} \cup \mathcal{D}_{\text{new}}$
  $\mathcal{X}_u^{(i)} \leftarrow \mathcal{X}_u^{(i)} \backslash \mathcal{D}_{\text{new}}$
  $\mathcal{D}_{\text{train_rm}}^{(i)} \leftarrow \left\{\boldsymbol{x}_j \in \mathcal{D}_{\text{train}}^{(i)} \;\middle|\; c_j^{(i)} > t^{(i)}\right\}$
  $\mathcal{D}_{\text{th_rm}}^{(i)} \leftarrow \left\{\boldsymbol{x}_j \in \mathcal{D}_{\text{th}}^{(i)} \;\middle|\; c_j^{(i)} > t^{(i)}\right\}$
  $\mathcal{D}_{\text{train}}^{(i)} \leftarrow \mathcal{D}_{\text{train}}^{(i)} \backslash \mathcal{D}_{\text{train_rm}}^{(i)}, \quad \mathcal{D}_{\text{th}}^{(i)} \leftarrow \mathcal{D}_{\text{th}}^{(i)} \backslash \mathcal{D}_{\text{th_rm}}^{(i)}$
  ▷ /*** Query new points ***/
  $(\mathcal{D}_{\text{train_qr}}^{(i)}, \mathcal{D}_{\text{th_qr}}^{(i)}) \leftarrow \text{RandomQuery}\left(\mathcal{X}_u^{(i)}, |\mathcal{D}_{\text{train_rm}}^{(i)}|, |\mathcal{D}_{\text{th_rm}}^{(i)}|\right)$ ▷ New queried points
  $\mathcal{D}_{\text{train}}^{(i+1)} \leftarrow \mathcal{D}_{\text{train}}^{(i)} \cup \mathcal{D}_{\text{train_qr}}^{(i)}, \quad \mathcal{D}_{\text{th}}^{(i+1)} \leftarrow \mathcal{D}_{\text{th}}^{(i)} \cup \mathcal{D}_{\text{th_qr}}^{(i)}$
  $\mathcal{X}_u^{(i+1)} \leftarrow \mathcal{X}_u^{(i)} \backslash (\mathcal{D}_{\text{train_qr}}^{(i)} \cup \mathcal{D}_{\text{th_qr}}^{(i)})$
  $N_{\text{budget}}^{(i+1)} \leftarrow N_{\text{budget}}^{(i)} - |\mathcal{D}_{\text{train_qr}}^{(i)}| - |\mathcal{D}_{\text{th_qr}}^{(i)}|$
  $i \leftarrow i + 1$
end while
return $\mathcal{D}_{\text{AL}}^{(i)}$

distributions are kept the same as $\mathcal{X}_u^{(i)}$. Both sets are then restored to their designated sizes by querying new human labels from $\mathcal{X}_u^{(i)}$. Auto-labeled points never enter $\mathcal{D}_{\text{train}}$ or $\mathcal{D}_{\text{th}}$. Steps (2)–(4) repeat until $\mathcal{X}_u^{(i)}$ is empty, the budget N_t is exhausted, or a maximum number of rounds is reached, as specified in Algorithm 2. The complete TBAL loop, combining AL-ERO with threshold estimation and unlabeled-pool updates, is shown in Algorithm 2.

For the full-loop tables, coverage is $|\mathcal{D}_{\mathrm{AL}}|/|\mathcal{X}_u|$: the denominator is the complete dataset, including initially human-labeled points, while the numerator counts only auto-labeled points. Error is the fraction of incorrect labels within $\mathcal{D}_{\mathrm{AL}}$. In contrast, first-round error-coverage curves are computed only on that round's unlabeled pool and normalize coverage by the size of that pool.

C More Experiment Setting and Results

C.1 Dataset Details

OrganACSMNIST combines three abdominal CT organ-classification datasets from MedMNIST v2—OrganAMNIST (axial), OrganCMNIST (coronal), and OrganSMNIST (sagittal)—which differ in imaging plane. The details of each dataset are summarized in Tab. 4.

Table 4: Dataset details for the experiments.

Dataset	**Size**	$\mathcal{D}_{\mathrm{train}}$	$\mathcal{D}_{\mathrm{th}}$	$\mathcal{D}_{\mathrm{hyp}}$	**Human-label quota**
OrganACSMNIST	107,624	10,000	5,000	3,000	50,000
PathMNIST	107,180	10,000	5,000	3,000	50,000
PubMed 200k RCT	2,270,286	10,000	5,000	3,000	100,000
20 Newsgroups	18,846	5,000	3,000	2,000	15,000

C.2 Method Details

CE, Squentropy, MDCA, CRL, FMFP, AURC, and Colander follow their original formulations, as described in Sec. 2; their hyperparameter search spaces are given in Tabs. 5 and 6 below.

C.2.1 AL-Surrogate

To fully compare our proposed method, we construct AL-Surrogate, targeting the same constrained problem as AL-ERO, Eq. (3.1): jointly optimizing classification and selection performance. The difference is that AL-Surrogate deals with selection indicator with sigmoid surrogate, as Colander does. This method isolates the effect of the exact reformulation from the benefit of joint classifier–confidence training, giving a controlled comparison against AL-ERO (Sec. D.1).

Concretely, AL-Surrogate optimizes the classifier directly on $\mathcal{D}_{\mathrm{train}}$: let $\mathcal{P}_{\mathrm{train}}(\boldsymbol{\theta}, t)$ and $\mathcal{E}_{\mathrm{train}}(\boldsymbol{\theta}, t)$ denote the coverage and error of Eq. (3.2), but with the selection indicator $\mathbb{1}\{c_i(\boldsymbol{\theta}) > t\}$ replaced throughout by the same sigmoid relaxation $\sigma(\alpha, c_i(\boldsymbol{\theta}) - t)$ used by Colander.

AL-Surrogate is trained by minimizing the penalized objective, adapting Colander's coverage-error loss to classifier training and adding the CE term:

$$\mathcal{L}_{\mathrm{Surr}} = -\mathcal{P}_{\mathrm{train}}(\boldsymbol{\theta}, t) + \mu\big(\mathcal{E}_{\mathrm{train}}(\boldsymbol{\theta}, t) - \hat{\epsilon}\big) + \beta\, \mathcal{L}_{\mathrm{CE}}, \tag{C.1}$$

where $\hat{\epsilon}$ is the fixed empirical target error. Following Colander's strategy, μ penalizes the error constraint; it is selected by hyperparameter search and held constant throughout training. Deployment thresholds are estimated on $\mathcal{D}_{\mathrm{th}}$ as in Appendix B.

AL-Surrogate+EP: AL-Surrogate+EP solves the same objective Eq. (C.1) as AL-Surrogate, but with the exact penalty method as its optimizer. The error constraint is handled as $\mu\big[\mathcal{E}_{\mathrm{train}}(\boldsymbol{\theta}, t) - \hat{\epsilon}\big]_+$. Rather than holding μ fixed, its value is adapted across outer iterations: the classifier is trained to minimize Eq. (C.1) with the current μ, after which μ is multiplied by 1.05 whenever the empirical training-set error exceeds $\hat{\epsilon}$, and otherwise left unchanged, mirroring the penalty update in Algorithm 1. This lets AL-Surrogate+EP be compared against AL-ERO under matched optimization. Deployment thresholds are re-estimated on $\mathcal{D}_{\mathrm{th}}$ for checkpoint selection and used to label data, as with other methods.

C.2.2 Common training and hyperparameter selection

Except for plain CE method, several classifier training methods are compared. Squentropy (Hui et al., 2023) augments cross-entropy with an average squared loss over the incorrect classes. Correctness

Ranking Loss (CRL) (Moon et al., 2020) introduces a ranking regularizer, encouraging confidence scores to follow the relative correctness of predictions. FMFP (Zhu et al., 2022) improves failure prediction by seeking flat minima via sharpness-aware minimization. Multi-class Difference in Confidence and Accuracy (MDCA) (Hebbalaguppe et al., 2022) adds a calibration regularizer that reduces the discrepancy between model confidence and empirical accuracy. AURC optimization (Zhou et al., 2024) instead directly targets the error-coverage tradeoff used in selective prediction. All these classifier training methods use the same frozen feature representation and linear classifier. CE is trained from scratch; the other methods start from a CE checkpoint trained on the same current $\mathcal{D}_{\mathrm{train}}$, using the selected CE hyperparameters. CE, Squentropy, MDCA, CRL, FMFP, AURC, and AL-Surrogate use a 200-epoch budget with Adam and cosine learning-rate decay. AL-ERO uses exact penalty method with 200 outer epochs and 1200 iterations within each outer epoch. After each outer iteration, a deployment threshold is re-estimated on $\mathcal{D}_{\mathrm{th}}$, and the classifier with the largest resulting coverage on $\mathcal{D}_{\mathrm{th}}$ is retained.

Hyperparameters for each method are selected by the same rule, for each dataset and target error, evaluated at the first TBAL round with five model seeds: among candidates whose mean $\mathcal{D}_{\mathrm{hyp}}$ error is under ϵ, we select the one with the largest mean coverage; if none is feasible, we select hyperparameters with the smallest mean error. Since every other method warm-starts from a CE checkpoint, CE hyperparameters are selected first.

Colander is a post-classifier method: at every TBAL round, half of the $\mathcal{D}_{\mathrm{train}}$ quota is held out as a selector-training set $\mathcal{D}_{\mathrm{se}}$, the classifier is trained by cross-entropy on the remaining half, following hyperparameter selection strategy as the CE baseline, and Colander's selector is then fit on $\mathcal{D}_{\mathrm{se}}$ by minimizing a sigmoid-relaxed version of Eq. (2.1), with its own hyperparameter search space given in Tab. 6. Both the classifier and the selector use the same 200 epoch with Adam and cosine learning-rate decay, and the checkpoint with the largest resulting $\mathcal{D}_{\mathrm{th}}$ coverage is retained; both are refit from scratch at every round. We also compare with other post-classifier optimization methods—Dirichlet (Kull et al., 2019), Temperature scaling (Guo et al., 2017), Scaling-binning (Kumar et al., 2019), and top-label histogram binning (Gupta & Ramdas, 2021). These all methods share the same experiment setting with Colander.

D Additional Experiments and Results

D.1 Exact Reformulation vs. Surrogate Approximation

To isolate the effect of the exact reformulation, we compare AL-ERO with AL-Surrogate+EP, a variant of AL-Surrogate solved using the same exact penalty method as AL-ERO. AL-Surrogate+EP retains the sigmoid approximation used by AL-Surrogate, while matching the optimization method of AL-ERO. Therefore, the two methods differ only in the treatment of the indicator functions: AL-ERO uses the exact reformulation in Lem. 3.1, whereas AL-Surrogate+EP uses sigmoid approximations. The results are reported in Tab. 7.

AL-ERO consistently achieves higher coverage than AL-Surrogate+EP across all four datasets and both target error levels, while both methods maintain auto-labeling error below the prescribed target. Since the two methods use the same auto-labeling objective and exact penalty method and differ only in the treatment of the indicator functions, these results further support the benefit of exact reformulation over smooth surrogate approximation for directly optimizing the TBAL objective.

D.2 Performance Across Foundation Models

To examine whether the performance of AL-ERO depends on the choice of pretrained representation, we repeat the OrganACSMNIST experiment at $\epsilon = 5\%$ using BioMedCLIP (Zhang et al., 2025a) and PubMedCLIP (Eslami et al., 2023) in place of the CLIP representation used in the main experiments. The results are reported in Tab. 8.

Across both alternative representations, AL-ERO achieves the highest coverage among all methods while maintaining error below the prescribed target. With BioMedCLIP, AL-ERO reaches 48.68% coverage, compared with 44.46% for the strongest baseline. With PubMedCLIP, AL-ERO achieves 61.68% coverage, compared with 58.14% for AL-Surrogate. Although the relative performance of the competing methods changes with the feature representation, AL-ERO remains consistently

Table 5: Classifier optimization method hyperparameter search spaces.

Method	Hyperparameter	Values
CE	learning rate	1e-5, 1e-4, 1e-3
	weight decay	1e-5, 1e-4, 1e-3
	batch size	256, 512
Squentropy	learning rate	1e-5, 1e-4, 1e-3
	Squentropy weight	0.1, 1, 10
	batch size	256, 512
MDCA	learning rate	1e-5, 1e-4, 1e-3
	calibration weight	1, 10, 100
	batch size	256, 512
CRL	learning rate	1e-5, 1e-4, 1e-3
	ranking weight	0.1, 0.5, 1
	batch size	256, 512
FMFP	learning rate	1e-5, 1e-4, 1e-3
	SAM radius	0.05, 0.1, 0.2
	batch size	256, 512
AURC	learning rate	1e-5, 1e-4, 1e-3
	AURC weight	0.1, 1, 10
	batch size	128, 256
AL-Surrogate	classifier learning rate	1e-5, 1e-4, 1e-3
	sigmoid α	1, 5
	error weight	10, 100
	CE weight	0.1, 1, 10
AL-Surrogate+EP	classifier learning rate	1e-5, 5e-5, 1e-4
	initial error weight	0.1, 1, 10
	CE weight	0.1, 1, 10
AL-ERO	classifier learning rate	1e-5, 5e-5, 1e-4
	s_i learning rate	1e-5, 5e-5, 1e-4
	CE weight	0.1, 1, 10

Table 6: Separate selector hyperparameter search spaces. Each selector is fit on $\mathcal{D}_{\mathrm{se}}$, one half of the $\mathcal{D}_{\mathrm{train}}$ quota, on top of a plain cross-entropy classifier whose own recipe is selected separately. Candidate configurations are compared on $\mathcal{D}_{\mathrm{hyp}}$.

Method	Hyperparameter	Values
Dirichlet	regularization strength	1e-3, 1e-2, 1e-1
Temperature scaling	learning rate	1e-4, 1e-3, 1e-2
	weight decay	0.1, 1
Scaling-binning	number of bins	10, 20, 25
	learning rate	1e-4, 1e-3, 1e-2
	weight decay	0.1, 1
Histogram binning	points per bin	25, 50
Colander	learning rate	1e-4, 1e-3, 1e-2
	weight decay	0.01, 0.1, 1
	error weight	10, 100
	sigmoid α	0.1, 1, 10

favorable, indicating that its improvement is not specific to the representation used in the main experiments.

Table 7: Exact reformulation vs. surrogate approximation: results under error constraints $\epsilon = 5\%$ (top) and $\epsilon = 10\%$ (bottom).

Method	OrganACSMNIST		PathMNIST		PubMed 200k RCT		20 Newsgroups	
	Cov	Err	Cov	Err	Cov	Err	Cov	Err
AL-Surrogate+EP	53.45 ± 0.13	3.73 ± 0.06	72.78 ± 0.11	3.97 ± 0.02	31.03 ± 0.51	3.98 ± 0.04	23.04 ± 0.05	4.66 ± 0.05
AL-ERO	$\mathbf{53.59 \pm 0.18}$	3.62 ± 0.05	$\mathbf{73.45 \pm 0.09}$	3.94 ± 0.04	$\mathbf{37.22 \pm 2.16}$	4.39 ± 0.25	$\mathbf{23.13 \pm 0.42}$	4.89 ± 0.20

Method	OrganACSMNIST		PathMNIST		PubMed 200k RCT		20 Newsgroups	
	Cov	Err	Cov	Err	Cov	Err	Cov	Err
AL-Surrogate+EP	66.44 ± 0.22	7.89 ± 0.09	82.96 ± 0.02	8.57 ± 0.04	62.06 ± 0.37	8.30 ± 0.04	30.72 ± 0.06	9.42 ± 0.10
AL-ERO	$\mathbf{68.15 \pm 0.34}$	8.52 ± 0.12	$\mathbf{83.19 \pm 0.01}$	8.31 ± 0.05	$\mathbf{62.95 \pm 0.71}$	8.48 ± 0.08	$\mathbf{31.11 \pm 0.09}$	9.72 ± 0.10

Table 8: Auto-labeling results on OrganACSMNIST using alternative medical-image embeddings in place of CLIP ViT-B/32, at target error $\epsilon = 5\%$.

(a) BioMedCLIP

Method	Cov	Err
CE	41.11 ± 0.33	4.45 ± 0.09
Squentropy	41.70 ± 0.68	4.60 ± 0.16
CRL	43.15 ± 0.29	4.47 ± 0.08
FMFP	42.26 ± 0.49	4.46 ± 0.13
MDCA	43.04 ± 0.39	4.51 ± 0.16
AURC	$\underline{44.46 \pm 0.70}$	4.67 ± 0.19
AL-Surrogate	42.12 ± 0.26	4.66 ± 0.02
AL-ERO	$\mathbf{48.68 \pm 0.36}$	4.28 ± 0.08

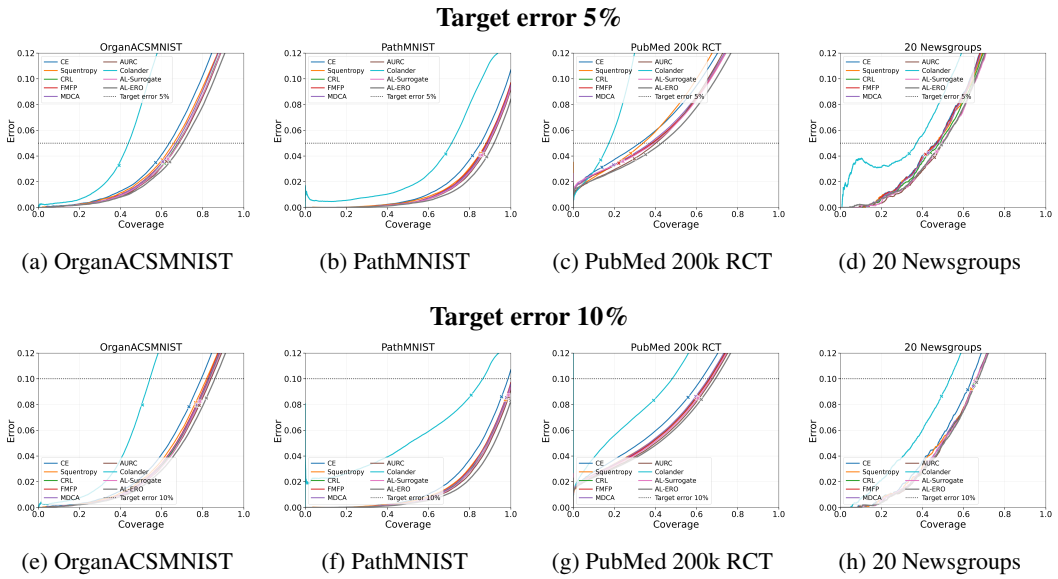
(b) PubMedCLIP

Method	Cov	Err
CE	52.39 ± 0.43	4.05 ± 0.06
Squentropy	56.39 ± 0.30	4.28 ± 0.06
CRL	57.03 ± 0.33	4.26 ± 0.10
FMFP	56.22 ± 0.26	4.29 ± 0.07
MDCA	56.60 ± 0.14	4.37 ± 0.13
AURC	57.69 ± 0.25	4.27 ± 0.18
AL-Surrogate	$\underline{58.14 \pm 0.30}$	4.53 ± 0.09
AL-ERO	$\mathbf{61.68 \pm 0.25}$	4.43 ± 0.06

D.3 Error-Coverage Curves at the First TBAL Iteration

Fig. 4 provides a more direct view of how confidence optimization affects threshold-based auto-labeling. For a fixed set of classifier predictions, the coverage-error tradeoff induced by thresholding is determined by the confidence ordering of samples rather than by the absolute scale of the scores. We therefore plot the error-coverage curve on the unlabeled pool at the first TBAL iteration ($i = 1$), which traces the auto-labeling performance obtained by sweeping the confidence threshold. The curves are shown for target errors 5% and 10%, with each curve aggregated over five seeds. We additionally mark with a cross the mean operating point obtained using the threshold estimated from $\mathcal{D}_{\mathrm{th}}$, allowing the deployed auto-labeling result to be viewed together with the full ranking-induced error-coverage tradeoff.

Across datasets, AL-ERO generally produces a more favorable error-coverage curve, achieving higher coverage at comparable error over substantial portions of the curve, particularly in the low-error region. This further shows that directly optimizing the TBAL objective improves the confidence ranking used for threshold-based selection, consistent with the coverage improvements observed at the $\mathcal{D}_{\mathrm{th}}$-estimated operating points.



(a) OrganACSMNIST (b) PathMNIST (c) PubMed 200k RCT (d) 20 Newsgroups

(e) OrganACSMNIST (f) PathMNIST (g) PubMed 200k RCT (h) 20 Newsgroups

Figure 4: Error-coverage curves at the first TBAL iteration for target error 5% (top) and 10% (bottom).