

Selective Annotation: Certified Accuracy, Minimal Human Cost

YIFAN WU, Shenzhen University, China and Shenzhen Institute of Computing Sciences, China

SHUHAO LIU*, Shenzhen Institute of Computing Sciences, China

YUANZHE CHEN, Shenzhen Institute of Computing Sciences, China

WENFEI FAN, Shenzhen Institute of Computing Sciences, China and University of Edinburgh, United Kingdom

This paper studies large-scale dataset labeling via *selective annotation*. Given an unlabeled dataset D , the goal is to label all samples in D with minimal human annotation cost, while ensuring a certified labeling accuracy. We show that this problem is intractable. We propose SALT (Selective Annotation for Labeling and Training), a human-in-the-loop framework for labeling. SALT incrementally trains a model M and introduces a novel *curvature-adaptive coverage condition* under which M is provably guaranteed to meet a target accuracy bound on D . It iteratively selects a small number of high-impact samples for annotation, guided by a unified coverage-uncertainty criterion derived from the theoretical bound. SALT further integrates multiple labeling functions through a logic-based correction mechanism to mitigate weak supervision noise, and leverages these enhanced weak signals to speed up early learning when annotated data is scarce. Using real-world datasets, we empirically show that SALT improves labeling accuracy by up to 72.4 points over active learning, saving human effort by over 75.7%.

CCS Concepts: • **Information systems** → **Data management systems**; • **Computing methodologies** → **Active learning settings**; • **Theory of computation** → *Machine learning theory*.

Additional Key Words and Phrases: Selective annotation, Data labeling, Machine learning, Human-in-the-loop data management, Weak supervision

ACM Reference Format:

Yifan Wu, Shuhao Liu, Yuanzhe Chen, and Wenfei Fan. 2026. Selective Annotation: Certified Accuracy, Minimal Human Cost. *Proc. ACM Manag. Data* 4, 6 (SIGMOD), Article 390 (December 2026), 46 pages. <https://doi.org/10.1145/3857218>

1 Introduction

Modern machine learning (ML) systems, particularly large language models (LLMs), rely on large-scale labeled datasets. Data labeling, *i.e.*, assigning predefined categories or annotations to raw samples, provides the supervision signal necessary for training, instruction tuning, and alignment. High-quality labeled datasets play a critical role in a wide range of applications such as autonomous driving, social media intent recognition and medical diagnosis [1], and their importance continues to grow as ML systems scale.

*Corresponding author.

Authors' Contact Information: Yifan Wu, wuyifan@sics.ac.cn, Shenzhen University, College of Computer Science and Software Engineering, Shenzhen, China and Shenzhen Institute of Computing Sciences, Shenzhen, China; Shuhao Liu, shuhao@sics.ac.cn, Shenzhen Institute of Computing Sciences, Shenzhen, China; Yuanzhe Chen, chenyuanzhe@sics.ac.cn, Shenzhen Institute of Computing Sciences, Shenzhen, China; Wenfei Fan, wenfei@inf.ed.ac.uk, Shenzhen Institute of Computing Sciences, Shenzhen, China and University of Edinburgh, Edinburgh, United Kingdom.



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

© 2026 Copyright held by the owner/author(s).

ACM 2836-6573/2026/12-ART390

<https://doi.org/10.1145/3857218>

In practice, training a production-grade ML model often requires a labeled dataset D containing millions of samples. For example, ImageNet [21] includes over 14 million human-annotated images collected over several years. Similarly, natural language processing systems depend on large-scale labeled corpora, and modern commercial LLMs are trained or tuned on tens to hundreds of millions of annotated text instances for tasks such as sentiment analysis, question answering, and instruction execution [30, 58, 86].

However, constructing such datasets relies heavily on human annotators, making it costly and time-consuming. Labeling one million text documents requires over 1,388 person-hours even under optimistic assumptions, and expert annotation can cost \$50–100 per instance [34]. In many domains, qualified annotators are scarce; diverting clinicians from patient care to label data is neither feasible nor sustainable. This highlights the need for labeling strategies that minimize human effort while preserving accuracy and reliability.

Selective annotation. The question is: *Can we label an entire dataset by annotating only a small subset, while provably meeting a target accuracy?* This gives rise to the *selective annotation problem*.

Given an unlabeled dataset D , the goal is to manually annotate only a small subset $D_S \subseteq D$ and automatically label the remaining samples, while meeting a prescribed target accuracy threshold over the entire dataset. The objective is to minimize human annotation effort while guaranteeing labeling quality. For simplicity, we focus on classification, one of the most common labeling tasks in practice. This is a data-quality problem: it delivers the labeled dataset itself as a *data asset of certified quality*, in line with data preparation and human-in-the-loop data management [8, 63].

A natural approach is to train a labeling model $\mathcal{M}$ that annotates samples in D with provable accuracy. However, no prior work targets this objective and thus falls short. (1) *Active learning* [10, 42, 81] reduces annotation cost but optimizes generalization rather than dataset-wide labeling accuracy on D , and assumes a fully labeled validation or test set for selection and stopping. (2) *Weak supervision* [8, 60, 63] scales labeling via heuristic signals, which are often noisy and can degrade model quality. (3) *Semi-supervised learning* [12, 73] expands labels via pseudo-labels, but errors can propagate and amplify over iterations. (4) *Unsupervised learning* [16, 50] groups similar samples without annotations, but the discovered clusters may not align with the target classes. This gap leads to three challenges:

- *Certified accuracy is missing.* No existing approach provides verifiable guarantees that the labeled dataset meets a target accuracy threshold, as true labels are unknown during the process.
- *Minimizing human effort is hard.* Selecting a subset $D_S \subseteq D$ that maximizes global impact is inherently combinatorial and intractable (Section 3), since the value of each sample depends on its interaction with others and the underlying data geometry.
- *Noise must be explicitly controlled.* Weak supervision improves scalability but introduces noise that, if uncorrected, can propagate and violate the objective accuracy bound.

SALT. The key idea is to view selective annotation as a *coverage problem in embedding space*: each annotated sample covers a local region where the model’s loss is provably bounded, and achieving global coverage over D yields a dataset-wide accuracy guarantee.

Based on this idea, we propose SALT (Selective Annotation for Labeling and Training), a framework that addresses the three challenges above: certified accuracy, minimal human effort, and noise tolerance. SALT trains a classification model $\mathcal{M}$ to automatically label D while minimizing human cost under a given accuracy target. SALT employs an iterative human-in-the-loop (HITL) procedure. At each round, it selects a subset $D_t \subseteq D$ for annotation, updates $\mathcal{M}$, and checks a coverage-based certification condition. The process terminates once the induced coverage spans D , at which point $\mathcal{M}$ is guaranteed to meet the given accuracy target on the entire dataset.

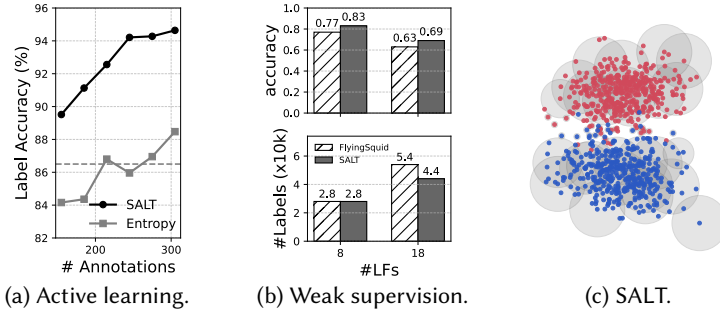


Fig. 1. Prior methods vs. SALT. (a) *Active learning* (e.g., Entropy [81]) optimizes *generalization* via uncertainty sampling, but offers *no dataset-level guarantee* on D and relies on *heuristic stopping* (e.g., budget or validation proxy). (b) *Weak supervision* (e.g., FlyingSquid [27]) optimizes *label aggregation* from heuristic sources, but provides *neither control over global accuracy* due to noise *nor well-defined stopping criterion*. (c) SALT induces a *coverage objective* aligned with dataset labeling, provides a *certified accuracy guarantee* via curvature-adaptive coverage, and admits an *explicit stopping rule*: it terminates once the certified regions fully cover D .

Figure 1 illustrates the difference between prior work and SALT.

At a high level, SALT makes the following contributions.

(1) *Certified selective annotation*. We introduce the first framework for selective annotation with certified dataset-wide accuracy. We derive *curvature-adaptive coverage*, where the size of the local cover induced by each annotated sample is determined by the local loss geometry. When these local covers jointly span the dataset, *i.e.*, a full coverage, we obtain a provable bound on global risk and labeling accuracy. This provides a verifiable stopping criterion, unlike prior methods that rely on heuristics without guarantees.

(2) *Coverage-driven selection*. We develop a selection strategy that *directly optimizes certified coverage*, rather than heuristic uncertainty or diversity. At each round, SALT selects hard samples that expand uncovered certified regions, progressing from global coverage to boundary refinement. This unifies uncertainty and diversity under a single objective, aligned with the accuracy guarantee.

(3) *Integration of weak signals*. We incorporate weak supervision via a logic-based correction framework that *controls noise without compromising guarantees*. By resolving conflicts among labeling functions using rules learned from human annotations, SALT produces higher-quality labels for efficient bootstrapping. This reduces human effort while preserving the certified accuracy guarantee.

(4) *Experimental evaluation*. Using real-world datasets, we validate the effectiveness and efficiency of SALT. (a) SALT achieves the highest labeling Normalized Cross-Entropy (NCE) [35] accuracy, outperforming active learning baselines by up to 72.4 points. (b) It reduces human annotation cost by up to 75.7% under high-accuracy targets. (c) Its weak-supervision module reduces the error rate of weak labels, measured against ground-truth labels, by up to 7.8 points, and further reduces annotation cost via effective bootstrapping. (d) It speeds up convergence and reduces end-to-end execution time by >44.0%, verifying strong efficiency and robustness.

Organization. Section 2 reviews related work on data labeling. Section 3 formulates the selective annotation problem, proves its intractability, and presents the SALT framework. Section 4 introduces the curvature-adaptive cover and derives the corresponding accuracy guarantee. Section 5 presents the coverage-driven selection strategy, and Section 6 describes the logic-based integration of weak supervision. Section 7 reports experimental results, and Section 8 concludes with a discussion of novelty and future directions.

2 Related Work

This section categorizes related work on data labeling.

Active learning seeks to train a model $\mathcal{M}$ that generalizes well with minimal annotation cost. In the standard setting, it iteratively selects batches of unlabeled samples for annotation, augments the training set, and updates $\mathcal{M}$ to guide subsequent selection. Selection strategies fall into 3 categories. (1) *Uncertainty-based methods* (e.g., Uncertainty [49], Margin [67, 74], Entropy [81], and others [28, 45, 76, 77]) prioritize samples for which the current model is most uncertain. (2) *Diversity-based methods* (e.g., CoreSet [69], TypiClust [44], and others [9, 18, 44, 71, 83, 87]) select representative subsets that cover the input space while reducing redundancy. (3) *Hybrid methods* (e.g., BADGE [6], UHerd [10], ALFA-Mix [59], and others [42, 51]) combine uncertainty and diversity to balance informativeness and coverage, particularly in batch selection.

Active learning is the area closest to selective annotation and thus the primary baseline family. Our setting differs in four ways. (1) SALT targets labeling accuracy on the *given* dataset D , not optimizing for generalization. (2) It provides a coverage-based certificate which active learning typically lacks. (3) Its coverage-driven strategy adaptively balances multiple criteria rather than relying on fixed heuristics. (4) It uses multiple labeling functions as additional supervision, beyond the standard active learning paradigm.

Weak supervision reduces reliance on large-scale manual annotation by employing labeling functions (LFs), e.g., heuristic rules or programmatic functions, to automatically assign labels to unlabeled data. However, such labeling functions are not always available (e.g., in visual tasks), and when they are, they are often noisy, incomplete and conflicting, which limits their reliability.

One line of work assists LF authoring. ActiveDP [32], for example, uses active learning to select instances from which the user authors LFs, then combines the LFs to label data at inference. (1) It seeks *broad-coverage weak labels* for model training, whereas SALT labels *every* sample in dataset D subject to a dataset-wide accuracy target. (2) It assumes that a user can derive a *useful* LF from a sample; this is more demanding than assigning class labels in SALT and may be infeasible when interpretable LFs are unavailable, e.g., in image classification. (3) It provides neither a formal accuracy bound nor a certified stopping condition, whereas SALT provides both.

Other methods improve weak-label quality: (1) *Rule- and distant supervision* [8, 23, 29, 36, 60, 63, 64, 75] encode domain knowledge or align data with external sources to generate heuristic labels. (2) *Label aggregation* [20, 27, 39] model the accuracy and correlations of multiple noisy sources to infer latent true labels, typically via generative or probabilistic models. (3) *Joint optimization and noise-aware learning* [61, 65] combine weak-label inference with model training and refine labels through iterative or end-to-end optimization. SALT optionally integrates weak supervision to speed up learning when reliable LFs are available. To ensure robustness, it employs a logic-based mechanism to detect and mitigate errors, preventing noise propagation. Unlike purely model-based aggregation, it explicitly reasons over heterogeneous and complementary LFs, improving reliability without affecting the overall accuracy guarantee.

Semi-supervised learning [12, 14, 47, 73, 82, 88] trains models on a small labeled set and a large unlabeled set, typically using pseudo-labeling to exploit data distribution and model confidence for better generalization. However, early errors may be reinforced, and these methods do not guarantee full-dataset accuracy. In contrast, SALT reliably labels a given dataset under explicit accuracy constraints, aiming to control dataset-wide labeling errors rather than improve generalization through pseudo-label expansion. Accordingly, SALT uses conservative sample selection and correction to limit error accumulation, reflecting a different objective and risk tolerance.

Coreset selection [4, 38, 41–43, 53, 69, 72, 78] selects a representative subset from a *fully labeled*

dataset to reduce training cost while preserving generalization. Unlike SALT, coreset methods assume full label access during selection, enabling label-aware criteria *e.g.*, class-balanced coverage, gradient matching, and validation-objective optimization. SALT instead selects samples before observing their labels and must rely solely on pre-label information, precluding criteria that require ground-truth labels for the dataset.

Crowdsourcing [20, 22, 39, 70] addresses *how* to obtain reliable labels from noisy annotators, via truth inference and worker modeling. In contrast, SALT studies *which* samples should be annotated and *when* annotation can stop with a certified accuracy guarantee. The two are complementary: crowdsourcing can provide the annotation mechanism within SALT, while SALT decides where human effort is spent.

Industrial annotation platforms (*e.g.*, ScaleAI [1] and SurgeAI [2]) provide workflow and quality-management infrastructure for labeling and AI-oriented data engineering. To our knowledge, they offer neither dataset-wide accuracy certification nor a provable stopping condition. SALT is orthogonal to these platforms and can serve as their sample-selection and termination component.

3 SALT for Selective Annotation

This section formulates and analyzes the selective annotation problem (Section 3.1), and presents SALT at a high level (Section 3.2).

3.1 The Selective Annotation Problem

Preliminaries. We start with basic notations and definitions.

Datasets. Let $D = \{x_i\}_{i=1}^n$ denote an unlabeled dataset with n samples. Each sample $x_i \in \mathcal{X}$ is a data point in the sample space $\mathcal{X}$, *e.g.*, an image, a document, or a relational tuple, *etc.*

Labeling. We consider classification tasks, where each sample is assigned one or more labels. We focus on *w.l.o.g.* single-label classification, where each $x_i \in D$ is assigned a label $y_i \in \{1, \dots, C\}$ from C classes. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ denote the fully labeled dataset derived from D . Following [19, 35, 54, 56], we assume an underlying conditional label distribution $\eta(x) = \Pr[y \mid x] \in \Delta^C$, where Δ^C is the C -dimensional probability simplex and $y \sim \eta(x)$.

Labeling models. We consider a multi-layer perceptron classifier $\mathcal{M}_\theta : \mathcal{X} \rightarrow \{1, \dots, C\}$, where θ is the model parameters (the weights and biases). The model consists of multiple fully connected layers followed by a softmax output layer. Given $x \in \mathcal{X}$, let $\mathbf{x} \in \mathbb{R}^k$ be its embedding. The model produces logits $z_\theta(\mathbf{x}) \in \mathbb{R}^C$, and the class-probability output is $p_\theta(x) = \text{softmax}(z_\theta(\mathbf{x})) \in \Delta^C$. When clear from context, we write p_θ as p and use x and $\mathbf{x}$ interchangeably.

Labeling accuracy. Given a labeling model $\mathcal{M}_\theta$ and an unlabeled dataset D , we define the labeling accuracy of $\mathcal{M}_\theta$ using the Normalized Cross-Entropy (NCE) [35], a standard metric for classification:

$$\text{acc}(\mathcal{M}_\theta, \mathcal{D}) := 1 - R_{\mathcal{D}}(\theta) / \ln C. \quad (1)$$

Here $R_{\mathcal{D}}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(x, y) \in \mathcal{D}} \ell(\theta; x, \eta)$ is the empirical risk over $\mathcal{D}$, where ℓ is the cross-entropy loss $\ell(\theta; x, \eta) = \sum_{k=1}^C -\eta_k \ln p_k$. Intuitively, this score serves as an accuracy metric with confidence awareness: 0 for random guessing and 1 for perfect predictions.

Problem. We now state the *selective annotation problem* (SAP).

- *Input:* An unlabeled dataset $D = \{x_i\}_{i=1}^n$, a classification model $\mathcal{M}$ to train, a learning algorithm A , and an accuracy tolerance ζ .
- *Output:* An annotated dataset $\mathcal{D}_S$ obtained from a subset S of D .
- *Objective:* Minimize the cardinality $|\mathcal{D}_S|$ of $\mathcal{D}_S$ subject to $\text{acc}(\mathcal{M}_{\theta_S}, \mathcal{D}) \geq \text{acc}(\mathcal{M}_{\theta_D}, \mathcal{D}) - \zeta$,

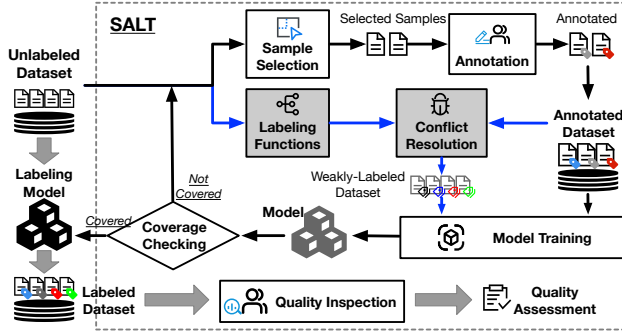


Fig. 2. The data labeling workflow with SALT.

where θ_S (resp. θ_D) denotes the parameters of $\mathcal{M}$ learned by A on $\mathcal{D}_S$ (resp. $\mathcal{D}$).

Intuitively, SAP aims to select a minimum subset $S \subseteq D$ for human annotation and construct a labeled subset $\mathcal{D}_S$. It then trains a labeling model $\mathcal{M}_{\theta_S}$ on $\mathcal{D}_S$, while requiring its accuracy on $\mathcal{D}$ to be within ζ of that of $\mathcal{M}_{\theta_D}$ trained on the fully labeled dataset $\mathcal{D}$. This formulation promotes scalable automated labeling by reducing human annotation effort without sacrificing labeling quality.

Note that SAP differs from unsupervised learning [16, 50]: its goal is to construct a labeled dataset for downstream supervised tasks, not to learn representations or discover latent structure in D . While unsupervised methods may group similar samples, the resulting clusters may not align with the target classes, and assigning semantic labels to clusters still requires human supervision.

Despite its appeal, SAP is intractable. Its decision version, denoted by DSAP, is to decide, given a budget N and a tolerance ζ on the labeling accuracy deviation, whether there exists a subset $S \subseteq D$ such that $|S| \leq N$ and $\text{acc}(\mathcal{M}_{\theta_S}, \mathcal{D}) \geq \text{acc}(\mathcal{M}_{\theta_D}, \mathcal{D}) - \zeta$.

Theorem 1: SAP is NP-hard. □

Proof sketch: We prove the NP-hardness by reduction from the set cover (see [80] for details), which is NP-complete (cf. [31]). □

In practice, the learned decision boundary of a model $\mathcal{M}$ is highly non-convex and depends sensitively on which samples are annotated. SAP is hard because the benefit of annotating a point does not depend on the point alone: it depends on how its label interacts with nearby samples, especially in class-overlap regions. Moreover, both the true label and the resulting gain are revealed only after paying the annotation cost. Hence, SAP is a combinatorial, delayed-feedback search problem, which explains its difficulty.

3.2 SALT: An Overview

Despite this intractability, we develop SALT, a human-in-the-loop framework for selective annotation with accuracy guarantees.

Integration with data labeling workflows. As shown in Figure 2, given an unlabeled dataset D , SALT produces a labeling model $\mathcal{M}_{\theta_S}$ trained using an annotated subset $\mathcal{D}_S$ from D . The model is then used to label the remaining samples in D , yielding a complete labeled dataset $\mathcal{D}_{\theta_S}$ with certified accuracy. This output can be passed directly to quality inspection, where auditors evaluate a random sample from $\mathcal{D}_{\theta_S}$ and issue the final quality report.

This deployment setting also explains why active learning is not a natural fit. Most active learning pipelines rely on a fully labeled validation or test set for selection and stopping, optimizing generalization on that proxy. In production labeling, however, such a reference set for the same dataset D is typically unavailable before labeling, and quality is assessed only after annotation via

sampled human review. SALT is designed for this setting: it optimizes quality directly on D and produces labels that can be audited immediately.

HITL pipeline. Figure 2 also shows the human-in-the-loop (HITL) pipeline of SALT. The system iteratively combines human annotations and weak supervision to improve the labeling model. Each round t uniformly performs selective annotation, weak-label refinement, model training and certified coverage checking, in four steps.

(1) Selective human annotation. Given the current model $\mathcal{M}_t$ and its certified coverage over D , SALT selects a small batch $\bar{D}_t \subseteq D$ for human annotation, to expand coverage with minimal labeling cost. Human annotators provide trusted labels to update the model.

We assume annotators are reliable, as is typical in production settings with qualified and properly incentivized experts. If this assumption does not hold, additional quality-control mechanisms, e.g., cross-validation, can be applied, but are beyond our scope.

(2) Weak label refinement. If LFs are available, SALT can incorporate weak supervision to accelerate model training (the blue arrows). It applies them to generate weak labels, then reconciles conflicts and filters unreliable outputs using the human annotations collected so far, by adopting a logic-based framework with Label Refining Rules. This improves scalability while controlling noise propagation.

(3) Model training. Using human labels and refined weak labels, SALT updates $\mathcal{M}_t$ to $\mathcal{M}_{t+1}$. Early rounds rely more on weak labels to get started, while later rounds rely more on human labels to improve quality and support the final guarantee.

(4) Coverage checking. Finally, SALT checks whether the updated model $\mathcal{M}_{t+1}$ satisfies a certified coverage condition. Each annotated sample defines a certified local cover in embedding space. When these local covers together cover D , SALT terminates with a dataset-wide accuracy guarantee; otherwise, it proceeds to the next round.

Key properties. This design addresses the challenges of SAP: coverage checking and its termination condition certify the labeling accuracy on D ; coverage-driven selection routes only the most influential uncovered samples to humans; and conflict resolution anchored by human-labeled samples suppresses the noise of weak signals.

4 Accuracy Guarantees

This section establishes our theoretical foundation for SALT.

4.1 Objective and Overview

Given a model $\mathcal{M}$ trained on a labeled subset $\mathcal{D}_S$, we want to develop a verifiable condition under which $\mathcal{M}$ achieves certified dataset-wide labeling accuracy on $\mathcal{D}$. This amounts to identifying regions where the model is reliable, by combining local stability with global coverage, which together capture how well $\mathcal{M}$ fits the entire dataset. The key idea is to control how far errors can propagate: if every point lies close to a labeled point with stable predictions, then the entire dataset is well controlled.

This requires combining two types of information: (1) local sensitivity of the model around each point, and (2) global coverage of the dataset. Existing active learning criteria capture only one at a time. Uncertainty-based methods [49, 67, 74, 76] focus on local difficulty but do not provide dataset-level guarantees, while diversity-based methods (e.g., k -center greedy [69]) capture global spread but ignore that representativeness varies across regions and models. As a result, neither alone yields a verifiable stopping rule for selective annotation, and naive hybrids still treat them as separate heuristics. The example below illustrates the main challenge: local difficulty and global coverage must be unified in a single object.

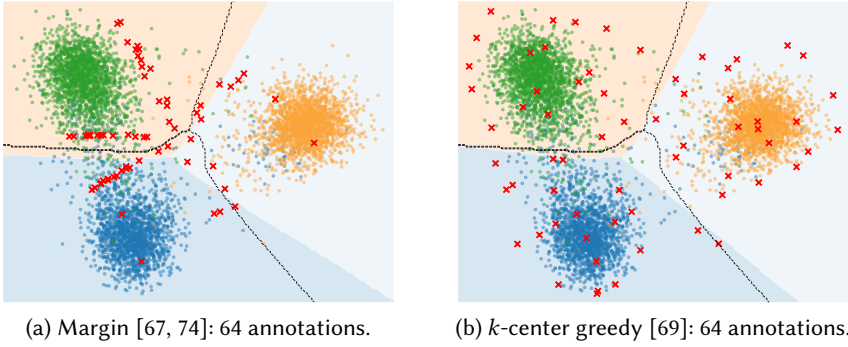


Fig. 3. Sample selection of different active learning strategies.

Example 1: Figure 3 shows a 2D projection of different sampling strategies on a three-class AGNews subset [37]. Given a budget of 64 annotations, (a) uncertainty-based Margin [74] prioritizes points near the decision boundary of the *current* model; however, it can distort the global geometry, *e.g.*, between the green and yellow classes, especially when the model is poorly initialized. (b) Diversity-based k -center greedy [69] captures global geometry better, but does not directly support the decision boundary refinement needed for high accuracy. Consequently, it may over-select points far from informative boundary regions, wasting human annotation effort. $\square$

Certified coverage framework: an overview. We recast selective annotation as *certified coverage* in embedding space. Each labeled point defines a *local cover*, *i.e.*, a neighborhood where the model’s loss changes only slightly. Global progress is then measured by the union of these covers. This provides a single geometric view that captures both local reliability and global coverage.

Local: curvature-adaptive influence. A labeled point $\mathbf{x}$ certifies the largest neighborhood where the loss stays within a given budget, defining its local region of confidence. Its size is *curvature adaptive*: in flat regions the loss changes slowly, so the region can expand, but in steep regions small changes can sharply increase loss, so the region must shrink. Thus, points near complex boundaries certify small covers, while points in stable regions certify larger ones.

Global: dataset-wide coverage. We extend this local notion to a global certificate by aggregating all such covers. An unlabeled point that is in at least one cover is *covered*, *i.e.*, its loss is close to that of a nearby labeled point. Uncovered points indicate where the model lacks guarantees and where new annotations are most useful. Once all points are covered, the desired accuracy guarantee follows.

4.2 Deriving Certified Coverage

We now formalize this intuition. We view the dataset as embedded points in $\mathbb{R}^k$ and derive the certificate in a local-to-global argument. Lemma 2 bounds the loss variation within a certified neighborhood of an annotated sample (steps (1)–(2)), making it a local representative of the points that it covers. Theorem 3 aggregates these local guarantees through a coverage assignment that associates every sample with such a representative; since each local loss deviation is bounded, their weighted sum bounds the dataset-wide risk gap (step (3)). Since accuracy (Equation 1) is a normalized form of empirical risk, the risk bound directly yields the accuracy guarantee (step (4)). Below we present a sketch of our main results, with full proofs and a visualized derivation chain in [80].

(1) Bounded loss variations. Let $\mathbf{x} \in \mathbb{R}^k$ be an embedded input sample. Write $\eta(\mathbf{x}) = \Pr[y \mid \mathbf{x}] \in \Delta^C$ for the conditional label distribution, and let $\ell(\mathbf{x}, \eta)$ be the cross-entropy loss. Given $\mathbf{x}$, the model $\mathcal{M}_\theta$ produces logits $z_\theta(\mathbf{x}) \in \mathbb{R}^C$ and class probabilities $p_\theta(\mathbf{x}) \in \Delta^C$. For two nearby embedded

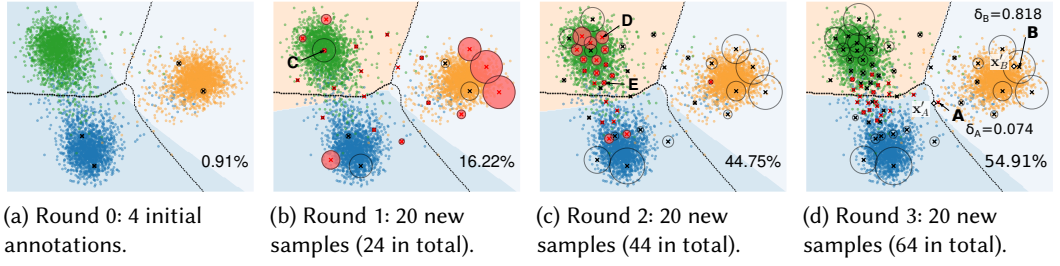


Fig. 4. Sample selection of SALT in 3 rounds. Newly annotated points are marked in red, with circles indicating their certified local covers.

points $\mathbf{x}$ and $\mathbf{x}'$, we want to quantify how much the loss can change inside a small neighborhood around $\mathbf{x}$. Under a standard smoothness assumption that $\eta(\cdot)$ is Lipschitz continuous [68], the next lemma controls this local loss variation.

Lemma 2: Assume that $\eta(\mathbf{x})$ is λ -Lipschitz continuous under $\|\cdot\|_2$. Fix $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^k$ with a maximum distance $\|\mathbf{x}' - \mathbf{x}\|_2 \leq \delta$. Then

$$|\ell(\mathbf{x}', \eta') - \ell(\mathbf{x}, \eta)| \leq b\delta + a\delta^2, \quad (2)$$

$$\text{with } a = \frac{1}{2}h\Gamma_\theta^2 + \lambda\Gamma_\theta, \quad b = \|\eta - p\|_2\Gamma_\theta + \lambda\|z\|_2.$$

Here $p = p_\theta(\mathbf{x})$, Γ_θ is the product of spectral norms across network layers of $\mathcal{M}_\theta$, and $h = \|H\|_2$ with $H = \text{diag}(q(\xi)) - q(\xi)q(\xi)^\top$ for some ξ on the segment between $\mathbf{x}$ and $\mathbf{x}'$. $\square$

Lemma 2 shows that within a ball $B(\mathbf{x}, \delta)$ around a labeled point, the loss variation is bounded by $e(\mathbf{x}, \delta, \theta) := a\delta^2 + b\delta$. This bound depends on the prediction mismatch $\|\eta - p\|_2$ and the local curvature h . They jointly control how far the point can certify its neighborhood.

Intuitively, this bound quantifies how “stable” the model is around $\mathbf{x}$: it is small when the model prediction already matches the local label distribution and the loss geometry is smooth, and large when the model is uncertain or the region is highly curved.

Proof sketch: Layer-wise Lipschitzness gives $\|z_\theta(\mathbf{x}') - z_\theta(\mathbf{x})\|_2 \leq \Gamma_\theta\delta$. A Taylor bound on the cross-entropy and the λ -Lipschitz continuity of $\eta(\cdot)$ then yield $|\ell(\mathbf{x}', \eta') - \ell(\mathbf{x}, \eta)| \leq b\delta + a\delta^2$. $\square$

(2) Local cover radius. We invert the loss bound to ask: how far can we move from a labeled point while keeping the loss within a tolerable budget? Fix a local loss budget $\epsilon > 0$. The *local cover radius* at $\mathbf{x}$ is the largest radius for which every point inside the ball remains within this ϵ . Equivalently, it is the maximum admissible neighborhood satisfying $e(\mathbf{x}, \delta, \theta) \leq \epsilon$. Solving for this yields

$$\delta_{\mathbf{x}}(\epsilon, \theta) = \frac{2\epsilon}{b + \sqrt{b^2 + 4a\epsilon}}. \quad (3)$$

This is the key curvature-adaptive quantity in SALT. It depends on the prediction mismatch $\|\eta - p\|_2$ and the curvature term h , and thus varies across the embedding space. It unifies confidence and representativeness: when $\|\eta - p\|_2$ is large or h is high, the denominator in Equation 3 gets larger, and $\delta_{\mathbf{x}}$ gets smaller, so the point certifies only a small neighborhood. When $\|\eta - p\|_2$ is small and curvature is mild, the denominator decreases, and $\delta_{\mathbf{x}}$ grows, allowing the point to represent a larger neighborhood.

Example 2: Figure 4d contrasts certified coverage with uncertainty- and diversity-based selection. Consider two illustrative unlabeled samples $\mathbf{x}'_A$ and $\mathbf{x}'_B$, each at distance $r = 0.2$ from its nearest annotated point, A and B, respectively. With $\epsilon = 0.05$, Equation 3 yields a small $\delta_A = 0.074$ at

boundary A , where $(\|\eta - p\|_2, h) = (0.115, 0.368)$, but a large $\delta_B = 0.818$ at stable-interior B , where $(\|\eta - p\|_2, h) = (0.013, 0.003)$. Prior methods fall short: uncertainty-based Margin prioritizes $\mathbf{x}'_A$ near the current decision boundary but provides no dataset-wide certificate, whereas diversity-based k -center greedy treats the two samples alike because both have distance r . SALT instead unifies local reliability and global coverage: it leaves $\mathbf{x}'_A$ uncovered since $r > \delta_A$, but certifies $\mathbf{x}'_B$ since $r \leq \delta_B$, guaranteeing that its loss deviation from B is at most ϵ . $\square$

(3) *Dataset-wide aggregation.* We now extend pointwise stability to a dataset-wide guarantee. Since each labeled point represents a different number of samples, we aggregate their contributions by weighting each center based on how many points it covers. Suppose that the annotated set $S \subseteq D$ induces local balls $B(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta))$ that cover D , i.e., $\mathcal{D} \subseteq \bigcup_{\mathbf{x}_i \in S} B(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta))$. To aggregate this cover, let $\tau : D \rightarrow S$ be a surjective map that assigns each $\mathbf{x}'_j \in D$ to a center $\mathbf{x}_i \in S$ with $\mathbf{x}'_j \in B(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta))$. The corresponding weight is

$$w_i = |\{\mathbf{x}'_j \in D \mid \tau(\mathbf{x}'_j) = \mathbf{x}_i\}|.$$

It counts the points covered by center $\mathbf{x}_i$. The empirical risk over $\mathcal{D}$ is $R_{\mathcal{D}}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} \ell(x, y)$. To relate this to training on $\mathcal{D}_S$, we define the *weighted empirical risk* over $\mathcal{D}_S$ as

$$\widehat{R}_{\mathcal{D}_S}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}_S} w_i \ell(\mathbf{x}_i, y_i).$$

This assigns each labeled point a weight equal to the number of points it represents under the cover.

Denote by $\theta_{\mathcal{D}} = \arg \min R_{\mathcal{D}}(\theta)$ and $\theta_S = \arg \min \widehat{R}_{\mathcal{D}_S}(\theta)$ the empirical risk minimizers (ERM) over labeled datasets $\mathcal{D}$ and $\mathcal{D}_S$, respectively. Assume uniform bounds $\Gamma_{\theta} \leq \Gamma$ on the product of layer spectral norms and $\|z(\mathbf{x}; \theta)\|_2 \leq B$ on the logit norm. We next show that complete certified coverage ensures that training on the labeled subset is nearly as effective as training on the full dataset, i.e., the full-data risk of θ_S remains close to that of $\theta_{\mathcal{D}}$.

Theorem 3: *Consider a model $\mathcal{M}_{\theta_S}$ trained over a labeled subset $\mathcal{D}_S$. Under the coverage condition $\mathcal{D} \subseteq \bigcup_{\mathbf{x}_i \in S} B(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta_S))$ with a surjective assignment induced by τ and weights w_i , we have:*

$$\begin{aligned} |R_{\mathcal{D}}(\theta_S) - R_{\mathcal{D}}(\theta_{\mathcal{D}})| &\leq \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}_i \in S} \left[w_i e(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta_S), \theta_S) \right. \\ &\quad \left. + w_i \widehat{e}(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta_S), \theta_{\mathcal{D}}) \right], \end{aligned} \quad (4)$$

where $\widehat{e}(\mathbf{x}, \delta, \theta) = (\sqrt{2}\Gamma + \lambda B)\delta + (\frac{1}{4}\Gamma^2 + \lambda\Gamma)\delta^2$. $\square$

Proof sketch: We split the risk gap into deviations between full-data and weighted risks. Since θ_S and $\theta_{\mathcal{D}}$ minimize $\widehat{R}_{\mathcal{D}_S}$ and $R_{\mathcal{D}}$, the gap $|R_{\mathcal{D}}(\theta_S) - R_{\mathcal{D}}(\theta_{\mathcal{D}})|$ is at most the sum of $|R_{\mathcal{D}}(\theta) - \widehat{R}_{\mathcal{D}_S}(\theta)|$ at $\theta = \theta_S$ and at $\theta = \theta_{\mathcal{D}}$. For each deviation, the assignment τ maps every point of D into the certified ball of its center, so Lemma 2 bounds its loss deviation from that center, and summing these bounds with weights w_i bounds the deviation. At θ_S this yields the e terms; at $\theta_{\mathcal{D}}$, which cannot be trained without the full labels, we bound $\|\eta - p\|_2 \leq \sqrt{2}$ and $h \leq 1/2$ uniformly, yielding $\widehat{e}$. $\square$

Theorem 3 is our main result, on which the certification framework relies. Intuitively, it connects local covers to a global guarantee. If every point in D is covered by a certified ball, then training on the labeled subset, with each center weighted by its coverage, closely matches training on the full dataset. Hence, complete certified coverage ensures that unlabeled data does not introduce additional risk, and as a result, further annotation becomes unnecessary.

The bound gets tighter as the local radii shrink. By Equation 3, $\delta_{\mathbf{x}}(\epsilon, \theta) \rightarrow 0$ as $\epsilon \rightarrow 0$, and e and $\widehat{e}$ scale as $O(\delta + \delta^2)$. Thus, smaller local variation yields a smaller global risk gap.

All terms of Theorem 3 are computable, and the bound can be checked directly; the corollary

below instantiates it for a well-trained model, giving the interpretable form in our stopping rule:

Corollary 4: *Under the coverage condition of Theorem 3, we have:*

$$|R_{\mathcal{D}}(\theta_S) - R_{\mathcal{D}}(\theta_{\mathcal{D}})| \leq 2\epsilon, \quad (5)$$

when θ_S and $\theta_{\mathcal{D}}$ recover the true conditional probabilities on S , i.e., $p_{\theta_S}(\mathbf{x}_i) = p_{\theta_{\mathcal{D}}}(\mathbf{x}_i) = \eta(\mathbf{x}_i)$ for every $\mathbf{x}_i \in S$. $\square$

Intuitively, this assumes that $\mathcal{M}_{\theta}$ is expressive enough and well-trained to capture the true distribution over $\mathcal{D}_S$, which is generally reasonable in practice as long as $\mathcal{M}_{\theta}$ has sufficient capacity.

(4) Certified accuracy. We convert the risk bound into the selective-annotation accuracy target. By $\text{acc}(\mathcal{M}_{\theta}, \mathcal{D}) = 1 - R_{\mathcal{D}}(\theta)/\ln C$,

$$\text{acc}(\mathcal{M}_{\theta_{\mathcal{D}}}, \mathcal{D}) - \text{acc}(\mathcal{M}_{\theta_S}, \mathcal{D}) \leq \frac{|R_{\mathcal{D}}(\theta_S) - R_{\mathcal{D}}(\theta_{\mathcal{D}})|}{\ln C}.$$

Thus, bounding the risk gap directly bounds the accuracy gap. To enforce slack ζ , it suffices to require the risk gap $\leq \zeta \ln C$. By Corollary 4, this can be done by setting $\epsilon = \frac{1}{2}\zeta \ln C$. Thus, ϵ is the local loss budget that ensures the desired global accuracy slack.

Termination condition for SALT. The stopping rule is now immediate. Given accuracy slack ζ , we set $\epsilon = \frac{1}{2}\zeta \ln C$, which splits the global accuracy budget evenly across local regions, so that local deviations do not accumulate beyond the target slack. SALT then checks whether the labeled set induces complete certified coverage. If every point in D is in $B(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta_S))$ for some $\mathbf{x}_i \in S$, then the model meets the accuracy target, and the HITL loop stops.

Practical concerns. The guarantee is a sufficient condition based on bounded spectral norms and Lipschitz continuity of $\eta(\cdot)$. These should be viewed as modeling assumptions for certification, not properties that always hold in practice. When local geometry is irregular or labels change sharply, the bound may be conservative and require smaller ϵ to achieve the same slack. This said, it remains useful when representations are smooth and local behavior is stable.

The Lipschitz constant λ is dataset-dependent. In practice, we estimate it via labeled samples: $\lambda = \max_{i \neq j} |\eta(x_i) - \eta(x_j)| / \|x_i - x_j\|$.

Beyond classification. The certification above is not specific to classification: it only sees samples as points in the embedding space, and a loss with bounded local variation (Lemma 2). To support a new labeling task, it suffices to replace the classification loss with a task-specific loss and re-derive the certified cover radius via Equation 3; the coverage-based selection and termination then carry over. For instance, a smooth regression loss (e.g., squared loss) yields a Lemma 2-style bound, turning the certification target into a bounded regression risk; structured tasks decomposable into local decisions, e.g., named entity recognition (per-token labeling) and entity resolution (per-pair classification), fit the framework easily.

5 Adaptive Sample Selection

The certified coverage framework from Section 4 provides a stopping rule for selective annotation: once the certified local covers induced by the labeled set fully cover D , the target labeling accuracy is guaranteed. The remaining challenge is thus algorithmic: how to expand certified coverage with as few annotations as possible.

This section formulates round-wise selective annotation and develops a greedy algorithm with a coverage-adaptive scoring rule that prioritizes samples contributing most to uncovered regions.

Progressing towards certified coverage. This problem is non-trivial because coverage depends on labels that are unknown before annotation. Once new annotations are obtained, they can change

Algorithm 1: SALT: sample selection at round t .**Input:** Unlabeled dataset D , annotated samples $\tilde{D}_{t-1}$, current model $\mathcal{M}_{t-1}$, tolerance ζ , round budget bgt.**Output:** Selected samples D_t .

```

/* Compute current coverage and identify uncovered, candidate samples. */
1  Compute  $\mathbb{B}_i \leftarrow B(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\frac{1}{2}\zeta \ln C, \mathcal{M}_{t-1}))$  for each  $\mathbf{x}_i$  in  $\bigcup_{i=1}^{t-1} D_i$ ;
2  Candidate samples  $\psi_0 \leftarrow D \setminus \bigcup_{i=1}^{t-1} \mathbb{B}_i$ ;  $\psi \leftarrow \psi_0$ ;  $D_t \leftarrow \emptyset$ ;
3  for  $k = 1$  to bgt do
4       $\mathbf{x}^{(k)} \leftarrow \arg \max_{\mathbf{x} \in \psi} \text{score}(\mathbf{x}, \psi)$ ; /* Greedy selection by scoring. */
5       $\mathbb{B}^{(k)} \leftarrow B(\mathbf{x}^{(k)}, \delta_{\mathbf{x}^{(k)}}(\frac{1}{2}\zeta \ln C, \mathcal{M}_{t-1}))$ ; /* Get pseudo coverage. */
6       $\psi \leftarrow \psi \setminus \mathbb{B}^{(k)}$ ; /* Update candidate samples. */
7  return  $D_t \leftarrow \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(\text{bgt})}\}$ ;

```

the model's predictions and the coverage geometry, so a batch that appears useful before annotation may no longer be optimal after retraining. Thus, SALT adopts an HITL procedure: at each round, it selects samples using the current model, obtains their labels from human annotators, updates the model with the newly labeled data, and then re-evaluates the certified coverage condition.

Following common practices in active learning, we assume a small labeled set for model initialization and a fixed annotation budget bgt in each round. Consider sample selection in round t :

- *Input:* Unlabeled D , annotated $\tilde{D}_{t-1}$ from previous rounds, model $\mathcal{M}_{t-1}$ trained on $\tilde{D}_{t-1}$, tolerance ζ , and round budget bgt.
- *Output:* A subset $D_t \subseteq D$ to be annotated in round t .
- *Objective:* To maximize the certified coverage attained by the updated model $\mathcal{M}_t$ trained on $\tilde{D}_t = \tilde{D}_{t-1} \cup D_t$.

This round-wise selective annotation problem (RSAP) is intractable, even under the assumption that the predicted labels of all samples in D_t exactly match the eventual human annotations.

Theorem 5: RSAP is NP-hard. □

Proof sketch: We prove the NP-hardness by reduction from set cover (see [80] for details), which is NP-complete (cf. [31]). □

Despite this, we develop a greedy algorithm for round-wise sample selection, guided by the certified coverage criterion.

- (1) It restricts the candidate pool to uncovered points, since points that are already covered are already controlled at the target tolerance and are therefore less likely to improve certified coverage.
- (2) It adapts selection to the current coverage state. Early rounds should favor points that cover large uncovered regions to quickly recover the dataset's coarse structure, while later rounds focus on harder regions near decision boundaries or high-curvature areas that require finer refinement. This transition should be driven by the coverage state itself, rather than by manually defined stages.

Algorithm. In each round t , SALT selects a batch of bgt samples to maximize coverage under model $\mathcal{M}_{t-1}$. Each candidate is scored by its expected contribution to coverage. The score initially favors samples covering many uncovered points and increasingly prioritizes harder regions with lower predictive quality as coverage grows.

Algorithm 1 constructs D_t through greedy coverage expansion. It first evaluates coverage under $\mathcal{M}_{t-1}$ by computing the radius $\delta_{\mathbf{x}_i}$ and coverage ball $\mathbb{B}_i$ for each labeled sample $\mathbf{x}_i$ (line 1). The remaining uncovered samples form the candidate pool ψ (line 2). The algorithm then selects bgt samples greedily from ψ . At each step, it chooses $\mathbf{x}^{(k)}$ with the highest score, balancing coverage gain against boundary refinement (line 4). It estimates the gain using a pseudo coverage ball $\mathbb{B}(\mathbf{x})$, treating

$\mathcal{M}_{t-1}(\mathbf{x})$ as a proxy label (line 5). After selecting $\mathbf{x}^{(k)}$, it removes the samples in its pseudo coverage region from ψ to avoid redundancy (line 6). After bgt steps, it returns D_t for annotation (line 7).

Coverage-adaptive scoring. The greedy procedure uses $\text{score}(\cdot)$ to rank samples $\mathbf{x}$ in ψ . Following the design above, the score balances global coverage expansion with predictive-quality improvement as coverage increases. Formally, we define the function as

$$\text{score}(\mathbf{x}, \psi) = \ln |\psi \cap \mathbb{B}(\mathbf{x})| + \alpha \left(1 - \frac{|\psi|}{|D|}\right) \ln |\eta'(\mathbf{x}) - p(\mathbf{x})|_2, \quad (6)$$

where $\mathbf{x} \in \psi$ is a candidate sample; $\mathbb{B}(\mathbf{x})$ is its pseudo coverage ball with radius $\delta_{\mathbf{x}}(\frac{1}{2}\zeta \ln C, \mathcal{M}_{t-1})$; $\eta'(\mathbf{x})$ is the one-hot vector corresponding to $\arg \max p(\mathbf{x})$; and α is a hyperparameter.

The first term, $\ln |\psi \cap \mathbb{B}(\mathbf{x})|$, estimates the coverage gain by counting the candidates, assuming that $\mathbf{x}$ is annotated consistently with $\mathcal{M}_{t-1}$. Maximizing this term therefore favors samples that expand coverage most. The second term, $\ln |\eta'(\mathbf{x}) - p(\mathbf{x})|_2$, measures predictive quality at $\mathbf{x}$. A larger value indicates that the prediction is farther from a confident one-hot distribution and thus identifies a lower-quality point, often near a decision boundary.

The factor $1 - \frac{|\psi|}{|D|}$, i.e., the current coverage ratio, modulates the quality term. At low coverage, this factor is small, so the coverage term dominates and favors rapid global expansion. As coverage grows, the factor increasingly emphasizes predictive quality.

Properties. Our selection strategy has the following properties.

Bounded approximation. Under the optimistic assumption that the predicted labels of selected D_t match their human annotations, Algorithm 1 is a $(1 - \frac{1}{e})$ -approximation algorithm for RSAP.

Theorem 6: *Given candidate samples ψ_0 at round t , define the expanded coverage of a subset A of ψ_0 as $f(A) := |\bigcup_{\mathbf{x} \in A} (\psi_0 \cap \mathbb{B}(\mathbf{x}))|$. Then we have $f(D_t) \geq (1 - \frac{1}{e}) \max_{A \subseteq D, |A| \leq \text{bgt}} f(A)$.* $\square$

Proof sketch: Greedy selection with $\text{score}(\cdot)$ maximizes the marginal gain of $f(\cdot)$ when $\alpha(1 - \frac{|\psi|}{|D|}) = 0$. It thus coincides with the classical greedy algorithm for set cover on universe ψ_0 , with collection of sets $\{\psi_0 \cap \mathbb{B}(\mathbf{x}) : \mathbf{x} \in \psi_0\}$ and $|A| \leq \text{bgt}$. Since $f(\cdot)$ is monotone and submodular, the standard result for greedy coverage [55] provides the constant approximation ratio $1 - \frac{1}{e}$. $\square$

Adaptive selection focus. The strategy adapts to the current coverage level. In the early stages, when coverage is low, $\alpha(1 - \frac{|\psi|}{|D|}) \approx 0$, so the first term dominates. Consequently, the greedy strategy reduces to classical coverage maximization and inherits its approximation guarantee, prioritizing samples with large coverage gain from dense uncovered regions that rapidly recover the dataset's coarse structure. In later stages, as coverage increases, the second term in Equation 6 becomes increasingly important, and the selection focus naturally shifts from expansion to refinement. At that point, the remaining candidates tend to concentrate near decision boundaries or in high-curvature regions. Annotating such hard points improves the model after retraining, which can further enlarge certified coverage in subsequent rounds.

Example 3: Continuing Example 2, Figure 4 shows the adaptive behavior of SALT: newly selected samples are highlighted as red crosses; we mark their certified coverage regions as red circles and the coverage ratios $1 - \frac{|\psi|}{|D|}$ in the right lower corners. With more annotations and growing coverages, SALT naturally shifts from reconstructing global data distribution to refining high-curvature decision-boundary, without manually specifying stage transitions.

In Round 1, SALT selects points such as C, whose pseudo radius (dashed circle) covers 767 samples, because the initially low coverage ratio ($< 1\%$) makes the coverage term dominate the score and favor large pseudo coverage gains. These coverage-oriented selections raise the coverage

ratio to 16.22%, automatically increasing the weight of the quality term in Round 2. Accordingly, SALT selects D and E in Round 2 because they combine large coverage gains with low predictive quality: D has a coverage gain of 3.892 and uncertainty of -2.414 , whereas E has 1.792 and -0.501 , respectively. In the final round, SALT selects points in high-curvature decision-boundary regions (e.g., A), even though each newly selected point may cover only a few samples (e.g., A covers 4 samples only), because the quality term now dominates the score. $\square$

Beyond diversity or uncertainty. Equation 6 unifies diversity and uncertainty through the same geometric quantity, namely, $\|\eta' - p\|_2$. A smaller discrepancy yields larger coverage, while a larger one signals low predictive quality and proximity to decision boundaries. Thus, coverage expansion and quality improvement are not separate heuristics, but two views of the same geometry.

Complexity. Let $n = |D|$ and $m = |\tilde{D}_{t-1}|$. At each round, computing coverage balls for the m labeled samples takes $O(m)$ time. Each of the bgt greedy steps then scores all candidates in ψ , yielding a worst-case time of $O(m + \text{bgt} \cdot n^2)$. In practice, the cost decreases across rounds as ψ shrinks with increasing coverage.

Hyper-parameter tuning. The scoring function introduces α to balance coverage gain and predictive quality (Equation 6). Its value is dataset-dependent, as the two terms may differ in scale. In practice, we set α so the two terms have comparable influence at the start of selection, preventing either from dominating early, while allowing the adaptive factor $1 - \frac{|\psi|}{|D|}$ to gradually shift the focus from coverage expansion to boundary refinement (see Section 7).

6 Integrating Weak Supervision

This section shows how SALT integrates weak supervision to reduce human cost when labeled data is scarce. It introduces a logic-based correction mechanism that leverages noisy labeling functions while controlling their errors, and supports effective early-stage learning without compromising the certified accuracy guarantee.

6.1 Weak Supervision with Labeling Functions

We first present weak supervision via labeling functions, which provide scalable but noisy labels to accelerate early-stage learning.

Weak supervision. Weak supervision [8, 60, 63] derives approximate labels from labeling functions (LFs), such as heuristics or external tools. For example, sentiment analysis may match keywords such as “excellent” to identify positive texts. In dataset labeling, LFs provide auxiliary supervision that is especially valuable when human labels are scarce: they expand limited “gold” labels in early rounds, speed up convergence, and reduce annotation cost.

Formally, let $D = \{x_i\}_{i=1}^n$ be an unlabeled dataset. Weak supervision produces a matrix $\Lambda \in \mathbb{W}^{n \times m}$ over m available LFs, where each LF $_j$ outputs a weak label $\lambda_{ij} \in \mathbb{W} = \{1, \dots, C, \perp\}$ for x_i . Here C is the number of possible classes, and $\perp$ denotes absence of a label assignment, i.e., LF $_j$ does not assign a label to x_i .

Caveats and remedies. Weak supervision is called “weak” precisely because individual LFs can have errors that are hard to detect: conflicts arise across LFs, and mislabels may propagate, distort the learned geometry, and ultimately violate the accuracy objective. SALT addresses this with two mechanisms: (1) logical rules (LRRs) resolve LF conflicts and filter unreliable signals by combining LF outputs, inter-sample relations, and gold-calibrated soft labels; and (2) the model is trained with both refined weak labels and human annotations, while the weight of weak supervision gradually decreases.

6.2 Logic-based Correction for Weak Labels

We now introduce a logic-based correction mechanism, based on LRRs, to resolve conflicts and reduce noise in weak labels. Note that LRRs serve only to improve weak-label quality and bootstrap early-stage learning when human labels are scarce; the certified guarantees of SALT are derived from human annotations and certified coverage, and do *not* depend on LRRs or weak labels.

Logic-based correction. We define LRRs, starting from predicates.

Predicates. A predicate over dataset D and its weak labels Λ is

$$p ::= D(x) \mid T(\text{LF}, x, w) \mid \neg T(\text{LF}, x, w) \mid M(x, x') \mid \text{Pr}(x, w) \otimes c,$$

where x, x' range over samples in D , referred to as variables bounded by $D(x)$ and $D(x')$; $T(\text{LF}, x, w)$ indicates that LF assigns label $w \in \mathbb{W}$ to x ; $\neg T(\text{LF}, x, w)$ indicates otherwise; $M(x, x')$ denotes a relation between samples (e.g., similar topics); and $\text{Pr}(x, w) \otimes c$ is a soft-label predicate, where $\text{Pr}(x, w)$ is the calibrated probability of x having label w , $\otimes \in \{>, <, \geq, \leq\}$, and $c \in [0, 1]$ is a threshold.

We elaborate the predicate types as follows.

(1) *Tool predicate* $T(\text{LF}, x, w)$. It returns true if $\text{LF}(x) = w$, making use of the output of an LF. LFs may include heuristics such as regular expressions or keywords for text classification, and metadata filters or pattern matchers for images [60]. More advanced LFs may also use frequency, co-occurrence or pattern order. More generally, SALT can treat outputs from existing weak supervision tools as LFs, e.g., WeaSEL [66], FlyingSquid [27] and MeTaL [7].

(2) *Relationship predicate* $M(x, x')$. It returns true if a model M determines that x and x' satisfy a pre-defined relation. For text, e.g., M_{topic} may identify whether two samples share the same topic [79]. For images, M_{geo} may use metadata to infer whether samples were captured near the same location [13]. Such relations propagate information across samples and can improve weak-label quality.

(3) *Soft label predicate* $\text{Pr}(x, w) \otimes c$. Here $\text{Pr}(x, w)$ is the probability of x having label w , predicted by a model $\mathcal{P}$ that fuses weak labels Λ and human annotations. The predicate returns true if $\text{Pr}(x, w) \otimes c$ holds, i.e., the probability satisfies the threshold c under operator $\otimes$.

The fusion model $\mathcal{P}$ follows the intuition behind Snorkel LabelModel [62]: agreement patterns among noisy LFs reveal source reliability and enable soft-label inference. We extend this with gold-label calibration, using the available human annotations as strong signals so that the estimated reliabilities are anchored by trusted labels rather than weak-label statistics alone.

Concretely, model $\mathcal{P}$ takes as input the weak label vector produced for a sample x_i by all LFs, i.e., the i -th row of Λ , and outputs a probability $\text{Pr}(x_i, w)$ for each label $w \in \mathbb{W}$. Internally, $\mathcal{P}$ introduces a latent true label and treats each LF as a noisy source with its own class-conditional error profile, so the posterior is obtained by combining the estimated reliabilities of all LFs.

We train $\mathcal{P}$ with a joint objective over weak and strong labels (see [80]). Weak labels learn LF reliabilities from agreement patterns, while human annotations serve as ground truth for calibration. As a result, $\mathcal{P}$ is more robust than naive voting, downweighting noise and producing calibrated soft labels suitable for LRRs.

Label Refining Rules. An LRR φ has the following form:

$$\varphi : X \rightarrow x.\text{label} = w,$$

where X is a conjunction of predicates over D and Λ , and x is a variable ranging over samples in D (i.e., $D(x)$ appears in X). The atom $x.\text{label} = w$ assigns a refined label $w \in \mathbb{W}$ to x . We refer to X as the *precondition* and $x.\text{label} = w$ as the *consequence* of φ .

An LRR φ is *conflict-resolving* if its consequence assigns a valid class label $w \in \{1, \dots, C\}$. If $w = \perp$, then φ is a *filtering rule*, as it blocks unreliable weak labels from propagating downstream.

Intuitively, an LRR encodes when a weak label should be trusted, corrected or filtered based

on a structured pattern of signals in X . Tool predicates indicate which LFs are applied, soft-label predicates summarize confidence from the fusion model, and relationship predicates provide supporting or contradicting evidence from related samples. In this sense, an LRR synthesizes *prior experience* about how these signals combine to produce reliable labels.

LRRs make a robust refinement layer. They (1) confirm a label if sources agree, (2) override weak heuristics with stronger evidence, (3) propagate support through relationships, or (4) filter labels (with $\perp$) when signals conflict. By resolving conflicts before training, LRRs yield more stable supervision for downstream learning.

Example 4: Below are LRRs mined from the THUCNews dataset [33]. They are representative rather than exhaustive: LRRs are discovered automatically, and the three rules here only illustrate the major categories of discovered predicates, *i.e.*, tool, relationship and soft-label predicates, respectively.

(1) $\varphi_1 : D(x) \wedge T(\text{LF}_{\text{finance}}, x, \text{finance}) \wedge T(\text{LF}_{\text{athletes}}, x, \text{sports}) \rightarrow x.\text{label} = \text{sports}$. This LRR states that a text sample x should be classified as sports when both financial expressions and athlete keywords are present, despite the fact that it has some finance-specific expressions. It suggests that, in THUCNews, sports keywords are more indicative than finance terms. This rule is used to resolve conflicts between LFs and produce a more reliable label.

(2) $\varphi_2 : D(x) \wedge D(x') \wedge T(\text{LF}_{\text{stock}}, x', \text{stocks}) \wedge M_{\text{topic}}(x, x') \rightarrow x.\text{label} = \text{stocks}$. This LRR assigns label stocks to x if it shares a topic with x' , which is identified as stocks due to stock-market signals. It propagates label evidence across related samples, reflecting that topic-consistent texts tend to share labels. It is used to transfer reliable signals and improve labeling consistency across samples.

(3) $\varphi_3 : D(x) \wedge T(\text{LF}_{\text{scitech}}, x, \text{tech}) \wedge \text{Pr}(x, \text{tech}) < 0.1 \rightarrow x.\text{label} = \perp$. This LRR filters the keyword-based label technology when the fusion model assigns low confidence (< 0.1). It removes unreliable weak labels for x when supporting evidence is weak. This rule prevents low-confidence signals from propagating into training. $\square$

Semantics. A valuation h of an LRR φ over a dataset D and its weak labels Λ is a mapping that assigns each variable x of φ to a sample $h(x)$ in D , and each variable w to a label in $\mathbb{W}$.

We say that valuation h satisfies a predicate p , denoted $h \models p$, if one of the following conditions holds: (1) If p is $D(x)$, then $h(x)$ is a sample of $\mathcal{D}$. (2) If p is $T(\text{LF}, x, w)$ (or $\neg T(\text{LF}, x, w)$), the tool LF assigns (or does not assign) w to the sample $h(x)$. (3) If $p = M(x, x')$, the model M predicts that $h(x)$ and $h(x')$ satisfy the target relation. (4) If $p = \text{Pr}(x, w) \otimes c$, the fusion model assigns the sample $h(x)$ a soft label w with confidence satisfying threshold c .

For precondition X , $h \models X$ iff $h \models p$ for all predicates p in X . For an LRR $\varphi = X \rightarrow q$, $h \models \varphi$ iff $h \models X$ implies $h \models q$. Dataset $\mathcal{D}$ with Λ satisfies φ , written $\mathcal{D} \models \varphi$, if all valuations h of φ over $\mathcal{D}$ and Λ satisfy it. For a set Σ of LRRs, we write $\mathcal{D} \models \Sigma$ if $\mathcal{D} \models \varphi$ for all $\varphi \in \Sigma$.

Remarks. In practice, we maintain an offline set Σ of LRRs for online HITL use, including both expert-specified and mined rules.

(a) *Rule mining.* We mine LRRs offline from prior similar labeling tasks, treating human annotations as ground truth. The mining process follows a levelwise search in the predicate space. While classic anti-monotonicity is not effective here due to soft predicates and cross-sample relations, we employ *conservative and adaptive pruning*: we apply loose support thresholds to retain candidate rules, and further filter them using confidence and coverage impact estimated on human-labeled data. Pruning is relaxed when support is low and gets stricter as support for an LRR increases. This preserves important rules and reduces expensive search (see [80] for details). Mined rules are thus validated against trusted human annotations before use; they are revalidated as annotations accumulate across HITL rounds, and rules whose confidence degrades are dropped.

(b) *Rule conflict handling.* Conflicts may arise when multiple LRRs apply to the same sample with different consequences. Although such cases are rarely encountered in practice, we include a lightweight handling strategy for completeness. For each sample, we rank the applicable rules by an evidence-aware score that combines rule confidence on human-labeled data, rule specificity, and agreement with the fusion model's soft labels. We then select the consequence of the highest-ranking rule. If the evidence is weak or the top candidates remain too close, we abstain and assign $\perp$ rather than forcing a label. Details are deferred to [80].

The learned rules also admit a consistency analysis for conflict detection, via standard dependency-analysis methods [24–26].

LRRs in the HITL pipeline. Weak-label refinement with LRRs integrates naturally into the SALT HITL pipeline, as the core of *Conflict Resolution* (Figure 2). It takes as input the current human annotations, the weak-label matrix Λ from all LFs, and a rule set Σ , and produces one refined weak label per sample in D . Human annotations always take precedence and are never overridden; conflicting weak labels are resolved only with validated LRRs, and are otherwise discarded (assigned $\perp$).

Operationally, the fusion model $\mathcal{P}$ is first updated using the current human annotations. Rule predicates are then evaluated using the updated soft labels from $\mathcal{P}$, LF outputs in Λ , and relationship-model outputs. When a sample matches an LRR, its consequence gives us a refined label, which is passed to the model-training stage.

These refined labels provide additional supervision, especially in early rounds when human annotations are scarce. By filtering noise and consolidating reliable signals, LRRs support more effective bootstrapping, improve the initial model quality, and reduce the number of samples that require human annotation.

Training with weak and strong signals. In each HITL round, after obtaining refined weak labels and human annotations, we train the labeling model $\mathcal{M}_\theta$ by combining both. Weak labels provide scalable but noisy supervision, while human annotations are reliable but limited. Thus, training should leverage weak labels for early bootstrapping without letting their noise dominate.

We adopt weighted training [11, 47]. Let $\mathcal{D}_W$ be the refined weakly labeled set and $\mathcal{D}_S$ be the human-labeled set, we optimize

$$\mathcal{L}(t) = e^{-\rho t} \sum_{(x, \tilde{y}) \in \mathcal{D}_W} \ell(x, \tilde{y}) + (1 - e^{-\rho t}) \sum_{(x, y) \in \mathcal{D}_S} \delta_x \ell(x, y)$$

as the epoch-dependent objective, where t is the training epoch, δ_x is the local cover radius of x , and $\rho > 0$ is the decay rate. The factor $e^{-\rho t}$ directly controls the weight of weak supervision at epoch t , decaying over time to shift training toward the more reliable $\mathcal{D}_S$. Hence the influence of weak labels diminishes across HITL rounds (as $\mathcal{D}_S$ grows); the final labeling accuracy is governed by the certified coverage framework rather than weak-label quality.

Complexity. The training is dominated by forward and backward passes of $\mathcal{M}_\theta$ over $\mathcal{D}_W$ and $\mathcal{D}_S$. For T epochs, the total cost is $O(T(|\mathcal{D}_W| + |\mathcal{D}_S|) \cdot C_M)$, where C_M is the per-sample cost of one optimization step. Thus, adding refined weak labels does not change the training complexity; it just increases the size of the training set.

7 Experimental Study

Using real-life datasets, we evaluated SALT for its (a) labeling accuracy, (b) human annotation cost, and (c) efficiency compared with active learning. We also empirically tested its sensitivity to hyperparameters and its effectiveness in weak-label integration.

Experimental settings. We start with the settings.

Datasets. We evaluated SALT on five real-world classification datasets: THUCNews [33], DBPedia [48],

Table 1. Summary of classification datasets.

Dataset	Type	#Samples	#Classes	#Init / bgt	α
THUCNews	Chinese news articles (~907 chars)	68.6k	14	10 / 40	3.0
DBPedia	Wiki articles (~48 words)	70k	14	10 / 30	3.0
RESISC45	Images: remote sensing	31.5k	45	100 / 100	5.0
CIFAR100	Images: vehicles/animals	50k	100	200 / 250	1.0
Mercari	E-commerce product (~30 words)	1.3M	10	10 / 500	5.0

RESISC45 [17], CIFAR100 [46], and Mercari [52]; Table 1 summarizes their characteristics and initial labeled sets. We used each dataset’s training split and treated its built-in labels as ground truth. To simulate selective annotation, we initially hid all labels and used an oracle to reveal the ground-truth label of each selected sample. We randomly sampled each initial labeled set once and shared it across all methods, following standard active-learning practice [6, 10].

Labeling models. For text datasets, we used Qwen text embedding v4 [84] to compute a 768-dimensional embedding for each document. For image datasets, we used DINOv2 [57] to extract a 768-dimensional feature vector from each image. We then fed these embeddings into a four-layer MLP for classification. In each round, the models of all methods were reinitialized randomly [6], i.e., we used cold-start rather than warm starting from the previous model [5].

Baselines. We tested SALT against SOTA active learning methods:

- *Uncertainty-based:* (1) Margin [67], selecting samples with the smallest gap between the top two predictions; and (2) Entropy [81], selecting samples with the highest predictive entropy.
- *Diversity-based ones,* including (3) KCGreedy [69], which prioritizes samples farthest from the labeled set; and (4) TypiClust [44], which selects samples difficult to distinguish from labeled ones.
- *Hybrid solutions,* including (5) BADGE [6], a k -means++ method based on gradient embeddings w.r.t. the model’s output weights; (6) UHerding [10], which favors samples farthest from labeled ones with uncertainty regularization; and (7) ALFAMix [59], which selects samples via feature-space interpolation.
- *Unsupervised* (8) TEMI [3], one of the SOTA deep-clustering methods as a zero-annotation reference. It self-distills probabilistic assignments from paired embeddings and evaluates NCE accuracy after Hungarian cluster-label matching with ground truth.
- (9) ActiveDP [32], which labels textual data using LFs interactively authored by users. Here we follow its original simulated-user protocol: given a sample, the user returns an unseen, low-error keyword LF, sampled in proportion to its coverage.

We further include (10) SALT_{noWS}, which disables weak-label integration in SALT; (11) SALT_{cov} and (12) SALT_{unc}, which remove the uncertainty and the coverage terms, respectively, in greedy scoring.

To test SALT in weak-label integration, we evaluated the SOTA weak supervision methods, including Snorkel [60], FlyingSquid [27], ActiveDP’s ConFusion [32], majority voting, and random.

SALT configurations. For coverage-adaptive scoring, we set the default round budget bgt and α separately for each dataset, as listed in Table 1. We also set the decay rate to $\rho = 0.5$ for weighted training with integrated weak labels and selected human annotations. The Lipschitz coefficient λ for each dataset is estimated per round.

For text classification datasets, we used a set of curated LFs from WRENCH [85] and a few keyword-based LFs generated by ChatGPT. Since there are no high-quality LFs available for RESISC45 and CIFAR100, we disabled weak-label integration in those cases.

To integrate weak labels over THUCNews and DBPedia, we used 21 and 16 mined LRRs respectively.

Testbed. Our testbed server is powered by 2× Intel Xeon Gold 6254 @3.10GHz CPUs, each with 16

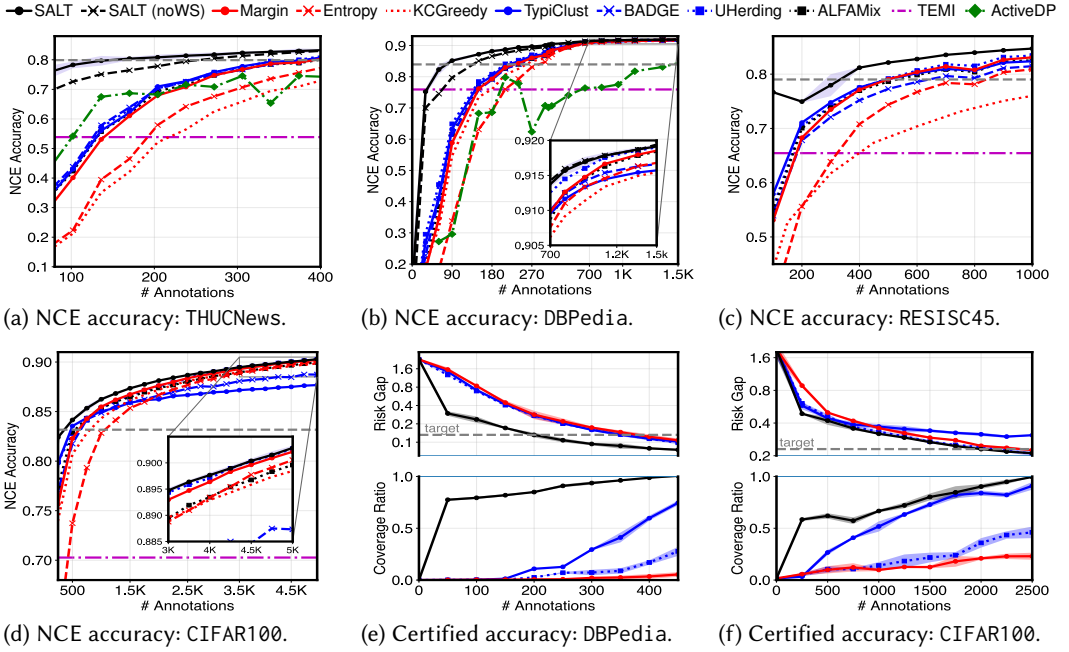


Fig. 5. NCE accuracy and certified accuracy of SALT. The shaded bands indicate its 95% confidence intervals.

physical cores and hyperthreading, and 64 GB DDR4 RAM. It also has an NVIDIA Tesla V100 32GB GPU. The operating system is Ubuntu 22.04.2 LTS with Linux kernel 5.15.0, CUDA 12.2, Python 3.10.19 with PyTorch 2.91. Each experiment is repeated five times, and the average is reported here.

Experimental results. We next report our findings. To verify that SALT addresses the core challenges outlined in Section 1, we evaluated the certified accuracy (Exp-1), the human annotation cost (Exp-2), and the noise control in weak-label integration (Exp-3). We further studied the overall efficiency of the SALT pipeline (Exp-4), its scalability to larger datasets (Exp-5), its sensitivity to hyperparameters (Exp-6), and ablation studies (Exp-7).

Exp-1: Accuracy. We compare labeling accuracy of SALT against active learning baselines and ActiveDP. We find the following.

Overall labeling accuracy. Figures 5a–5d report labeling NCE accuracy across datasets. As annotations increase, SALT is consistently the most accurate, with larger gains on more challenging datasets. Narrow 95% confidence intervals confirm SALT’s stability; baseline variances are similar, but their intervals are omitted for legibility.

(1) On the 14-class DBPedia, SALT is 16.7–72.4 points more accurate than active learning in the low-label setting (e.g., < 120 annotations). As more labels are added, the gap narrows to 2.1–6.1 points, but SALT remains the best throughout. The same pattern appears across datasets. With only 320 annotations over the 14-class THUCNews, SALT reaches 82% NCE accuracy, 6.2–19.5 points higher than active learning without weak supervision; its final accuracy is 1.9–9.2 points higher than the baselines. These results show that coverage-driven selection improves dataset-wide labeling quality, prioritizing samples that are both informative and representative.

(2) The final gap is smaller on both image datasets (0.36–7.8 points), due to their class-balanced distribution and well-separated clusters in the embedding space. Thus, most methods become competitive with sufficient labels. This said, SALT remains up to 31.2 percentage points more

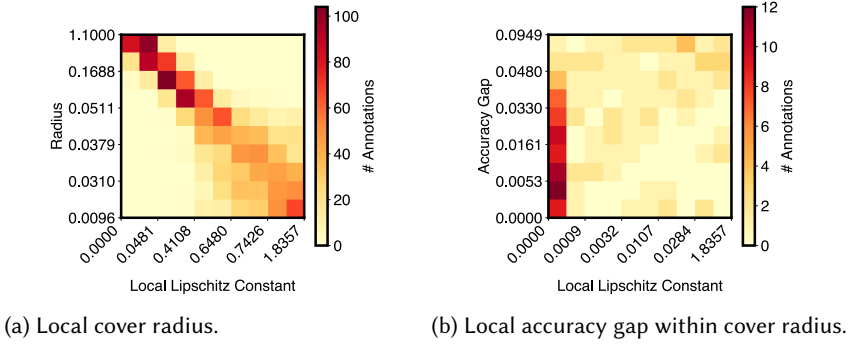


Fig. 6. Sensitivity to local Lipschitzness.

accurate with fewer than 300 annotations.

(3) The absolute accuracy gap among all methods narrows in the high-label setting, while SALT remains the most accurate throughout. This is expected. The labeling accuracy of all methods is upper-bounded by the accuracy of the model trained on the fully annotated dataset. With more annotations, all methods approach this common upper bound, and thus the performance gap naturally diminishes; convergence is faster on the class-balanced CIFAR100, where all methods recover more quickly from suboptimal early selections.

(4) As a zero-annotation reference, TEMI attains only 53.9–75.9% NCE accuracy across datasets, far below the accuracy targets, since the clusters discovered without supervision do not necessarily align with the target classes. Hence, certified dataset labeling requires human supervision, whose cost SALT minimizes.

(5) Figures 5a–5b also show that SALT beats ActiveDP by 7.8–43.9 and 6.6–55.6 percentage points, respectively, across common budgets. ActiveDP’s keyword LFs cannot simultaneously provide high precision and broad coverage; additional coarse and correlated LFs therefore limit its label model, whereas SALT learns directly from instance labels. These results already favor ActiveDP because we conservatively count creating one LF as one annotation, although authoring a rule requires more effort than assigning one class label.

Certified accuracy. We also validated the certified accuracy guarantee. Figures 5e–5f show the risk gap (*i.e.*, $|R_{\mathcal{D}}(\theta_S) - R_{\mathcal{D}}(\theta_{\mathcal{D}})|$) against coverage ratio on DBPedia and CIFAR100. We set $\zeta = 0.1$ for both datasets, such that the target risk gap (the dashed horizontal line) is $0.1 \ln C$ by Corollary 4. For SALT, the risk gap falls below the target after 150 and 2000 annotations on DBPedia and CIFAR100, respectively, at coverage ratios of 90.9% and 81.0%. Full coverage is reached later at around 420 and 2500 annotations, confirming Corollary 4: full coverage suffices to meet the accuracy target.

In contrast, active learning methods fail to meet the coverage condition efficiently. The best baseline on DBPedia (resp. CIFAR100) is diversity-based TypiClust, which reaches a coverage ratio of 74.4% (resp. 91.2%) with 450 (resp. 2500) annotations, but it has the worst risk gap. Uncertainty and hybrid methods fail to achieve high coverage, due to the lack of support within the dense regions far from the decision boundary. This justifies coverage-driven selection.

Overall, these results empirically validate both the correctness and practical value of the certified accuracy guarantee.

Sensitivity to local Lipschitzness. We studied the sensitivity of SALT to local Lipschitzness on CIFAR100 after 7 annotation rounds.

Figure 6a shows that most samples lie in relatively flat regions with small local Lipschitz constants.

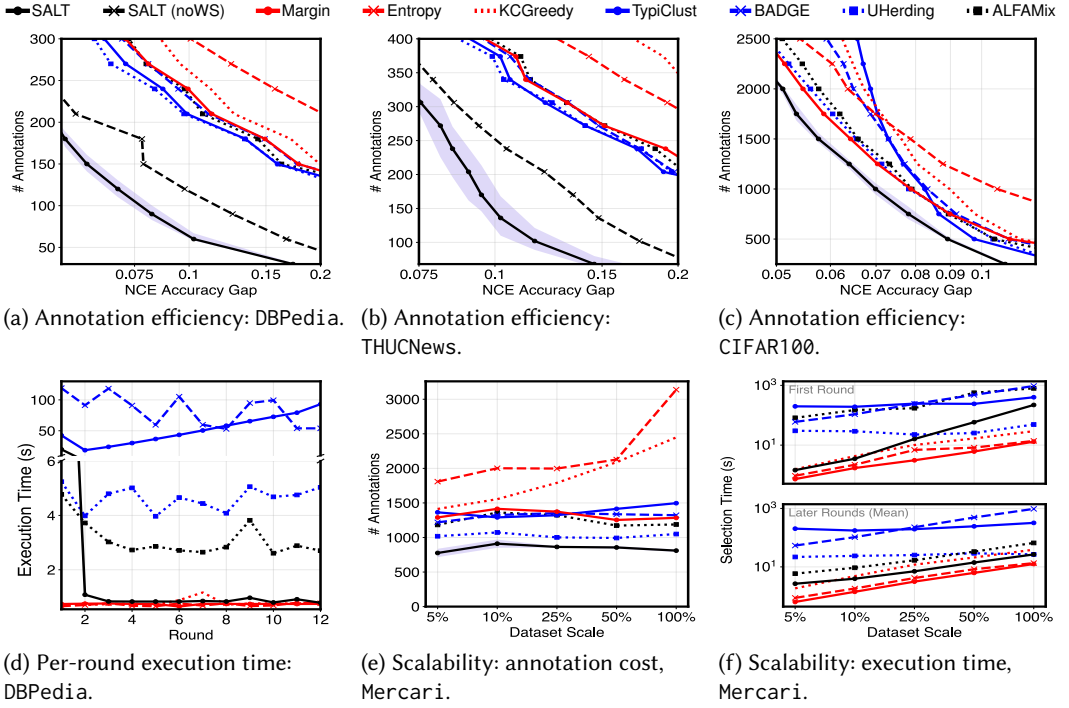


Fig. 7. Labeling efficiency, execution time, and scalability of SALT.

The heatmap further illustrates how the local cover radius varies with Lipschitzness. As expected, the two are strongly correlated: regions with larger local Lipschitz constants are steeper and therefore admit smaller local cover radii; in contrast, flatter regions admit larger ones.

Figure 6b shows a different trend. The local accuracy gap within each coverage ball has little correlation with Lipschitzness. This supports our theory: while local curvature affects the size of coverage balls, their accuracy guarantees remain stable across regions.

Exp-2: Human annotation cost. SALT’s higher labeling accuracy translates into lower human cost. Figures 5a-5d mark the target tolerance $\zeta = 0.1$. Across all datasets, SALT requires 25.9–75.7% fewer annotations than active learning baselines to reach this accuracy.

Figures 7a-7c further examine the annotation cost under varying accuracy tolerance ζ . SALT consistently reduces human effort. For instance, at $\zeta = 0.1$ on DBPedia, THUCNews and CIFAR100, SALT requires only ~64, ~146 and ~973 annotations (0.1%, 0.2% and 1.9% of D), while active learning baselines use 3.26–4.11 $\times$, >2.50 $\times$ and 1.35–3.01 $\times$ more, respectively. Under a looser tolerance, the gap is larger: at $\zeta = 0.15$, SALT is expected to reduce human cost by 75.3–83.9% on DBPedia and THUCNews. This validates the coverage-adaptive greedy strategy, which prioritizes samples that are both informative and representative, thus improving annotation efficiency.

The performance gap is smaller on CIFAR100 than on text datasets. This is because the much larger annotation budget on CIFAR100 allows all methods to better correct early selection bias and converge over time. Moreover, its class-balanced distribution (*i.e.*, 500 samples per class) enables recovery from suboptimal early selections. In contrast, DBPedia has greater class overlap and a more skewed distribution, making sample selection quality critical throughout the annotation process; thus SALT is more effective.

For all methods, human cost increases rapidly as ζ decreases. This is expected and reflects diminishing returns in selective annotation: once easy and well-covered regions are labeled, further gains rely on hard boundary cases, which require denser human supervision.

Exp-3: Weak label integration. To evaluate noise control in isolation, we decoupled the weak-label integration component from the HITL pipeline and compare SALT against representative weak supervision methods as a micro-benchmark (Figures 8a-8b). Since weak labels are hard rather than probabilistic, we assess their quality by the *error rate*, *i.e.*, the fraction of weak labels that disagree with the ground-truth. On THUCNews and DBPedia, SALT covers 31.6k and 51.6k samples, yielding 11.0% and 3.2% lower coverage than weak supervision, respectively. This reduction is expected: unlike existing methods that aggregate most available weak signals, SALT uses filtering LRRs to reject labels when evidence is insufficient, removing low-confidence labels before downstream model training.

Despite covering fewer samples, SALT produces cleaner weak labels, reducing the error rate of weak supervision methods by 1.8–5.9 and 3.6–7.8 percentage points on THUCNews and DBPedia, respectively. These results confirm the effectiveness of LRRs: rather than simply combining LF outputs, LRRs reconcile supportive signals and suppress unreliable ones, retaining a smaller but much cleaner label set that is better suited for bootstrapping model training.

Exp-4: Efficiency. Figures 7d compares SALT with active learning methods in terms of execution time per round on DBPedia. We exclude model training, as it is a fixed overhead shared by all methods under the same budget. The results are consistent on other datasets.

(1) In the early rounds, SALT spends more time on selecting samples, because its greedy selection procedure has $O(m + \text{bgt}n^2)$ complexity with n remaining candidates. However, this overhead gets smaller rapidly as the HITL process proceeds. Once more samples are covered, the candidate pool becomes much smaller, and the round cost drops accordingly. For example, the execution time of round 2 (coverage ratio=77.0%) is only 5.5% of that of round 1 on DBPedia, which is consistent with our theoretical analysis.

(2) To reach $\zeta = 0.1$, SALT takes 62.5%, 58.7% and 44.0% less time than UHerding, KCGreedy and ALFAMix, respectively. It is magnitudes faster than BADGE and TypiClust. This improvement comes from two sources. First, as discussed above, the cost of each SALT round decreases over time. Second, and more importantly, SALT reaches the same target accuracy in much fewer rounds than the baselines. Hence, its higher selection overhead in early rounds is amortized by faster convergence of the overall HITL workflow.

(3) Margin and Entropy use cheaper selection heuristics and are 0.04–0.37s faster than SALT per round. However, reaching the same target accuracy requires much more annotations. As human labeling dominates HITL cost, their savings are outweighed by the additional annotation effort, making SALT more efficient overall.

Exp-5: Scalability. We evaluate scalability on Mercari, using class-stratified random subsets of 5%, 10%, 25% and 50% of its samples.

(1) *Annotation cost.* Figure 7e reports the human annotation cost required to reach $\zeta = 0.1$. As the pool grows by 20 $\times$, the cost of SALT stays flat within 17.4%, and is the lowest at every scale (13.8–23.8% below the best baseline). In contrast, Entropy and KCGreedy need 1.7 $\times$ more annotations at full scale than at 5%, so SALT’s advantage widens with the dataset. A larger pool represents the distribution more densely in the same embedding space, giving SALT better representatives of important regions and placing more samples within each selected point’s certified neighborhood. Thus, each annotation covers more samples.

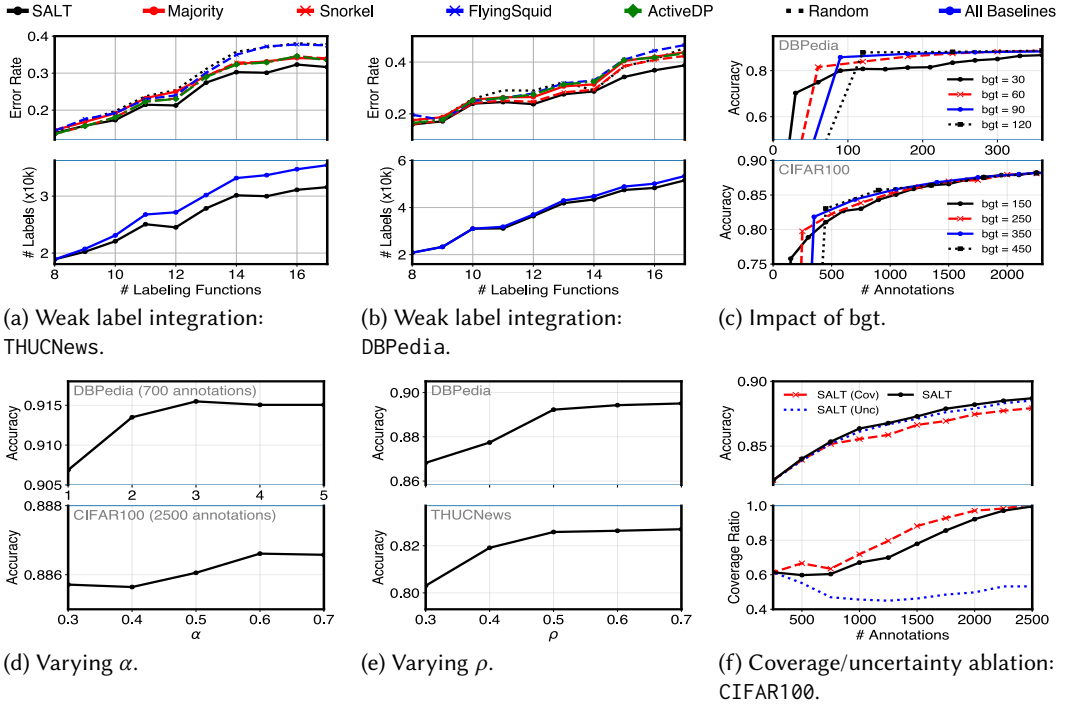


Fig. 8. Weak label integration, hyperparameter sensitivity, and ablation study of SALT.

(2) *Execution time.* Figure 7f reports the first-round and average later-round run times. Under the same $20\times$ growth, SALT's first-round time grows by $151.3\times$ (from 1.5 to 220.8 s), still well below the worst-case $O(m + \text{bgt} \cdot n^2)$ cost, as the first round scores all samples. The average later-round time instead grows sublinearly ($10.0\times$), since certified coverage rapidly shrinks the candidate pool. Consistent with Exp-4, Margin and Entropy are fast and take higher HITL cost; the other baselines are slower than SALT by $1.1\text{--}36.2\times$.

Exp-6: Hyperparameter settings. We studied the sensitivity of SALT to its main hyperparameters, including the round budget bgt , the weight α in coverage-adaptive scoring, and the decay rate ρ .

Impact of bgt . Figure 8c shows that, given the same total annotation budget, a larger bgt usually improves early accuracy. This is because early SALT rounds prioritize coverage expansion, so larger batches cover the embedding space more effectively and provide a stronger initial training signal. As annotations accumulate, the accuracy differences across bgt values diminish. In practice, it is preferable to use a larger bgt in early rounds and a relatively smaller one later, so as to obtain faster feedback on the accuracy certificate.

Impact of α . Figure 8d shows the final labeling accuracy of SALT is robust to different uncertainty weights α in our selection heuristic: on DBPedia (resp. CIFAR100), accuracy stays within $\pm 0.8\%$ (resp. $\pm 0.1\%$) of the best setting when $\alpha \in [1, 5]$ (resp. $[0.3, 0.7]$). Outside this range, it degrades in both directions: a small α overemphasizes coverage and underexplores boundary regions, while a large α overfocuses on hard but redundant samples, reducing annotation efficiency. In practice, we set α by a brief validation sweep and pick the value that best balances annotation-accuracy tradeoff.

Impact of ρ . Figure 8e varies the decay rate ρ in the weighted training objective and reports final

labeling accuracy after 10 rounds. Accuracy levels off when $\rho \geq 0.5$, but drops for smaller values. A small ρ slows the decay of weak-label weights, allowing noisy signals to persist and bias training. We recommend $\rho = 0.5$, which balances early bootstrapping with reliable convergence.

Exp-7: Ablation study. Finally, we studied the impact of key components in the SALT pipeline, by testing their ablations.

Impact of weak signals. Figures 5a–5b show that incorporating LFs allows SALT to converge faster than $\text{SALT}_{\text{noWS}}$, with the largest gains in the low-label setting (≤ 120 and ≤ 60 annotations, respectively). On THUCNews, SALT is 5.2% more accurate, and on DBPedia by up to 6.4%. After 8 rounds of annotation, the gap between SALT and $\text{SALT}_{\text{noWS}}$ is reduced to within $\pm 0.1\%$, a negligible difference since human annotations become dominant in training.

Impact of the coverage term in sample selection. Figure 8f compares SALT with SALT_{unc} , its variant without coverage (i.e., pure uncertainty sampling), on CIFAR100. SALT consistently beats SALT_{unc} in labeling accuracy. SALT_{unc} yields low and unstable coverage, reaching only 53.2% even with 2.5k annotations. It thus reduces to uncertainty active learning, making the certified stopping rule much less practical. This shows that the coverage term is necessary to effectively connect certified coverage to actual labeling accuracy.

Impact of the uncertainty term in sample selection. Figure 8f also includes SALT_{cov} , which removes the uncertainty term. As shown there, SALT_{cov} matches SALT in early rounds but falls behind later, showing a 0.75–0.95% accuracy deficit after more than 1k annotations. Although SALT_{cov} attains a higher coverage ratio, removing the uncertainty term prevents it from concentrating on hard boundary regions once the model becomes better calibrated.

Summary. SALT has the highest labeling accuracy across text and image datasets, outperforming active learning by up to 72.4 points in NCE accuracy, and reduces human annotations by up to 75.7% under high-accuracy targets ($\zeta \leq 0.1$). LRR-based refinement yields cleaner weak labels, reducing their error rate by up to 7.8 percentage points, and faster convergence amortizes the higher early selection cost, cutting total execution time by >44.0%. SALT is robust to hyperparameters: larger bgt improves accuracy, accuracy is consistent for $\rho \geq 0.5$ (recommended $\rho = 0.5$), and performance is stable for small variations in α , which should be set per dataset. Its ablations confirm that coverage and uncertainty are complementary, and that LRR-filtered weak supervision is essential for efficient bootstrapping without compromising the certified accuracy guarantee.

8 Conclusion

The novelty of SALT lies in a certified coverage formulation that connects local loss stability to dataset-wide accuracy guarantees, a curvature-adaptive local cover that unifies uncertainty and diversity into a single selection criterion, a global risk bound yielding the first accuracy-certified stopping rule, and a logic-based correction that integrates weak supervision to improve efficiency while controlling noise. Though developed for selective annotation, these techniques can be transferred to other data-management tasks: the certified coverage framework applies to any embedding-based labeling, independent of the models, and LRRs offer a general mechanism for reconciling conflicting signals, common in data integration and cleaning. Our experiments have verified that SALT is promising in practice.

Future work will explore (a) more automated methods for detecting and correcting mislabels and unlabeled data to further minimize human annotation, (b) cross-modal labeling beyond a uniform D , and (c) domain-specific labeling for vertical-domain LLM training.

References

- [1] 2026. Data labeling: The authoritative guide. <https://scale.com/guides/data-labeling-annotation-guide>.
- [2] 2026. SurgeAI products. <https://surgehq.ai/products>.
- [3] Nikolas Adaloglou, Felix Michels, Hamza Kalisch, and Markus Kollmann. 2023. Exploring the Limits of Deep Image Clustering using Pretrained Models. In *BMVC*. BMVA. <https://papers.bmvc2023.org/0297.pdf>
- [4] Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. 2020. Contextual diversity for active learning. In *ECCV*. Springer, 137–153. doi:10.1007/978-3-030-58517-4_9
- [5] Jordan Ash and Ryan P Adams. 2020. On warm-starting neural network training. In *NeurIPS*. 3884–3894. <https://proceedings.neurips.cc/paper/2020/hash/288cd2567953f06e460a33951f55daaf-Abstract.html>
- [6] Jordan Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2021. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. In *ICLR*. <https://openreview.net/forum?id=ryghZJBKPS>
- [7] Stephen H Bach, Bryan He, Alexander Ratner, and Christopher Ré. 2017. Learning the structure of generative models without labeled data. In *ICML*. <https://proceedings.mlr.press/v70/bach17a.html>
- [8] Stephen H Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alex Ratner, Braden Hancock, Houman Alborzi, Rahul Kuchha, Chris Ré, and Rob Malkin. 2019. Snorkel DryBell: A case study in deploying weak supervision at industrial scale. In *SIGMOD*. 362–375. doi:10.1145/3299869.3314036
- [9] Wonho Bae, Junhyug Noh, and Danica J Sutherland. 2024. Generalized coverage for more robust low-budget active learning. In *ECCV*. doi:10.1007/978-3-031-73010-8_19
- [10] Wonho Bae, Gabriel L Oliveira, and Danica J Sutherland. 2025. Uncertainty herding: One active learning method for all label budgets. In *ICLR*. <https://openreview.net/forum?id=UgPoHhYQ2U>
- [11] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML*. doi:10.1145/1553374.1553380
- [12] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*. https://proceedings.neurips.cc/paper_files/paper/2019/hash/1cd138d0499a68f4bb72bee04bbec2d7-Abstract.html
- [13] Gabriele Berton, Riccardo Mereu, Gabriele Trivigno, Carlo Masone, Gabriela Csorba, Torsten Sattler, and Barbara Caputo. 2022. Deep visual geo-localization benchmark. In *CVPR*. doi:10.1109/CVPR52688.2022.00532
- [14] Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT*. doi:10.1145/279943.279962
- [15] Leo Breiman, Jerome Friedman, Charles J. Stone, and R. A. Olshen. 1984. *Classification and Regression Trees*. Taylor & Francis. doi:10.1201/9781315139470
- [16] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep Clustering for Unsupervised Learning of Visual Features. In *ECCV*. doi:10.1007/978-3-030-01264-9_9
- [17] Gong Cheng, Junwei Han, and Xiaoqiang Lu. 2017. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proc. IEEE* 105, 10 (Oct 2017), 1865–1883. doi:10.1109/JPROC.2017.2675998
- [18] Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. 2021. Batch active learning at scale. In *NeurIPS*. https://proceedings.neurips.cc/paper_files/paper/2021/hash/64254db8396e404d9223914a0bd355d2-Abstract.html
- [19] Katherine M. Collins, Umang Bhatt, and Adrian Weller. 2022. Eliciting and Learning with Soft Labels from Every Annotator. In *HCOMP*. doi:10.1609/hcomp.v10i1.21986
- [20] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 28, 1 (1979), 20–28. doi:10.2307/2346806
- [21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*. doi:10.1109/CVPR.2009.5206848
- [22] Mohamad Dolatshah, Mathew Teoh, Jiannan Wang, and Jian Pei. 2018. Cleaning Crowdsourced Labels Using Oracles For Statistical Classification. *PVLDB* 12, 4 (2018), 376–389. doi:10.14778/3297753.3297758
- [23] Miao Fan, Deli Zhao, Qiang Zhou, Zhiyuan Liu, Fang Zheng, and Edward Y Chang. 2014. Distant supervision for relation extraction with matrix completion. In *ACL*. doi:10.3115/v1/P14-1079
- [24] Wenfei Fan, Floris Geerts, Xibei Jia, and Anastasios Kementsietsidis. 2008. Conditional Functional Dependencies for Capturing Data Inconsistencies. *TODS* 33, 1 (2008), 6:1–6:48. doi:10.1145/1366102.1366103
- [25] Wenfei Fan, Ruochun Jin, Muyang Liu, Ping Lu, Chao Tian, and Jingren Zhou. 2020. Capturing Associations in Graphs. *PVLDB* 13, 11 (2020), 1863–1876. doi:10.14778/3407790.3407795
- [26] Wenfei Fan, Ping Lu, and Chao Tian. 2020. Unifying logic rules and machine learning for entity enhancing. *Sci. China Inf. Sci.* 63, 7 (2020). doi:10.1007/s11432-020-2917-1
- [27] Daniel Fu, Mayee Chen, Frederic Sala, Sarah Hooper, Kayvon Fatahalian, and Christopher Ré. 2020. Fast and three-rious:

- Speeding up weak supervision with triplet methods. In *ICML*. <https://proceedings.mlr.press/v119/fu20a.html>
- [28] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep Bayesian active learning with image data. In *ICML*. <https://proceedings.mlr.press/v70/gal17a.html>
- [29] Sainyam Galhotra, Behzad Golshan, and Wang-Chiew Tan. 2021. Adaptive rule discovery for labeling text data. In *SIGMOD*. 2217–2225. doi:10.1145/3448016.3457334
- [30] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. <https://pile.eleuther.ai>. doi:10.48550/arXiv.2101.00027
- [31] Michael Garey and David Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company. <https://dl.acm.org/doi/10.5555/574848>
- [32] Naiqing Guan and Nick Koudas. 2024. ActiveDP: Bridging Active Learning and Data Programming. In *EDBT*. 515–521. doi:10.48786/edbt.2024.44
- [33] Zhipeng Guo, Yu Zhao, Yabin Zheng, Xiance Si, Zhiyuan Liu, and Maosong Sun. 2026. THUCTC: An efficient Chinese text classifier. Online. <https://github.com/thunlp/THUCTC>.
- [34] HeroHunt.ai. 2026. The ultimate AI data labeling industry overview (2026). <https://www.herohunt.ai/blog/the-ultimate-ai-data-labeling-industry-overview>.
- [35] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015). doi:10.48550/arXiv.1503.02531
- [36] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL-HLT*. <https://aclanthology.org/P11-1055/>
- [37] HuggingFace. 2006. AGNews dataset. <https://huggingface.co/datasets/sentence-transformers/agnnews>.
- [38] Nikolai Kalischek, Jan D Wegner, and Konrad Schindler. 2021. In the light of feature distributions: moment matching for neural style transfer. In *CVPR*. doi:10.1109/CVPR46437.2021.00926
- [39] David Karger, Sewoong Oh, and Devavrat Shah. 2011. Iterative learning for reliable crowdsourcing systems. In *NIPS*. <https://proceedings.neurips.cc/paper/2011/hash/c667d53acd899a97a85de0c201ba99be-Abstract.html>
- [40] Richard M Karp. 2009. Reducibility among combinatorial problems. In *50 Years of Integer Programming 1958-2008: from the Early Years to the State-of-the-Art*. Springer, 219–241. doi:10.1007/978-3-540-68279-0_8
- [41] Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. 2021. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *ICML*. <https://proceedings.mlr.press/v139/killamsetty21a.html>
- [42] Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. 2021. Glisten: Generalization based data subset selection for efficient and robust learning. In *AAAI*. doi:10.1609/aaai.v35i9.16988
- [43] Krishnateja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh Iyer. 2021. Retrieve: Coreset selection for efficient and robust semi-supervised learning. In *NeurIPS*. <https://proceedings.neurips.cc/paper/2021/hash/793bc52a941b3951dfdb85fb04f9d06-Abstract.html>
- [44] Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. 2021. Task-aware variational adversarial active learning. In *CVPR*. doi:10.1109/CVPR46437.2021.00807
- [45] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. 2019. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *NeurIPS*. https://proceedings.neurips.cc/paper_files/paper/2019/hash/95323660ed2124450caa2c2c46b5ed90-Abstract.html
- [46] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning multiple layers of features from tiny images*. Technical Report. University of Toronto, Toronto, Ontario. <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>
- [47] Dong-Hyun Lee. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*. https://openreview.net/forum?id=3iGjh_NmoG
- [48] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia: A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195. doi:10.3233/SW-140134
- [49] David D Lewis. 1995. A sequential algorithm for training text classifiers: Corrigendum and additional data. *ACM SIGIR Forum* 29, 2 (1995), 13–19. doi:10.1145/219587.219592
- [50] James MacQueen. 1967. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1. 281–297. <https://digicoll.lib.berkeley.edu/record/113015>
- [51] Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active Learning by Acquiring Contrastive Examples. In *EMNLP*. doi:10.18653/v1/2021.emnlp-main.51
- [52] Mercari, Inc. 2017. Mercari Price Suggestion Challenge. Kaggle. <https://www.kaggle.com/competitions/mercari-price-suggestion-challenge> Accessed: 2026-07-28.

- [53] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. 2020. Coresets for data-efficient training of machine learning models. In *ICML*. <https://proceedings.mlr.press/v119/mirzasoleiman20a.html>
- [54] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help?. In *NeurIPS*. <https://proceedings.neurips.cc/paper/2019/hash/f1748d6b0fd9d39f71450117eba2725-Abstract.html>
- [55] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming* 14, 1 (1978), 265–294. doi:10.1007/BF01588971
- [56] Quang Nguyen, Hamed Valizadegan, and Milos Hauskrecht. 2014. Learning classification models with soft-label information. *Journal of the American Medical Informatics Association* 21, 3 (2014), 501–508. doi:10.1136/amiajnl-2013-001964
- [57] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. DINOv2: Learning Robust Visual Features without Supervision. *TMLR* (2024). <https://openreview.net/forum?id=a68SUt6zFt>
- [58] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*. doi:10.52202/068431-2011
- [59] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Reza Haffari, Anton Van Den Hengel, and Javen Qinfeng Shi. 2022. Active Learning by Feature Mixing. In *CVPR*. doi:10.1109/CVPR52688.2022.01192
- [60] Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. *PVLDB* 11, 3 (2017), 269–282. doi:10.14778/3157794.3157797
- [61] Alex Ratner, Braden Hancock, Jared Dunnmon, Roger Goldman, and Christopher Ré. 2018. Snorkel MeTaL: Weak supervision for multi-task learning. In *DEEM*. doi:10.1145/3209889.3209898
- [62] Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. 2019. Training complex models with multi-task weak supervision. In *AAAI*. doi:10.1609/aaai.v33i01.33014763
- [63] Alexander J Ratner, Stephen H Bach, Henry R Ehrenberg, and Christopher Ré. 2017. Snorkel: Fast training set generation for information extraction. In *SIGMOD*. doi:10.1145/3035918.3056442
- [64] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming: Creating large training sets, quickly. In *NIPS*. <https://proceedings.neurips.cc/paper/2016/hash/6709e8d64a5f47269ed5cea9f625f7ab-Abstract.html>
- [65] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2015. Training Deep Neural Networks on Noisy Labels with Bootstrapping. In *ICLR*. <https://research.google/pubs/training-deep-neural-networks-on-noisy-labels-with-bootstrapping/>
- [66] Salva Rühling Cachay, Benedikt Boecking, and Artur Dubrawski. 2021. End-to-end weak supervision. In *NeurIPS*. <https://proceedings.neurips.cc/paper/2021/hash/0e674a918ebca3f78bfe02e2f387689d-Abstract.html>
- [67] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active hidden markov models for information extraction. In *IDA*. doi:10.1007/3-540-44816-0_31
- [68] Micheál Ó Searcóid. 2007. *Metric spaces*. Springer. doi:10.1007/978-1-84628-627-8
- [69] Ozan Sener and Silvio Savarese. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *ICLR*. <https://openreview.net/forum?id=H1aIuk-RW>
- [70] Victor S. Sheng and Jing Zhang. 2019. Machine Learning with Crowdsourcing: A Brief Summary of the Past Research and Future Directions. In *AAAI*. 9837–9843. doi:10.1609/aaai.v33i01.33019837
- [71] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. 2019. Variational adversarial active learning. In *ICCV*. doi:10.1109/ICCV.2019.00607
- [72] Samarth Sinha, Han Zhang, Anirudh Goyal, Yoshua Bengio, Hugo Larochelle, and Augustus Odena. 2020. Small-GAN: Speeding up gan training using core-sets. In *ICML*. <https://proceedings.mlr.press/v119/sinha20b.html>
- [73] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*. <https://proceedings.neurips.cc/paper/2020/hash/06964dce9addb1c5cb5d6e3d9838f733-Abstract.html>
- [74] Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *JMLR* 2, Nov (2001), 45–66. <https://www.jmlr.org/papers/v2/tong01a.html>
- [75] Paroma Varma and Christopher Ré. 2018. Snuba: Automating weak supervision to label training data. *PVLDB* 12, 3 (2018), 223–236. doi:10.14778/3291264.3291268
- [76] Dan Wang and Yi Shang. 2014. A new active labeling method for deep learning. In *IJCNN*. doi:10.1109/IJCNN.2014.6889457
- [77] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. 2016. Cost-effective active learning for deep image

- classification. *TCSVT* 27, 12 (2016), 2591–2600. doi:10.1109/TCSVT.2016.2589879
- [78] Kai Wei, Rishabh Iyer, and Jeff Bilmes. 2015. Submodularity in data subset selection and active learning. In *ICML*. <https://proceedings.mlr.press/v37/wei15.html>
- [79] Xiaobao Wu, Thong Nguyen, and Anh Tuan Luu. 2024. A survey on neural topic models: Methods, applications, and challenges. *Artificial Intelligence Review* 57, 2 (2024), 18. doi:10.1007/s10462-023-10661-7
- [80] Yifan Wu, Shuhao Liu, Yuanzhe Chen, and Wenfei Fan. 2026. *Full version with appendices*. <https://shuhaoliu.github.io/assets/papers/sigmod27-salt-full.pdf>
- [81] Binhui Xie, Longhui Yuan, Shuang Li, Chi Harold Liu, Xinjing Cheng, and Guoren Wang. 2022. Active learning for domain adaptation: An energy-based approach. In *AAAI*. doi:10.1609/aaai.v36i8.20850
- [82] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *CVPR*. doi:10.1109/CVPR42600.2020.01070
- [83] Ofer Yehuda, Avihu Dekel, Guy Hacohen, and Daphna Weinshall. 2022. Active learning through a covering lens. In *NeurIPS*. doi:10.52202/068431-1624
- [84] Peng Yu, En Xu, Bin Chen, Haibiao Chen, and Yinfei Xu. 2025. QZhou-Embedding Technical Report. *arXiv preprint arXiv:2508.21632* (2025). doi:10.48550/arXiv.2508.21632
- [85] Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. 2021. WRENCH: A Comprehensive Benchmark for Weak Supervision. In *NeurIPS*. <https://openreview.net/forum?id=Q9SKS5k8io>
- [86] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Guoyin Wang, and Fei Wu. 2026. Instruction tuning for large language models: A survey. *Comput. Surveys* 58, 7 (2026), 1–36. doi:10.1145/3777411
- [87] Haizhong Zheng, Rui Liu, Fan Lai, and Atul Prakash. 2023. Coverage-centric coreset selection for high pruning rates. In *ICLR*. <https://openreview.net/forum?id=QwKvL6wC8Yi>
- [88] Xiaojin Zhu and Zoubin Ghahramani. 2002. *Learning from labeled and unlabeled data with label propagation*. Technical Report. Carnegie Mellon University. <https://www.cs.cmu.edu/~zhuxj/pub/CMU-CALD-02-107.pdf>

A Proof of Theorem 1

Theorem 1. SAP is NP-hard. □

Proof: Recall the selective annotation problem (SAP):

- *Input:* An unlabeled dataset $D = \{x_i\}_{i=1}^n$, a classification model $\mathcal{M}$ to be trained, a learning algorithm A that trains $\mathcal{M}$ on a given dataset, and a target accuracy threshold ζ .
- *Output:* An annotated dataset $\mathcal{D}_S$ obtained from a subset S of D .
- *Objective:* To minimize the cardinality of $\mathcal{D}_S$, subject to $\text{acc}(\mathcal{M}_{\theta_S}, \mathcal{D}) \geq \text{acc}(\mathcal{M}_{\theta_D}, \mathcal{D}) - \zeta$, where $\mathcal{M}_{\theta_S}$ (resp. $\mathcal{M}_{\theta_D}$) is the model $\mathcal{M}$ trained on $\mathcal{D}_S$ and $\mathcal{D}$, respectively.

Since the accuracy of a model is determined by its loss on the dataset, we can equivalently reformulate SAP as follows:

- *Objective*:* To minimize the cardinality of $\mathcal{D}_S$, subject to $R_{\mathcal{D}}(\theta_S) \leq R_{\mathcal{D}}(\theta_D) + \tau$, where $R_{\mathcal{D}}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} \ell(x, y)$ is the empirical risk over $\mathcal{D}$, ℓ is the cross-entropy loss function $\ell(x, \eta) = \sum_{k=1}^C -\eta_k \ln p_k$, and $\tau = \zeta \ln C$.

We investigate the decision version of SAP, denoted by DSAP:

- *Input:* An unlabeled dataset $D = \{x_i\}_{i=1}^n$, a classification model $\mathcal{M}$, a learning algorithm A that trains $\mathcal{M}$ on a given dataset, a budget N for annotation, and a target accuracy threshold τ .
- *Problem:* Is there an annotated subset $\mathcal{D}_S$ with $|\mathcal{D}_S| \leq N$ such that $R_{\mathcal{D}}(\theta_S) \leq R_{\mathcal{D}}(\theta_D) + \tau$.

Assume that the learning algorithm A is deterministic and runs in polynomial time (PTIME). We prove that DSAP is NP-complete.

Upper bound. We first provide an NP algorithm for DSAP.

Given such inputs, the algorithm works as follows:

- (1) It first guesses a subset S with $|S| \leq N$.
- (2) It computes θ_S and θ_D by performing the learning algorithm A on $\mathcal{D}_S$ and $\mathcal{D}$, respectively.
- (3) It evaluates $R_{\mathcal{D}}(\theta_S)$ and $R_{\mathcal{D}}(\theta_D)$, and checks whether the inequality holds, if so, return true. Otherwise, reject the guess.

Step 2 runs in PTIME by assumption. Step 3 computes the loss after forward propagation, which consists of some linear operations and activation functions, and runs in PTIME. Thus DSAP is in NP.

Lower bound. We prove NP-hardness of DSAP by a reduction from SET-COVER problem, which is to decide, given a universe $U = \{u_1, \dots, u_m\}$, a collection of subsets $C = \{C_1, \dots, C_n\}$ where $C_i \subseteq U$, and an integer k , whether there exists a subcollection of C of size at most k that covers U . SET-COVER is NP-complete [40].

Intuitively, we construct a dataset $\mathcal{D}$ based on the SET-COVER instance such that each set C_i corresponds to a set example x_i^{set} in $\mathcal{D}$, and each element u_j corresponds to an element example x_j^{elem} in $\mathcal{D}$. We show that if there exists a set cover of size at most k , then we can select the corresponding set examples and one guard example to achieve a loss close to the optimal loss. On the other hand, if there is no set cover of size at most k , then any subset of examples with size at most $k + 1$ will have loss on element examples significantly larger than the optimal loss. Thus, by setting an appropriate threshold τ and learning rate η , we can ensure that there exists a subset of examples with loss less than $R_{\mathcal{D}}(\theta_D) + \tau$ if and only if there exists a set cover of size at most k . We construct the reduction as follows.

Model. We fix the model class $\mathcal{M}$ to be binary logistic regression. The model parameter is $\theta \in \mathbb{R}^d$, and the prediction function is

$$p_{\theta}(x) = \sigma(\theta^{\top} x), \quad \sigma(t) = \frac{1}{1 + e^{-t}}.$$

The loss function is the cross-entropy loss

$$\ell(x, y, \theta) = -y \ln \sigma(\theta^\top x) - (1 - y) \ln(1 - \sigma(\theta^\top x)).$$

This construction is equivalent to a softmax classifier with two classes. Let the model parameters be $W = (w_0, w_1) \in \mathbb{R}^{2 \times m}$. Define

$$p(y = c \mid x) = \frac{\exp(w_c^\top x)}{\exp(w_0^\top x) + \exp(w_1^\top x)}.$$

Define the reparameterization $\theta := w_1 - w_0$. Then we have

$$p(y = 1 \mid x) = \frac{1}{1 + \exp(-\theta^\top x)} = \sigma(\theta^\top x),$$

which coincides exactly with logistic regression.

Dataset $\mathcal{D}$. Given an instance (U, C, k) of SET-COVER with $U = \{u_1, \dots, u_m\}$ and $C = \{C_1, \dots, C_n\}$, we construct a dataset $\mathcal{D} \subset \mathbb{R}^{m+1} \times \{0, 1\}$ with the ground truth labels as follows.

- *Set examples:* For each $C_i \in C$, create one example

$$x_i^{\text{set}} = (2\mathbf{1}_{u_j \in C_i}, 0), \quad y = 1.$$

- *Element examples:* For each $u_j \in U$, create one example

$$x_j^{\text{elem}} = (e_j, 1), \quad y = 1.$$

- *Guard examples:* Create $(m + 2)T$ copies of

$$x^{\text{guard}} = (0^{(m)}, 1/T), \quad y = 0.$$

Here T is an integer greater than m, n and $k + 1$, for instance, $T = m + n + k + 1$. The budget is set to $N = k + 1$, and $|D| = n + m + (m + 2)T$.

Intuitively, the three example types play the following roles. The first m coordinates of θ record the elements of U , and the last coordinate separates element examples from guards. Training on a set example x_i^{set} pushes up the coordinates of the elements in C_i , so an element example x_j^{elem} is classified correctly iff some selected set covers u_j . The guard examples pin the last coordinate down: without them, a selection could cheat by including element examples, which push the last coordinate up and thus classify *all* element examples correctly without covering anything. Since guards constitute the vast majority of $\mathcal{D}$, misclassifying them is prohibitively costly, while satisfying them costs only one selected guard example.

Learning algorithm. The learning algorithm A initializes $\theta^{(0)} = 0$ and performs exactly one step of gradient descent with a fixed sufficiently large learning rate $\eta > 0$, which will be specified later:

$$\theta_S = \theta^{(0)} - \eta \nabla R_S(\theta^{(0)}).$$

This algorithm runs in PTIME w.r.t. the size of the training set.

Learning rate η and threshold τ . We construct η and τ based on the following lemmas, which will be proved later. In words: training on the full dataset yields arbitrarily small empirical loss (Lemma 7); so does training on the examples of a set cover plus one guard example (Lemma 8); but if the selected sets fail to cover some element, the corresponding element example is badly misclassified and the loss blows up (Lemma 9). The learning rate η and threshold τ are chosen to separate these two regimes.

Lemma 7: *The loss $R_{\mathcal{D}}(\theta_{\mathcal{D}})$ of the model trained on the full dataset $\mathcal{D}$ is upper bounded by*

$$R_{\mathcal{D}}(\theta_{\mathcal{D}}) < \ln(1 + \exp(-\eta/T|\mathcal{D}|))$$

which approaches 0 as $\eta \rightarrow \infty$. □

Lemma 8: *If C' is a solution to the SET-COVER instance, let S be the subset of $\mathcal{D}$ corresponding to C' and one guard example, then the loss of the model trained on S is upper bounded by*

$$R_{\mathcal{D}}(\theta_S) < \ln(1 + \exp(-\eta/2T^2(k+1))),$$

which also approaches 0 as $\eta \rightarrow \infty$. $\square$

Lemma 9: If S is a subset of $\mathcal{D}$ with $|S| \leq k+1$. Suppose S contains no x^{elem} and at least one x^{guard} , and the corresponding sets do not cover U , then the loss of the model trained on S is lower bounded

$$R_{\mathcal{D}}(\theta_S) > 1/|\mathcal{D}| \ln(1 + \exp(\eta/2T|S|)) > \frac{\eta}{2T(k+1)|\mathcal{D}|},$$

which approaches ∞ as $\eta \rightarrow \infty$. $\square$

Thus, we set $\eta = T^2(k+1)|\mathcal{D}|$ and $\tau = 0.1$, which ensures that the loss $R_{\mathcal{D}}(\theta_{\mathcal{D}})$ of the model trained on the full dataset and $R_{\mathcal{D}}(\theta_S)$ for any S corresponding to a valid set cover is less than 0.001. On the other hand, for any S in Lemma 9 that does not correspond to a valid set cover, the loss $R_{\mathcal{D}}(\theta_S)$ is greater than $\frac{1}{2} \ln 2$.

Correctness of the reduction. We first show that the above reduction can be computed in PTIME in the size of the SET-COVER instance (U, C, k) . Let $|U| = m$ and $|C| = n$, and set $T = m + n + k + 1$ and $N = k + 1$. The constructed labeled dataset $\mathcal{D} \subset \mathbb{R}^{m+1} \times \{0, 1\}$ has

$$|\mathcal{D}| = n + m + (m+2)T = O((m+n+k)^2),$$

which is polynomial in the input size.

Set examples x_i^{set} can be generated in PTIME in the size of C . Constructing the m element examples takes $O(m)$ time. Constructing the $(m+2)T$ guard examples takes $O((m+2)T)$ time;

Finally, the learning rate and threshold are computed as $\eta = T^2(k+1)|\mathcal{D}|$ and $\tau = 0.1$. These quantities can be computed in PTIME. The learning algorithm A works in PTIME in the size of $\mathcal{D}$. Thus the reduction from SET-COVER to DSAP runs in PTIME.

We next show that the reduction is correct.

($\Rightarrow$) If there exists a set cover C' of size at most k , then we can construct a subset S of $\mathcal{D}$ consisting of the examples corresponding to the sets in C' and one guard example. By Lemma 8, the loss of the model trained on S is less than 0.001, which is less than $R_{\mathcal{D}}(\theta_{\mathcal{D}}) + \tau$.

($\Leftarrow$) If there exists a subset S of $\mathcal{D}$ with $|S| \leq k+1$ such that $R_{\mathcal{D}}(\theta_S) \leq R_{\mathcal{D}}(\theta_{\mathcal{D}}) + \tau$, then we claim that the sets corresponding to the examples in S must cover U . The argument formalizes the intuition above: we first rule out the “cheating” selections that contain element examples, by showing that they misclassify the guard majority and thus cannot meet the threshold; the remaining selections are subject to Lemma 9, which forces them to cover U .

We first show that S must satisfy the conditions in Lemma 9. Let a denote the number of x^{elem} examples in S , and b denote the number of x^{guard} examples in S . If $b \leq Ta$, we calculate the last coordinate of θ_S as follows. We first compute θ_S by performing one step of gradient descent, which can be written as

$$\theta_S = \eta/|S| \sum_{(x,y) \in S} (y - 1/2)x.$$

Thus the last coordinate of θ_S is $a/2 - b/2T$. Since $b \leq Ta$, we have $a/2 - b/2T \geq 0$. Then we evaluate the loss $l(x^{\text{guard}}, \theta_S)$, which is $\ln(1 + \exp(\theta_S^\top x^{\text{guard}}))$. Since $\theta_S^\top x^{\text{guard}} \geq 0$, we have $l(x^{\text{guard}}, \theta_S) \geq \ln 2$. Finally, we have

$$R_{\mathcal{D}}(\theta_S) \geq \frac{mT + 2T}{n + m + (m+2)T} \ln 2 \geq \frac{1}{2} \ln 2 > R_{\mathcal{D}}(\theta_{\mathcal{D}}) + \tau,$$

which contradicts the assumption.

Thus $b > Ta$. Since $|S| \leq k+1$, we have $a < b/T \leq (k+1)/T < 1$. This implies that S contains no x^{elem} examples and at least one x^{guard} example. By Lemma 9, we conclude that the sets corresponding to the examples in S must cover U . Proof of Lemmas 7-8. We first prove Lemma 7.

Since $\theta^{(0)} = 0$, we have $p_{\theta^{(0)}}(x) = 1/2$ for all $x \in \mathcal{D}$. Thus, $\theta_{\mathcal{D}}$ can be written as

$$\theta_{\mathcal{D}} = \eta/|\mathcal{D}| \sum_{(x,y) \in \mathcal{D}} (y - 1/2)x.$$

Substituting the examples in $\mathcal{D}$ into the above equation, we have

$$\theta_{\mathcal{D}} = \eta/|\mathcal{D}|([r_1, \dots, r_m, 0] + [1/2, \dots, 1/2, -1]),$$

where $r_j > 0$ is the number of sets in $\mathcal{C}$ that contain element j . After the single gradient step, θ is simply the sum of the training examples weighted by $\pm\eta/2$ according to their labels; every element coordinate of $\theta_{\mathcal{D}}$ is thus positive and its last coordinate is negative, so all three example types are classified correctly with comfortable margins. Then we evaluate the loss of each example in $\mathcal{D}$ under $\theta_{\mathcal{D}}$:

$$\begin{aligned} l(x_i^{\text{set}}, \theta_{\mathcal{D}}) &< \ln(1 + e^{-3\eta/|\mathcal{D}|}), \\ l(x_j^{\text{elem}}, \theta_{\mathcal{D}}) &< \ln(1 + e^{-\eta/2|\mathcal{D}|}), \\ l(x^{\text{guard}}, \theta_{\mathcal{D}}) &< \ln(1 + e^{-\eta/T|\mathcal{D}|}). \end{aligned}$$

Since $T = m + n + k + 1 \geq 4$, we have $R_{\mathcal{D}}(\theta_{\mathcal{D}}) < \ln(1 + e^{-\eta/T|\mathcal{D}|})$.

The proof of Lemma 8 is similar. Since $\mathcal{C}'$ is a set cover, we have $r_j \geq 1$ for all j . Thus, we can write θ_S as

$$\theta_S = \eta/|S|[r_1, \dots, r_m, -\frac{1}{2T}].$$

Then we evaluate the loss of each example in $\mathcal{D}$ under θ_S ; they are both less than $\ln(1 + e^{-\eta/2T^2|S|})$. Thus, we have $R_{\mathcal{D}}(\theta_S) < \ln(1 + e^{-\eta/2T^2(k+1)})$.

Proof of Lemma 9. Let b denote the number of x^{guard} examples in S . Since S contains no x^{elem} examples, and at least one x^{guard} example, we have $b \geq 1$. Then we can write θ_S as

$$\theta_S = \eta/|S|[r_1, \dots, r_m, -\frac{b}{2T}].$$

Since the sets corresponding to the examples in S do not cover U , there exists some j such that $r_j = 0$. Then we evaluate the loss of x_j^{elem} under θ_S , which is given by

$$\theta_S^\top x_j^{\text{elem}} = -\eta b/2T|S|, \quad l(x_j^{\text{elem}}, \theta_S) = \ln(1 + e^{\eta b/2T|S|}).$$

Thus, we conclude that

$$R_{\mathcal{D}}(\theta_S) \geq \frac{1}{|\mathcal{D}|} \ln(1 + e^{\eta/2T|S|}). \quad \square$$

B Roadmap: the Accuracy Guarantee

This section gives a high-level overview of how SALT derives its dataset-wide accuracy guarantee, before the formal proofs in Appendices C–E; Figure 9 visualizes the derivation chain.

The goal. SALT annotates only a small subset S of the dataset $\mathcal{D}$ and trains a model $\mathcal{M}_{\theta_S}$ on it to label the rest. The guarantee to establish is that $\mathcal{M}_{\theta_S}$ labels $\mathcal{D}$ almost as accurately as the ideal model $\mathcal{M}_{\theta_{\mathcal{D}}}$ that would be obtained if the entire $\mathcal{D}$ were fully annotated: the accuracy gap is at most a user-specified slack ζ . The difficulty is that this gap cannot be measured directly, since training $\mathcal{M}_{\theta_{\mathcal{D}}}$ would require annotating all elements of $\mathcal{D}$. We therefore need a condition that (a) is checkable from the annotated samples alone, and (b) provably implies the accuracy target.

The derivation builds this condition in four steps.

Step 1: an annotated sample vouches for its neighbors. The loss is smooth over the embedding space: samples that are close have similar losses. Lemma 2 quantifies this: within distance δ of an annotated sample, the loss deviates by at most an explicit function of δ , whose coefficients measure how well the model already fits that sample and how curved the loss surface is around it. Hence an annotation is a certificate for its entire neighborhood.

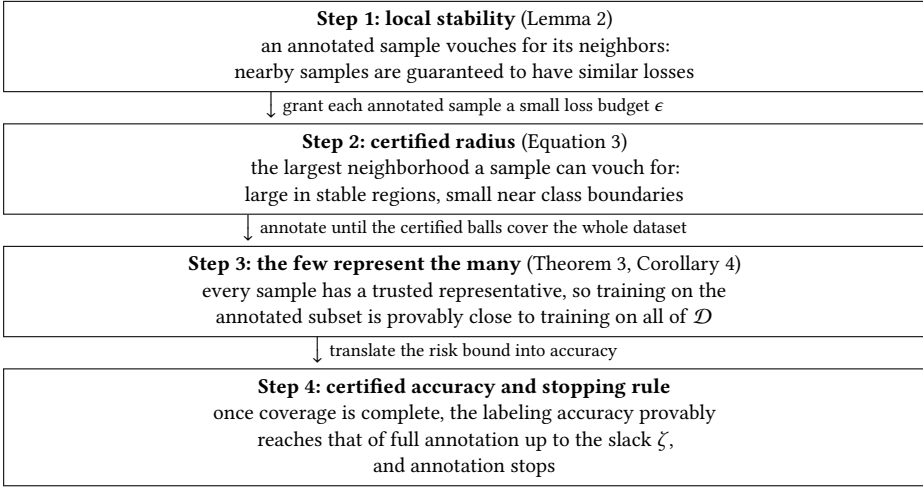


Fig. 9. The derivation chain of the accuracy guarantee.

Step 2: a certified radius for each sample. Granting each annotated sample a small loss budget ϵ , we invert the bound of Step 1 to obtain its *certified radius* (Equation 3): the largest distance within which the loss is guaranteed to deviate by no more than ϵ . The radius adapts to the local geometry: samples in flat, confidently fitted regions certify large balls, whereas samples near class boundaries certify small ones, matching the varying circle sizes in Figure 4.

Step 3: the few represent the many. Suppose the certified balls of the annotated samples cover the whole dataset. Then every sample has a representative in S whose loss is within ϵ of its own. Weighting each representative by the number of samples it covers, the training objective on the small set S becomes a faithful proxy for the objective on the entire $\mathcal{D}$. Theorem 3 turns this into a bound on the risk gap between $\mathcal{M}_{\theta_S}$ and $\mathcal{M}_{\theta_{\mathcal{D}}}$, which Corollary 4 simplifies to 2ϵ , *i.e.*, one local budget for each of the two models.

Step 4: from risk to accuracy, and when to stop. Since the labeling accuracy is a rescaled form of the risk (Equation 1), a risk gap of 2ϵ is an accuracy gap of $2\epsilon/\ln C$; choosing $\epsilon = \frac{1}{2}\zeta \ln C$ thus certifies the slack ζ . This also gives the stopping rule of SALT: annotate until the certified balls cover $\mathcal{D}$. Complete coverage is directly checkable and, by the chain above, implies the accuracy target, so annotation can safely stop.

C Proof of Lemma 2

Lemma 2. Assume $\eta(\mathbf{x})$ is λ -Lipschitz continuous under $\|\cdot\|_2$. Fix $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^k$ with a maximum distance $\|\mathbf{x}' - \mathbf{x}\|_2 \leq \delta$. Then

$$|\ell(\mathbf{x}', \eta') - \ell(\mathbf{x}, \eta)| \leq b\delta + a\delta^2, \quad (7)$$

$$\text{with } a = \frac{1}{2}h\Gamma_\theta^2 + \lambda\Gamma_\theta, \quad b = \|\eta - p\|_2\Gamma_\theta + \lambda\|z\|_2.$$

Here $p = p_\theta(\mathbf{x})$, Γ_θ is the product of spectral norms across network layers of $\mathcal{M}_\theta$, and $h = \|H\|_2$ with $H = \text{diag}(q(\xi)) - q(\xi)q(\xi)^\top$ for some ξ on the segment between $\mathbf{x}$ and $\mathbf{x}'$. $\square$

Proof: We proceed in three steps.

(1) Bounding the logit deviation. Let $z = z_\theta(\mathbf{x})$ be the logits of $\mathcal{M}_\theta$ at $\mathbf{x}$, z' be the logits at $\mathbf{x}'$, $h_l(\mathbf{x})$ the output of the l -th layer at $\mathbf{x}$, and W_{l+1} and b_{l+1} be the weight and bias of the $(l+1)$ -th layer, respectively. Then we have the following:

$$h_{l+1}(\mathbf{x}) = A(W_{l+1}h_l(\mathbf{x}) + b_{l+1}),$$

where A is the activation function. Thus, we have

$$\begin{aligned} & \|h_{l+1}(\mathbf{x}') - h_{l+1}(\mathbf{x})\|_2 \\ & \leq \lambda^A \|(W_{l+1}h_l(\mathbf{x}') + b_{l+1}) - (W_{l+1}h_l(\mathbf{x}) + b_{l+1})\|_2 \\ & = \lambda^A \|W_{l+1}(h_l(\mathbf{x}') - h_l(\mathbf{x}))\|_2 \\ & \leq \lambda^A \|W_{l+1}\|_2 \|h_l(\mathbf{x}') - h_l(\mathbf{x})\|_2, \end{aligned}$$

where λ^A is the Lipschitz constant of the activation function A , and $\|W_{l+1}\|_2$ is the spectral norm of the weight matrix W_{l+1} . In particular, when A is ReLU, we have $\lambda^A = 1$. By induction, we have

$$\|z' - z\|_2 \leq \Gamma_\theta \|\mathbf{x}' - \mathbf{x}\|_2 \leq \Gamma_\theta \delta.$$

That is, each layer can stretch distances by at most its spectral norm, so the whole network stretches the input perturbation δ by at most the product Γ_θ of these norms: nearby inputs yield nearby logits.

(2) Decomposing the loss deviation. We can write the loss function $\ell(\mathbf{x}, \eta)$ as

$$\ell(\mathbf{x}, \eta) = - \sum_{k=1}^C \eta_k \ln p_k = - \sum_{k=1}^C \eta_k z_k + \ln \left(\sum_{k=1}^C e^{z_k} \right).$$

Thus the deviation of the loss can be written as

$$\begin{aligned} \ell(\mathbf{x}', \eta') - \ell(\mathbf{x}, \eta) &= - \sum_{k=1}^C \eta_k z'_k + \ln \left(\sum_{k=1}^C e^{z'_k} \right) \\ &\quad - \left(- \sum_{k=1}^C \eta_k z_k + \ln \left(\sum_{k=1}^C e^{z_k} \right) \right) \\ &\quad + \sum_{k=1}^C (\eta_k - \eta'_k) z'_k. \end{aligned}$$

Intuitively, the loss can change for two reasons: the logits move from z to z' while the target distribution stays η (the first two lines), and the target itself drifts from η to η' (the last term). We bound the two sources separately.

(3) Bounding the two terms. Let $g(z) = - \sum_{k=1}^C \eta_k z_k + \ln \left(\sum_{k=1}^C e^{z_k} \right)$. We have

$$\nabla g(z) = -\eta + p, \quad H(z) = \nabla^2 g(z) = \text{diag}(p) - pp^\top.$$

By Taylor expansion with remainder, we have

$$\begin{aligned} g(z') - g(z) &= \nabla g(z)^\top (z' - z) + \frac{1}{2} (z' - z)^\top H(\xi) (z' - z) \\ &\leq \|\eta - p\|_2 \|z' - z\|_2 + \frac{1}{2} h \|z' - z\|_2^2 \\ &\leq \|\eta - p\|_2 \Gamma_\theta \delta + \frac{1}{2} h \Gamma_\theta^2 \delta^2, \end{aligned}$$

where ξ is some point on the segment between z and z' , and $h = \|H(\xi)\|_2$. The first-order term is driven by the prediction mismatch $\|\eta - p\|_2$, and the second-order term by the curvature h of the log-sum-exp function; these are exactly the two coefficients appearing in the lemma. For the last term, we have

$$\begin{aligned}
 \sum_{k=1}^C (\eta_k - \eta'_k) z'_k &= \sum_{k=1}^C (\eta_k - \eta'_k) (z'_k - z_k + z_k) \\
 &\leq \left| \sum_{k=1}^C (\eta_k - \eta'_k) (z'_k - z_k) \right| + \left| \sum_{k=1}^C (\eta_k - \eta'_k) z_k \right| \\
 &\leq \lambda \Gamma_\theta \delta^2 + \lambda \|z\|_2 \delta.
 \end{aligned}$$

Both inequalities use the λ -Lipschitz continuity of $\eta(\cdot)$: $\|\eta - \eta'\|_2 \leq \lambda \delta$: the target distribution cannot drift faster than the inputs move.

Combining the above bounds, we have

$$|\ell(\mathbf{x}', \eta') - \ell(\mathbf{x}, \eta)| \leq b\delta + a\delta^2,$$

where $a = \frac{1}{2}h\Gamma_\theta^2 + \lambda\Gamma_\theta$, $b = \|\eta - p\|_2\Gamma_\theta + \lambda\|z\|_2$.

As stated in Step 1 of the roadmap (Appendix B), the bound is small exactly when the model already fits the sample (small $\|\eta - p\|_2$) and the local loss surface is flat (small h). $\square$

D Proof of Theorem 3

Theorem 3. Consider a model $\mathcal{M}_{\theta_S}$ trained over a labeled subset $\mathcal{D}_S$. Under the coverage condition $\mathcal{D} \subseteq \bigcup_{\mathbf{x}_i \in S} B(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta_S))$ with a surjective assignment induced by τ and weights w_i , we have:

$$\begin{aligned}
 |R_{\mathcal{D}}(\theta_S) - R_{\mathcal{D}}(\theta_{\mathcal{D}})| &\leq \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}_i \in S} \left[w_i e(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta_S), \theta_S) \right. \\
 &\quad \left. + w_i \widehat{e}(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta_S), \theta_{\mathcal{D}}) \right],
 \end{aligned} \tag{4}$$

where $\widehat{e}(\mathbf{x}, \delta, \theta) = (\sqrt{2}\Gamma + \lambda B)\delta + (\frac{1}{4}\Gamma^2 + \lambda\Gamma)\delta^2$. $\square$

Proof: The roadmap of the proof is as follows. We first apply a triangle decomposition to separate the target gap into two deviations between the full-data empirical risk and the weighted risk on $\mathcal{D}_S$:

$$|R_{\mathcal{D}}(\theta_S) - R_{\mathcal{D}}(\theta_{\mathcal{D}})| \leq \left| R_{\mathcal{D}}(\theta_S) - \widehat{R}_{\mathcal{D}_S}(\theta_S) \right| + \left| \widehat{R}_{\mathcal{D}_S}(\theta_{\mathcal{D}}) - R_{\mathcal{D}}(\theta_{\mathcal{D}}) \right|.$$

Then, for any fixed θ , we use the coverage assignment τ to associate each full-data point with a labeled center and then apply Lemma 2 to control the center-to-point loss deviation by the corresponding local variation term, which yields a weighted sum of $e(\cdot)$ contributions. Finally, we instantiate this fixed- θ bound at $\theta = \theta_S$ and $\theta = \theta_{\mathcal{D}}$. Because some factors under $\theta_{\mathcal{D}}$ are inaccessible, we upper bound them by $\|\eta - p\|_2 \leq \sqrt{2}$ and $h \leq 1/2$, which produces the surrogate term $\widehat{e}$ and completes the bound in Equation 4.

We next present the details.

(1) Triangle decomposition. First, by triangle inequality, we have

$$\begin{aligned}
 R_{\mathcal{D}}(\theta_S) - R_{\mathcal{D}}(\theta_{\mathcal{D}}) &= R_{\mathcal{D}}(\theta_S) - \widehat{R}_{\mathcal{D}_S}(\theta_S) + \widehat{R}_{\mathcal{D}_S}(\theta_S) - \widehat{R}_{\mathcal{D}_S}(\theta_{\mathcal{D}}) \\
 &\quad + \widehat{R}_{\mathcal{D}_S}(\theta_{\mathcal{D}}) - R_{\mathcal{D}}(\theta_{\mathcal{D}}).
 \end{aligned}$$

Since θ_S is the minimizer of $\widehat{R}_{\mathcal{D}_S}(\theta)$, we have the middle term $\widehat{R}_{\mathcal{D}_S}(\theta_S) - \widehat{R}_{\mathcal{D}_S}(\theta_{\mathcal{D}}) \leq 0$. Thus, we have

$$R_{\mathcal{D}}(\theta_S) - R_{\mathcal{D}}(\theta_{\mathcal{D}}) \leq |R_{\mathcal{D}}(\theta_S) - \widehat{R}_{\mathcal{D}_S}(\theta_S)| + |\widehat{R}_{\mathcal{D}_S}(\theta_{\mathcal{D}}) - R_{\mathcal{D}}(\theta_{\mathcal{D}})|.$$

In words, the middle term drops because θ_S is, by definition, the best model for the weighted objective $\widehat{R}_{\mathcal{D}_S}$; what remains is to show that $\widehat{R}_{\mathcal{D}_S}$ approximates the full-data risk $R_{\mathcal{D}}$ well, at each of the two models of interest.

(2) Bounding the deviation for a fixed model. Next, we upper bound the two terms on the right-hand

side of the above inequality. Since D is covered by the union of the balls centered at $\mathbf{x}_i \in S$ with radius $\delta_{\mathbf{x}_i}(\epsilon, \theta_S)$, we can use the assignment τ to associate each $\mathbf{x} \in D$ with a center $\mathbf{x}_i \in S$ such that $\mathbf{x} \in B(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta_S))$. Thus for any fixed $\hat{\theta}$, and $\mathbf{x} \in B(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta_S))$, we can apply Lemma 2 to upper bound the loss deviation as follows:

$$|\ell(\mathbf{x}, \eta) - \ell(\mathbf{x}_i, \eta_i)| \leq e(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta_S), \hat{\theta}),$$

where η_i is $\mathbf{x}_i$'s ground truth label and $e(\cdot)$ is defined in Lemma 2.

Thus, the total deviation between the full-data empirical risk and the weighted risk on $\mathcal{D}_S$ can be upper bounded by

$$|R_{\mathcal{D}}(\hat{\theta}) - \hat{R}_{\mathcal{D}_S}(\hat{\theta})| \leq \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}_i \in S} w_i e(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta_S), \hat{\theta}),$$

where w_i is the number of points in $\mathcal{D}$ that are associated with $\mathbf{x}_i$ by the assignment τ . This is the ‘‘few represent the many’’ step of the roadmap (Appendix B): grouping the points of $\mathcal{D}$ by their assigned centers, the group of the w_i points represented by $\mathbf{x}_i$ contributes at most w_i times the local variation bound of its center.

(3) Instantiating at θ_S and $\theta_{\mathcal{D}}$. By substituting $\hat{\theta} = \theta_S$, the first term is bounded by

$$|R_{\mathcal{D}}(\theta_S) - \hat{R}_{\mathcal{D}_S}(\theta_S)| \leq \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}_i \in S} w_i e(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta_S), \theta_S).$$

For the second term, since we cannot access the model parameters $\theta_{\mathcal{D}}$ trained on the full dataset, we upper bound the factors $\|\eta - p\|_2$ and h in $e(\cdot)$. Indeed, evaluating these factors at $\theta_{\mathcal{D}}$ would require the full annotation that SALT precisely avoids; we therefore replace them with worst-case constants valid for *any* model, at the price of a looser but computable surrogate $\hat{e}$. For $\|\eta - p\|_2$, we have $\|\eta - p\|_2 \leq \sqrt{2}$ since both η and p are probability vectors.

For h , we have $h = \|H\|_2$ where $H = \text{diag}(q) - qq^T$ for some probability vector q . By a standard fact in matrix analysis, we have

$$\|H\|_2 \leq \sqrt{\|H\|_1 \|H\|_{\infty}},$$

where $\|H\|_1$ and $\|H\|_{\infty}$ are the maximum absolute column sum and row sum of H , respectively. Since H is symmetric, $\|H\|_1 = \|H\|_{\infty}$. Thus, we only need to upper bound $\|H\|_1$. For any column j ,

$$\sum_{k=1}^C |H_{kj}| = q_j(1 - q_j) + \sum_{k \neq j} q_k q_j = 2q_j(1 - q_j) \leq 1/2,$$

where the last inequality is due to the fact that $q_j(1 - q_j)$ is maximized at $q_j = 1/2$. Thus, we have $h = \|H\|_2 \leq 1/2$. By substituting the above bounds into $e(\cdot)$, we have $e(\mathbf{x}, \delta, \theta_{\mathcal{D}}) \leq (\sqrt{2}\Gamma + \lambda B)\delta + (\frac{1}{4}\Gamma^2 + \lambda\Gamma)\delta^2 := \hat{e}(\mathbf{x}, \delta, \theta_{\mathcal{D}})$.

Thus, the second term is bounded by

$$|\hat{R}_{\mathcal{D}_S}(\theta_{\mathcal{D}}) - R_{\mathcal{D}}(\theta_{\mathcal{D}})| \leq \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}_i \in S} w_i \hat{e}(\mathbf{x}_i, \delta_{\mathbf{x}_i}(\epsilon, \theta_S), \theta_{\mathcal{D}}),$$

which completes the proof. $\square$

E Proof of Corollary 4

Corollary 4. Under the coverage condition of Theorem 3, we have:

$$|R_{\mathcal{D}}(\theta_S) - R_{\mathcal{D}}(\theta_{\mathcal{D}})| \leq 2\epsilon, \quad (8)$$

when θ_S and $\theta_{\mathcal{D}}$ recover the true conditional probabilities on S , i.e., $p_{\theta_S}(\mathbf{x}_i) = p_{\theta_{\mathcal{D}}}(\mathbf{x}_i) = \eta(\mathbf{x}_i)$ for every $\mathbf{x}_i \in S$. $\square$

Proof: We instantiate the bound in Theorem 3 under the assumption $p_{\theta_S}(\mathbf{x}_i) = p_{\theta_D}(\mathbf{x}_i) = \eta(\mathbf{x}_i)$ for every $\mathbf{x}_i \in S$, and show that each of the two terms in the bound is at most ϵ . Intuitively, each term costs one local loss budget: the first by the very definition of the certified radius, and the second because the well-trained assumption removes the prediction-mismatch slack from the surrogate $\widehat{e}$, so the same radius keeps the ideal model within budget as well. Throughout the proof, we write $\delta_i := \delta_{\mathbf{x}_i}(\epsilon, \theta_S)$ for brevity.

By Theorem 3, we have

$$\begin{aligned} |R_D(\theta_S) - R_D(\theta_D)| &\leq \underbrace{\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}_i \in S} w_i e(\mathbf{x}_i, \delta_i, \theta_S)}_{(I)} \\ &\quad + \underbrace{\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}_i \in S} w_i \widehat{e}(\mathbf{x}_i, \delta_i, \theta_D)}_{(II)}. \end{aligned}$$

Since the assignment τ is surjective, every point of $\mathcal{D}$ is associated to exactly one center, so $\sum_{\mathbf{x}_i \in S} w_i = |\mathcal{D}|$.

(1) Bounding (I). By the definition of the local cover radius in Equation 3, δ_i is the largest δ satisfying $e(\mathbf{x}_i, \delta, \theta_S) \leq \epsilon$ at the model θ_S that defines the coverage. Thus we obtain

$$e(\mathbf{x}_i, \delta_i, \theta_S) \leq \epsilon, \quad \forall \mathbf{x}_i \in S.$$

Summing with weights w_i and using $\sum_i w_i = |\mathcal{D}|$,

$$(I) \leq \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}_i \in S} w_i \epsilon = \epsilon.$$

(2) Bounding (II). Recall from the proof of Theorem 3 that the surrogate $\widehat{e}$ was obtained from the exact bound

$$\begin{aligned} e(\mathbf{x}_i, \delta, \theta_D) &= (\|\eta(\mathbf{x}_i) - p_{\theta_D}(\mathbf{x}_i)\|_2 \Gamma_{\theta_D} + \lambda \|z(\mathbf{x}_i; \theta_D)\|_2) \delta \\ &\quad + \left(\frac{1}{2} h_D \Gamma_{\theta_D}^2 + \lambda \Gamma_{\theta_D}\right) \delta^2 \end{aligned}$$

by replacing the two inaccessible factors at θ_D with their worst-case upper bounds:

$$\|\eta(\mathbf{x}_i) - p_{\theta_D}(\mathbf{x}_i)\|_2 \leq \sqrt{2}, \quad h_D \leq \frac{1}{2}.$$

Under the corollary's assumption, $p_{\theta_D}(\mathbf{x}_i) = \eta(\mathbf{x}_i)$ at every center $\mathbf{x}_i \in S$, so the first slack is in fact zero: the $\sqrt{2} \Gamma$ contribution vanishes. Hence the tightened surrogate is

$$\widehat{e}(\mathbf{x}_i, \delta, \theta_D) \leq \lambda B \delta + \left(\frac{1}{4} \Gamma^2 + \lambda \Gamma\right) \delta^2.$$

Symmetrically, under the assumption $p_{\theta_S}(\mathbf{x}_i) = \eta(\mathbf{x}_i)$, the coefficients a, b in $e(\mathbf{x}_i, \cdot, \theta_S)$ simplify to $b_S = \lambda \|z(\mathbf{x}_i; \theta_S)\|_2$ and $a_S = \frac{1}{2} h_S \Gamma_{\theta_S}^2 + \lambda \Gamma_{\theta_S}$, and by $\|z\|_2 \leq B$, $\Gamma_{\theta_S} \leq \Gamma$, and $h_S \leq 1/2$, we also have $b_S \leq \lambda B$ and $a_S \leq \frac{1}{4} \Gamma^2 + \lambda \Gamma$. Consequently, $e(\mathbf{x}_i, \delta, \theta_S)$ and the tightened $\widehat{e}(\mathbf{x}_i, \delta, \theta_D)$ share the same polynomial upper bound in δ . Applying Equation 3 with these uniform coefficients, the radius δ_i at which the radius condition is saturated satisfies

$$\widehat{e}(\mathbf{x}_i, \delta_i, \theta_D) \leq \epsilon, \quad \forall \mathbf{x}_i \in S.$$

Summing with weights and using $\sum_i w_i = |\mathcal{D}|$,

$$(II) \leq \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}_i \in S} w_i \epsilon = \epsilon.$$

(3) Combining. Adding the two bounds, we obtain

$$|R_D(\theta_S) - R_D(\theta_D)| \leq (I) + (II) \leq \epsilon + \epsilon = 2\epsilon,$$

which is the claimed bound in Equation 5. Together with the risk-to-accuracy conversion of Step 4 of the roadmap (Section 4.2), *i.e.*, setting $\epsilon = \frac{1}{2}\zeta \ln C$, this completes the derivation chain and yields the certified accuracy guarantee of SALT. $\square$

F Proof of Theorem 5

Theorem 5. RSAP is NP-hard. $\square$

Proof: We first recall the problem RSAP:

- *Input:* Unlabeled D , annotated $\tilde{D}_{t-1}$ from previous rounds, model $\mathcal{M}_{t-1}$ trained on $\tilde{D}_{t-1}$, tolerance ζ , and round budget bgt .
- *Output:* A subset $D_t \subseteq D$ to be annotated in round t .
- *Objective:* To maximize the certified coverage attained by the updated model $\mathcal{M}_t$ trained on $\tilde{D}_t = \tilde{D}_{t-1} \cup D_t$.

To simplify discussion, we assume that the predicted labels of all samples in $\tilde{D}_{t-1}$ and D match the eventual human annotations. We will show that even this restricted version of RSAP is NP-hard. In this setting, for each point $\mathbf{x}$ in D and $\tilde{D}_{t-1}$, the certified radius is fixed, which can be obtained by directly applying Equation 3. Thus the set of points covered by $\mathbf{x}$, denoted by $S_{\mathbf{x}}$, is known. In other words, once the certified radii are fixed, annotating a candidate $\mathbf{x}$ simply “buys” a known set $S_{\mathbf{x}}$ of covered points, and the problem asks to cover the most points with bgt purchases, *i.e.*, a maximum-coverage problem. This restricted version of RSAP is equivalent to:

- *Input:* A set D , a collection of subsets $\{S_{\mathbf{x}} \mid \mathbf{x} \in D\}$ with $\mathbf{x} \in S_{\mathbf{x}}$, and a budget bgt .
- *Output:* A subset $D_t \subseteq D$, with $|D_t| \leq \text{bgt}$.
- *Objective:* To maximize $|\bigcup_{\mathbf{x} \in D_t} S_{\mathbf{x}}|$.

We will show the NP-hardness of RSAP by showing its deterministic version DRSAP is NP-complete, which is defined as follows:

- *Input:* A set D , a collection of subsets $\{S_{\mathbf{x}} \mid \mathbf{x} \in D\}$ with $\mathbf{x} \in S_{\mathbf{x}}$, a budget bgt , and an integer A .
- *Problem:* Is there a subset $D_t \subset D$ such that $|D_t| \leq \text{bgt}$ and $|\bigcup_{\mathbf{x} \in D_t} S_{\mathbf{x}}| \geq A$?

We now prove that DRSAP is NP-complete.

Upper bound. We first provide an NP algorithm for DRSAP.

Given such inputs, the algorithm works as follows:

- (1) It first guesses a subset D_t with $|D_t| \leq \text{bgt}$.
- (2) It computes the union $\bigcup_{\mathbf{x} \in D_t} S_{\mathbf{x}}$.
- (3) It checks whether $|\bigcup_{\mathbf{x} \in D_t} S_{\mathbf{x}}| \geq A$, if so, return true. Step 2 and Step 3 both run in PTIME. Thus DRSAP is in NP.

Lower bound. We prove NP-hardness of DRSAP by a reduction from SET-COVER problem (see the proof of Theorem 1 for SET-COVER).

Intuitively, given such an instance of the SET-COVER, we construct a set $D = U \cup \{x_1, \dots, x_n\}$, where x_i represents the corresponding set C_i in $\mathcal{C}$. We construct $S_{\mathbf{x}}$ for each $\mathbf{x} \in D$, such that the SET-COVER has a solution if and only if there exists $|D_t| \leq k$ and $|\bigcup_{\mathbf{x} \in D_t} S_{\mathbf{x}}| \geq m + k$. We give the reduction as follows.

Constructions. Given $U = \{u_1, \dots, u_m\}$ and $\{C_1, \dots, C_n\}$, we let

$$D = U \cup \{x_1, \dots, x_n\},$$

where x_i represents the corresponding set C_i .

For each $\mathbf{x} \in D$, we construct

$$S_{u_j} = \{u_j\}, \quad S_{x_i} = x_i \cup C_i,$$

which meets the requirement that $x \in S_x$.

We set $\text{bgt} = k$ and $A = m + k$.

Correctness of the reduction. The above reduction is clearly in PTIME. We next show that the reduction is correct.

($\Rightarrow$) If there exists a set cover $\{C_{i_1}, \dots, C_{i_r}\}$ with $r \leq k$, we let D_t be the subset of $\{x_1, \dots, x_n\}$ containing $x_{i_1}, \dots, x_{i_r}$ and $|D_t| = k$. Since $\bigcup_{l=1}^r C_{i_l} = U$, we conclude that $|\bigcup_{x \in D_t} S_x| = m + k$.

($\Leftarrow$) If there exists D_t with $|D_t| \leq k$ and $|\bigcup_{x \in D_t} S_x| \geq m + k$, then $U \subset \bigcup_{x \in D_t} S_x$. Intuitively, k picks can gather at most the k picked set-vertices themselves plus the m elements they cover, so reaching the target $m + k$ leaves no slack: every pick must be a set-vertex and the chosen sets must cover all of U . Otherwise, we have $|\bigcup_{x \in D_t} S_x|$ is at most $m - 1 + k$. Moreover, we must have $|D_t| = k$ and $D_t \cap U = \emptyset$, otherwise $|\bigcup_{x \in D_t} S_x|$ will be at most $m + k - 1$. Thus the corresponding set $\{C_i \mid x_i \in D_t\}$ is a cover of U with size k . $\square$

G Proof of Theorem 6

Theorem 6. Given candidate samples ψ_0 at round t , define the expanded coverage of a subset A of ψ_0 as $f(A) := |\bigcup_{x \in A} (\psi_0 \cap \mathbb{B}(x))|$. Then we have $f(D_t) \geq (1 - \frac{1}{e}) \max_{A \subseteq D, |A| \leq \text{bgt}} f(A)$. $\square$

Proof: We show that when $\alpha(1 - \frac{|\psi|}{|D|}) = 0$, Algorithm 1 reduces to the classical greedy algorithm for maximum coverage, and the approximation guarantee follows from the standard result for greedy maximization of monotone submodular functions [55].

Intuitively, coverage is a task with diminishing returns: the more points are already covered, the fewer new points any additional ball can contribute. This is precisely monotone submodularity, for which greedy selection is $(1 - 1/e)$ -optimal; the crux is thus to verify that the scoring of Algorithm 1 indeed picks, at every step, the ball that covers the most new points.

Setup. For each candidate $x \in \psi_0$, define $S_x := \psi_0 \cap \mathbb{B}(x)$ as the set of candidates covered by the ball of x . The coverage function can then be written as $f(A) = |\bigcup_{x \in A} S_x|$ for any $A \subseteq \psi_0$. This is exactly the maximum coverage objective over universe ψ_0 with collection $\mathcal{S} = \{S_x : x \in \psi_0\}$ and cardinality constraint $|A| \leq \text{bgt}$.

Monotonicity. For any $A \subseteq B \subseteq \psi_0$, we have $\bigcup_{x \in A} S_x \subseteq \bigcup_{x \in B} S_x$. Thus, $f(A) \leq f(B)$, i.e., function f is monotone.

Submodularity. For any $A \subseteq B \subseteq \psi_0$ and any $x \notin B$, the marginal gain of adding x satisfies

$$\begin{aligned} f(A \cup \{x\}) - f(A) &= |S_x \setminus \bigcup_{y \in A} S_y| \\ &\geq |S_x \setminus \bigcup_{y \in B} S_y| = f(B \cup \{x\}) - f(B), \end{aligned}$$

where the inequality holds because $\bigcup_{y \in A} S_y \subseteq \bigcup_{y \in B} S_y$, so removing a larger set from S_x leaves no more elements. Intuitively, the marginal value of a ball only shrinks when the selection grows, i.e., diminishing returns.

Greedy algorithm. We now show that Algorithm 1 reduces to a greedy coverage maximization algorithm. When $\alpha(1 - \frac{|\psi|}{|D|}) = 0$, the scoring function (Equation 6) reduces to $\text{score}(x, \psi) = \ln |\psi \cap \mathbb{B}(x)|$. Since $\ln$ is strictly monotone increasing, the candidate maximizing this score also maximizes $|\psi \cap \mathbb{B}(x)|$.

We show that at each greedy step k , $|\psi \cap \mathbb{B}(x)|$ equals the marginal coverage gain $f(D_t^{k-1} \cup \{x\}) - f(D_t^{k-1})$, where $D_t^{k-1} = \{x^{(1)}, \dots, x^{(k-1)}\}$ denotes the samples selected in the first $k-1$ steps. By line 6 of Algorithm 1, the candidate pool at step k is $\psi = \psi_0 \setminus \bigcup_{j=1}^{k-1} \mathbb{B}(x^{(j)})$. For any $x \in \psi$,

$$\begin{aligned}
|\psi \cap \mathbb{B}(\mathbf{x})| &= \left| (\psi_0 \cap \mathbb{B}(\mathbf{x})) \setminus \bigcup_{j=1}^{k-1} \mathbb{B}(\mathbf{x}^{(j)}) \right| \\
&= \left| S_{\mathbf{x}} \setminus \bigcup_{j=1}^{k-1} S_{\mathbf{x}^{(j)}} \right| = f(D_t^{k-1} \cup \{\mathbf{x}\}) - f(D_t^{k-1}).
\end{aligned}$$

The second equality uses $S_{\mathbf{x}} \subseteq \psi_0$: for any $\mathbf{z} \in S_{\mathbf{x}}$, $\mathbf{z} \in \mathbb{B}(\mathbf{x}^{(j)})$ if and only if $\mathbf{z} \in \psi_0 \cap \mathbb{B}(\mathbf{x}^{(j)}) = S_{\mathbf{x}^{(j)}}$. Hence Algorithm 1 selects at each step the candidate with the largest marginal gain of f , which is exactly the classical greedy strategy for maximum coverage.

Approximation guarantee. As f is monotone and submodular, and Algorithm 1 greedily maximizes the marginal gain of f subject to $|D_t| \leq \text{bgt}$, the standard result for greedy maximization of monotone submodular functions under a cardinality constraint [55] yields

$$f(D_t) \geq \left(1 - \frac{1}{e}\right) \max_{A \subseteq D, |A| \leq \text{bgt}} f(A).$$

□

H Training a Fusion Model

Recall the fusion model $\mathcal{P}$, which takes weak label results of LFs and generates robust fused weak labels. To train $\mathcal{P}$ effectively, we extend the idea in Snorkel LabelModel [62] by incorporating human annotated labels as strong labels to improve the estimation of labeling function reliability. This method prioritizes high-confidence supervision signals and gradually resorts to probabilistic fusion.

Let Y_i denote the latent true label of sample i . Each labeling function can be modeled by a confusion matrix describing the probability of emitting label $\hat{y}$ given the true label y :

$$\mu_{y\hat{y}}^{(j)} = \Pr(\lambda_{ij} = \hat{y} | Y_i = y).$$

The idea of probabilistic label fusion is that the true labels are unobserved and the parameters are estimated by maximizing the marginal likelihood of the observed labeling function outputs [62], which can be expressed by $\mu_{y\hat{y}}^{(j)}$:

$$\Pr(\lambda_i) = \sum_{y=1}^K \Pr(Y_i = y) \prod_{j=1}^m \mu_{y, \lambda_{ij}}^{(j)}.$$

Here K is the number of classes, and $\Pr(Y_i = y)$ is the class distribution of the dataset. For human annotated samples, the strong label are treated as true labels. Therefore, we split the likelihood function into two parts and balance them using a parameter β :

$$\begin{aligned}
\mathcal{L}(\mu) &= (1 - \beta) \sum_{i \notin D_t^h} \sum_{y=1}^K \Pr(Y_i = y) \prod_{j=1}^m \mu_{y, \lambda_{ij}}^{(j)} \\
&\quad + \beta \sum_{i \in D_t^h} \Pr(Y_i) \prod_{j=1}^m \mu_{Y_i, \lambda_{ij}}^{(j)}.
\end{aligned}$$

This objective allows the strong label to directly supervise the estimation of labeling function confusion matrices, while the unlabeled data provides additional statistical structure through labeling function agreement patterns. To maximize the likelihood function, parameter μ can be determined using gradient-based optimization. Finally, the fused soft labels can be computed using Bayes rule:

$$\Pr(Y_i = y | \lambda_i) = \frac{\Pr(Y_i = y) \prod_{j=1}^m \mu_{y, \lambda_{ij}}^{(j)}}{\sum_{y'=1}^K \Pr(Y_i = y') \prod_{j=1}^m \mu_{y', \lambda_{ij}}^{(j)}}.$$

I Discovery of LRRs

We mine LRRs offline from prior similar labeling tasks, treating human annotations as ground truth. The goal is to construct an offline rule set Σ that can later be reused in online HITL labeling.

Overview. In general, the LRR mining procedure is adapted from conventional levelwise rule mining algorithms. One major distinction lies in pruning: support is still anti-monotone here, but, as shown below, is not effective and prunes very little.

Support of LRRs. Let $\varphi : X \rightarrow x.\text{label} = w$ be an LRR mined over a human-labeled dataset D_H . Following standard rule mining [25, 26], we measure how often φ is correct by the *support* supp of φ in D_H .

We first collect the samples on which φ fires, *i.e.*, its *firing set*

$$\text{fireset}(\varphi, D_H) = \{a \in D_H \mid \exists h, h(x) = a, h \models \varphi\},$$

where h ranges over the valuations of φ over D_H . The *support* of φ is then the number of samples in its firing set whose consequence is confirmed by the human annotation:

$$\text{supp}(\varphi, D_H) = |\{a \in \text{fireset}(\varphi, D_H) \mid y_a = w\}|,$$

where y_a denotes the annotated label of a ; Intuitively, support counts the samples that φ labels correctly.

Proposition 10: *The support of LRRs is anti-monotone.* □

Proof: Consider two LRRs $\varphi' : X' \rightarrow q$ and $\varphi : X \rightarrow q$. We say that φ' extends φ if X' retains all predicates of X and adds more (possibly over new variables). Thus, the anti-monotonicity of support holds if $\text{supp}(\varphi', D_H) \leq \text{supp}(\varphi, D_H)$. As φ and φ' share the same consequence q , it suffices to show $\text{fireset}(\varphi', D_H) \subseteq \text{fireset}(\varphi, D_H)$; intersecting both sides with the samples annotated w preserves the inclusion, and so does taking cardinalities.

Consider a sample $a \in \text{fireset}(\varphi', D_H)$, witnessed by a valuation h' with $h'(x) = a$ and $h' \models \varphi'$. Let h be the restriction of h' to the variables of φ . Every predicate p of X also occurs in X' , and whether p is satisfied depends only on the samples assigned to the variables of p , which h and h' assign identically. Hence $h \models p$ for all p in X , *i.e.*, $h \models \varphi$. Since $h(x) = a$, we have that $a \in \text{fireset}(\varphi, D_H)$.

This proves the anti-monotonicity of support. □

Remarks on pruning effectiveness. Anti-monotonicity is what licenses levelwise pruning: once the support of a conjunction X falls below a threshold σ , no extension of X can recover, and the entire branch is discarded. While the inequality of Proposition 10 holds in our setting, it is almost always *tight*, and hence a branch is rarely discarded. There are three reasons for this.

(1) Cross-sample predicates hardly filter anything. Adding $M(x, x')$ drops a sample a from the firing set only if a has *no* related sample at all in D_H . Relation models such as topic similarity or geo-proximity are dense, *i.e.*, almost every sample has some related sample. Such a predicate therefore changes which evidence supports a , and the support of a child stays close to that of its parent.

(2) Soft predicates come with a cut-off threshold c . A conjunction is extended by a family $\text{Pr}(x, w) \otimes c$ indexed by a continuous c , whose value is derived from the data (step (3) below). Loosening c recovers nearly all the support of the parent, so every branch retains children above σ . Worse still, support pruning bites only at tight thresholds, which are exactly the discriminative ones.

(3) Useful LRRs are often rare. The most valuable ones correct a systematic but infrequent conflict among LFs, or filter a small unreliable subpopulation. Hence σ has to be kept low so as not to lose them, which disables support pruning even further. What separates a good LRR from a bad one

here is confidence and coverage impact rather than frequency, and neither is anti-monotone, since specializing a rule can raise its confidence arbitrarily.

Proposition 10 thus remains sound, though. The mining procedure below keeps the support filter, which is cheap and safe, with a loose σ , and does the actual pruning with confidence and coverage impact measured on human-labeled data (step (4)).

LRR mining procedure. It runs the following steps iteratively:

(1) Predicate construction. We first instantiate a finite pool of candidate predicates from the available weak-supervision signals.

Tool predicates $T(\text{LF}, x, w)$ and $\neg T(\text{LF}, x, w)$ are existing LFs. For example, for text classification, we may include regular-expression and keyword-based labelers as well as outputs from other weak supervision tools. When the task lacks rich keyword labelers, an LLM can be used to analyze the dataset and propose a keyword dictionary, which is then converted into keyword-based LFs.

We also instantiate relationship predicates $M(x, x')$ from available relation detectors, and soft predicates $\text{Pr}(x, w) \otimes c$ from the fusion model. This yields a predicate pool that captures weak labels, cross-sample evidence, and gold-calibrated soft confidence.

(2) Levelwise search. Starting from single predicates, we perform a bottom-up search over conjunctions in the predicate space. At level ℓ , the algorithm joins surviving $(\ell - 1)$ -predicate conjunctions that share a common prefix, canonicalizes the resulting conjunctions to avoid duplicates, and uses them as candidate preconditions.

For each precondition X , we form candidate LRRs by attaching a consequence $x.\text{label} = w$ for each valid class label w , and, when abstention is preferable, the filtering consequence $x.\text{label} = \perp$. Evaluation is performed on the human-labeled data D_H .

For a conflict-resolving rule $\varphi_w : X \rightarrow x.\text{label} = w$, we measure how reliably the precondition X predicts the consequence on human-labeled data, using the same accounting as supp:

$$\text{conf}(\varphi_w) = \frac{\text{supp}(\varphi_w, D_H)}{|\text{fireset}(\varphi_w, D_H)| + \epsilon}, \quad (9)$$

where $\epsilon > 0$ is a smoothing constant. That is, among the samples on which φ_w fires, $\text{conf}(\varphi_w)$ is the fraction whose annotation confirms the consequence.

For filtering rules with consequence $\perp$, this counts how often the rule fires on samples whose weak labels are contradicted by the human annotation or remain too inconsistent to trust.

(3) Threshold selection for soft predicates. Predicates of the form $\text{Pr}(x, w) \otimes c$ introduce a continuous threshold parameter, so they cannot be enumerated directly. To discretize this space, we collect the fusion-model scores observed on human-labeled data, sort them, and consider split points between consecutive values. For a target consequence label w , we treat samples with human label w as positives and the remaining samples as negatives. We then adapt the Gini-impurity criterion from decision-tree learning [15] to choose the threshold c that best separates these two groups. This converts a continuous family of soft predicates into a small set of data-dependent predicates with strong predictive utility, while reducing overfitting and avoiding exhaustive threshold search.

(4) Conservative and adaptive pruning. By Proposition 10 above, support pruning is sound but not very helpful in our setting. Thus we use conservative and adaptive pruning instead of aggressive frequency-only pruning. At each level, we first apply a loose support threshold σ to retain potentially useful candidates. We then filter induced rules using confidence and estimated coverage impact on human-labeled data, where coverage impact measures how much a rule helps confirm, correct or filter weak labels on the samples it covers. Rules with negligible confidence or impact are removed, while rare but high-value candidates are allowed to survive. Pruning is therefore relaxed when support is low, so that uncommon but important patterns are not discarded too early, and becomes

stricter as support for an LRR increases. In practice, this strategy preserves important rules while substantially reducing search cost.

Output. The search terminates when no new conjunction survives or a preset maximum rule length is reached. The remaining LRRs are added to Σ together with expert-specified rules and later reused for online weak-label correction.

J Handling Conflicts of LRRs

Conflicts arise when multiple applicable LRRs fire on the same sample but imply different labels. Although such cases are rare in our setting, we resolve them with an evidence-aware ranking strategy instead of a fixed global priority order.

Applicable rules. Let Σ be the set of LRRs used. For a sample x , let $\Sigma(x)$ denote the set of learned LRRs whose preconditions are satisfied on x . Equivalently, $\Sigma(x)$ is the subset of Σ that fires on x under the current weak labels, inter-sample relations, and soft-label predicates. If all rules in $\Sigma(x)$ imply the same label, we directly use that label. Conflict handling is only needed when at least two rules in $\Sigma(x)$ imply different consequences.

Resolution score. For each applicable LRR $\varphi \in \Sigma(x)$, we compute

$$\text{score}(\varphi, x) = \alpha \cdot \text{conf}(\varphi) + \beta \cdot \text{spec}(\varphi) + \delta \cdot \text{soft}(\varphi, x),$$

where $\alpha, \beta, \gamma, \delta \geq 0$ are normalized weights. The three terms capture complementary evidence from global rule quality, rule structure, and sample-specific model support, respectively. In practice, these terms are normalized to comparable scales before aggregation so that no single component dominates purely due to magnitude.

(1) Rule confidence. Term $\text{conf}(\varphi)$ measures the reliability of LRR φ on the human-labeled data collected so far. Computed by Equation 9, it favors LRRs that are empirically accurate on trusted annotations. It provides a precision-style estimate of how often φ produces the correct consequence when it fires on labeled samples.

(2) Rule specificity. Term $\text{spec}(\varphi)$ prefers LRRs with more specific predicates in the precondition, since they usually capture narrower and more reliable conditions. More specifically, let $X = p_1 \wedge \dots \wedge p_k$ be the precondition of rule φ , where $p_1, \dots, p_k$ are the predicates. For each predicate p_j , we assign an informativeness weight $\omega(p_j) \in [0, 1]$, where more selective predicates receive larger weights. For example, a predicate with a stricter threshold or a rarer matching condition is treated as more informative than a common or weakly constrained predicate. We define

$$\text{spec}(\varphi) = \frac{\sum_{j=1}^k \omega(p_j)}{k_{\max}},$$

where $k_{\max}$ is the maximum possible total weight under the allowed rule template. Under this definition, an LRR with more informative predicates or a longer conjunction of selective conditions receives a larger specificity score. This term helps prevent a broad rule with high coverage from overriding a more precise rule that matches the current sample more tightly.

(3) Agreement with soft labels. Term $\text{soft}(\varphi, x)$ measures whether the rule consequence agrees with the fusion model's prediction for the same sample. If rule φ predicts label w_φ , we use the calibrated soft label $\text{Pr}(x, w_\varphi)$ from model $\mathcal{P}$ as the agreement score, or its normalized variant. This term injects sample-level evidence into conflict resolution, which is especially helpful when several rules have similar global reliability. It also couples conflict handling with the calibrated fusion model, so the final decision is informed by both symbolic rules and probabilistic supervision.

Decision rule and abstention. We select

Table 2. Complete set of 21 LRRs discovered on THUCNews.

#	LRR	Explanation
1	$D(x) \wedge T(\text{LF}_{\text{lottery}}, x, \text{lottery}) \rightarrow x.\text{label} = \text{lottery}$	Trusts distinctive lottery and draw-result cues.
2	$D(x) \wedge T(\text{LF}_{\text{zodiac}}, x, \text{constellation}) \rightarrow x.\text{label} = \text{constellation}$	Trusts horoscope and zodiac cues for constellation news.
3	$D(x) \wedge \Pr(x, \text{tech}) > 0.93 \rightarrow x.\text{label} = \text{tech}$	Accepts a technology label only at near-certain fused confidence.
4	$D(x) \wedge T(\text{LF}_{\text{housing}}, x, \text{realty}) \wedge \Pr(x, \text{realty}) \geq 0.73 \rightarrow x.\text{label} = \text{realty}$	Confirms property-market cues with moderate model support.
5	$D(x) \wedge T(\text{LF}_{\text{furnishing}}, x, \text{home}) \wedge \neg T(\text{LF}_{\text{property}}, x, \text{realty}) \rightarrow x.\text{label} = \text{home}$	Separates home-design articles from real-estate reporting.
6	$D(x) \wedge T(\text{LF}_{\text{school}}, x, \text{education}) \wedge \Pr(x, \text{education}) > 0.81 \rightarrow x.\text{label} = \text{education}$	Requires school-related cues and strong education confidence.
7	$D(x) \wedge T(\text{LF}_{\text{style}}, x, \text{fashion}) \wedge \Pr(x, \text{fashion}) \geq 0.68 \rightarrow x.\text{label} = \text{fashion}$	Combines clothing and style cues with model support.
8	$D(x) \wedge T(\text{LF}_{\text{public}}, x, \text{society}) \wedge \neg T(\text{LF}_{\text{government}}, x, \text{politics}) \rightarrow x.\text{label} = \text{society}$	Treats public-interest reporting without governmental cues as society news.
9	$D(x) \wedge T(\text{LF}_{\text{government}}, x, \text{politics}) \wedge \Pr(x, \text{politics}) > 0.77 \rightarrow x.\text{label} = \text{politics}$	Confirms government and policy cues as politics.
10	$D(x) \wedge T(\text{LF}_{\text{celebrity}}, x, \text{entertainment}) \wedge \neg T(\text{LF}_{\text{match}}, x, \text{sports}) \rightarrow x.\text{label} = \text{entertainment}$	Distinguishes celebrity coverage from athlete or match reporting.
11	$D(x) \wedge T(\text{LF}_{\text{gameplay}}, x, \text{game}) \wedge \Pr(x, \text{game}) \geq 0.71 \rightarrow x.\text{label} = \text{game}$	Confirms gameplay and game-title cues as game news.
12	$D(x) \wedge T(\text{LF}_{\text{macro}}, x, \text{finance}) \wedge \Pr(x, \text{finance}) > 0.79 \rightarrow x.\text{label} = \text{finance}$	Confirms macroeconomic and banking cues as finance.
13	$D(x) \wedge T(\text{LF}_{\text{finance}}, x, \text{finance}) \wedge T(\text{LF}_{\text{athletes}}, x, \text{sports}) \rightarrow x.\text{label} = \text{sports}$	Rule φ_1 in Example 4; specific sports cues override incidental finance expressions.
14	$D(x) \wedge T(\text{LF}_{\text{scitech}}, x, \text{tech}) \wedge \Pr(x, \text{tech}) < 0.1 \rightarrow x.\text{label} = \perp$	Rule φ_3 in Example 4; it filters a technology keyword label contradicted by the fusion model.
15	$D(x) \wedge T(\text{LF}_{\text{gossip}}, x, \text{entertainment}) \wedge \Pr(x, \text{entertainment}) \leq 0.16 \rightarrow x.\text{label} = \perp$	Filters a gossip cue when entertainment confidence remains very low.
16	$D(x) \wedge T(\text{LF}_{\text{headline}}, x, \text{stocks}) \wedge \Pr(x, \text{stocks}) < 0.24 \rightarrow x.\text{label} = \perp$	Filters an ambiguous stock headline unsupported by the fusion model.
17	$D(x) \wedge D(x') \wedge T(\text{LF}_{\text{stock}}, x', \text{stocks}) \wedge M_{\text{topic}}(x, x') \rightarrow x.\text{label} = \text{stocks}$	Rule φ_2 in Example 4; it propagates a stock label between same-topic articles.
18	$D(x) \wedge D(x') \wedge T(\text{LF}_{\text{teacher}}, x', \text{education}) \wedge M_{\text{topic}}(x, x') \rightarrow x.\text{label} = \text{education}$	Transfers education evidence from a same-topic article.
19	$D(x) \wedge D(x') \wedge T(\text{LF}_{\text{tournament}}, x', \text{sports}) \wedge M_{\text{event}}(x, x') \rightarrow x.\text{label} = \text{sports}$	Propagates tournament evidence across reports of the same event.
20	$D(x) \wedge D(x') \wedge T(\text{LF}_{\text{development}}, x', \text{realty}) \wedge M_{\text{event}}(x, x') \rightarrow x.\text{label} = \text{realty}$	Transfers realty evidence between articles about the same development.
21	$D(x) \wedge D(x') \wedge \Pr(x', \text{politics}) \geq 0.88 \wedge M_{\text{event}}(x, x') \rightarrow x.\text{label} = \text{politics}$	Uses a high-confidence report about the same political event as evidence.

$$\varphi^*(x) = \arg \max_{\varphi \in \Sigma(x)} \text{score}(\varphi, x)$$

and use the consequence of $\varphi^*(x)$ when it is sufficiently better supported than the alternatives. Let φ_1 and φ_2 be the top two rules under $\text{score}(\cdot, x)$. If $\text{score}(\varphi_1, x) - \text{score}(\varphi_2, x) < \tau$ for a small margin τ , or if the winning rule still has weak evidence (e.g., low $\text{conf}(\varphi_1)$ or low $\Pr(x, w_{\varphi_1})$), we abstain and assign $\perp$. This filtering step avoids forcing a brittle decision when the evidence is ambiguous. Operationally, it turns unresolved conflicts into abstentions, which is preferable to propagating a potentially incorrect weak label.

Why not fixed priority. Fixed-priority resolution can over-trust a broad or stale rule even when sample-specific evidence points elsewhere. Our ranking strategy combines rule-level reliability, rule specificity, and sample-level soft-label evidence. Thus it is more robust to noisy rules and less likely to propagate uncertain labels.

Table 3. Complete set of 16 LRRs discovered on DBPedia.

#	LRR	Explanation
1	$D(x) \wedge \Pr(x, w) \geq 0.9 \rightarrow x.\text{label} = w$	Accept any entity type when the fused posterior is near-certain.
2	$D(x) \wedge T(\text{LF}_{\text{species}}, x, \text{Animal}) \rightarrow x.\text{label} = \text{Animal}$	Trusts distinctive zoological taxonomy and species cues.
3	$D(x) \wedge T(\text{LF}_{\text{botanical}}, x, \text{Plant}) \rightarrow x.\text{label} = \text{Plant}$	Trusts botanical taxonomy and flora cues.
4	$D(x) \wedge T(\text{LF}_{\text{campus}}, x, \text{EducationalInstitution}) \rightarrow x.\text{label} = \text{EducationalInstitution}$	Uses explicit university, school, and campus cues.
5	$D(x) \wedge T(\text{LF}_{\text{vehicle}}, x, \text{MeanOfTransportation}) \rightarrow x.\text{label} = \text{MeanOfTransportation}$	Uses vehicle specifications and route terminology.
6	$D(x) \wedge T(\text{LF}_{\text{settlement}}, x, \text{Village}) \rightarrow x.\text{label} = \text{Village}$	Uses settlement and municipality expressions as village evidence.
7	$D(x) \wedge T(\text{LF}_{\text{founder}}, x, \text{Company}) \wedge \neg T(\text{LF}_{\text{campus}}, x, \text{EducationalInstitution}) \rightarrow x.\text{label} = \text{Company}$	Treats founder cues as corporate evidence when campus cues are absent.
8	$D(x) \wedge T(\text{LF}_{\text{track}}, x, \text{Album}) \wedge \Pr(x, \text{Album}) \geq 0.68 \rightarrow x.\text{label} = \text{Album}$	Confirms a track-based label with moderately strong album confidence.
9	$D(x) \wedge T(\text{LF}_{\text{occupation}}, x, \text{Artist}) \wedge \Pr(x, \text{Artist}) \leq 0.17 \rightarrow x.\text{label} = \perp$	Filters a creative-occupation cue when the artist posterior is negligible.
10	$D(x) \wedge T(\text{LF}_{\text{office}}, x, \text{OfficeHolder}) \wedge \Pr(x, \text{OfficeHolder}) > 0.74 \rightarrow x.\text{label} = \text{OfficeHolder}$	Confirms government-office cues with strong calibrated support.
11	$D(x) \wedge D(x') \wedge \Pr(x', \text{Athlete}) > 0.82 \wedge M_{\text{template}}(x, x') \rightarrow x.\text{label} = \text{Athlete}$	Transfers athlete evidence between entries with the same infobox template.
12	$D(x) \wedge D(x') \wedge T(\text{LF}_{\text{geography}}, x', \text{NaturalPlace}) \wedge M_{\text{category}}(x, x') \rightarrow x.\text{label} = \text{NaturalPlace}$	Propagates geographic evidence between entries sharing an ontology category.
13	$D(x) \wedge T(\text{LF}_{\text{sponsor}}, x, \text{Company}) \wedge T(\text{LF}_{\text{capacity}}, x, \text{Building}) \wedge T(\text{LF}_{\text{location}}, x, \text{Building}) \rightarrow x.\text{label} = \text{Building}$	Capacity and location cues override a sponsor-based company cue.
14	$D(x) \wedge D(x') \wedge T(\text{LF}_{\text{director}}, x', \text{Film}) \wedge M_{\text{series}}(x, x') \wedge \Pr(x, \text{Film}) \geq 0.61 \rightarrow x.\text{label} = \text{Film}$	Combines a related entry's director cue with moderate film confidence.
15	$D(x) \wedge D(x') \wedge T(\text{LF}_{\text{author}}, x', \text{WrittenWork}) \wedge M_{\text{series}}(x, x') \wedge \neg T(\text{LF}_{\text{track}}, x, \text{Album}) \rightarrow x.\text{label} = \text{WrittenWork}$	Propagates written-work evidence when the target lacks an album cue.
16	$D(x) \wedge T(\text{LF}_{\text{genre}}, x, \text{Film}) \wedge T(\text{LF}_{\text{track}}, x, \text{Album}) \wedge T(\text{LF}_{\text{author}}, x, \text{WrittenWork}) \wedge \Pr(x, \text{Film}) < 0.4 \wedge \Pr(x, \text{Album}) < 0.4 \rightarrow x.\text{label} = \perp$	Filters irreconcilable creative-work cues when neither leading type is credible.

K Discovered LRRs in Experiments

In our experiments, SALT mines 21 and 16 LRRs on THUCNews and DBPedia, respectively. Example 4 explains three representative THUCNews rules. Here we list complete discovered rule sets, and briefly summarize their predicate-category distributions.

THUCNews is a news classification dataset with 14 classes: finance, lottery, realty, stocks, home, education, tech, society, fashion, politics, sports, constellation, game, and entertainment.

DBPedia is a text classification dataset with 14 classes from DBPedia 2014: Company, EducationalInstitution, Artist, Athlete, MeanOfTransportation, Building, OfficeHolder, NaturalPlace, Village, Animal, Plant, Album, Film, and WrittenWork.

The complete sets of the 21 and 16 LRRs discovered on THUCNews and DBPedia are listed in Tables 2 and 3, respectively. An LF subscript names the cue used by that keyword or regular-expression labeling function; M_{topic} and M_{event} catch same-topic and same-event relations between news articles, respectively, while M_{template} , M_{category} , and M_{series} denote shared infobox-template, ontology-category, and series relations between DBPedia entries, respectively.

Analysis. The rule sets exhibit the following structural patterns.

- *Predicate categories.* Among the 21 THUCNews rules, 19 use tool predicates, 11 use soft-label predicates, five use relationship predicates, and three use negated tool predicates. Among the 16 DBPedia rules, the corresponding counts are 14, seven, four, and two. These counts are non-exclusive because one LRR may combine multiple predicate categories.

- *Filtering rules.* Three THUCNews rules and two DBPedia rules have $\perp$ as their consequence. These rules discard weak labels whose cues conflict with low fusion-model confidence.
- *Precondition size.* The THUCNews set contains three rules with two predicates, 13 with three predicates, and five with four predicates. The DBPedia set contains six rules with two predicates, four with three, three with four, two with five, and one with six. Thus, 16 of 21 and 10 of 16 rules contain at most three predicates.
- *Most common forms.* Short confirmation and confidence-gating rules form the majority: they either confirm a distinctive tool cue or require soft-label confidence to support or reject that cue. Longer conflict-resolution and cross-sample propagation rules form a smaller minority.
- *Relationship evidence.* Five THUCNews rules and four DBPedia rules use relationship predicates to transfer evidence between related samples. They cover same-topic and same-event news relations, as well as shared template, category, and series relations between DBPedia entries.
- *Levelwise nonredundancy.* The levelwise search canonicalizes candidate conjunctions and prunes redundant candidates. Consequently, a direct comparison of the retained rules finds zero pairs with duplicate or strictly subsumed preconditions.

Received 17 April 2026; revised 19 August 2026; accepted 12 September 2026