ANCHOR-BASED CONFORMAL PREDICTION UNDER NOISY ANNOTATIONS IN SINGLE-CELL DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning predictive models from noisy annotations is a challenge in modern machine learning, particularly in domains where labels are obtained from multiple imperfect annotators. In this work, we introduce an anchor-based conformal prediction framework that provides rigorous uncertainty guarantees even in the presence of label noise. Our method identifies pseudo-anchors by selecting samples with strong agreement across annotators, uses these anchors to train a base predictor, and calibrates top-k conformal sets to ensure valid coverage. This construction produces prediction sets that are both reliable and compact, while explicitly accounting for annotation disagreement. Our results show that anchor-guided conformal prediction attains coverage close to nominal targets while producing smaller prediction sets and maintaining robustness in the presence of noisy labels. Although evaluated on single-cell data, the framework more generally offers a principled way to integrate multiple noisy annotator signals with conformal prediction, enabling reliable uncertainty estimates under imperfect supervision. This enables reliable uncertainty estimates in settings where ground-truth labels are scarce, expensive to obtain, or inherently ambiguous, and highlights how conformal methods can be applied to more realistic and noisy supervision scenarios.

1 INTRODUCTION

Learning under noisy supervision has been extensively studied along several complementary line. Classical strategies include noise-robust losses and label-correction techniques (Natarajan et al., 2013; Patrini et al., 2017b), as well as importance reweighting to correct risk under class-conditional noise (Liu & Tao, 2015). Robust training schemes further mitigate overfitting to corrupted labels via noise-tolerant objectives, bootstrapping, sample selection, and curriculum learning (Reed et al., 2014; Ghosh et al., 2017; Zhang & Sabuncu, 2018; Jiang et al., 2018; Han et al., 2018). Other approaches detect and relabel suspected errors using confident estimates or mixture modeling (Northcutt et al., 2021; Li et al., 2020), and anchor-based methods leverage (approximate) class-pure examples to identify noise transitions, with recent work exploring weaker conditions (Xia et al., 2019). Broad surveys synthesize these strands and practical considerations for modern deep learning (Han et al., 2020). A complementary direction explicitly models annotators, treating labels as noisy signals from multiple sources and inferring latent ground truth probabilistically (Dawid & Skene, 1979; Raykar et al., 2010; Rodrigues & Pereira, 2018). While effective in crowd-sourced or multi-annotator settings, these methods often assume parametric error forms and may not capture structured biological ambiguity (e.g., overlapping cell states) in single-cell data.

Conformal prediction provides an orthogonal perspective, focusing on uncertainty quantification with minimal assumptions (exchangeability of data) yet guaranteeing finite-sample validity (Vovk et al., 2005; Shafer & Vovk, 2008). Recent advances have extended conformal prediction to modern learning settings, including split conformal prediction (Lei et al., 2018), conformalized quantile regression (Romano et al., 2019), distributional conformal prediction (Chernozhukov et al., 2021), deep classifiers (Romano et al., 2020), covariate shift (Tibshirani et al., 2019), and multi-label outputs (Angelopoulos et al., 2022). However, conformal prediction under noisy annotations has received limited attention, and methods that integrate annotator disagreement into the calibration process remain scarce.

1.1 Motivation and Related Work

Our motivating application arises in single-cell transcriptomics, where each cell is represented by a high-dimensional vector of gene expression measurements, with tens of thousands of genes measured across hundreds of thousands of cells. A central task is to assign each cell to a cell type. In practice, however, the "true" cell type is typically unknown. Researchers therefore run clustering algorithms and use their outputs as proxy labels. Different methods often disagree on both the number of clusters and the cell assignments, highlighting inherent uncertainty in the annotation process. Treating any one clustering as ground truth risks propagating errors throughout downstream analyses and undermining reproducibility.

We highlight representative approaches to cell-type classification rather than an exhaustive review. Early tools rely on unsupervised clustering (e.g., k-means, Louvain, hierarchical) followed by marker-based annotation (Kiselev et al., 2019; Abdelaal et al., 2019). Supervised frameworks such as SingleCellNet (Tan & Cahan, 2019) and ACTINN (Ma & Pellegrini, 2019) train neural networks to predict cell types across datasets. More recently, deep learning has been leveraged to capture complex gene–cell dependencies, including convolutional and recurrent networks (Jia & Benson, 2020) and graph neural networks (GNNs) that model cell–cell similarity graphs (Brendel et al., 2022). A recent contribution, scCopulaGNN, combines copula theory with GNNs to model non-linear dependencies and cell–cell relationships in scRNA-seq data, achieving competitive performance on several benchmark datasets (Min et al., 2024). While these methods deliver strong predictive performance, they typically assume access to high-quality reference annotations, and more importantly, they do not directly address either annotation noise or the need for calibrated uncertainty.

Our work builds on these threads by addressing the dual challenges of label noise and predictive uncertainty in single-cell classification. Unlike existing single-cell classifiers, we do not treat annotator disagreement as noise to be eliminated; instead, we leverage regions of annotator agreement to identify anchor samples, and integrate these anchors into a conformal prediction framework that provides rigorous, distribution-free uncertainty guarantees. This bridges the gap between annotation-robust learning with uncertainty-aware prediction, enabling reliable cell-type assignment in the presence of ambiguous or noisy supervision.

1.2 Our Contributions

A central ingredient of our approach is the use of anchor points and their variants – pseudo–anchor points: instances that can be assigned to a class with near certainty (exactly or approximately). Leveraging this property, we learn the latent class distribution and estimate *instance-dependent* noise transition models, thereby bypassing the need to observe true labels for general instances. We combine anchors/pseudo-anchors with deep neural networks that jointly model annotator skill and class structure, yielding a flexible representation of the annotation process.

Point predictions alone are insufficient for deployment; predictions must include quantified uncertainty to reflect randomness from data collection and learning. We therefore integrate conformal prediction to produce prediction sets with rigorous, distribution-free coverage guarantees. Rather than outputting a single predicted label, the model returns a set of plausible labels that contains the truth with high probability. These sets are calibrated on held-out data and adapt to annotator noise, providing valid and efficient uncertainty quantification. This makes predictions not only accurate but also trustworthy—especially in biomedical and scientific applications. We make the following notable contributions:

- We extend the conformal prediction framework to data with noisy labels and formalize true-label prediction with multiple noisy annotators. We introduce an anchor-based framework that couples deep neural models of annotator skill and class structure with data-driven anchor (and pseudo-anchor) identification, capturing complex annotation processes common in biomedicine and beyond.
- We integrate conformal prediction to deliver distribution-free, calibrated prediction sets. The resulting method is robust to annotation noise, flexible in modeling annotator behavior, and principled in its treatment of uncertainty. Although we focus on single-cell classification, the framework applies broadly to medical imaging, crowdsourcing, and natural language processing, where noisy labels are the norm.

- We establish theoretical guarantees and validate the method on two single-cell RNA-seq datasets, showing strong performance.

2 Problem Setup and Methods

Let $\mathrm{x}_i \in \mathbb{R}^p$ denote the feature vector for cell i (e.g., gene expression), associated with an unobserved true class $y_i \in [K]$ (e.g., cell type), where $[K] = \{1, \ldots, K\}$. For each cell we observe a vector of noisy labels $\widetilde{\mathrm{y}}_i = (\widetilde{\mathrm{y}}_i^{(1)}, \ldots, \widetilde{\mathrm{y}}_i^{(R_i)})$ from R_i annotators; for ease of exposition we assume $R_i \equiv R$ (which matches our application datasets). We use uppercase letters X_i, Y_i and $\widetilde{Y}$ (with the subscript i sometimes omitted) for the corresponding random variables. We propose a method for predicting latent true labels from multiple noisy annotators with distribution-free uncertainty guarantees, using class-specific anchor points: an instance x is an anchor for class k if $\mathbb{P}(Y = k \mid X = \mathrm{x}) = 1$.

Warm-up and pipeline. In a warm-up stage, we train base models on noisy data (Liu & Tao, 2015) and identify class-specific anchor sets $D_{0,k}$; let $D_0 = \bigcup_{k\in[K]} D_{0,k}$. For model development, each $D_{0,k}$ is split into index sets $\mathcal{A}_k^t$ (training) and $\mathcal{A}_k^c$ (calibration/hold-out), and we define

$$D_0^t = \bigcup_{k\in[K]} \{(\mathrm{x}_i, \widetilde{\mathrm{y}}_i, y_i{=}k) : i \in \mathcal{A}_k^t\}, \qquad D_0^c = \bigcup_{k\in[K]} \{(\mathrm{x}_i, \widetilde{\mathrm{y}}_i, y_i{=}k) : i \in \mathcal{A}_k^c\}.$$

We then (i) learn an annotator-dependent transition model parameterized by two deep networks, (ii) form likelihood-based class scores for point prediction, and (iii) calibrate top-k prediction sets on held-out anchors. Our full training and inference pipeline is summarized in Algorithm 1.

Annotator transition model. We model annotator behavior and class dependence via two feature maps, $\psi^{\mathrm{A}}(\mathrm{x})$ and $\psi^{\mathrm{C}}(\mathrm{x})$, implemented as feedforward neural networks. For annotator $r \in [R]$ and class $j \in [K]$, the probability of reporting class j when the true class is k is modeled by the softmax transition:

$$\mathbb{P}\Big(\widetilde{\mathrm{Y}}^{(r)} = j \mid \mathrm{Y} = k, \mathrm{X} = \mathrm{x}\Big) = \frac{\exp\{\langle \alpha_j^{(r)} \psi^{\mathrm{A}}(\mathrm{x})\rangle + \langle \beta_j^{(k)} \psi^{\mathrm{C}}(\mathrm{x})\rangle\}}{\sum_{\ell=1}^{K} \exp\big\{\langle \alpha_\ell^{(r)} \psi^{\mathrm{A}}(\mathrm{x})\rangle + \langle \beta_\ell^{(k)},\, \psi^{\mathrm{C}}(\mathrm{x})\rangle\big\}}, \tag{1}$$

where $\alpha_j^{(r)}$ captures annotator–specific effects and $\beta_j^{(k)}$ captures class–specific structure. This parameterization disentangles annotator-specific effects (via $\alpha^{(r)} \equiv \{\alpha_j^{(r)}\}_{j=1}^R$ and ψ^{A}) from class-specific structure (via $\beta^{(k)} \equiv \{\beta_j^{(k)}\}_{j=1}^K$ and ψ^{C}).

Anchors-based likelihood training. Given D_0^t, estimate the model parameters $\theta = (\theta_{\mathrm{A}}, \theta_{\mathrm{C}}, \{\alpha^{(r)}\}_{r=1}^R, \{\beta^{(k)}\}_{k=1}^K)$ by maximizing the log-likelihood of observed annotator labels under 1:

$$\widehat{\theta} = \arg\max_{\theta} \sum_{k\in[K]} \sum_{i\in\mathcal{A}_k^t} \sum_{r=1}^{R} \sum_{j=1}^{K} \mathbf{1}\{\widetilde{\mathrm{y}}_i^{(r)} = j\} \log P_\theta\big(\widetilde{\mathrm{Y}}^{(r)} = j \,\big|\, \mathrm{Y} = k, \mathrm{X} = \mathrm{x}_i\big), \tag{2}$$

where $P_\theta\big(\widetilde{\mathrm{Y}}^{(r)} = j \,\big|\, \mathrm{Y} = k, \mathrm{X} = \mathrm{x}_i\big)$ denotes the probability (1) parameterized by θ. For inference, we define the annotator-conditional term and the joint class score. Using the fitted parameters $\widehat{\theta}$, set

$$\tau_{kj}^{(r)}(\mathrm{x}) \;\equiv\; P_{\widehat{\theta}}\big(\widetilde{\mathrm{Y}}^{(r)} = j \,\big|\, \mathrm{Y} = k, \mathrm{X} = \mathrm{x}\big), \qquad r \in [R],\ j \in [K], \tag{3}$$

and for an observed pair $(\mathrm{x}_i, \widetilde{\mathrm{y}}_i)$ define

$$\tau_k(\mathrm{x}_i, \widetilde{\mathrm{y}}_i) \;\equiv\; \prod_{r=1}^{R} \prod_{j=1}^{K} \Big\{\tau_{kj}^{(r)}(\mathrm{x}_i)\Big\}^{\mathbf{1}\{\widetilde{\mathrm{y}}_i^{(r)} = j\}}. \tag{4}$$

Equivalently, viewing the score as a plug-in likelihood under $\widehat{\theta}$ yields

$$\widehat{\tau}_k(\mathrm{x}, \widetilde{\mathrm{y}}) \;\equiv\; \prod_{r=1}^{R} \prod_{j=1}^{K} \Big\{P_{\widehat{\theta}}\big(\widetilde{\mathrm{Y}}^{(r)} = j \,\big|\, \mathrm{Y} = k, \mathrm{X} = \mathrm{x}\big)\Big\}^{\mathbf{1}\{\widetilde{\mathrm{y}}^{(r)} = j\}}. \tag{5}$$

The product form in 4 assumes conditional independence of annotators given (Y, X): $\mathbb{P}(\tilde{\mathrm{y}}^{(1)}, \ldots, \tilde{\mathrm{y}}^{(R)} \mid Y = k, X = \mathrm{x}) = \prod_{r=1}^{R} \mathbb{P}(\tilde{\mathrm{y}}^{(r)} \mid Y = k, X = \mathrm{x})$. The point predictor takes the maximum-score class:

$$\widehat{h}(\mathrm{x}, \widetilde{\mathrm{y}}) \;=\; \arg\max_{k \in [K]} \; \widehat{\tau}_k(\mathrm{x}, \widetilde{\mathrm{y}}).$$

Conformal Prediction. To quantify uncertainty, we calibrate top-k prediction sets on the held-out anchors D_0^c. For each $(\mathrm{x}_i, \widetilde{\mathrm{y}}_i, y_i) \in D_0^c$, compute the class scores $\{\widehat{\tau}_k(\mathrm{x}_i, \widetilde{\mathrm{y}}_i)\}_{k \in [K]}$, sort them in decreasing order to obtain $\hat{y}^{(1)}, \ldots, \hat{y}^{(K)}$, and form the nested sets $C(\mathrm{x}_i, \widetilde{\mathrm{y}}_i; k) = \{\hat{y}^{(1)}, \ldots, \hat{y}^{(k)}\}$. Let $k_i = \min\{k : y_i \in C(\mathrm{x}_i, \tilde{\mathrm{y}}_i; k)\}$ be the smallest set size that captures the true label. Given a target miscoverage $\alpha \in (0, 1)$, choose $\widehat{k}_c(\alpha)$ as the $\lceil |D_0^c| + 1 \rceil (1 - \alpha)$ quantile of $\{k_i : i \in \bigcup_{k \in [K]} \mathcal{A}_k^c\}$. At deployment, the calibrated set predictor returns $\mathcal{C}_\alpha(\mathrm{x}, \widetilde{\mathrm{y}}) = C(\mathrm{x}, \widetilde{\mathrm{y}}; \widehat{k}_c(\alpha))$, which achieves marginal coverage at least $1 - \alpha$ under exchangeability, while keeping sets as small as possible. The procedure for constructing top-k conformal sets is given in Algorithm 2.

Assumptions and relaxations. Our procedure is developed under a conditional-independence assumption: given the features, annotators label an instance independently. Formally, for any $x \in \mathcal{X}$, $\mathbb{P}(\widetilde{Y}^{(1)}, \ldots, \widetilde{Y}^{(R)} \mid X = x) = \prod_{r=1}^{R} \mathbb{P}(\widetilde{Y}^{(r)} \mid X = x)$, which reduces learning the label-noise process to estimating the annotator-specific transition model (1). When this assumption is substantially violated (e.g., correlated annotators), one can replace the product form with a joint model $\mathbb{P}(\widetilde{Y}^{(1)}, \ldots, \widetilde{Y}^{(R)} \mid X = x)$ or incorporate a shared latent factor to capture dependence across annotators. Within the pool of anchor points D_0, the paired variables $\{X_i, \widetilde{Y}_i\}$ are assumed independent across i; they need not be identically distributed, though our coverage statements apply to the anchor subset that is exchangeable with the corresponding test points. When the size of D_0 is small in applications, we enlarge it with pseudo-anchors: an instance x is a δ-pseudo-anchor for class k if $\mathbb{P}(Y = k \mid X = \mathrm{x}) \geq 1 - \delta$ for $0 \leq \delta < 1$, with $\delta = 0$ recovering an anchor; see Appendix B for details.

Algorithm 1: Anchor–Guided Training and Point Prediction

Input: Anchors $D_0 = \bigcup_{k \in [K]} D_{0,k}$ with index splits $\mathcal{A}_k^t, \mathcal{A}_k^c$ for each $k \in [K]$.

Output: MLE $\widehat{\theta}$ and point predictor $\widehat{h}(\mathrm{x}, \widetilde{\mathrm{y}})$.

Train.;

Maximize the objective in (2) on $D_0^t = \bigcup_{k \in [K]} \{(\mathrm{x}_i, \widetilde{\mathrm{y}}_i, y_i{=}k) : i \in \mathcal{A}_k^t\}$, using the transition form (1).

Score.;

For any $(\mathrm{x}, \widetilde{\mathrm{y}})$, compute $\tau_{kj}^{(r)}(\mathrm{x})$ as in (3) and the class score $\tau_k(\mathrm{x}, \widetilde{\mathrm{y}})$ via (4); equivalently use $\widehat{\tau}_k$ from (5).

Predict.;

Set $\widehat{h}(\mathrm{x}, \widetilde{\mathrm{y}}) = \arg\max_{k \in [K]} \widehat{\tau}_k(\mathrm{x}, \widetilde{\mathrm{y}})$.

Algorithm 2: Top-k Conformal Set Prediction

Input: Calibration anchors $D_0^c = \bigcup_{k \in [K]} \{(\mathrm{x}_i, \widetilde{\mathrm{y}}_i, y_i{=}k) : i \in \mathcal{A}_k^c\}$; target miscoverage $\alpha \in (0, 1)$.

Output: Set predictor $(\mathrm{x}, \widetilde{\mathrm{y}}) \mapsto \mathcal{C}_\alpha(\mathrm{x}, \widetilde{\mathrm{y}})$.

For each $(\mathrm{x}_i, \widetilde{\mathrm{y}}_i, y_i) \in D_0^c$, compute $\{\widehat{\tau}_k(\mathrm{x}_i, \widetilde{\mathrm{y}}_i)\}_{k \in [K]}$ (cf. (5)), order classes to form $C(\mathrm{x}_i, \widetilde{\mathrm{y}}_i; k)$, and record the minimal k_i with $y_i \in C(\mathrm{x}_i, \widetilde{\mathrm{y}}_i; k_i)$.

Choose $\widehat{k}_c(\alpha)$ by the $\lceil |D_0^c| {+} 1 \rceil (1 - \alpha)$ quantile rule described in Section 2.

At test time, output $\mathcal{C}_\alpha(\mathrm{x}, \widetilde{\mathrm{y}}) = C(\mathrm{x}, \widetilde{\mathrm{y}}; \widehat{k}_c(\alpha))$.

3 THEORETICAL GUARANTEE

We begin by formalizing when anchor points exist and how they can be identified from observable quantities. Theorem 1 provides a necessary-and-sufficient characterization under a mild separability assumption on annotators.

Theorem 1. *Assume that for any $r \in [R]$, $k \in [K]$, and $\mathrm{x} \in \mathcal{X}$,*

$$\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = k|\mathrm{Y} = k, \mathrm{X} = \mathrm{x}) > \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = k|\mathrm{Y} = j, \mathrm{X} = \mathrm{x}) \quad \textit{for all} \quad j \neq k. \tag{6}$$

Then for an $\mathrm{x} \in \mathcal{X}$,

$$\mathbb{P}(\mathrm{Y} = k|\mathrm{X} = \mathrm{x}) = 1 \quad \textit{iff} \quad \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = k|\mathrm{X} = \mathrm{x}) = \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = k|\mathrm{Y} = k, \mathrm{X} = \mathrm{x}),$$

where "iff" means "if and only if".

Assumption (6) states that, conditional on any input x, annotator r is more likely to assign label k when the true class is k than when it is any other class, i.e., the annotator is better than chance for class k at x (non-degenerate). This is reasonable in settings where annotators are not incompetent (see Appendix B). The equivalence in Theorem 1 has two key implications: (i) anchors (and practical pseudo-anchors, as discussed in Appendix B) can be discovered directly from data without observing true labels, and (ii) at anchors, annotator-specific noise transitions are identified, enabling consistent estimation of instance-dependent transition models.

Theorem 2. *Assume the condition in Theorem 1, and further assume that*

$$\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = k|\mathrm{Y} = k, \mathrm{X} = \mathrm{x}(k)) = \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = k|\mathrm{X} = \mathrm{x}(k)) \tag{7}$$

holds for

$$\mathrm{x}(k) = \mathrm{argmax}_{\mathrm{x}} \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = k|\mathrm{X} = \mathrm{x}), \tag{8}$$

where $k \in [K]$. Then for this k,

$$\mathbb{P}(\mathrm{Y} = k|\mathrm{X} = \mathrm{x}(k)) = 1.$$

That is, $\mathrm{x}(k)$ is an anchor point for class k.

This theorem follows directly from Theorem 1 and yields a practical anchor–point identification rule: apply (8) to the observed annotations and inputs (essentially a majority–voting argmax over $\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = k \mid \mathrm{X} = \mathrm{x})$). Related proposals (e.g.,Li et al. (2020) Liu & Tao (2015); Patrini et al. (2017a)) implicitly rely on consistent estimation of $\mathbb{P}(\widetilde{\mathrm{Y}} = \widetilde{\mathrm{y}} \mid \mathrm{X} = \mathrm{x})$ in the presence of anchors, but typically do not state the additional condition (7), which is needed to justify the validity of the procedure based on (8). (7) asserts that, at $\mathrm{x}(k)$, the input alone is sufficient for predicting $\widetilde{\mathrm{Y}}^{(r)} = k$ (i.e., it captures all information about the true label $\mathrm{Y} = k$ relevant to the annotator's output). This is plausible in practice—for example, a diagnostic biomarker that determines a test's positive call for disease k, or spam detection where predictions rely solely on message features. Importantly, (7) is required only at the argmax points $\mathrm{x}(k)$ selected by (8), not for all $\mathrm{x} \in \mathcal{X}$; it is thus weaker than the global *nondifferential misclassification* condition

$$\mathbb{P}\Big(\widetilde{\mathrm{Y}}^{(r)} = j \mid \mathrm{Y} = k,\, \mathrm{X} = \mathrm{x}\Big) = \mathbb{P}\Big(\widetilde{\mathrm{Y}}^{(r)} = j \mid \mathrm{X} = \mathrm{x}\Big) \quad \text{for all } j, k \in [K], \tag{9}$$

which states conditional independence between $\widetilde{\mathrm{Y}}$ and Y, given X.

Building on the foundation set by Theorems 1 and 2, our anchor-based likelihood estimators are consistent under standard regularity using the likelihood theory, and consequently, we can establish distribution-free validity of our conformal top-k prediction sets calibrated on (pseudo-)anchors, including monotonicity in α and finite-sample marginal coverage at level $1 - \alpha$, as stated in the following theorem.

Theorem 3. *Suppose we have a future input $\mathrm{X} = \mathrm{x}$ with crowdsourced label $\widetilde{\mathrm{y}}$. Then*

(a) $\mathcal{C}(\mathrm{x}_i, \widetilde{\mathrm{y}}_i; k_1) \subset \mathcal{C}(\mathrm{x}_i, \widetilde{\mathrm{y}}_i; k_2)$ for $k_1 \leq k_2$;

(b) For any $\alpha \in (0, 1)$, $P\{\mathrm{Y} \in \mathcal{C}(\mathrm{x}, \widetilde{\mathrm{y}}; k^c(\alpha)\} \geq 1 - \alpha$.

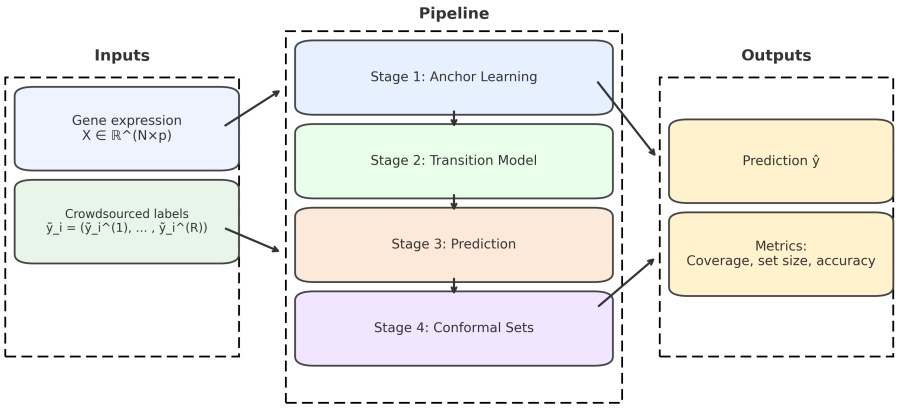


Figure 1: Overview of the proposed anchor-based conformal prediction pipeline. The framework identifies anchors from annotator agreement, trains a base model on these anchors, and calibrates conformal prediction sets to handle label noise.

4 Implementation Procedures

We develop a procedure for predicting latent true labels from multiple noisy annotators while providing distribution-free uncertainty guarantees. The approach proceeds by constructing per-class anchor sets from high-agreement subsets of annotations, learning an annotator-dependent transition model parameterized by two deep networks, forming likelihood-based class scores for point prediction, and calibrating top-k prediction sets on held-out anchors. As illustrated in Figure 1, our method consists of four main stages: anchor selection, base predictor training, anchor-guided calibration, and conformal prediction set generation.

4.1 Data and Preprocessing

We evaluate on two single-cell RNA-seq datasets: *Baron3* (Baron et al., 2016) and *PBMC2* (Stuart et al., 2020). The *Baron3* data were collected from pancreatic islets of a 38-year-old male (BMI $= 27.5$; non-diabetic). The *PBMC2* data comprise peripheral blood mononuclear cells from a healthy donor. Both datasets contain substantial cellular and genetic information and are labeled for classification. In terms of gene counts, they are comparable, with counts ranging from 20,000 to 24,000, indicating consistency in gene capture across studies. From each raw expression matrix we selected 1,200 highly variable genes (HVGs). Counts were transformed using the centered log-ratio (CLR) to mitigate compositional effects and used to construct a k-nearest neighbor (kNN) graph. Across experiments, we used stratified splits, Adam optimization with early stopping on validation loss, and cross-entropy as the primary objective.

4.2 Base Predictors

We evaluate our anchor-based conformal prediction framework with several choices of the base predictor f for cell-type classification. Within the family of graph neural networks (GNNs), the Graph Convolutional Network (GCN) updates node representations by aggregating neighborhood features through learned filters (Gao et al., 2023). The Graph Attention Network (GAT) extends this by assigning attention weights, allowing the predictor to emphasize more informative neighbors (Liu & Zhou, 2020). GraphSAGE (Graph Sample and Aggregate) provides an inductive variant that samples neighborhoods and aggregates features to construct low-dimensional node embeddings suitable for large graphs (Hamilton et al., 2018). As a non-graph baseline, we also consider a Multi-Layer Perceptron (MLP), a standard feed-forward predictor with fully connected layers (Gharehbaghi, 2023).

Table 1: Anchor counts and proportions (Prop) per cell type for both datasets.

(a) Dataset 1: *Baron3*

Cell type	Total	Anchors	Prop
t_cell	81	6	0.0741
macrophage	117	7	0.0598
epsilon	132	7	0.0530
mast	78	4	0.0513
schwann	86	4	0.0465
quiescent_stellate	96	4	0.0417
name	107	4	0.0374
gamma	133	4	0.0301
ductal	218	6	0.0275
beta	334	5	0.0150
delta	292	4	0.0137
endothelial	160	2	0.0125
activated_stellate	565	4	0.0071
alpha	457	3	0.0066
acinar	737	4	0.0054

(b) Dataset 2: *PBMC2*

Cell type	Total	Anchors	Prop
B cell	250	215	0.8600
cMono	409	349	0.8533
ncMono	119	92	0.7731
CD4 T cell	1238	872	0.7044
NK cell	270	185	0.6852
CD8 T cell	676	434	0.6420
cDC	20	12	0.6000
pDC	12	7	0.5833
Plasma cell	6	0	0.0000

5 Analysis Results

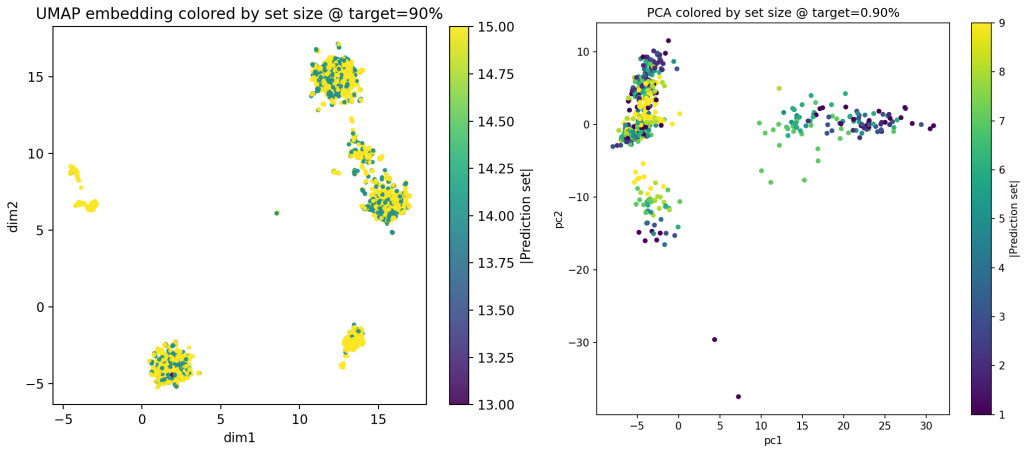
We evaluated the proposed anchor-based conformal prediction framework on two single-cell datasets with noisy annotations. Our primary focus is **Top-k** conformal prediction; the adaptive prediction sets (**APS**) method serves as a conservative comparator.

Baron3. Agreement-based anchor selection yielded 68 high-confidence cells across 15 cell types. Anchor representation was heterogeneous: relatively enriched in T cells (7.4%) and macrophages (6.0%), but sparse in acinar and α cells (both $< 1\%$), reflecting variation in annotator agreement (Table 1). Using the anchor-calibration split, **Top-k** conformal sets closely tracked nominal coverage $(1 - \alpha)$: at 80%, 85%, 90%, and 95% targets, empirical coverage was 0.80, 0.88, 0.95, and 0.95, respectively. By contrast, **APS** achieved highly conservative coverage ($\geq 96\%$ across targets) but produced much larger prediction sets. Visualization on a low-dimensional embedding further supports these patterns: under **APS**, nearly all cells attain maximal set sizes, whereas our **Top-k** set sizes vary smoothly across clusters, with smaller sets in well-separated endocrine populations and larger sets in ambiguous ductal and stellate regions (Figure 2).

PBMC2. Anchor selection again revealed substantial heterogeneity (Table 1): some immune subtypes exhibited relatively high anchor proportions, whereas others had very few, underscoring differences in annotator consistency. As in *Baron3*, **APS** achieved near-perfect coverage across all nominal levels but at the cost of inflated set sizes, often approaching the entire label space. In contrast, our **Top-k** delivered coverage much closer to the target values while maintaining smaller, more interpretable sets. Compact sets concentrated in well-defined clusters, whereas ambiguous regions yielded larger sets, as expected.

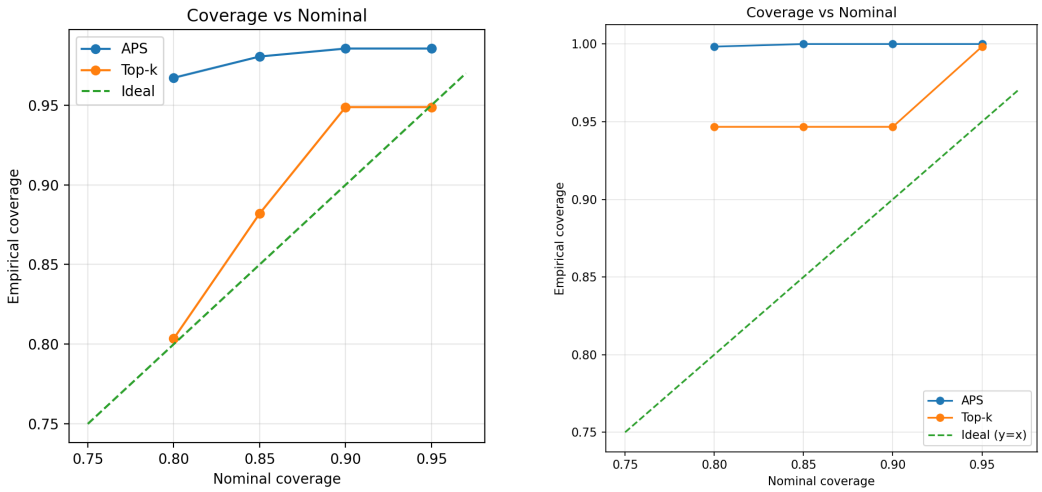
To quantify this trade-off across datasets, Table 2 reports average conformal set sizes. **APS** consistently produced very large sets (near the total number of classes), whereas **Top-k** yielded compact and interpretable sets (average sizes 12–14 in *Baron3* and substantially smaller in *PBMC2*). Together with the anchor statistics, these results indicate that anchors not only capture annotator agreement but also enable calibration procedures that produce valid, biologically meaningful, and compact conformal prediction sets. Overall, anchor-guided **Top-k** maintains reliable calibration with practical utility, while **APS** serves as a conservative upper baseline. Additional per-class results, including confusion matrices (Figure 4) and detailed classification metrics (Tables 3–4), are provided in Appendix D.

Set size analysis across datasets. Table 2 summarizes the average conformal prediction set sizes obtained from the anchor-based calibration procedure in both datasets. In *Baron3*, **APS** produced



(a) Dataset 1: *Baron3* (b) Dataset 2: *PBMC2*

Figure 2: Low-dimensional embedding (UMAP or PCA fallback) colored by conformal set size at 90% target. Regions with higher ambiguity receive larger sets; well-separated clusters receive smaller sets.



(a) Dataset 1: *Baron3* (b) Dataset 2: *PBMC2*

Figure 3: Empirical vs nominal coverage on two datasets. APS is highly conservative, while Top-k tracks nominal levels more closely.

nearly maximal set sizes (14–15 labels on average), confirming its conservative nature and limited informativeness. By contrast, in *PBMC2*, the anchor-based procedure yielded much more compact sets, often of size one across most targets, with only a modest increase at the 95% coverage level. This difference illustrates how anchor prevalence and annotator agreement directly influence calibration outcomes: when anchors are sparse (*Baron3*), conformal sets inflate toward the full label space, whereas when anchors are abundant and reliable (*PBMC2*), prediction sets remain small and interpretable. Together, these results reinforce that anchors provide a flexible mechanism for trading off between validity and efficiency across datasets of differing annotation quality. They highlight the expected trade-off: **APS** provides conservative coverage with inflated sets, while **Top-k** achieves coverage close to nominal with more compact and interpretable sets.

Table 2: Empirical coverage and average set size for APS and Top-k conformal prediction across targets on both datasets. Top-k provides compact sets while APS is more efficient than APS but still less conservative.

Method	Dataset	Target	Empirical Coverage	Avg. Set Size
APS	*Baron3*	0.80	0.967	14.49
		0.85	0.981	14.68
		0.90	0.986	14.74
		0.95	0.986	14.74
	PBMC2	0.80	0.998	4.40
		0.85	1.000	4.94
		0.90	1.000	5.45
		0.95	1.000	6.20
Top-k	*Baron3*	0.80	0.804	12.00
		0.85	0.882	13.00
		0.90	0.949	14.00
		0.95	0.949	14.00
	PBMC2	0.80	0.947	1.00
		0.85	0.947	1.00
		0.90	0.947	1.00
		0.95	0.998	2.00

Extended baseline comparisons on the *Baron3* dataset are provided in Appendix D (Table 5), showing that our anchor-based conformal prediction achieves high ROC-AUC while maintaining competitive PR-AUC against state-of-the-art classifiers.

6 Conclusion and Discussion

We introduced an anchor-based conformal prediction framework for classification with noisy annotations. By leveraging agreement-based pseudo-anchors to guide calibration, our method provides rigorous uncertainty guarantees while remaining robust to annotator disagreement. The application to two single-cell datasets confirmed that anchor-guided **Top-k** tracked nominal coverage while yielding smaller sets. Visual analyses demonstrated that prediction set sizes aligned with biological structure, expanding in ambiguous regions and contracting in well-separated clusters, thereby providing not only validity but also meaningful interpretability. These findings suggest that anchor-based conformal prediction offers a practical solution for integrating noisy labels in biomedical applications, where annotator variability is common and rigorous uncertainty quantification is essential. While demonstrated here on single-cell classification, the framework applies broadly to other domains where multiple imperfect annotations are available(e.g., medical diagnosis, crowdsourced vision/NLP). In high-stakes settings such as biomedicine, our guarantees support trustworthy deployment while explicitly accounting for annotator variability.

Promising directions include semi- and weakly supervised extensions that leverage unlabeled data, multi-modal integration, and active/online anchor discovery for dynamic annotator pools. Scaling-efficient calibration (e.g., cross-conformal and streaming variants), fairness-aware uncertainty quantification, and robustness to distribution shift and label bias are additional priorities for real-world deployment.

7 Ethics Statement

This paper introduces an anchor-based conformal prediction framework for handling noisy annotations in single-cell data. The goal of this work is to improve the reliability and robustness of predictive modeling in biomedical research, thereby advancing our understanding of cellular heterogeneity and disease mechanisms. We have carefully considered the ethical implications and do not anticipate any direct negative consequences arising from this work. While potential downstream applications may involve clinical or biomedical decision-making, this study is methodological in nature and not directly applied to patient care. We are committed to the responsible communication and use of our methods and encourage their application in ways that respect ethical standards in biomedical research and data privacy.

8 Reproducibility Statement

We have taken steps to ensure the reproducibility of our work. All datasets used in this study are clearly referenced in the paper. Descriptions of preprocessing procedures, model architectures, training protocols, and evaluation metrics are provided in the Methods section and Appendix. To further support transparency and facilitate future research, we will release an open-source implementation of our framework on GitHub upon acceptance of the paper.

References

Tamim Abdelaal, Laura Michielsen, Daniel Cats, Diana Hoogduin, Huipeng Mei, Marcel J T Reinders, and Ahmed Mahfouz. A comparative study of cell type annotation tools for single-cell rna-seq data. *Genome Biology*, 20:264, 2019.

Anastasios N Angelopoulos et al. Conformal prediction for multi-label classification. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

Maayan Baron, Adrian Veres, Samuel Wolock, Anthony Faust, Renaud Gaujoux, Alessandro Vetere, John Ryu, Bridget Wagner, Shai Shen-Orr, Allon Klein, et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Systems*, 3(4):346–360, 2016.

Maximilian Brendel, Chuan Su, Zhiguang Bai, Hongyu Zhang, Olivier Elemento, and Fei Wang. Application of deep learning on single-cell rna sequencing data analysis: A review. *Genomics, Proteomics & Bioinformatics*, 20(5):814–835, 2022.

Victor Chernozhukov, Kaspar Wüthrich, and Ying Zhu. Distributional conformal prediction. *arXiv preprint arXiv: 1909.07889*, 2021.

A Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C*, 1979.

Hong Gao, Bo Zhang, Liang Liu, Shuo Li, Xinyu Gao, and Bin Yu. A universal framework for single-cell multi-omics data integration with graph convolutional networks. *Briefings in Bioinformatics*, 24(3):bbad081, 2023. doi: 10.1093/bib/bbad081.

Amin Gharehbaghi. Multi-layer perceptron (mlp) neural networks for time series classification. In *Deep Learning in Time Series Analysis*, pp. 81–88. Springer, 2023.

Aritra Ghosh, Naresh Manwani, and P. S. Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1025–1035, 2018.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Bo Han, Quanming Yao, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. A survey of learning with noisy labels. *arXiv preprint arXiv:2012.02513*, 2020.

Junteng Jia and Austin R Benson. Residual correlation in graph neural network regression. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 588–598, 2020.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

Vladimir Y Kiselev, Tallulah S Andrews, and Martin Hemberg. Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 20(5):273–282, 2019.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113 (523):1094–1111, 2018. Split conformal prediction; earlier preprints circulated circa 2013.

Junnan Li, Richard Socher, and Steven C. H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, 2015.

Zonghan Liu and Jie Zhou. *Graph attention networks*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2020.

F. Ma and M. Pellegrini. Actinn: Automated identification of cell types in single cell rna sequencing. *Bioinformatics*, 36(2):533–538, 2019.

Shijie Min, Leann Lac, and Pingzhao Hu. A copula-infused graph neural network for cell type classification in single-cell rna sequencing data. *Bioinformatics*, 2024. URL `https://github.com/shijiemin/scCopulaGNN`. In press.

Nagarajan Natarajan et al. Learning with noisy labels. In *NeurIPS*, 2013.

Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Machine Learning Research*, 22(259):1–73, 2021.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017a.

Giorgio Patrini et al. Making deep neural networks robust to label noise: a loss correction approach. In *CVPR*, 2017b.

Vikas C Raykar et al. Learning from crowds. *JMLR*, 2010.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

Filipe Rodrigues and Francisco Pereira. Deep learning from crowds. In *AAAI*, 2018.

Yaniv Romano, Evan Patterson, and Emmanuel J. Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Yaniv Romano et al. Classification with valid and adaptive coverage. In *NeurIPS*, 2020.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 2008.

Tim Stuart, Abhishek Srivastava, Caleb Lareau, and Rahul Satija. Multimodal single-cell chromatin analysis with signac. *bioRxiv*, pp. 2020–11, 2020.

Y. Tan and P. Cahan. Singlecellnet: a computational tool to classify single cell rna-seq data across platforms and across species. *Cell Systems*, 9(2):207–213, 2019.

Ryan J Tibshirani et al. Conformal prediction under covariate shift. In *NeurIPS*, 2019.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.

Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

APPENDICES: TECHNICAL DETAILS AND EXTENDED ANALYSIS RESULTS

A PROOFS OF THEOREMS

First, we comment that in defining anchor points, it is implicitly assumed that an instance x can be an anchor point for at most one class, with the property that if $\mathbb{P}(\mathrm{Y}=k|\mathrm{X}=\mathrm{x})=1$, then $\mathbb{P}(\mathrm{Y}=j|\mathrm{X}=\mathrm{x})=0$ for any $j\neq k$. However, each class can have multiple anchor points; having $\mathbb{P}(\mathrm{Y}=k|\mathrm{X}=\mathrm{x})=1$ does not exclude $\mathbb{P}(\mathrm{Y}=k|\mathrm{X}=\mathrm{x}^*)=1$ for those instances x^* that are not identical to x.

Proof of Theorem 1: First we show the '$\Longrightarrow$" direction, which is immediate from the following derivations:

$$\begin{aligned}
& \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)}=\tilde{\mathrm{y}}^{(r)}|\mathrm{X}=\mathrm{x}) \\
= & \sum_{j\in\mathcal{Y}}\Big\{\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)}=\tilde{\mathrm{y}}^{(r)}|\mathrm{Y}=j,\mathrm{X}=\mathrm{x})\mathbb{P}(\mathrm{Y}=j|\mathrm{X}=\mathrm{x})\Big\} \\
= & \sum_{j\neq k}\Big\{\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)}=\tilde{\mathrm{y}}^{(r)}|\mathrm{Y}=j,\mathrm{X}=\mathrm{x})\mathbb{P}(\mathrm{Y}=j|\mathrm{X}=\mathrm{x})\Big\} \\
& +\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)}=\tilde{\mathrm{y}}^{(r)}|\mathrm{Y}=k,\mathrm{X}=\mathrm{x})\mathbb{P}(\mathrm{Y}=k|\mathrm{X}=\mathrm{x}) \\
= & \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)}=\tilde{\mathrm{y}}^{(r)}|\mathrm{Y}=k,\mathrm{X}=\mathrm{x}),
\end{aligned} \tag{A.1}$$

where we use the conditions for anchor points.

Next, we show the "$\Longleftarrow$" direction. Indeed, applying (A.1) to the condition

$$\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)}=k|\mathrm{X}=\mathrm{x})=\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)}=k|\mathrm{Y}=k,\mathrm{X}=\mathrm{x})$$

leads to

$$\begin{aligned}
& \sum_{j\neq k}\mathbb{P}(\tilde{\mathrm{Y}}^{(r)}=k|\mathrm{Y}=j,\mathrm{X}=\mathrm{x})\mathbb{P}(\mathrm{Y}=j|\mathrm{X}=\mathrm{x}) \\
& +\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)}=k|\mathrm{Y}=k,\mathrm{X}=\mathrm{x})\{\mathbb{P}(\mathrm{Y}=k|\mathrm{X}=\mathrm{x})-1\}=0,
\end{aligned}$$

which is equivalently written as

$$\sum_{j\neq k}\Big\{\mathbb{P}(\tilde{\mathrm{Y}}^{(r)}=k|\mathrm{Y}=j,\mathrm{X}=\mathrm{x})-\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)}=k|\mathrm{Y}=k,\mathrm{X}=\mathrm{x})\Big\}\mathbb{P}(\mathrm{Y}=j|\mathrm{X}=\mathrm{x})=0.$$

By the assumption (6) and the fact that $\mathbb{P}(\mathrm{Y}=j|\mathrm{X}=\mathrm{x})\geq 0$ for all $j\neq k$, we conclude that

$$\mathbb{P}(\mathrm{Y}=j|\mathrm{X}=\mathrm{x})=0 \quad \text{ for all } \quad j\neq k,$$

and thus yielding

$$\mathbb{P}(\mathrm{Y}=k|\mathrm{X}=\mathrm{x})=1.$$

□

To prove Theorem 3, we first show the following lemma.

Lemma 1. *Suppose $\{c_1,\ldots,c_m;c_{m+1}\}$ is a sequence of constants, taking values in $[K]$. For $k\in[K]$ and $\alpha\in(0,1)$, define*

$$\mathcal{I}_k=\{i\in[m]:c_i\leq k\},$$
$$k(\alpha)=\inf\{k\in[K]:|\mathcal{I}_k|\geq(m+1)(1-\alpha)\},$$

and

$$\mathcal{J}=\{i\in[m]:c_i<c_{m+1}\}.$$

Then "$c_{m+1}>k(\alpha)$" iff "$|\mathcal{J}|>(m+1)(1-\alpha)$".

Proof. Show "$\Longrightarrow$":

For any $i_0 \in \mathcal{I}_{k(\alpha)}$, we have that $c_{i_0} \leq k(\alpha)$. Then by the condition $c_{m+1} > k(\alpha)$, $c_{i_0} < c_{m+1}$, leading to $i_0 \in \mathcal{J}$ by definition of $\mathcal{J}$. Therefore,

$$\mathcal{I}_{k(\alpha)} \subset \mathcal{J},$$

yielding $|\mathcal{I}_{k(\alpha)}| \leq |\mathcal{J}|$. Then applying definition of $k(\alpha)$ shows $(m+1)(1-\alpha) \leq |\mathcal{J}|$.

Show "$\Longleftarrow$": We show the conclusion by contradiction. If the conclusion does not hold, then $c_{m+1} \leq k(\alpha)$, implying that $c_i < k(\alpha)$ for any $i \in \mathcal{J}$. Consequently, $\max\{c_i : i \in \mathcal{J}\} < k(\alpha)$. Thus, there exists k_0 such that $\max\{c_i : i \in \mathcal{J}\} < k_0 < k(\alpha)$, showing that

$$\mathcal{J} \subset \mathcal{I}_{k_0}. \tag{A.2}$$

By the condition $|\mathcal{J}| > (m+1)(1-\alpha)$, we obtain $|\mathcal{I}_{k_0}| > (m+1)(1-\alpha)$. On the other hand, by the definition of $k(\alpha)$, we conclude that $k(\alpha) \leq k_0$, which is a contradiction to (A.2). □

Proof of Theorem 3. **(a).** This is immediate by construction.

(b). For any $(\mathrm{x}, \mathrm{y}) \in \overline{\mathcal{D}}_0^c$, define the calibration score

$$s(\mathrm{x}, \widetilde{\mathrm{y}}; \mathrm{y}) = \inf\{k : \mathrm{y} \in \mathcal{C}(\mathrm{x}, \widetilde{\mathrm{y}}; k)\}.$$

Then for any $k \in [K]$,

$$s(\mathrm{x}, \widetilde{\mathrm{y}}; \mathrm{y}) \leq k \Longleftrightarrow \mathrm{y} \in \mathcal{C}(\mathrm{x}, \widetilde{\mathrm{y}}; k). \tag{A.3}$$

Then for any $\alpha \in (0, 1)$,

$$\begin{aligned} \widehat{k}^c(\alpha) &= \inf\{k : |\{i \in \mathcal{A}^c : \mathrm{y}_i \in \mathcal{C}(\mathrm{x}_i, \widetilde{\mathrm{y}}_i; k)\}| \geq (n^c+1)(1-\alpha)\} \\ &= \inf\{k : |\{i \in \mathcal{A}^c : s(\mathrm{x}_i, \widetilde{\mathrm{y}}_i; \mathrm{y}_i) \leq k\}| \geq (n^c+1)(1-\alpha)\} \end{aligned}$$

Because $\{s(\mathrm{X}_i, \widetilde{\mathrm{Y}_i}; \mathrm{Y}_i)\}_{i=1}^{n^c}$ and $s(\mathrm{X}_{N+1}, \widetilde{\mathrm{Y}}_{N+1}; \mathrm{Y}_{N+1})$ are exchangeable random variables, so

$$|\{i \in \mathcal{A}^c : s(\mathrm{X}_{N+1}, \widetilde{\mathrm{Y}}_{N+1}; \mathrm{Y}_{N+1}) > s(\mathrm{x}_i, \widetilde{\mathrm{y}_i}; \mathrm{y}_i)\}|$$

is stochastically dominated by the discrete uniform distribution on $\{0, 1, \ldots, n\}$. Consequently, by (A.3) and Lemma 1,

$$\begin{aligned} & \mathbb{P}\{\mathrm{Y}_{N+1} \notin \mathcal{C}(\mathrm{X}_{N+1}, \widetilde{\mathrm{Y}}_{N+1}; k^c(\alpha)\} \\ = \ & \mathbb{P}\{s(\mathrm{X}_{N+1}, \widetilde{\mathrm{Y}}_{N+1}; \mathrm{Y}_{N+1}) > k^c(\alpha)\} \\ = \ & \mathbb{P}\{|\{i \in \mathcal{A}^c : s(\mathrm{X}_{N+1}, \widetilde{\mathrm{Y}}_{N+1}; \mathrm{Y}_{N+1}) > s(\mathrm{x}_i, \widetilde{\mathrm{y}_i}; \mathrm{y}_i)\}| > (n^c+1)(1-\alpha)\} \\ \leq \ & \mathbb{P}\{U > (n^c+1)(1-\alpha)\} \\ \leq \ & \alpha, \end{aligned}$$

where the second last step is due to the uniform distribution derived from the exchangeablity. and U represents a discrete random variable following a uniform distribution on $\{0, 1, \ldots, n\}$.

□

B Details about Pseudo-Anchor Points

Here, we show results for pseudo anchor points.

Theorem 4. *For $r \in [R]$ and $k \in [K]$, let $p_{rk}(x) = \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)} | \mathrm{Y} = k, \mathrm{X} = \mathrm{x})$. If x is a δ-pseudo anchor point, then*

(i) $(1-\delta)p_{rk}(x) \leq \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} | \mathrm{X} = \mathrm{x}) \leq (K-1)\delta + p_{rk}(x)$;

(ii) $\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|\mathrm{X} = \mathrm{x}) - (K-1)\delta \leq p_{rk}(x) \leq \frac{1}{1-\delta}\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|\mathrm{X} = \mathrm{x})$.

Proof. Assume that x is a δ-pseudo anchor point for class k. Then expression (A.1) becomes

$$\begin{aligned} & \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|\mathrm{X} = \mathrm{x}) \\ = & \sum_{j\in\mathcal{Y}} \Big\{\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|\mathrm{Y} = j, \mathrm{X} = \mathrm{x})\mathbb{P}(Y = j|\mathrm{X} = \mathrm{x})\Big\} \\ = & \sum_{j\neq k} \Big\{\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|\mathrm{Y} = j, \mathbf{x})\mathbb{P}(\mathrm{Y} = j|\mathrm{X} = \mathrm{x})\Big\} \\ & + \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|Y = k, \mathrm{X} = \mathrm{x})\mathbb{P}(\mathrm{Y} = k|\mathrm{X} = \mathrm{x}) \end{aligned} \tag{B.1}$$

Noting that all probabilities in the first term of (B.1) are nonnegative, then by definition of the δ-pseudo anchor point for x, we obtain that

$$\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)}|\mathrm{X} = \mathrm{x}) \geq (1-\delta)\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|\mathrm{Y} = k, \mathrm{X} = \mathrm{x}). \tag{B.2}$$

On the other hand, if x is a δ-pseudo anchor point, then

$$\mathbb{P}(\mathrm{Y} = j|\mathrm{X} = \mathrm{x}) \leq \delta \quad \text{for any} \quad j \neq k,$$

therefore, by that all conditional probabilities in (B.1) are between 0 and 1, we obtain that

$$\begin{aligned} & \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|\mathrm{X} = \mathrm{x}) \\ \leq & \sum_{j\neq k} \Big\{\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|\mathrm{Y} = j, \mathbf{x}) \times \delta\Big\} + p_{rk}(x)\mathbb{P}(\mathrm{Y} = k|\mathrm{X} = \mathrm{x}) \\ \leq & \delta \sum_{j\neq k} \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|\mathrm{Y} = j, \mathbf{x}) + p_{rk}(x) \\ = & (K-1)\delta + p_{rk}(x). \end{aligned} \tag{B.3}$$

Combining (B.2) and (B.3) gives us

$$(1-\delta)p_{rk}(x) \leq \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|\mathrm{X} = \mathrm{x}) \leq (K-1)\delta + p_{rk}(x),$$

leading to

$$\begin{aligned} & \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|\mathrm{X} = \mathrm{x}) - (K-1)\delta \\ \leq & \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|Y = k, \mathrm{X} = \mathrm{x}) \\ \leq & \frac{1}{1-\delta}\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|\mathrm{X} = \mathrm{x}). \end{aligned} \tag{B.4}$$

□

Remark. The inequalities in (B.4) has important implications. In the degenerate situation with $\delta = 0$, i.e., x is an anchor point, (B.4) recovers the identity:

$$\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|Y = k, \mathrm{X} = \mathrm{x}) = \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|\mathrm{X} = \mathrm{x}).$$

When δ is extremely small such that $(K-1)\delta$ is close to 0 and $\frac{1}{1-\delta}$ is close to 1, we have that

$$\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|Y = k, \mathrm{X} = \mathrm{x}) \approx \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = \tilde{\mathrm{y}}^{(r)}|\mathrm{X} = \mathrm{x}).$$

, showing that a pseudo-anchor point can be practically regarded as an anchor point.

C Deep Neural Networks

We describe architectures ψ_A, ψ_C, and ψ_S in detail. Let

$$\mathcal{S}^{K-1} = \left\{ (s_1, \ldots, s_K)^{\mathrm{T}} : s_j \geq 1 \quad \text{for} \quad j \in [K] \quad \text{and} \quad \sum_{j=1}^{K} s_j = 1 \right\}$$

denote the $(K-1)$-dimensional simplex, and let

$$G : \mathbb{R}^K \longrightarrow \mathcal{S}^{K-1}$$

denote a softmax function, given by

$$G(z) = \begin{pmatrix} \frac{\exp(z_1)}{\sum_{j=1}^{K} \exp(z_j)} \\ \frac{\exp(z_2)}{\sum_{j=1}^{K} \exp(z_j)} \\ \vdots \\ \frac{\exp(z_K)}{\sum_{j=1}^{K} \exp(z_j)} \end{pmatrix} \quad \text{for} \quad z = (z_1, \ldots, z_K)^{\mathrm{T}}.$$

For $r \in [R]$, we now describe the conditional probability mass function of $\widetilde{\mathrm{Y}}^{(r)}$, given Y and X. Specifically, for $k \in [K]$, we specify the vector of the conditional probability mass functions of $\widetilde{\mathrm{Y}}^{(r)}$, given $\mathrm{Y} = k$ and $\mathrm{X} = \mathrm{x}$ as:

$$\begin{pmatrix} \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = 1 | \mathrm{Y} = k, \mathrm{X} = \mathrm{x}) \\ \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = 2 | \mathrm{Y} = k, \mathrm{X} = \mathrm{x}) \\ \vdots \\ \mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = K | \mathrm{Y} = k, \mathrm{X} = \mathrm{x}) \end{pmatrix} = G \left\{ \begin{pmatrix} \alpha_1^{(r)} \\ \alpha_2^{(r)} \\ \vdots \\ \alpha_K^{(r)} \end{pmatrix} \psi^{\mathrm{A}}(\mathrm{x}) + \begin{pmatrix} \beta_1^{(k)} \\ \beta_2^{(k)} \\ \vdots \\ \beta_K^{(k)} \end{pmatrix} \psi^{\mathrm{C}}(\mathrm{x}) \right\}$$

where $\alpha^{(r)} \triangleq (\alpha_1^{(r)}, \ldots, \alpha_K^{(r)})^{\mathrm{T}}$ and $\beta^{(k)} \triangleq (\beta_1^{(k)}, \ldots, \beta_1^{(k)})^{\mathrm{T}}$ are weights; and $\psi^{\mathrm{A}}(\mathrm{x})$ and $\psi^{\mathrm{C}}(\mathrm{x})$ are functions facilitating the dependence on the annotator's skills and the class label.

Expressing this elementwisely, we obtain that for $j \in [K]$,

$$\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = j | \mathrm{Y} = k, \mathrm{X} = \mathrm{x}) = \frac{\exp\{< \alpha_j^{(r)}, \psi^{\mathrm{A}}(\mathrm{x}) > + < \beta_j^{(k)}, \psi^{\mathrm{C}}(\mathrm{x}) >\}}{\sum_{l=1}^{K} \exp\{< \alpha_l^{(r)}, \psi^{\mathrm{A}}(\mathrm{x}) > + < \beta_l^{(k)}, \psi^{\mathrm{C}}(\mathrm{x}) >\}}. \tag{C.1}$$

Here, the weights $\alpha^{(r)}$ and $\beta^{(k)}$ and the functions $\psi^{\mathrm{A}}(\mathrm{x}_i)$ and $\psi^{\mathrm{C}}(\mathrm{x})$ are unknown, which need to be trained using the data $\overline{\mathcal{D}}_{0,k}$ or its subset.

To flexibly reflect possibly different effects of the annotator expertise (r) and the ground truth (k) in the annotation process, we employ deep neural network (DNN) architectures to describe $\psi^{\mathrm{A}}(\mathrm{x}_i)$ and $\psi^{\mathrm{C}}(\mathrm{x}_i)$. Specifically, we specify $\psi^{\mathrm{A}}(\mathrm{x}_i)$ as a network with an input layer and an output layer that are linked by $H^{\mathrm{A}} - 1$ hidden layers, where the hth hidden layer has L_h^{A} nodes for $h = 1, \ldots, H^{\mathrm{A}} - 1$. Let $L^{\mathrm{A}} = (L_0^{\mathrm{A}}, L_1^{\mathrm{A}}, \ldots, L_{H^{\mathrm{A}}}^{\mathrm{A}})^{\mathrm{T}}$ denote the width vector for the network, with $L_0^{\mathrm{A}} = p$ for the input layer that records measurements of p elements of x_i, and $L_{H^{\mathrm{A}}}^{\mathrm{A}} = 1$ for the output layer. The network architecture $\{H^{\mathrm{A}}, L^{\mathrm{A}}\}$ is characterized by a sequence of linear and nonlinear functions, approximating $\psi^{\mathrm{A}}(\mathrm{x})$ by

$$\widehat{\psi}^{\mathrm{A}}(\theta^{\mathrm{A}}; \mathrm{x}) \triangleq W_{H^{\mathrm{A}}}^{\mathrm{A}} \sigma_{H^{\mathrm{A}}-1}^{\mathrm{A}} \Big[\cdots \sigma_2^{\mathrm{A}} \Big\{ W_2^{\mathrm{A}} \sigma_1^{\mathrm{A}} (W_1^{\mathrm{A}} \mathrm{x} + b_1^{\mathrm{A}}) + b_2^{\mathrm{A}} \Big\} + b_3^{\mathrm{A}} \cdots \Big] + b_{H^{\mathrm{A}}}^{\mathrm{A}}, \tag{C.2}$$

or equivalently,

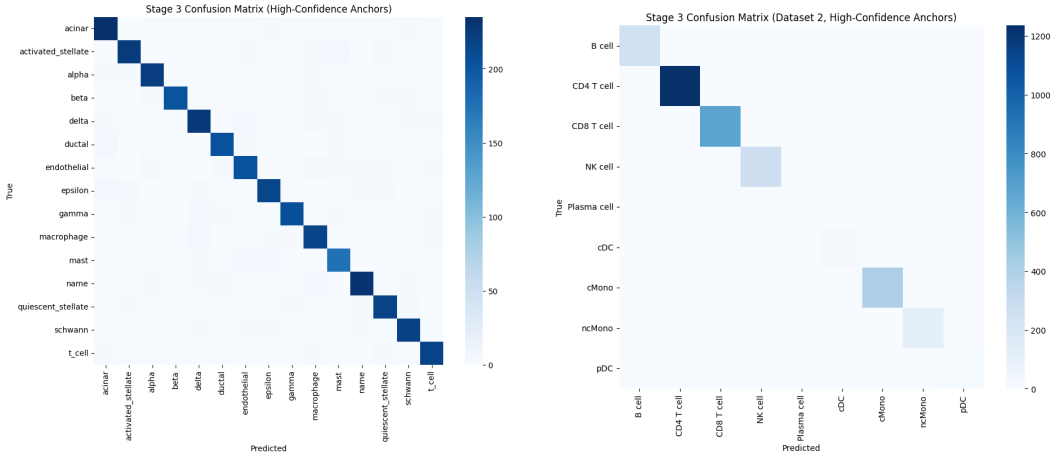
$$\widehat{\psi}^{\mathrm{A}}(\theta; \mathrm{x}) \triangleq g(H^{\mathrm{A}}; \mathrm{x}),$$

where the g functions is determined by the recursive equation

$$g(j; \mathrm{x}) = W_j^{\mathrm{A}} g(j-1; \mathrm{x}) + b_j^{\mathrm{A}} \quad \text{for} \quad j = 2, \ldots, H^{\mathrm{A}};$$

with

$$g(1; \mathrm{x}) = \sigma_1^{\mathrm{A}}(W_1^{\mathrm{A}} \mathrm{x} + b_1^{\mathrm{A}}).$$



(a) Dataset 1: *Baron3*

(b) Dataset 2: *PBMC2*

Figure 4: Confusion Matrices of predicted cells vs true cell types for both datasets

Here, for $h = 1, \ldots, H^{\mathrm{A}}$, W_h^{A} is an $L_h^{\mathrm{A}} \times L_{h-1}^{\mathrm{A}}$ weight matrix, $b_h^{\mathrm{A}} \in \mathbb{R}^{L_h^{\mathrm{A}}}$ is the bias vector in layer h, θ^{A} is the parameter vector formed by stacking $\{W_h^{\mathrm{A}}, b_h^{\mathrm{A}}\}_{h=1}^{H^{\mathrm{A}}}$ from bottom to top, σ_h^{A} is a user-specified activation function that operates elementwise (e.g, a ReLu function), and x is a p-dimensional argument.

Analogously, we specify $\psi^{\mathrm{C}}(\mathrm{x})$ as a network, using similar notation but replacing the subscript $_{\mathrm{A}}$ with $_{\mathrm{C}}$ for the relevant quantities. Let $\theta = (\theta^{\mathrm{AT}}, \theta^{\mathrm{CT}}; \alpha^{(r)}, \beta^{(k)} : r \in [R], k \in [K])^{\mathrm{T}}$. As a result, the conditional probability in (C.1) is modeled as follows:

$$\mathbb{P}(\widetilde{\mathrm{Y}}^{(r)} = j | \mathrm{Y} = k, \mathrm{X} = \mathrm{x}) = \frac{\exp\{< \alpha_j^{(r)}, \widehat{\psi}^{\mathrm{A}}(\theta^{\mathrm{A}}; \mathrm{x}) > + < \beta_j^{(k)}, \widehat{\psi}^{\mathrm{C}}(\theta^{\mathrm{C}}; \mathrm{x}) >\}}{\sum_{l=1}^{K} \exp\{< \alpha_l^{(r)}, \widehat{\psi}^{\mathrm{A}}(\theta^{\mathrm{A}}; \mathrm{x}) > + < \beta_l^{(k)}, \widehat{\psi}^{\mathrm{C}}(\theta^{\mathrm{C}}; \mathrm{x}) >\}}.$$

D Extended Results

We include per-class coverage tables, histograms, and plots to supplement the analysis results in Section 5.

Table 3: Confusion matrix of predicted vs. true cell types

	B cell	CD4 T cell	CD8 T cell	NK cell	Plasma cell	cDC	cMono	ncMono	pDC
B cell	75	0	0	0	0	0	0	0	0
CD4 T cell	0	363	8	0	0	0	0	0	0
CD8 T cell	0	18	181	4	0	0	0	0	0
NK cell	0	0	5	76	0	0	0	0	0
Plasma cell	0	0	0	0	2	0	0	0	0
cDC	0	0	0	0	0	6	0	0	0
cMono	0	0	0	0	0	0	123	0	0
ncMono	0	0	0	0	0	0	0	36	0
pDC	0	0	0	0	0	1	0	0	2

Table 4: Per-class classification report

	B cell	CD4 T cell	CD8 T cell	NK cell	Plasma cell	cDC	cMono	ncMono	pDC	Accuracy	Macro Avg	Weighted Avg
Precision	1.000	0.953	0.933	0.950	1.000	0.857	1.000	1.000	1.000	0.960	0.966	0.960
Recall	1.000	0.978	0.892	0.938	1.000	1.000	1.000	1.000	0.667	0.960	0.942	0.960
F1-score	1.000	0.965	0.912	0.944	1.000	0.923	1.000	1.000	0.800	0.960	0.949	0.960
Support	75	371	203	81	2	6	123	36	3	0.960	900	900

Table 5: Performance comparison on the *Baron3* dataset. Our method is the proposed anchor-based conformal prediction (Top-k). Metrics include ROC-AUC and PR-AUC.

Model	ROC-AUC	PR-AUC
GCN	0.9942	0.9815
MLP	0.9881	0.9799
GAT	0.9867	0.9738
GraphSAGE	0.9909	0.9803
SingleCellNet	0.9866	0.9756
ACTINN	0.9889	0.9804
Ours (Anchor-based CP)	**0.9953**	**0.9803**