Domain Adaptive and Fine-grained Anomaly Detection for Single-cell Sequencing Data and Beyond

Kaichen Xu[1*], **Yueyang Ding**[1*], **Suyang Hou**[2], **Weiqiang Zhan**[1], **Nisang Chen**[1], **Jun Wang**[3], **Xiaobo Sun**[1†]

[1]School of Statistics and Mathematics, Zhongnan University of Economics and Law
[2]School of Information Engineering, Zhongnan University of Economics and Law
[3]iWudao Tech
{kaichenxu, yueyangding, suyang, weiqiangzhan, nisangchen}@stu.zuel.edu.cn, xsun28@gmail.com, jwang@iwudao.tech

Abstract

Fined-grained anomalous cell detection from affected tissues is critical for clinical diagnosis and pathological research. Single-cell sequencing data provide unprecedented opportunities for this task. However, current anomaly detection methods struggle to handle domain shifts prevalent in multi-sample and multi-domain single-cell sequencing data, leading to suboptimal performance. Moreover, these methods fall short of distinguishing anomalous cells into pathologically distinct subtypes. In response, we propose ACSleuth, a novel, reconstruction deviation-guided generative framework that integrates the detection, domain adaptation, and fine-grained annotating of anomalous cells into a methodologically cohesive workflow. Notably, we present the first theoretical analysis of using reconstruction deviations output by generative models for anomaly detection in lieu of domain shifts. This analysis informs us to develop a novel and superior maximum mean discrepancy-based anomaly scorer in ACSleuth. Extensive benchmarks over various single-cell data and other types of tabular data demonstrate ACSleuth's superiority over the state-of-the-art methods in identifying and subtyping anomalies in multi-sample and multi-domain contexts. Our code is available at https://github.com/Catchxu/ACsleuth.

1 Introduction

The detection and differentiation of anomalous cells (ACs) from affected tissues, which we refer to as Fine-grained Anomalous Cell Detection (FACD), is critical for investigating the pathological heterogeneity of diseases, significantly contributing to the clinical diagnostics, biomedical research, and the development of targeted therapies [Aevermann *et al.*, 2018]. Single-cell (SC) sequencing data, referred to as SC data for brevity, such as single-cell RNA-sequencing (scRNA-seq) and single-cell ATAC-sequencing (scATAC-seq) data, provide unprecedented opportunities for FACD analysis from distinct perspectives [Zhai *et al.*, 2023b; Pouyan *et al.*, 2016; Macnair and Robinson, 2023]. For example, a typical scRNA-seq dataset is organized as a tabular matrix $\mathbf{X} \in \mathbb{R}^{N \times G}$, where $\mathbf{X}_{i,j}$ represents the expression read counts of the j-th gene in the i-th cell. This dataset characterizes gene expression levels within single cells, thus facilitating the FACD.

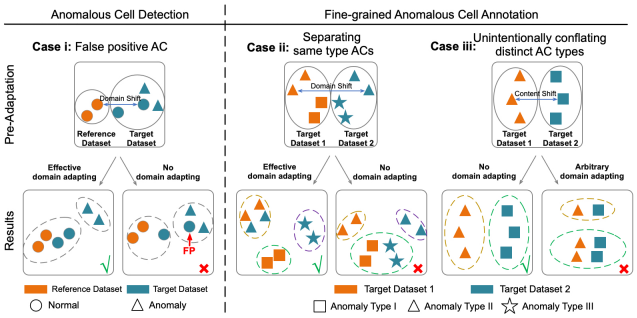

Figure 1: Three error types in multi-sample and multi-domain FACD analysis due to domain shift and sample-specific AC types. Domain shift represents non-biological variations caused by technical differences among samples, while content shift represents true biological variations.

The FACD workflow is a two-step process: the detection of anomalous cells as a whole followed by their fine-grained annotating. The detection step aims to identify ACs in target datasets by comparing them to "normality" defined by a reference dataset containing normal cells only. The subsequent fine-grained annotating step involves clustering the detected ACs into different groups, each representing a biologically distinct subtype. However, de novo FACD using SC data presents a significant challenge, especially in multi-sample contexts, primarily due to various types of cross-sample **Domain Shifts** (DS), i.e., batch effects, stemming from non-biological variations in data types, sequencing technologies, and experimental conditions [Holmström *et al.*, 2022; Zhao *et al.*, 2020]. These DS often intermingle with content shifts, which originate from biological variations and represent true signals crucial for AC detection, leading to several types of errors in FACD analysis as shown in Figure 1: **i**)

*Equal contribution.
†Corresponding author.

False positive AC. Normal cells in the target dataset can be erroneously labeled as anomalous due to the DS relative to the reference dataset; **ii) Separating same type ACs**. ACs of the same type across different target datasets may appear very different due to DS, leading to their misclassification as distinct types; **iii) Unintentionally conflating distinct AC types**. Efforts to mitigate DS can inadvertently diminish content shifts critical for distinguishing between sample-specific AC types, leading to their merging as one type.

Anomaly detection (AD) methods generally involve assigning an anomaly score to each target instance, followed by setting a threshold for anomaly determination as per the user's tolerance for false positives [Fernández-Saúco *et al.*, 2019]. This strategy is followed by approaches tailored for AC detection, including uncertainty-based and generative methods [Li *et al.*, 2022]. The former methods, exemplified by scmap [Kiselev *et al.*, 2018], assign anomaly scores based on cluster assignment uncertainty, while the latter methods, such as CAMLU [Li *et al.*, 2022], attempt to reconstruct cells and output reconstruction deviations as anomaly scores. These methods either are susceptible to DS or fail to provide fine-grained annotation, i.e., labeling all ACs as "unassigned" [Petegrosso *et al.*, 2020; Zhai *et al.*, 2023a]. To mitigate DS, domain adaptation methods like MNN [Haghverdi *et al.*, 2018] are often used before AC detection, yet with limited effectiveness owing to their sensitivity to ACs [Yang *et al.*, 2020]. For fine-grained AC annotation, researchers often resort to clustering methods, which are not specifically oriented towards this purpose and thus struggle to capture subtle differences among ACs. Particularly, this separation of fine-grained AC annotation as an independent clustering task can disrupt the methodological coherence of the workflow, leading to the overall underperformance of FACD [Chen *et al.*, 2020]. MARS [Brbić *et al.*, 2020] and scGAD (also known as scPOT) [Zhai *et al.*, 2023b; Zhai *et al.*, 2023a] represent the only two available methods that integrate AC detection and fine-grained annotation. Yet, MARS, lacking a domain adaptation mechanism, exhibits compromised performance in lieu of DS [Zhai *et al.*, 2023a]. scGAD falls short of handling multiple target datasets due to its reliance on MNN for domain adaptation, which is effective primarily in single target dataset scenarios [Yu *et al.*, 2023a].

Inspired by the versatility of generative adversarial networks (GAN) in AD [Zenati *et al.*, 2018] and domain adaptation-related style-transfer [Zhu *et al.*, 2017] tasks, we propose ACSleuth, an innovative GAN-based framework designed to integrate AC detection, domain adaptation, and fine-grained annotation into a cohesive workflow, as shown in Figure 2. For AC detection (**Phase I**), a specialized GAN module (*module I*) is trained on the reference dataset and then applied to target datasets to discriminate between normal and anomalous cells based on their anomaly scores. *Module I* differs from existing GAN-based AD methods in incorporating a memory block to reduce mode collapse risk [Gong *et al.*, 2019] and featuring a novel maximum mean discrepancy (MMD)-based anomaly scorer. This scorer translates cell reconstruction deviations into anomaly scores, whose efficacy in facilitating AD is theoretically proved in **Theorem 3.3**. In the subsequent **Phase II**, ACSleuth utilizes a second GAN module (*module II*) to pinpoint "kin" normal cell pairs between the reference and target datasets, from which DS inherent in each target dataset can be learned as a matrix for domain adaptation. This approach prevails over existing domain adaptation methods for SC data, e.g., MNN, in preserving the data's original scale and semantic integrity, simultaneously adapting multiple datasets, and being resilient against complications caused by sample-specific AC types. Finally, ACSleuth performs a self-paced deep clustering (**Phase III**) on fusions of embeddings and reconstruction deviations of domain-adapted ACs, obtaining iteratively enhanced fine-grained annotations. Overall, our main contributions are:

- We propose an innovative framework for FACD in SC data, which integrates AC detection, domain adaptation, and fine-grained annotation into a cohesive workflow.
- We provide the first theoretical analysis of using reconstruction deviations for AD in lieu of DS (**Theorem 3.1-3.3**). This analysis also informs us to develop a novel and effective MMD-based anomaly scorer.
- We introduce a novel domain adaptation method in multi-sample and multi-domain contexts.
- We innovatively fuse reconstruction deviations with domain-adapted cell embeddings as clustering inputs for enhanced AC fine-grained annotations.
- ACSleuth has been rigorously benchmarked in various experimental scenarios regarding DS types, target sample quantity, and sample-specific AC types. ACSleuth consistently outperforms the state-of-the-art (SOTA) methods across all scenarios. Furthermore, ACSleuth is also applicable to other types of tabular data.

2 Related Works

2.1 Anomalous Single Cell Detection

AD methods tailored for SC data, such as scmap, CHETAH [De Kanter *et al.*, 2019], and scPred [Alquicira-Hernandez *et al.*, 2019], typically involve cell classification or clustering, labeling cells with uncertain assignments as ACs. These methods suffer from high false positive rates, as the assignment uncertainty might stem from similarities among normal cell types rather than the presence of ACs [Li *et al.*, 2022]. To circumvent this issue, CAMLU directly discriminates ACs from normal cells using informative genes selected based on their reconstruction deviations. Moreover, AD methods for general types of tabular data, categorized as generative, one-class classification, and contrastive approaches, are also adaptable to SC data. Generative methods like RCA [Liu *et al.*, 2021] and ALAD [Zenati *et al.*, 2018] detect anomalies based on instance reconstruction deviations, using collaborative autoencoders (AE) and a GAN model, respectively. DeepSVDD [Ruff *et al.*, 2018], a one-class classification method, learns a finite-sized "normality" hypersphere in a latent space and labels outside instances as anomalies. Contrastive methods ICL [Shenkar and Wolf, 2021] and NeuTraL [Qiu *et al.*, 2021] generate different views for contrastive learning using mutual information-based mappings and learnable neural transformations, respectively, with anomalies identified as those with large contrastive losses.

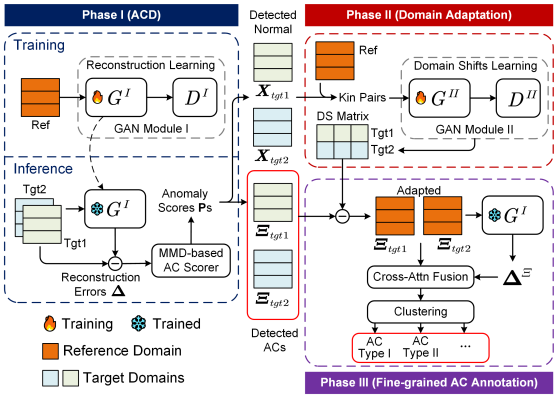

Figure 2: The workflow of ACSleuth.

Lastly, a recent study SLAD [Xu *et al.*, 2023] introduces scale learning to identify anomalies as those unfitted to the scale distribution of inlier representations. A common limitation across these methods is their inability to account for DS between reference and target datasets and to further categorize ACs into fine-grained subtypes.

2.2 Fine-grained Anomalous Cell Annotation

As a task following AC detection, fine-grained AC annotation in SC data is often approached as an independent clustering task using general-purpose clustering methods like Leiden [Traag *et al.*, 2019], EDESC [Cai *et al.*, 2022], and DFCN [Tu *et al.*, 2021], or methods specifically designed for SC data, such as scTAG [Yu *et al.*, 2022], Seurat V5 [Hao *et al.*, 2023], and scDEFR [Cheng *et al.*, 2023]. EDESC, DFCN, scTAG, and scDEFR, which are based on DEC [Xie *et al.*, 2016], adopt a schema of joint optimization of instance representation learning and discriminatively boosted clustering, yet differ in their specific approaches for learning representations and calculating cluster soft assignments. Seurat V5, on the other hand, first builds a KNN graph based on cell-cell similarities and then applies community detection clustering using the Louvain algorithm. A significant challenge with these methods is their dependency on the accuracy of the AC detection in the first place. Treating AC detection and fine-grained annotation as separate tasks can disrupt the methodological coherence of the FACD workflow, potentially compromising its overall performance. To tackle this issue, scGAD and MARS integrate both tasks into a unified process. MARS, for example, learns a latent space where cells positioned near landmarks of distinct anomaly subtypes are assigned corresponding anomaly annotations. While these methods have shown promising results when all ACs are from the same domain, their efficacy can be undermined in the presence of DS. To tackle this, scGAD employs the MNN to adapt DS before fine-grained annotating. However, MNN's limitation to a single reference and target dataset makes scGAD less suitable for multi-sample and multi-domain contexts.

3 Methods

ACSleuth's workflow (Figure 2) comprises three phases corresponding to AC detection (**Phase I**), domain adaptation (**Phase II**), and fine-grained annotation (**Phase III**).

3.1 Anomalous Cell Detection (Phase I)

In Phase I, a GAN model (*module I*) is initially trained to reconstruct normal cells in the reference dataset, and then applied to target datasets. Anomaly scores, derived from reconstruction deviations, are then used to identify ACs. Let $\boldsymbol{x}_i \in \mathbb{R}^{N_{gene}}$ denote the gene expression vector of a cell $i \in S^k$, S^k denote the single-cell set in domain k, N_{gene} the number of genes. Then we have the following assumption [Hornung *et al.*, 2016]:

Assumption 3.1.

$$\boldsymbol{x}_i = \boldsymbol{x}_i^* + \boldsymbol{b}^k + \boldsymbol{\epsilon}_i, \tag{1}$$

where $\boldsymbol{x}_i^*$ denotes cell i's biological content, $\boldsymbol{b}^k$ the non-biological content specific to domain k, $\boldsymbol{\epsilon}_i \sim N(0, \boldsymbol{\sigma}_i^2)$ a random Gaussian noise.

GAN Model Training. *Module I* comprises a generator G^I and a discriminator D^I. The generator itself is composed of three components: an MLP-based encoder ($G_E^I : \mathbb{R}^{N_{gene}} \to \mathbb{R}^p$), an MLP-based decoder ($G_D^I : \mathbb{R}^p \to \mathbb{R}^{N_{gene}}$), and a queue-based memory block ($\boldsymbol{Q} \in \mathbb{R}^{N_{mem} \times p}$), where p is the cell embedding dimension, N_{mem} is the memory block size. The embedding of any cell i can be expressed as:

$$\boldsymbol{z}_i = G_E^I(\boldsymbol{x}_i) \in \mathbb{R}^p. \tag{2}$$

The memory block provides an attention-based means to reconstruct $\boldsymbol{z}_i$ as $\widetilde{\boldsymbol{z}}_i$:

$$\widetilde{\boldsymbol{z}}_i = \boldsymbol{Q}^T \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{z}_i}{\tau}\right) \in \mathbb{R}^p, \tag{3}$$

where τ is a temperature hyperparameter. During the training, $\boldsymbol{Q}$ is dynamically updated by enqueuing the most recent $\widetilde{\boldsymbol{z}}$ and dequeuing the oldest ones to maintain a balance between preserving learned features and adapting to new data, thus reducing the risk of mode collapse. G_D^I reconstructs cell i from $\widetilde{\boldsymbol{z}}_i$ as $\widehat{\boldsymbol{x}}_i$:

$$\widehat{\boldsymbol{x}}_i = G_D^I(\widetilde{\boldsymbol{z}}_i) \in \mathbb{R}^{N_{gene}}. \tag{4}$$

The discriminator D^I comprises an encoder, similar to G_E^I, and an MLP-based classifier. D^I is trained to distinguish between $\boldsymbol{x}$ and $\widehat{\boldsymbol{x}}$. The loss functions for the generator ($\mathcal{L}_{G^I}$) and discriminator ($\mathcal{L}_{D^I}$) are defined as:

$$\mathcal{L}_{G^I} = \alpha\mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{adv}} = \alpha\mathbb{E}\left[\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|_1\right] - \beta\mathbb{E}\left[D^I(\widehat{\boldsymbol{x}})\right], \tag{5}$$

$$\mathcal{L}_{D^I} = \mathbb{E}\left[D^I(\widehat{\boldsymbol{x}}) - D^I(\boldsymbol{x})\right] + \lambda\mathbb{E}\left[\left(\|\nabla_{\widetilde{\boldsymbol{x}}} D^I(\widetilde{\boldsymbol{x}})\|_2 - 1\right)^2\right], \tag{6}$$

where $\widetilde{\boldsymbol{x}} = \epsilon\widehat{\boldsymbol{x}} + (1-\epsilon)\boldsymbol{x}$, $\epsilon \in (0, 1)$. Here, $\mathcal{L}_{\text{rec}}$ represents the reconstruction loss, $\mathcal{L}_{\text{adv}}$ the adversarial loss, and $\alpha, \beta, \lambda \geq 0$ the weights of three loss terms. A gradient penalty term applied to $\widetilde{\boldsymbol{x}}$ ensures the Lipschitz continuity of the discriminator [Gulrajani *et al.*, 2017].

MMD-based anomalous cell scorer. Let $\boldsymbol{x}_i$ denote a normal cell, and $\boldsymbol{\xi}_j$ an anomalous cell, $\boldsymbol{\delta}_i^x := \boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i$, $\boldsymbol{\delta}_j^\xi := \boldsymbol{\xi}_j - \widehat{\boldsymbol{\xi}}_j$ denote their reconstruction deviations. The scorer aims to utilize the MMD metric to translate $\boldsymbol{\delta}$s into anomaly scores that can represent the maximized discrepancy between

$\boldsymbol{\delta}^x$ and $\boldsymbol{\delta}^\xi$ in the Reproduced Hilbert Kernel Space (RHKS), thus more effective facilitating the differentiation between normal and anomalous cells. Specifically, given the distributions $\boldsymbol{\delta}_i^x \sim p, \boldsymbol{\delta}_j^\xi \sim q$, the MMD metric is calculated as [Gretton *et al.*, 2012]:

$$\begin{aligned} MMD^2\left(\boldsymbol{\delta}_i^x, \boldsymbol{\delta}_j^\xi\right) &= \|\frac{1}{m}\sum_{i=1}^{m}\phi(\boldsymbol{\delta}_i^x) - \frac{1}{n}\sum_{j=1}^{n}\phi(\boldsymbol{\delta}_j^\xi)\|_{\mathcal{H}}^2 \\ &= \mathbb{E}_{\boldsymbol{\delta}^x,(\boldsymbol{\delta}^x)'\sim p}\left[k(\boldsymbol{\delta}^x,(\boldsymbol{\delta}^x)')\right] + \mathbb{E}_{\boldsymbol{\delta}^\xi,(\boldsymbol{\delta}^\xi)'\sim q}\left[k(\boldsymbol{\delta}^\xi,(\boldsymbol{\delta}^\xi)')\right] \\ &- 2\mathbb{E}_{\boldsymbol{\delta}^x\sim p,\boldsymbol{\delta}^\xi\sim q}\left[k(\boldsymbol{\delta}^x,\boldsymbol{\delta}^\xi)\right], \end{aligned} \tag{7}$$

where k is a positive definite kernel function, m and n denote the actual yet unknown numbers of inliers and anomalies in the target dataset. An AC scorer function can be trained by minimizing the loss function:

$$\min_{\boldsymbol{\delta}_m^x,\boldsymbol{\delta}_n^\xi} \quad \mathcal{L}_p\left(\boldsymbol{\delta}_m^x,\boldsymbol{\delta}_n^\xi\right) = -MMD^2\left(\boldsymbol{\delta}_m^x,\boldsymbol{\delta}_n^\xi\right), \tag{8}$$

where $\boldsymbol{\delta}_m^x = \{\boldsymbol{\delta}_i^x | \boldsymbol{\delta}_i^x \in \mathbb{R}^{N_{gene}}, i = 1,2,\cdots,m]\}$, $\boldsymbol{\delta}_n^\xi = \{\boldsymbol{\delta}_j^\xi | \boldsymbol{\delta}_j^\xi \in \mathbb{R}^{N_{gene}}, j = 1,2,\cdots,n\}$ denote the sets of reconstruction deviations of inliers and anomalies, respectively.

As shown in the theoretical analysis in section 3.4, given a linear kernel k, the loss function in equation (8) can be reformulated as:

$$\mathcal{L}_p\left(\boldsymbol{\delta}_m^x,\boldsymbol{\delta}_n^\xi\right) = -\sum_{i}^{m+n}\sum_{j\neq i}^{m+n} k(\boldsymbol{\delta}_i,\boldsymbol{\delta}_j)\gamma_c(p_i,p_j), \tag{9}$$

where $p_i \coloneqq f_p(\boldsymbol{\delta}_i) \in (0,1)$ is instance i's anomaly score, and f_p is a trainable MLP-based function. $\gamma_c(p_i,p_j) : (0,1)\times(0,1)\to\mathbb{R}$ defines a mapping function as:

$$\begin{aligned} \gamma_c(p_i,p_j) \coloneqq &\frac{\sin\pi p_i \sin\pi p_j}{\pi^2}\left(\frac{[m(m-1)]^{-1}}{p_i p_j} - \frac{(mn)^{-1}}{(p_i-1)p_j}\right. \\ &\left. - \frac{(mn)^{-1}}{p_i(p_j-1)} + \frac{[n(n-1)]^{-1}}{(p_i-1)(p_j-1)}\right). \end{aligned} \tag{10}$$

Here, $\widetilde{n} = \sum_i^{m+n} p_i$ and $\widetilde{m} = (m+n) - \widetilde{n}$ are used in place of the unknown m and n. During the training, $p, \widetilde{n}$ and $\widetilde{m}$ are iteratively updated until convergence. Upon training completion, anomalies can be identified based on their anomaly scores p. Particularly, the property of $\mathcal{L}_p$'s bounded variation across domains endows our AC scorer's robustness against DS, as formalized in section 3.4.

3.2 Multi-sample Domain adaptation (Phase II)

This phase aims to adapt non-biological DS among ACs to expose their biological differences for fine-grained cell annotating. Initially, ACs detected in Phase I are excluded from the datasets to eliminate their interference. The DS of N_{target} datasets relative to the reference dataset are then learned as a matrix $\boldsymbol{S} \in \mathbb{R}^{N_{target}\times p}$ through the generator (G^{II}) of the second GAN module (*module II*), which has an encoder (G_E^{II}) and a decoder (G_D^{II}) similar to those in *module I*. A target cell $\boldsymbol{x}$ is adapted to the reference domain as follows:

$$\begin{aligned} \boldsymbol{z} = G_E^{II}(\boldsymbol{x}) \in \mathbb{R}^p, \quad \widetilde{\boldsymbol{z}} = \boldsymbol{z} - \boldsymbol{S}^T\boldsymbol{b}, \\ \widehat{\boldsymbol{x}} = G_D^{II}(\widetilde{\boldsymbol{z}}), \end{aligned} \tag{11}$$

where $\boldsymbol{b} \in \{0,1\}^{N_{target}}$ denotes the cell's one-hot encoded domain-identity vector, $\widehat{\boldsymbol{x}}$ represents the cell post-domain adaptation. The discriminator of *module II*, D^{II}, is trained to discriminate $\widehat{\boldsymbol{x}}$ as either from the reference domain or not. The loss functions of G^{II} and D^{II} are:

$$\mathcal{L}_{G^{II}} = \alpha\mathbb{E}\left[\|\boldsymbol{x}^+ - \widehat{\boldsymbol{x}}\|_1\right] - \beta\mathbb{E}\left[D^{II}(\widehat{\boldsymbol{x}})\right], \tag{12}$$

$$\begin{aligned} \mathcal{L}_{D^{II}} = &\mathbb{E}\left[D^{II}(\widehat{\boldsymbol{x}})\right] - \mathbb{E}\left[D^{II}(\boldsymbol{x}^+)\right] + \\ &\lambda\mathbb{E}\left[\left(\|\nabla_{\widetilde{\boldsymbol{x}}}D^{II}(\widetilde{\boldsymbol{x}})\|_2 - 1\right)^2\right], \end{aligned} \tag{13}$$

where $\widetilde{\boldsymbol{x}} = \epsilon\widehat{\boldsymbol{x}} + (1-\epsilon)\boldsymbol{x}^+$, $\epsilon \in (0,1)$. $\boldsymbol{x}^+$ is a "kin" cell from the reference dataset that is most similar to $\boldsymbol{x}$ in their biological contents. For each target cell $\boldsymbol{x}_i$, its "kin" reference cell $\boldsymbol{x}_i^+$ is identified as:

$$\boldsymbol{x}_i^+ = \underset{\boldsymbol{x}_j^+\in\mathcal{X}_{ref}}{\arg\max} K\left(G_E^{II}(\boldsymbol{x}_i), G_E^{II}(\boldsymbol{x}_j^+)\right), \tag{14}$$

where $\mathcal{X}_{ref}$ denotes the reference dataset, K is a Gaussian kernel function as in [You *et al.*, 2018].

3.3 Fine-grained Anomalous Cell Annotation (Phase III)

In Phase III, we first obtain *module I*-generated reconstruction deviations (denoted as $\boldsymbol{\Delta} \in \mathbb{R}^{N_{an}\times N_{gene}}$) of ACs post-domain adaptation (denoted as $\boldsymbol{\Xi} \in \mathbb{R}^{N_{an}\times N_{gene}}$), where N_{an} is the number of detected ACs:

$$\widehat{\boldsymbol{\Xi}} = G_D^I\left(G_E^I(\boldsymbol{\Xi})\right), \quad \boldsymbol{\Delta} = \boldsymbol{\Xi} - \widehat{\boldsymbol{\Xi}}. \tag{15}$$

Next, AC embeddings and reconstruction deviations are fused using a cross-attention fusion block:

$$\begin{aligned} \boldsymbol{\Psi} &= \text{Multi-Attn}(\boldsymbol{\Xi}\boldsymbol{W}^Q, \boldsymbol{\Delta}\boldsymbol{W}^K, \boldsymbol{\Delta}\boldsymbol{W}^V) \in \mathbb{R}^{N_{an}\times d}, \\ \boldsymbol{Z} &= \text{LayerNorm}(\boldsymbol{\Xi}\boldsymbol{W}^Q + \boldsymbol{\Psi}\boldsymbol{W}^\Psi), \\ \boldsymbol{Z}^* &= \text{LayerNorm}(\boldsymbol{Z} + \text{FFN}(\boldsymbol{Z})), \end{aligned} \tag{16}$$

where $\boldsymbol{W}^Q, \boldsymbol{W}^K, \boldsymbol{W}^V \in \mathbb{R}^{N_{gene}\times d}$, and $\boldsymbol{W}^\Psi \in \mathbb{R}^{d\times d}$ are trainable weight matrices. The resulting $\boldsymbol{Z}^*$ contains more expressive AC representations, on which a self-paced clustering [Li *et al.*, 2020] is conducted to cluster ACs into fine-grained groups. Cell i's soft assignment score to group j is calculated using a Cauchy kernel:

$$q_{i,j} = \frac{\left(1 + \|\boldsymbol{z}_i^* - \boldsymbol{\mu}_j\|^2/\nu\right)^{-1}}{\sum_{j'}\left(1 + \|\boldsymbol{z}_i^* - \boldsymbol{\mu}_{j'}\|^2/\nu\right)^{-1}}, \tag{17}$$

where ν denotes the degree of freedom. $\boldsymbol{\mu}_j$ is the centroid of group j, initialized by K-means clustering on $\boldsymbol{Z}^*$. The clustering loss function, a KL-divergence $\mathcal{L}_C$, is calculated on q and an auxiliary target distribution p, defined as:

$$p_{i,j} = \frac{q_{i,j}^2/\sum_i q_{i,j}}{\sum_j\left(q_{i,j}^2/\sum_i q_{i,j}\right)}, \tag{18}$$

$$\mathcal{L}_C = \sum_i\sum_j p_{i,j} log\left(\frac{p_{i,j}}{q_{i,j}}\right). \tag{19}$$

$\mathcal{L}_C$ overweights ACs with high-confident cluster assignments in updating the parameters of the cross-attention fusion block and the cluster centroids. Consequently, harder-to-cluster AC embeddings in $\boldsymbol{Z}^*$ are incrementally transformed into easier ones in iterations, which continues until the changes in anomalies' hard assignments fall below a threshold or a predetermined number of iterations is reached. The number of AC clusters is either known as a priori or automatically inferred (see Appendix B).

3.4 Theoretical Analysis

In this section, we provide a theoretical analysis on the existence of $\gamma_c(p_i, p_j)$ in equation (9) and the variation boundary of L_p across domains. Following the definitions and notations in section 3.1, we have the following theorem (detailed proof in Appendix A.1) :

Theorem 3.1. *Let m and n denote the actual numbers of inliers and anomalies in the target dataset, respectively. Let $\boldsymbol{\delta}_i \in \boldsymbol{\delta}_m^x \cup \boldsymbol{\delta}_n^\xi$, as defined in equation* (8)*. Define $s_i \coloneqq f_s(\boldsymbol{\delta}_i) \in \{0, 1\}$. The equation* (8) *can be rewritten as:*

$$\min_{f_s} \quad \mathcal{L}_p\left(\boldsymbol{\delta}_m^x, \boldsymbol{\delta}_n^\xi\right) = -\sum_{i}^{m+n}\sum_{j\neq i}^{m+n} k(\boldsymbol{\delta}_i, \boldsymbol{\delta}_j)\gamma(s_i, s_j), \quad (20)$$

where

$$\gamma(s_i, s_j) = \begin{cases} \frac{1}{m(m-1)}, & s_i = s_j = 0 \\ \frac{1}{n(n-1)}, & s_i = s_j = 1 \\ \frac{-1}{mn}, & s_i \neq s_j \end{cases} \quad (21)$$

If $s_i = 1$, instance i is annotated as anomalous, or normal otherwise.

To ensure a smooth backpropagation of the loss gradients, f_s is replaced with a continuous function $f_p : \mathbb{R}^p \to (0, 1)$, and $\gamma(s_i, s_j) : \{0, 1\} \times \{0, 1\} \to \mathbb{R}$ with a continuous mapping function $\gamma_c(p_i, p_j) : (0, 1) \times (0, 1) \to \mathbb{R}$, where $p_i \coloneqq f_p(\boldsymbol{\delta}_i)$ is the anomaly score for instance i. Theorem 3.2 guarantees the existence of γ_c (proof in Appendix A.2):

Theorem 3.2. *For any discrete mapping function $\gamma(s_i, s_j) : \{0, 1\} \times \{0, 1\} \to \mathbb{R}$, there exists a continuous function γ_c that satisfies:*

(1) $\forall s_i, s_j \in \{0, 1\}$, $\gamma_c(s_i, s_j) = \gamma(s_i, s_j)$.

(2) γ_c is analytic everywhere within its domain.

For γ in Equation (21)*, a potential γ_c is specified in Equation* (10)*.*

Theorem 3.3 addresses the bounded variation of $\mathcal{L}_p$ across domains (proof in Appendix A.3).

Theorem 3.3. *Given Assumption 3.1 and a linear kernel-induced MMD, the reconstruction deviations of m inliers ($\boldsymbol{\delta}_m^x$) and n anomalies ($\boldsymbol{\delta}_n^\xi$) in any domain satisfy:*

$$\begin{aligned} &\mathbb{P}\left(\left|\mathcal{L}_p(\boldsymbol{\delta}_m^x, \boldsymbol{\delta}_n^\xi) - \mathcal{L}_p(P_{\boldsymbol{\delta}^{x*}}, P_{\boldsymbol{\delta}^{\xi*}})\right| \geq \varepsilon\right) \\ &= \mathbb{P}\left(\left|MMD^2(\boldsymbol{\delta}_m^x, \boldsymbol{\delta}_n^\xi) - MMD^2(P_{\boldsymbol{\delta}^{x*}}, P_{\boldsymbol{\delta}^{\xi*}})\right| \geq \varepsilon\right) \\ &\leq \alpha \exp\left(-\frac{\beta C}{1+C} n\varepsilon^2\right), \end{aligned} \quad (22)$$

where $C = \frac{m}{n} \geq 1$, α, β are constants, ε denotes domain-associated variation of $\mathcal{L}_p$. $P_{\boldsymbol{\delta}^{x*}}$ and $P_{\boldsymbol{\delta}^{\xi*}}$ denote the domain-shift-free distributions of reconstruction deviations of inliers and anomalies, respectively.

Domain Shift Type	Ref Dataset	Target Dataset (Anomaly Type)	Experiment ID
Experimental Condition (**Type I**)	a-TME-0	a-TME-1 (Tumor)	1
	a-Brain-0	a-Brain-1 (Cerebellar granule)	2
		a-Brain-2 (Microglia)	3
		a-Brain-1 (Cerebellar granule) a-Brain-2 (Microglia)	4
	r-PBMC-0	r-PBMC-1 (B cell)	5
		r-PBMC-2 (Natural killer)	6
		r-PBMC-1 (B cell) r-PBMC-2 (Natural killer)	7
	r-Cancer-0	r-Cancer-1 (Epithelial, Immune tumor)	8
		r-Cancer-2 (Epithelial, Stromal tumor)	9
		r-Cancer-1 (Epithelial, Immune tumor) r-Cancer-2 (Epithelial, Stromal tumor)	10
	r-CSCC-0	r-CSCC-1 (Basal, Cycling, Keratinocyte tumor)	11
		r-CSCC-2 (Basal, Cyclng, Keratinocyte tumor)	12
		r-CSCC-1 (Basal, Cycling, Keratinocyte tumor) r-CSCC-2 (Basal, Cycling, Keratinocyte tumor) r-CSCC-3 (Basal, Cycling, Keratinocyte tumor)	13
Sequencing Technology (**Type II**)	r-Pan-0	r-Pan-1 (Delta, Acinar cell)	14
		r-Pan-2 (Delta, Acinar cell)	15
		r-Pan-1 (Delta, Acinar cell) r-Pan-2 (Delta, Acinar cell)	16
Data Type (**Type III**)	r-PBMC-0	a-TME-1 (Tumor)	17
Protocol Type (**Type IV**)	KDDRev-0	KDDRev-1 (DOS,PROBING) KDDRev-2 (DOS,R2L,U2R,PROBING)	18

Table 1: Experimental scenarios. For experiments 1-17, the dataset name is formatted as (data type)-(data name)-(dataset ID). For data type, a and r denote scATAC-seq and scRNA-seq, respectively. Dataset ID 0 is reserved for reference datasets.

4 Experiments

4.1 Experimental Settings

Datasets. scRNA-seq datasets used in this study include three 10x Peripheral Blood Mononuclear Cell (PBMC) datasets, three 10x lung cancer (Cancer) datasets, four 10x Cutaneous Squamous Cell Carcinoma (CSCC) datasets, and three Pancreatic (Pan) datasets sequenced with different technologies (InDrop, Smart-Seq2, and CEL-Seq2). scATAC-seq datasets include two 10x Human Tumor Microenvironment (TME) datasets and three 10x Mouse Brain (Brain) datasets. We use the KDDRev dataset for fine-grained cyber-intrusion detection. See Appendix C for more details.

Experimental Scenarios. Experiments are carefully designed for scenarios regarding DS types, target dataset quantity, and dataset-specific AC types (table 1). For SC data (i.e., experiments 1-17), DS is categorized into three types based on their origins: variations in experimental conditions (**Type I**), sequencing technologies (**Type II**), and data types (**Type III**). For cyber-intrusion detection data (i.e., experiment 18), DS stems from variations in connection protocol types (**Type IV**). Experiments 1-3, 5-6, 8-9, 11-12, 14-15, and 17 focus on a single target dataset. In cases involving more than one target dataset, experiments 13 and 16 feature target datasets with identical AC types, experiments 10 and 18 include both shared and dataset-specific anomaly types, while experiments 4 and 7 are set up with exclusively dataset-specific AC types.

Method	Experiment ID																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
SLAD	.52(.02)	.70(.00)	.63(.01)	.58(.01)	.77(.01)	**.88(.01)**	.82(.01)	.93(.00)	.92(.01)	.93(.01)	.59(.01)	.68(.02)	.60(.02)	.78(.02)	.81(.04)	.62(.01)	.49(.03)
ICL	.78(.06)	.70(.01)	.54(.01)	.50(.01)	.65(.03)	.72(.04)	.68(.03)	.93(.01)	.93(.01)	.93(.01)	.37(.01)	.33(.01)	.14(.02)	.77(.02)	.84(.01)	.62(.01)	.50(.05)
NeuTraL	.49(.02)	.67(.01)	.70(.02)	.56(.01)	.71(.01)	.74(.02)	.71(.01)	.96(.01)	.95(.03)	.95(.01)	.25(.05)	.20(.04)	.19(.04)	.53(.04)	.47(.06)	.51(.05)	.59(.07)
RCA	.37(.04)	.69(.02)	.77(.03)	.69(.01)	.72(.01)	.79(.01)	.77(.02)	.84(.03)	.75(.05)	.81(.04)	.33(.02)	.33(.03)	.21(.01)	.66(.04)	.72(.02)	.81(.02)	.37(.04)
CAMLU	.66(.12)	.69(.02)	**.79(.09)**	.55(.02)	.70(.05)	.82(.04)	.71(.05)	.90(.00)	.85(.13)	.88(.01)	.80(.02)	**.79(.00)**	**.78(.01)**	.55(.06)	.62(.03)	.53(.05)	.58(.00)
ALAD	.36(.18)	.40(.09)	.46(.18)	.45(.08)	.42(.05)	.33(.08)	.33(.02)	.22(.07)	.16(.08)	.20(.08)	.53(.09)	.48(.13)	.51(.13)	.33(.04)	.41(.05)	.38(.04)	.60(.07)
DeepSVDD	.57(.02)	.55(.01)	.61(.02)	.51(.01)	.62(.02)	.42(.02)	.48(.02)	.44(.14)	.64(.04)	.56(.08)	.47(.03)	.53(.02)	.50(.02)	.62(.01)	.43(.06)	.53(.03)	.62(.03)
scmap	.50(.00)	.59(.00)	.56(.00)	.57(.00)	.70(.00)	.83(.00)	.76(.00)	.53(.00)	.47(.00)	.50(.00)	.57(.00)	.65(.00)	.67(.00)	.77(.00)	.54(.00)	.78(.00)	.50(.00)
ACSleuth	**.83(.02)**	**.74(.02)**	.72(.03)	**.71(.01)**	**.78(.03)**	.85(.00)	**.83(.02)**	**.97(.01)**	**.95(.02)**	**.96(.01)**	**.81(.02)**	.70(.02)	.69(.03)	**.85(.02)**	**.93(.03)**	**.88(.03)**	**.74(.03)**

Table 2: The AC detection performance across 17 experiments is evaluated using averaged AUC scores with standard deviations indicated in parentheses. The best and second best scores for each experiment are **bolded** and underlined, respectively.

Method	Experiment ID										
	4	7	8	9	10	11	12	13	14	15	16
scGAD	.18(.01)	.13(.08)	.51(.02)	.55(.00)	.42(.01)	.27(.01)	.18(.00)	.16(.01)	.11(.01)	.16(.01)	.13(.01)
MARS	.19(.01)	.28(.02)	.34(.00)	.40(.00)	.28(.02)	.23(.01)	.24(.01)	.23(.01)	.21(.01)	.27(.02)	.20(.01)
SLAD-EDESC	.19(.06)	.27(.13)	.61(.28)	.64(.05)	.36(.14)	.17(.02)	.05(.04)	.05(.04)	.10(.05)	.15(.10)	.17(.10)
SLAD-DFCN	.26(.01)	.41(.02)	.71(.02)	.60(.01)	.37(.04)	.08(.03)	.04(.02)	.05(.03)	.05(.04)	.29(.04)	.02(.01)
SLAD-scTAG	.07(.05)	.10(.05)	.53(.05)	.50(.15)	.23(.07)	.20(.02)	.03(.02)	.02(.01)	.03(.01)	.11(.01)	.16(.05)
SLAD-Leiden	.11(.01)	.25(.02)	.10(.04)	.07(.03)	.27(.10)	.20(.00)	.13(.01)	.02(.00)	.20(.02)	.31(.03)	.13(.01)
SLAD-Seurat V5	.11(.01)	.40(.02)	.31(.02)	.27(.01)	.31(.01)	.22(.01)	.04(.01)	.03(.00)	.02(.01)	.00(.00)	.00(.00)
ACSleuth	**.36(.03)**	**.47(.06)**	**.79(.02)**	**.69(.03)**	**.74(.03)**	**.61(.02)**	**.52(.07)**	**.58(.04)**	**.31(.07)**	**.65(.04)**	**.43(.06)**

Table 3: The overall FACD performance across 11 experiments, evaluated using average F1×NMI scores with standard deviations indicated in parentheses. The best and second best scores for each experiment are **bolded** and underlined, respectively.

Baselines. The baselines for AC detection cover the SOTA methods across three major categories: contrastive methods (ICL [Shenkar and Wolf, 2021] and NeuTraL [Qiu *et al.*, 2021]), generative methods (CAMLU [Li *et al.*, 2022], RCA [Liu *et al.*, 2021], and ALAD [Zenati *et al.*, 2018]), and one-class classification method (DeepSVDD [Ruff *et al.*, 2018]). We also include SLAD [Xu *et al.*, 2023], a cutting-edge scale-learning method, and scmap [Kiselev *et al.*, 2018], an uncertainty-based method tailored for scRNA-seq. For overall FACD, the baselines encompass five composite methods that combine SLAD for AC detection with EDESC [Cai *et al.*, 2022], DFCN [Tu *et al.*, 2021], scTAG [Yu *et al.*, 2022], Leiden [Traag *et al.*, 2019], and Seurat V5 [Hao *et al.*, 2023] for fine-grained AC annotation, where the first three are deep clustering methods, and the last two are for SC clustering. We also include MARS [Brbić *et al.*, 2020] and scGAD [Zhai *et al.*, 2023b], the only two methods available that integrate AC detection and fine-grained annotation. For fine-grained cyber-intrusion detection, MARS, scGAD, and three composite methods, SLAD-EDESC, SLAD-DFCN, and SLAD-Leiden, serve as baselines.

Evaluation Protocols. Both the AUC and F1 scores are used to evaluate AD accuracy. For a fair comparison across methods, the F1 score is calculated with a deliberately chosen anomaly score threshold such that the number of samples exceeding this threshold (i.e., labeled as anomalies) matches the actual number of true anomalies [Shenkar and Wolf, 2021]. The normalized mutual information (NMI) is used to evaluate fine-grained anomaly annotation in both SC and cyber-intrusion data. As the overall performance of anomaly analysis hinges on both anomaly detection and fine-grained annotation, it is evaluated using the product of the F1 and NMI scores. The reported metrics represent averaged results, along with standard deviations obtained over ten independent runs.

Implementation Details. GAN models in *module I* and *module II* share the same architecture—the encoder: Input-512-256-256-256-256-256; the decoder is symmetric to the encoder; the discriminator: Input-512-64-64-64. The MMD-based AC scorer in *module I* is a three-layer MLP with an Input-512-256-1 configuration. The memory block in *module I* has a 512×256 dimension. The style block in *module II* has a $N_{tgt} \times 256$ dimension, where N_{tgt} is the number of target datasets. The weight parameters α, β, and λ in GAN's loss functions are set to 50, 1, and 10, respectively. The cross-attention block in Phase III includes two attention heads of dimension 256. The mini-batch size of two GAN models is 256, while the MMD-based AC scorer and the clustering module utilize the full batch. The training is conducted using an Adam optimizer with a default learning rate of 3e-4.

4.2 Results

Anomalous Cell Detection. Table 2 showcases ACSleuth's superiority over the competing methods in accurate AC detection across various scenarios (Table 1). ACSleuth ranks first 13 times, and among the top two performers 16 times. Significantly, ACSleuth consistently leads in experiments (4, 7, and 10) involving multiple target datasets and dataset-specific AC types. Moreover, the addition of a new target dataset (a-Brain-1) with its unique AC type in experiment 4, compared to experiment 3, highlights ACSleuth as the only method that

maintains its performance level amidst the newly introduced DS and dataset-specific AC type. Furthermore, ACSleuth significantly outperforms other methods in experiments 14-17 featuring DS (**Type II** and **III**), with an average leap of 11.92% in AUC scores. This underscores the essential role of domain adaptation in ensuring accurate AC detection in such complex scenarios. Collectively, these findings affirm ACSleuth's efficacy in AD, especially in scenarios involving significant DS and dataset-specific AC types.

Overall Performance of Fine-grained Anomalous Cell Detection. Here, the overall performance of ACSleuth in FACD is compared to the baseline methods, using the product of F1 (for AC detection) and NMI (for fine-grained AC annotating) scores as the evaluation metric. Table 3 showcases that ACSleuth consistently outperforms the top-performing baseline methods across 11 experiments, achieving an average leap of 74.13% in F1×NMI scores. These results reveal several specific strengths of our method: Experiments 4 and 7-13 demonstrate ACSleuth's excelling in handling DS **Type I**, while experiments 14-16 showcase its robustness against DS **Type II**. Moreover, experiments 4, 7, and 10 highlight ACSleuth's ability to manage complications arising from sample-specific AC types. In contrast, all baseline methods exhibit suboptimal performance, primarily due to their lack of a domain adaptation step. Although scGAD integrates MNN for domain adaptation, its performance is compromised in scenarios involving sample-specific AC types (see **Case ii** in Figure 1), as indicated in experiments 4 and 7. Finally, ACSleuth, scGAD, and MARS, which unify the AC detection and fine-grained annotation, emerge as at least two among the top three performers in seven out of the eleven experiments, suggesting the importance of maintaining methodological coherence of the workflow.

Experiment ID	Metrics	*ACSleuth*	*scGAD*	*MARS*	*SLAD*		
					EDESC	*DFCN*	*Leiden*
18	F1	.72(.02)	.41(.08)	.70(.01)	.44(.02)		
	NMI	.73(.02)	.49(.05)	.70(.01)	.47(.06)	.32(.04)	.50(.00)
	F1×NMI	.53(.03)	.20(.04)	.49(.01)	.20(.02)	.15(.01)	.22(.01)

Table 4: The detection and differentiation of cyber-intrusions are evaluated using F1 and NMI scores, respectively. Overall performance is measured in F1×NMI scores. Each cell displays the average score of 10 runs, with standard deviation noted in parenthesis.

Fine-grained Cyber-intrusion Detection. Here, we assess ACSleuth's applicability to fine-grained cyber-intrusion detection in the presence of DS rooted in variations in connection protocol types. The reference dataset includes UDP records only, while the two target datasets contain ICMP and TCP records, respectively. Table 4 showcases ACSleuth outperforms the competing methods in detecting and distinguishing various types of cyber-intrusions in terms of F1×NMI scores. Intriguingly, since R2L and U2R anomaly types are unique to KDDRev-2, using domain adaptation methods like MNN, which are sensitive to complications caused by dataset-specific anomaly types, can deteriorate performance in both AD and fine-grained annotation (see **Case ii** in Figure 1). This is supported by scGAD's low-performance ranking in both tasks. Conversely, ACSleuth's superior performance suggests its domain adaptation strategy is robust against such complications. Furthermore, these results also indicate ACSleuth's greater versatility, compared to MARS and scGAD, in handling general tabular data.

4.3 Ablation Study

A series of ablation studies are conducted to evaluate the roles of ACSleuth's key components in AC detection and fine-grained annotating (Table 5). The MMD-based anomaly scorer's efficacy is assessed by substituting it with two other commonly used anomaly scoring functions in reconstruction deviation-guided generative models [Schlegl *et al.*, 2017; Zenati *et al.*, 2018]: $F_G(\boldsymbol{x}_i) \coloneqq \|\boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i\|_2$ and $F_D(\boldsymbol{x}_i) \coloneqq \|D^I(\boldsymbol{x}_i) - D^I(\widehat{\boldsymbol{x}}_i)\|_2$, where $\widehat{\boldsymbol{x}}_i$ is the reconstructed $\boldsymbol{x}_i$ by *module I*. Our scorer outperforms F_G and F_D in detecting ACs, as evidenced by its average AUC improvement of 12.07% over F_G and 22.16% over F_D. Removing the memory block in *module I* (w/o MB) leads to a notable decrease in anomaly detection accuracy, averaging a 19.89% reduction in AUC scores. The essential role of the domain adaptation step in Phase II (w/o DA) in ensuring accurate fine-grained AC annotation is reflected by the observation that its omission leads to an average decline of 36.49% in F1×NMI scores. Lastly, using solely domain-adapted cell embeddings (w/o FB), rather than their fusions with reconstruction deviations, for the clustering in Phase III reduces F1×NMI scores by 14.84% on average.

Task	Ablation	Experiment ID			
		4	**10**	**13**	**16**
ACD	w/ F_G	.59(.02)	.88(.03)	.66(.01)	.77(.02)
	w/ F_D	.64(.05)	.58(.08)	.65(.05)	.83(.08)
	w/o MB	.67(.01)	.89(.03)	.61(.02)	.76(.02)
	Full	.71(.01)	.96(.04)	.69(.03)	.88(.03)
FACD	w/o DA	.22(.06)	.50(.11)	.39(.09)	.25(.08)
	w/o FB	.34(.02)	.65(.07)	.46(.05)	.34(.03)
	Full	.36(.03)	.74(.03)	.58(.04)	.43(.06)

Table 5: The upper table shows ACD results in average AUC scores. The lower table shows overall FACD results in average F1×NMI scores. "Full" is the complete ACSleuth model. "w/ F_G" and "w/ F_D" substitute the MMD-based scorer with the F_G and F_D scoring functions, respectively. "w/o MB" removes the memory block from *module I*. "w/o DA" omits the domain adaptation phase. "w/o FB" uses domain-adapted cell embeddings only for clustering.

5 Conclusion

In this paper, We propose ACSleuth, a novel method for FACD in multi-sample and multi-domain contexts. ACSleuth provides a generative framework that integrates AC detection, domain adaptation, and fine-grained annotation into a methodological cohesive workflow. Our theoretical analysis corroborates ACSleuth's resilience against DS. Extensive experiments, designed for scenarios involving various real datasets and DS types, have shown ACSleuth's superiority over the SOTA methods in identifying and differentiating ACs. Our method is also versatile for tabular data types beyond SC.

Acknowledgments

The project is funded by the Excellent Young Scientist Fund of Wuhan City (Grant No. 21129040740) to X.S.

References

[Aevermann *et al.*, 2018] Brian D. Aevermann, Mark Novotny, Trygve E Bakken, Jeremy A. Miller, Alexander D. Diehl, David Osumi-Sutherland, Roger S. Lasken, Ed S. Lein, and Richard H. Scheuermann. Cell type discovery using single-cell transcriptomics: implications for ontological representation. *Human Molecular Genetics*, 27:R40 – R47, 2018.

[Alquicira-Hernandez *et al.*, 2019] Jose Alquicira-Hernandez, Anuja Sathe, Hanlee P Ji, Quan Nguyen, and Joseph E Powell. scpred: accurate supervised method for cell-type classification from single-cell rna-seq data. *Genome Biology*, 20(1):1–17, 2019.

[Amdeberhan *et al.*, 2012] Tewodros Amdeberhan, Olivier Espinosa, Ivan Gonzalez, Marshall Harrison, Victor H Moll, and Armin Straub. Ramanujan's master theorem. *The Ramanujan Journal*, 29:103–120, 2012.

[Baron *et al.*, 2016] Maayan Baron, Adrian Veres, Samuel L Wolock, Aubrey L Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K Wagner, Shai S Shen-Orr, Allon M 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.

[Bischoff *et al.*, 2022] Philip Bischoff, Alexandra Trinks, Jennifer Wiederspahn, Benedikt Obermayer, Jan Patrick Pett, Philipp Jurmeister, Aron Elsner, Tomasz Dziodzio, Jens-Carsten Rückert, Jens Neudecker, et al. The single-cell transcriptional landscape of lung carcinoid tumors. *International Journal of Cancer*, 150(12):2058–2071, 2022.

[Bradshaw and Vignat, 2023] Zachary P Bradshaw and Christophe Vignat. An operational calculus generalization of ramanujan's master theorem. *Journal of Mathematical Analysis and Applications*, 523(2):127029, 2023.

[Brbić *et al.*, 2020] Maria Brbić, Marinka Zitnik, Sheng Wang, Angela O Pisco, Russ B Altman, Spyros Darmanis, and Jure Leskovec. Mars: discovering novel cell types across heterogeneous single-cell experiments. *Nature Methods*, 17(12):1200–1206, 2020.

[Cai *et al.*, 2022] Jinyu Cai, Jicong Fan, Wenzhong Guo, Shiping Wang, Yunhe Zhang, and Zhao Zhang. Efficient deep embedded subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2022.

[Chen *et al.*, 2020] Liang Chen, Yuyao Zhai, Qiuyan He, Weinan Wang, and Minghua Deng. Integrating deep supervised, self-supervised and unsupervised learning for single-cell rna-seq clustering and annotation. *Genes*, 11(7):792, 2020.

[Cheng *et al.*, 2023] Yue Cheng, Yanchi Su, Zhuohan Yu, Yanchun Liang, Ka-Chun Wong, and Xiangtao Li. Unsupervised deep embedded fusion representation of single-cell transcriptomics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5036–5044, 2023.

[Cusanovich *et al.*, 2018] Darren A Cusanovich, Andrew J Hill, Delasa Aghamirzaie, Riza M Daza, Hannah A Pliner, Joel B Berletch, Galina N Filippova, Xingfan Huang, Lena Christiansen, William S DeWitt, et al. A single-cell atlas of in vivo mammalian chromatin accessibility. *Cell*, 174(5):1309–1324, 2018.

[De Kanter *et al.*, 2019] Jurrian K De Kanter, Philip Lijnzaad, Tito Candelli, Thanasis Margaritis, and Frank CP Holstege. Chetah: a selective, hierarchical cell type identification method for single-cell rna sequencing. *Nucleic Acids Research*, 47(16):e95–e95, 2019.

[Fernández-Saúco *et al.*, 2019] Igr Alexánder Fernández-Saúco, Niusvel Acosta-Mendoza, Andrés Gago-Alonso, and Edel Bartolo García-Reyes. Computing anomaly score threshold with autoencoders pipeline. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 23rd Iberoamerican Congress, CIARP 2018, Madrid, Spain, November 19-22, 2018, Proceedings 23*, pages 237–244. Springer, 2019.

[Gong *et al.*, 2019] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1705–1714, 2019.

[Gretton *et al.*, 2012] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[Gulrajani *et al.*, 2017] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in Neural Information Processing Systems*, 30, 2017.

[Haghverdi *et al.*, 2018] Laleh Haghverdi, Aaron TL Lun, Michael D Morgan, and John C Marioni. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nature Biotechnology*, 36(5):421–427, 2018.

[Hao *et al.*, 2023] Yuhan Hao, Tim Stuart, Madeline H Kowalski, Saket Choudhary, Paul Hoffman, Austin Hartman, Avi Srivastava, Gesmira Molla, Shaista Madad, Carlos Fernandez-Granda, et al. Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nature Biotechnology*, pages 1–12, 2023.

[Hettich, 1999] Seth Hettich. The uci kdd archive. *http://kdd. ics. uci. edu*, 1999.

[Hoeffding, 1994] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *The Collected Works of Wassily Hoeffding*, pages 409–426, 1994.

[Holmström *et al.*, 2022] Susanna Holmström, Sampsa Hautaniemi, and Antti Häkkinen. POIBM: batch correction of heterogeneous RNA-seq datasets through latent sample matching. *Bioinformatics*, 38(9):2474–2480, 02 2022.

[Hornung *et al.*, 2016] Roman Hornung, Anne-Laure Boulesteix, and David Causeur. Combining location-and-scale batch effect adjustment with data cleaning by latent factor adjustment. *BMC Bioinformatics*, 17:1–19, 2016.

[Ji *et al.*, 2020] Andrew L Ji, Adam J Rubin, Kim Thrane, Sizun Jiang, David L Reynolds, Robin M Meyers, Margaret G Guo, Benson M George, Annelie Mollbrink, Joseph Bergenstråhle, et al. Multimodal analysis of composition and spatial architecture in human squamous cell carcinoma. *Cell*, 182(2):497–514, 2020.

[Kiselev *et al.*, 2018] Vladimir Yu Kiselev, Andrew Yiu, and Martin Hemberg. scmap: projection of single-cell rna-seq data across data sets. *Nature Methods*, 15(5):359–362, 2018.

[Li *et al.*, 2020] Xiangjie Li, Kui Wang, Yafei Lyu, Huize Pan, Jingxiao Zhang, Dwight Stambolian, Katalin Susztak, Muredach P Reilly, Gang Hu, and Mingyao Li. Deep learning enables accurate clustering with batch effect removal in single-cell rna-seq analysis. *Nature Communications*, 11(1):2338, 2020.

[Li *et al.*, 2022] Ziyi Li, Yizhuo Wang, Irene Ganan-Gomez, Simona Colla, and Kim-Anh Do. A machine learning-based method for automatically identifying novel cells in annotating single-cell rna-seq data. *Bioinformatics*, 38(21):4885–4892, 2022.

[Liu *et al.*, 2021] Boyang Liu, Ding Wang, Kaixiang Lin, Pang-Ning Tan, and Jiayu Zhou. Rca: A deep collaborative autoencoder approach for anomaly detection. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, 2021.

[Macnair and Robinson, 2023] Will Macnair and Mark Robinson. Sampleqc: robust multivariate, multi-cell type, multi-sample quality control for single-cell data. *Genome Biology*, 24(1):23, 2023.

[Petegrosso *et al.*, 2020] Raphael Petegrosso, Zhuliu Li, and Rui Kuang. Machine learning and statistical methods for clustering single-cell rna-sequencing data. *Briefings in Bioinformatics*, 21(4):1209–1223, 2020.

[Pouyan *et al.*, 2016] Maziyar Baran Pouyan, Vasu Jindal, Javad Birjandtalab, and Mehrdad Nourani. Single and multi-subject clustering of flow cytometry data for cell-type identification and anomaly detection. *BMC Medical Genomics*, 9(2):99–110, 2016.

[Qiu *et al.*, 2021] Chen Qiu, Timo Pfrommer, Marius Kloft, Stephan Mandt, and Maja Rudolph. Neural transformation learning for deep anomaly detection beyond images. In *International Conference on Machine Learning*, pages 8703–8714. PMLR, 2021.

[Ruff *et al.*, 2018] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on Machine Learning*, pages 4393–4402. PMLR, 2018.

[Satpathy *et al.*, 2019] Ansuman T Satpathy, Jeffrey M Granja, Kathryn E Yost, Yanyan Qi, Francesca Meschi, Geoffrey P McDermott, Brett N Olsen, Maxwell R Mumbach, Sarah E Pierce, M Ryan Corces, et al. Massively parallel single-cell chromatin landscapes of human immune cell development and intratumoral t cell exhaustion. *Nature Biotechnology*, 37(8):925–936, 2019.

[Schlegl *et al.*, 2017] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pages 146–157. Springer, 2017.

[Shenkar and Wolf, 2021] Tom Shenkar and Lior Wolf. Anomaly detection for tabular data with internal contrastive learning. In *International Conference on Learning Representations*, 2021.

[Traag *et al.*, 2019] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1):5233, 2019.

[Tu *et al.*, 2021] Wenxuan Tu, Sihang Zhou, Xinwang Liu, Xifeng Guo, Zhiping Cai, En Zhu, and Jieren Cheng. Deep fusion clustering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9978–9987, 2021.

[Xie *et al.*, 2016] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, pages 478–487. PMLR, 2016.

[Xu *et al.*, 2023] Hongzuo Xu, Yijie Wang, Juhui Wei, Songlei Jian, Yizhou Li, and Ning Liu. Fascinating supervisory signals and where to find them: Deep anomaly detection with scale learning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[Yang *et al.*, 2020] Yuchen Yang, Gang Li, Huijun Qian, Kirk C Wilhelmsen, Yin Shen, and Yun Li. SMNN: batch effect correction for single-cell RNA-seq data via supervised mutual nearest neighbor detection. *Briefings in Bioinformatics*, 22(3):bbaa097, 06 2020.

[You *et al.*, 2018] Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. GraphRNN: Generating realistic graphs with deep auto-regressive models. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5708–5717. PMLR, 10–15 Jul 2018.

[Yu *et al.*, 2022] Zhuohan Yu, Yifu Lu, Yunhe Wang, Fan Tang, Ka-Chun Wong, and Xiangtao Li. Zinb-based graph

embedding autoencoder for single-cell rna-seq interpretations. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 36, pages 4671–4679, 2022.

[Yu *et al.*, 2023a] Xiaokang Yu, Xinyi Xu, Jingxiao Zhang, and Xiangjie Li. Batch alignment of single-cell transcriptomics data using deep metric learning. *Nature Communications*, 14(1):960, 2023.

[Yu *et al.*, 2023b] Xiaokang Yu, Xinyi Xu, Jingxiao Zhang, and Xiangjie Li. Batch alignment of single-cell transcriptomics data using deep metric learning. *Nature Communications*, 14(1):960, 2023.

[Zenati *et al.*, 2018] Houssam Zenati, Manon Romain, Chuan-Sheng Foo, Bruno Lecouat, and Vijay Chandrasekhar. Adversarially learned anomaly detection. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 727–736. IEEE, 2018.

[Zhai *et al.*, 2023a] Yuyao Zhai, Liang Chen, and Min Deng. Realistic cell type annotation and discovery for single-cell rna-seq data. In *International Joint Conference on Artificial Intelligence*, 2023.

[Zhai *et al.*, 2023b] Yuyao Zhai, Liang Chen, and Min Deng. scgad: a new task and end-to-end framework for generalized cell type annotation and discovery. *Briefings in Bioinformatics*, 2023.

[Zhao *et al.*, 2020] Sicheng Zhao, Bo Li, Pengfei Xu, and Kurt Keutzer. Multi-source domain adaptation in the deep learning era: A systematic survey. *arXiv preprint arXiv:2002.12169*, 2020.

[Zheng *et al.*, 2017] GXY Zheng, JM Terry, P Belgrader, P Ryvkin, ZW Bent, R Wilson, SB Ziraldo, TD Wheeler, GP McDermott, J Zhu, et al. Massively parallel digital transcriptional profiling of single cells. nat commun 8: 14049. *Data Set5. Putative master regulators in the trans-eQTL (expression quantitative trait loci) hotspots Figure S*, 1, 2017.

[Zhu *et al.*, 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

A Theorem Proofs

A.1 Proof of Theorem 3.1

Proof. Gretton et al [Gretton *et al.*, 2012] provide an unbiased empirical MMD for samples:

$$\begin{aligned} & MMD^2\left(\boldsymbol{\delta}_m^x, \boldsymbol{\delta}_n^\xi\right) \\ &= \frac{1}{m(m-1)} \sum_i^m \sum_{j \neq i}^m k(\boldsymbol{\delta}_i^x, \boldsymbol{\delta}_j^x) + \frac{1}{n(n-1)} \sum_i^n \sum_{j \neq i}^n k(\boldsymbol{\delta}_i^\xi, \boldsymbol{\delta}_j^\xi) - \frac{2}{mn} \sum_i^m \sum_j^n k(\boldsymbol{\delta}_i^x, \boldsymbol{\delta}_j^\xi) \\ &= \sum_i^{m+n} \sum_{j \neq i}^{m+n} k(\boldsymbol{\delta}_i, \boldsymbol{\delta}_j) \gamma(s_i, s_j), \end{aligned} \tag{23}$$

where the adjustment coefficient γ is defined as:

$$\gamma(s_i, s_j) = \begin{cases} \frac{1}{m(m-1)}, & s_i = s_j = 0. \\ \frac{1}{n(n-1)}, & s_i = s_j = 1. \\ \frac{-1}{mn}, & s_i \neq s_j. \end{cases} \tag{24}$$

If $s_i = 1$, instance i is annotated as anomalous, or normal otherwise. □

A.2 Proof of Theorem 3.2

Proof. We define the two-dimensional sequence $\{\gamma_{m,n}\}_{m,n \in \mathbb{Z}}$ as:

$$\gamma_{0,0} = \frac{1}{m(m-1)}, \quad \gamma_{0,1} = \gamma_{1,0} = \frac{-1}{mn}, \quad \gamma_{1,1} = \frac{1}{n(n-1)}, \quad \forall i, j \geq 2, \gamma_{i,j} = 0 \tag{25}$$

The ordinary generating function $H(x, y)$ for $\gamma_{m,n}$ is:

$$H(x, y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \gamma_{i,j} x^i y^j, \tag{26}$$

where $(x, y) \in \mathbb{D} \coloneqq [0, 1] \times [0, 1]$. In this case, $H(x, y)$ is a formal power series, and γ_{ij} corresponds to the coefficients of $x^i y^j$. Note that $H(-x, -y)$ satisfies:

$$H(-x, -y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \gamma_{ij} \Gamma(i+1) \Gamma(j+1) \frac{(-1)^i (-1)^j}{i! j!} x^i y^j. \tag{27}$$

By Lemma A.1, which is the extension of Ramanujan's master theorem in the k-dimensional case [Bradshaw and Vignat, 2023; Amdeberhan *et al.*, 2012], the k-dimensional Mellin transform follows:

$$\begin{aligned} \mathcal{M}[H(-x, -y)](s, t) &\coloneqq \int_{\mathbb{D}} x^{s-1} y^{t-1} H(-x, -y) dx dy \\ &= \Gamma(s) \Gamma(t) \Gamma(1-s) \Gamma(1-t) \gamma_c(-s, -t) \\ &= \frac{\pi^2}{\sin \pi s \sin \pi t} \gamma_c(-s, -t), \end{aligned} \tag{28}$$

where $\gamma_c(s, t)$ is exactly the extension of sequence γ_{ij} in the continuous scenario. By solving the definite integral, we can obtain:

$$\begin{aligned} \gamma_c(-s, -t) &= \frac{\sin \pi s \sin \pi t}{\pi^2} \int_{\mathbb{D}} x^{s-1} y^{t-1} H(-x, -y) dx dy \\ &= \frac{\sin \pi s \sin \pi t}{\pi^2} \int_{\mathbb{D}} \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \gamma_{ij} x^{i+s-1} y^{j+t-1} dx dy \\ &= \frac{\sin \pi s \sin \pi t}{\pi^2} \int_{\mathbb{D}} (\gamma_{0,0} x^{s-1} y^{t-1} + \gamma_{1,0} x^s y^{t-1} + \gamma_{0,1} x^{s-1} y^t + \gamma_{1,1} x^s y^t) dx dy \\ &= \frac{\sin \pi s \sin \pi t}{\pi^2} \left(\gamma_{0,0} \frac{x^s}{s} \Big|_0^1 \frac{y^t}{t} \Big|_0^1 + \gamma_{1,0} \frac{x^{s+1}}{s+1} \Big|_0^1 \frac{y^t}{t} \Big|_0^1 + \gamma_{0,1} \frac{x^s}{s} \Big|_0^1 \frac{y^{t+1}}{t+1} \Big|_0^1 + \gamma_{1,1} \frac{x^{s+1}}{s+1} \Big|_0^1 \frac{y^{t+1}}{t+1} \Big|_0^1 \right) \\ &= \frac{\sin \pi s \sin \pi t}{\pi^2} \left(\frac{[m(m-1)]^{-1}}{st} + \frac{(mn)^{-1}}{(s+1)t} + \frac{(mn)^{-1}}{s(t+1)} + \frac{[n(n-1)]^{-1}}{(s+1)(t+1)}. \right) \end{aligned} \tag{29}$$

By replacing $-s$ and $-t$, **??** is proven. □

A.3 Proof of Theorem 3.3

Proof. Let S^r and S^t denote the sample sets in reference domain r and target domain t, respectively. $\boldsymbol{\zeta} \in S^r$ denote a reference sample. Given a proper reconstruction function $G(\cdot)$ trained on S^r exclusively, under Assumption 3.1 in the main text, $\boldsymbol{\zeta}$ and its reconstruction $\widehat{\boldsymbol{\zeta}}$ satisfy:

$$\begin{cases} \boldsymbol{\zeta} = \boldsymbol{\zeta}^* + \boldsymbol{b}^r + \boldsymbol{\epsilon}, \\ \widehat{\boldsymbol{\zeta}} \coloneqq G(\boldsymbol{\zeta}) = \widehat{\boldsymbol{\zeta}}^* + \boldsymbol{b}^r, \end{cases} \tag{30}$$

where $\widehat{\boldsymbol{\zeta}}^* \sim P_{\widehat{\boldsymbol{\zeta}}^*}, \boldsymbol{b}^r \sim P_{\boldsymbol{b}^r}$. The random error term $\boldsymbol{\epsilon}$ is ignored in $\widehat{\boldsymbol{\zeta}}$, as it cannot be learned by $G(\cdot)$. Given an inlier $\boldsymbol{x}_i$ and anomaly $\boldsymbol{\xi}_j$ in S^t, $\exists \boldsymbol{\zeta}_i, \boldsymbol{\zeta}_j \in S^r$ whose reconstructions by $G(\cdot)$ are equivalent to those of $\boldsymbol{x}_i$ and $\boldsymbol{\xi}_j$:

$$\begin{cases} \boldsymbol{x}_i = \boldsymbol{x}_i^* + \boldsymbol{b}^t + \boldsymbol{\epsilon}_i, \\ \widehat{\boldsymbol{x}}_i = G(\boldsymbol{x}_i) = G(\boldsymbol{\zeta}_i) = \widehat{\boldsymbol{\zeta}}_i^* + \boldsymbol{b}^r, \\ \boldsymbol{\xi}_j = \boldsymbol{\xi}_j^* + \boldsymbol{b}^t + \boldsymbol{\epsilon}_j, \\ \widehat{\boldsymbol{\xi}}_j = G(\boldsymbol{\xi}_j) = G(\boldsymbol{\zeta}_j) = \widehat{\boldsymbol{\zeta}}_j^* + \boldsymbol{b}^r, \end{cases} \tag{31}$$

The reconstruction deviations of $\boldsymbol{x}_i$ and $\boldsymbol{\xi}_j$ satisfy:

$$\begin{cases} \boldsymbol{\delta}_i^x = \boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i = \boldsymbol{x}_i^* - \widehat{\boldsymbol{\zeta}}_i^* + \boldsymbol{b}^t - \boldsymbol{b}^r + \boldsymbol{\epsilon}_i, \\ \boldsymbol{\delta}_j^\xi = \boldsymbol{\xi}_j - \widehat{\boldsymbol{\xi}}_j = \boldsymbol{\xi}_j^* - \widehat{\boldsymbol{\zeta}}_j^* + \boldsymbol{b}^t - \boldsymbol{b}^r + \boldsymbol{\epsilon}_j. \end{cases} \tag{32}$$

For a more concise representation, we define:

$$\begin{cases} \boldsymbol{\delta}_i^{x*} = \boldsymbol{x}_i^* - \widehat{\boldsymbol{\zeta}}_i^*, \quad \boldsymbol{\delta}_i^b = \boldsymbol{b}^t - \boldsymbol{b}^r + \boldsymbol{\epsilon}_i. \\ \boldsymbol{\delta}_j^{\xi*} = \boldsymbol{\xi}_j^* - \widehat{\boldsymbol{\zeta}}_j^*, \quad \boldsymbol{\delta}_j^b = \boldsymbol{b}^t - \boldsymbol{b}^r + \boldsymbol{\epsilon}_j. \end{cases} \tag{33}$$

Given a linear kernel k, it can be expanded as follows:

$$\begin{aligned} k(\boldsymbol{\delta}_i^x, \boldsymbol{\delta}_j^x) &= k(\boldsymbol{\delta}_i^{x*}, \boldsymbol{\delta}_j^{x*}) + k(\boldsymbol{\delta}_i^b, \boldsymbol{\delta}_j^b) + k(\boldsymbol{\delta}_i^{x*}, \boldsymbol{\delta}_j^b) + k(\boldsymbol{\delta}_i^b, \boldsymbol{\delta}_j^{x*}), \\ k(\boldsymbol{\delta}_i^\xi, \boldsymbol{\delta}_j^\xi) &= k(\boldsymbol{\delta}_i^{\xi*}, \boldsymbol{\delta}_j^{\xi*}) + k(\boldsymbol{\delta}_i^b, \boldsymbol{\delta}_j^b) + k(\boldsymbol{\delta}_i^{\xi*}, \boldsymbol{\delta}_j^b) + k(\boldsymbol{\delta}_i^b, \boldsymbol{\delta}_j^{\xi*}), \\ k(\boldsymbol{\delta}_i^x, \boldsymbol{\delta}_j^\xi) &= k(\boldsymbol{\delta}_i^{x*}, \boldsymbol{\delta}_j^{\xi*}) + k(\boldsymbol{\delta}_i^b, \boldsymbol{\delta}_j^b) + k(\boldsymbol{\delta}_i^{x*}, \boldsymbol{\delta}_j^b) + k(\boldsymbol{\delta}_i^b, \boldsymbol{\delta}_j^{\xi*}). \end{aligned} \tag{34}$$

Following (23), we have:

$$\begin{aligned} &MMD^2\left(\boldsymbol{\delta}_m^x, \boldsymbol{\delta}_n^\xi\right) \\ &= \frac{1}{m(m-1)} \sum_i^m \sum_{j \neq i}^m k(\boldsymbol{\delta}_i^x, \boldsymbol{\delta}_j^x) + \frac{1}{n(n-1)} \sum_i^n \sum_{j \neq i}^n k(\boldsymbol{\delta}_i^\xi, \boldsymbol{\delta}_j^\xi) - \frac{2}{mn} \sum_i^m \sum_j^n k(\boldsymbol{\delta}_i^x, \boldsymbol{\delta}_j^\xi) \\ &= MMD^2\left(\boldsymbol{\delta}_m^{x*}, \boldsymbol{\delta}_n^{\xi*}\right) + MMD^2\left(\boldsymbol{\delta}_m^b, \boldsymbol{\delta}_n^b\right) + 2R_{mn}^x + 2R_{mn}^\xi. \end{aligned} \tag{35}$$

where $\boldsymbol{\delta}_m^\cdot = \{\boldsymbol{\delta}_i^\cdot | \boldsymbol{\delta}_i^\cdot \in \mathbb{R}^d, \cdot \in \{x, x*, b\}, i = 1, 2, \cdots, m\}$ and $\boldsymbol{\delta}_n^\cdot = \{\boldsymbol{\delta}_j^\cdot | \boldsymbol{\delta}_j^\cdot \in \mathbb{R}^d, \cdot \in \{\xi, \xi*, b\}, j = 1, 2, \cdots, n\}$. The remainder terms R_{mn}^x and R_{mn}^ξ can be expressed as:

$$\begin{aligned} R_{mn}^x &\coloneqq \frac{1}{m(m-1)} \sum_i^m \sum_{j \neq i}^m \boldsymbol{\delta}_i^b \cdot \boldsymbol{\delta}_j^{x*} - \frac{1}{n^2} \sum_i^n \sum_j^n \boldsymbol{\delta}_i^b \cdot \boldsymbol{\delta}_j^{x*}, \\ R_{mn}^\xi &\coloneqq \frac{1}{n(n-1)} \sum_i^m \sum_{j \neq i}^m \boldsymbol{\delta}_i^b \cdot \boldsymbol{\delta}_j^{\xi*} - \frac{1}{m^2} \sum_i^m \sum_j^m \boldsymbol{\delta}_i^b \cdot \boldsymbol{\delta}_j^{\xi*}. \end{aligned} \tag{36}$$

Next, we discuss the asymptotic convergence property of each term in equation (35). For the first and second terms, following Lemma A.2 [Gretton *et al.*, 2012], we have:

$$\begin{aligned} \mathbb{P}\left(\left|MMD^2(\boldsymbol{\delta}_m^{x*}, \boldsymbol{\delta}_n^{\xi*}) - MMD^2(P_{\boldsymbol{\delta}^{x*}}, P_{\boldsymbol{\delta}^{\xi*}})\right| \geq \varepsilon\right) &\leq 2\exp\left(\frac{-Cn\varepsilon^2}{8(1+C){K_+^x}^2}\right), \\ \mathbb{P}\left(\left|MMD^2(\boldsymbol{\delta}_m^b, \boldsymbol{\delta}_n^b) - 0\right| \geq \varepsilon\right) &\leq 2\exp\left(\frac{-Cn\varepsilon^2}{8(1+C){K_+^b}^2}\right), \end{aligned} \tag{37}$$

where $K_+^x \coloneqq \sup_{ij} k(\boldsymbol{\delta}_i^{x*}, \boldsymbol{\delta}_j^{x*})$, $K_+^b \coloneqq \sup_{ij} k(\boldsymbol{\delta}_i^b, \boldsymbol{\delta}_j^b)$.

We then discuss the asymptotic convergence property of R_{mn}^x in (35). Given $\boldsymbol{\delta}^b$ and $\boldsymbol{\delta}^{x*}$ are independent, we define a random variable $y \coloneqq \boldsymbol{\delta}^b \cdot \boldsymbol{\delta}^{x*} \in \mathbb{R}$. According to Lemma A.3 and Hoeffding's Inequality [Hoeffding, 1994], R_{mn}^x is asymptotically bounded as shown below:

$$\begin{aligned}
\mathbb{P}(\left|R_{mn}^x\right| \geq \varepsilon) &= \mathbb{P}\left(\left|\frac{1}{m(m-1)}\sum_{i=1}^{m(m-1)} y_i - \frac{1}{n^2}\sum_{j=1}^{n^2} y_j\right| \geq \varepsilon\right) \\
&= \mathbb{P}\left(\left|\frac{1}{m(m-1)}\sum_{i=1}^{m(m-1)} y_i - \mathbb{E}(y) + \mathbb{E}(y) - \frac{1}{n^2}\sum_{j=1}^{n^2} y_j\right| \geq \varepsilon\right) \\
&\leq \mathbb{P}\left(\left|\frac{1}{m(m-1)}\sum_{i=1}^{m(m-1)} y_i - \mathbb{E}(y)\right| \geq \frac{\varepsilon}{2}\right) + \mathbb{P}\left(\left|\frac{1}{n^2}\sum_{j=1}^{n^2} y_j - \mathbb{E}(y)\right| \geq \frac{\varepsilon}{2}\right) \\
&\leq 2\exp\left(\frac{-2m(m-1)(\varepsilon/2)^2}{{K_+^y}^2}\right) + 2\exp\left(\frac{-2n^2(\varepsilon/2)^2}{{K_+^y}^2}\right) \\
&\leq 4\exp\left(\frac{-m(m-1)\varepsilon^2}{2{K_+^y}^2}\right),
\end{aligned} \tag{38}$$

where $K_+^y \coloneqq \sup_{i,j}(y_i - y_j)$. Similarly, for R_{mn}^ξ in equation (35), we define $\theta \coloneqq \boldsymbol{\delta}^b \cdot \boldsymbol{\delta}^{\xi*} \in \mathbb{R}$ and $K_+^\theta \coloneqq \sup_{i,j}(\theta_i - \theta_j)$. Then, R_{mn}^ξ is asymptotically bounded as follows:

$$\mathbb{P}(\left|R_{mn}^\xi\right| \geq \varepsilon) \leq 4\exp\left(\frac{-m^2\varepsilon^2}{2{K_+^\theta}^2}\right). \tag{39}$$

By Lemma A.3 and equations (37), (38), and (39), we have:

$$\begin{aligned}
&\mathbb{P}\left(\left|MMD^2(\boldsymbol{\delta}_m^x, \boldsymbol{\delta}_n^\xi) - MMD^2(P_{\boldsymbol{\delta}^{x*}}, P_{\boldsymbol{\delta}^{\xi*}})\right| \geq \varepsilon\right) \\
&\leq \mathbb{P}\left(\left|MMD^2(\boldsymbol{\delta}_m^{x*}, \boldsymbol{\delta}_n^{\xi*}) - MMD^2(P_{\boldsymbol{\delta}^{x*}}, P_{\boldsymbol{\delta}^{\xi*}})\right| \geq \frac{\varepsilon}{4}\right) \\
&\quad + \mathbb{P}\left(\left|MMD^2(\boldsymbol{\delta}_m^b, \boldsymbol{\delta}_n^b)\right| \geq \frac{\varepsilon}{4}\right) + \mathbb{P}(\left|2R_{mn}^x\right| \geq \frac{\varepsilon}{4}) + \mathbb{P}(\left|2R_{mn}^\xi\right| \geq \frac{\varepsilon}{4}) \\
&\leq 4\exp\left(\frac{-Cn\varepsilon^2}{128(1+C)K_+^2}\right) + 4\exp\left(\frac{-m(m-1)\varepsilon^2}{128K_+^2}\right) + 4\exp\left(\frac{-m^2\varepsilon^2}{128K_+^2}\right),
\end{aligned} \tag{40}$$

where $K_+ \coloneqq \max\{K_+^x, K_+^b, K_+^\xi, K_+^\theta\}$. When $m > 1 + (1+C)^{-1}$, the following inequality always holds[1]:

$$\frac{Cn}{1+C} < m(m-1) < m^2. \tag{41}$$

Finally, we have:

$$\mathbb{P}\left(\left|MMD^2(\boldsymbol{\delta}_m^x, \boldsymbol{\delta}_n^\xi) - MMD^2(P_{\boldsymbol{\delta}^{x*}}, P_{\boldsymbol{\delta}^{\xi*}})\right| \geq \varepsilon\right) \leq 12\exp\left(\frac{-Cn\varepsilon^2}{128(1+C)K_+^2}\right). \tag{42}$$

If $\alpha \coloneqq 12, \beta \coloneqq (128K_+^2)^{-1}$, **??** is proven. □

A.4 Lemmas

Lemma A.1 (k-dimensional Ramanujan's master theorem [Bradshaw and Vignat, 2023; Amdeberhan *et al.*, 2012])**.** *If a complex-valued function $f(x_1, \cdots, x_k)$ has an expansion:*

$$f(x_1, \cdots, x_k) = \sum_{n_1, \cdots, n_k}^{\infty} g(n_1, \cdots, n_k) \prod_{i=1}^{k} \frac{(-1)^{n_i}}{n_i!} x_i^{n_i}, \tag{43}$$

[1] In fact, this condition is always satisfied as long as there are more than only two normal samples.

where $g(n_1, \cdots, n_k)$ is a continuously analytic function everywhere, then the k-dimensional Mellin transform satisfies a multivariate version of Ramanujan's master theorem as follows:

$$\begin{aligned}\mathcal{M}[f(x_1, \cdots, x_k)](s_1, \cdots, s_k) &:= \int_{\mathbb{R}_+^k} \prod_{i=1}^{k} x_i^{s_i - 1} f(x_1, \cdots, x_k) dx_1 \cdots dx_k \\ &= \prod_{i=1}^{k} \Gamma(s_i) g(-s_1, \cdots, -s_k).\end{aligned} \tag{44}$$

The integral is convergent when $0 < Re(s_i) < 1, \forall i \in \{1, \cdots, k\}$.

Lemma A.2 (Asymptotic convergence of MMD [Gretton *et al.*, 2012])**.** *Given two sample sets $\boldsymbol{X} = \{\boldsymbol{x}_i | \boldsymbol{x}_i \sim p\}$, $\boldsymbol{Y} = \{\boldsymbol{y}_j | \boldsymbol{y}_j \sim q\}$, where p and q are the underlying distributions, respectively. Assume $0 \leq k(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq K$. We have:*

$$\mathbb{P}\left(\left|MMD^2(\boldsymbol{X}, \boldsymbol{Y}) - MMD^2(p, q)\right| \geq \varepsilon\right) \leq 2 \exp\left(\frac{-\varepsilon^2 mn}{8K^2(m+n)}\right), \tag{45}$$

Lemma A.3. *For any random variables $x_1, x_2, \cdots, x_k \in \mathbb{R}$, they always satisfy:*

$$\mathbb{P}\left(\left|\sum_{i=1}^{k} x_i\right| \geq \varepsilon\right) \leq \mathbb{P}\left(\sum_{i=1}^{k} |x_i| \geq \varepsilon\right) \leq \sum_{i=1}^{k} \mathbb{P}\left(|x_i| \geq \frac{\varepsilon}{k}\right), \tag{46}$$

where $\varepsilon \geq 0$, $k \in \mathbb{Z}_+$.

Proof. According to the triangle inequality, we have:

$$\left|\sum_{i=1}^{k} x_i\right| \leq \sum_{i=1}^{k} |x_i|, \tag{47}$$

which also indicates

$$\left\{\left|\sum_{i=1}^{k} x_i\right| \geq \varepsilon\right\} \subset \left\{\sum_{i=1}^{k} |x_i| \geq \varepsilon\right\}. \tag{48}$$

The following equation always holds:

$$\left\{\sum_{i=1}^{k} |x_i| \geq \varepsilon\right\} \subset \bigcup_{i=1}^{k} \left\{|x_i| \geq \frac{\varepsilon}{k}\right\}. \tag{49}$$

Thus, we have:

$$\mathbb{P}\left(\left|\sum_{i=1}^{k} x_i\right| \geq \varepsilon\right) \leq \mathbb{P}\left(\sum_{i=1}^{k} |x_i| \geq \varepsilon\right) \leq \sum_{i=1}^{k} \mathbb{P}\left(|x_i| \geq \frac{\varepsilon}{k}\right). \tag{50}$$

□

B Automatically Infer Anomaly Cluster Numbers

In case the true number of anomaly subtypes is unknown, ACSleuth can estimate this number based on a cell-cell similarity matrix calculated on fused anomalous cell embeddings (see Section 3.3 in the main text)[Yu *et al.*, 2023b]. Initially, give the fused anomalous cell embedding matrix $\Xi \in \mathbb{R}^{N_{an} \times p}$, where N_{an} denotes the anomaly cell numbers and p the embedding dimensions, is subjected to row normalization followed by the calculation of a cell-cell similarity matrix $\mathbb{S} \in \mathbb{R}^{N_{an} \times N_{an}}$ as:

$$\mathbb{S} = \frac{\Xi \Xi^T}{\max\{\Xi \Xi^T\}}. \tag{51}$$

Next, $\mathbb{S}$ is transformed to a graph Laplacian matrix $\mathbb{L} \in \mathbb{R}^{N_{an} \times N_{an}}$ as:

$$\mathbb{S}' = \mathbb{S} + \mathbb{S}^2, \tag{52}$$

$$\mathbb{L} = \mathbb{D}^{-\frac{1}{2}} \mathbb{S}' \mathbb{D}^{-\frac{1}{2}}. \tag{53}$$

Here, $\mathbb{D} \in \mathbb{R}^{N_{an} \times N_{an}}$ represents the degree matrix of $\mathbb{S}$, and $\mathbb{S}'$ aims to enhance the similarity structure. Then, the eigenvalues of $\mathbb{L}$ are ranked as $\lambda_{(1)} \leq \lambda_{(2)} \leq \cdots \leq \lambda_{(N_{an})}$. Finally, the number of subtypes can be inferred as:

$$n_{inf} = \text{argmax}_i \left\{\lambda_{(i)} - \lambda_{(i-1)}\right\}, \quad i = 2,\ 3,\ \cdots,\ N_{an}. \tag{54}$$

C Datasets

C.1 Datasets Detailed Information

Data Source	Dataset Name	#Instance	Sequencing Technology	Anomaly Ratio (%)
TME [Satpathy *et al.*, 2019]	a-TME-0	3559	10x	-
	a-TME-1	2968		60.24
Brain [Cusanovich *et al.*, 2018]	a-Brain-0	3203	Hi-Seq	-
	a-Brain-1	3448		39.07
	a-Brain-2	3750		7.31
PBMC [Zheng *et al.*, 2017]	r-PBMC-0	4698	10x	-
	r-PBMC-1	3684		13.14
	r-PBMC-2	3253		12.73
Cancer [Bischoff *et al.*, 2022]	r-Cancer-0	8104	10x	-
	r-Cancer-1	7721		50.03
	r-Cancer-2	4950		58.12
CSCC [Ji *et al.*, 2020]	r-CSCC-0	9242	10x	-
	r-CSCC-1	1409		73.46
	r-CSCC-2	2681		22.68
	r-CSCC-3	2091		32.52
Pancreas [Baron *et al.*, 2016]	r-Pan-0	7010	InDrop	-
	r-Pan-1	3514	Smart-Seq2	8.51
	r-Pan-2	3072	CEL-Seq2	13.41
KDDRev [Hettich, 1999]	KDDRev-0	7278	-	-
	KDDRev-1	5832		12.00
	KDDRev-2	815		87.48

Table 6: Datasets detailed information. The dataset name is formatted as (data type)-(data name)-(dataset ID). For data type, a and r denote scATAC-seq and scRNA-seq, respectively. Dataset ID 0 is reserved for reference datasets. #Instance denotes the number of instances in each dataset, and the anomaly ratio is the proportion of true anomalies.

C.2 KDDCUP-Rev Dataset

The KDD-Rev dataset is derived from the KDDCUP99 10 percent dataset, which contains 34 continuous and 7 categorical attributes. Domain shifts exist between records of different connection protocol types. In our settings, the reference dataset only includes UDP records, while the two target datasets contain ICMP and TCP records. There are four types of cyber-intrusions: DOS, R2L, U2R, and PROBING.