ACsleuth: Domain Adaptive and Fine-grained Anomalous Cell Detection for Single-cell Multiomics

Author Name
Affiliation
email@example.com

Abstract

Fined-grained anomaly detection at the cellular level is a crucial phase in diagnosing conditions and pathological analyses. Since the single-cell RNA sequencing (scRNA-seq) data analysis has been greatly prompted by the development of deep learning, we can effectively detect anomalous cells and explore their detail types. However, most of works simply focus on detecting anomalies within normal samples, overlooking the opportunity for more detailed distinctions of them. Moreover, the essence of both anomaly and fine-grained detection is to learn distinct representations for different sample categories, suggesting a straightforward pipeline of sequentially detecting anomalies then subtyping them. Thus, we introduce an innovative workflow called ACsleuth, aiming at fine-grained Anomalous cell detection for single-cell multiomics, which further contains a batch correction module to alleviate bias inherent in the raw data. By employing a GAN model to reconstruct normal samples and leveraging the reconstruction error as the assessment criterion for the anomaly detector, we then utilize the DESC clustering method for scRNA-seq data to explore subtypes of anomalies. Extensive experiments improve the superiority of ACsleuth both of the comparison with the state-of-the-art anomaly detection methods in detecting anomalies and simple combinations of anomaly detection and clustering methods in fine-grained anomalous detection tasks.

1 Introduction

Recently, single-cell RNA sequencing (scRNA-seq) technologies has developed rapidly, which allows us to better explore tissue heterogeneity at the cellular level. It yields a gene expression matrix, where each vector represents the expression values of specific genes across cells. In addition, detecting diseased cells, commonly referred to anomaly detection, constitutes a crucial phase in diagnosing conditions and conducting pathological analyses. Moreover, fine-grained detection of diseased cells, which can also be called subtyping, enables a more comprehensive understanding of disease classifications and facilitates the development of nuanced and efficacious treatment strategies. Therefore, leveraging the gene expression matrix derived from scRNA-seq empowers us to enhance detecting anomalous cells, then delve deeper into the exploration of their fine-grained subtypes.

Motivated by deep learning, previous studies for scRNA-seq data mainly focus on cell clustering and cell type annotation. The former aims to identify cell groups such as sczi-Desk [Chen *et al.*, 2020] and scCNC [Wang *et al.*, 2022], and the latter simply annotates these groups based on marker genes or another prior information such as ItClust [Hu *et al.*, 2020], scArches [Lotfollahi *et al.*, 2022]. However, both of aforementioned tasks are not ideally suited for fine-grained anomalous cells detection. Anomaly detection naturally exhibits class-imbalance characteristics, while the challenge in fine-grained anomaly detection lies in acquiring more precise representations of distinct cell types due to their inherent similarity. Nevertheless, employing reference-based methods can be challenging, given the presence of technical variation (e.g., different laboratory conditions) across diverse studies, commonly referred recognized as batch effects in molecular biology literature [Cheng *et al.*, 2023]. The intricate nature of scRNA-seq data, characterized by noise, batch effects, high dimensionality and sparsity [Amodio *et al.*, 2019] poses challenges that impede these deep methods from achieving optimal performance directly based on the raw data.

Domain adaptation is a crucial consideration in biological data analysis. Under the same sequencing technique, one of most typical instances is correcting batch effect, as we have discussed above. For example, the gene expression of cells from the same tissue but obtained from different patients, may be interfered by pronounced batch effects. Additionally, data derived from different sequencing technologies exhibits heterogeneous distributions, while compared with scRNA-seq,other sequencing technologies like scATAC-seq are even rarer [Stuart *et al.*, 2021]. This dataset shift is commonly denoted as domain bias in computer vision, encompassing aspects such as style differences,diverse sensory devices, etc [Wang and Deng, 2018]. Both batch effects within the same sequencing technique data and the bias arising from different sequencing technologies contribute to domain bias [Lu *et al.*, 2021]. However, most previous studies failed to recognize such shared characteristics between these two biases and

tended to address them independently, which are neither elegant nor efficient.

Besides, most of anomaly detection studies work solely on identifying anomalies within normal samples, which often treating it as a class-imbalance binary classification task [Aggarwal, 2017]. However, these studies overlook the opportunity for more detailed classification of anomalies, such as distinguishing between different types of tumors or various criminal behaviors. This limitation results in underutilization of the available data and hinders the ability to learn more granular representations of the anomalous samples. Furthermore, the core objective of both anomaly detection and fine-grained detection is to acquire a more nuanced representation for each type of samples. This shared optimalization goal suggests that these two tasks can seamlessly integrate into an entire pipeline. Starting with raw data encompassing both normal and anomalous samples, the pipeline can effectively distinguish anomalies of various subtypes.

To address these limitations, we propose an comprehensive workflow consists of fine-grained anomalous cell detection and batch correction for scRNA-seq data, named ACsleuth, aiming at subtyping anomalies from the raw data which contains various normal and diseased types of cells. Our contributions are summarized as follows:

- Methodologically, we introduce an innovative workflow called ACsleuth, which contains fine-grained anomalous cell detection. The anomaly detector is first trained unsupervisedly on the reference dataset which has only normal samples, then we learn the batch effects on the normal samples identified by the former detector and use it for the batch correction on anomaly sets sequentially, finally subtyping anomalies simply by a recent single-cell clustering method.

- Innovatively, we achieve domain adaptation by tackling batch effects in the target dataset after removing anomalous samples and using datasets derived from different sequencing technologies as the reference and the target data. For the former, ACsleuth involves leveraging a wasserstein distance as the domain transfer loss in MMD, to learn the domain bias with solely normal cells then subsequently applying bias correction on the detected anomalies before fine-grained subtyping them. For the latter, we simply correct the domain bias in the cross-domain anomaly detection task to obtain distinct representations for anomalous cells.

- Empirically, we conduct extensive experiments to validate the exceptional performance of ACsleuth. Across evaluations on three distinct scRNA-seq datasets and one scATAC-seq dataset for cross-domain anomaly detection task within different numbers of highly variable genes, ACsleuth consistently raised state-of-the-art in the majority of anomaly detection and fine-grained detection.

2 Related Works

2.1 Anomaly Detection

Anomaly detection usually relies on learning distinct representations between normal samples and anomalies. Many studies focus on representing normal samples and anomalies are detected by filtering representations that are dissimilar to normal ones. One straightforward manner is one-class classification, which aims to train a model that can accurately describe normal samples and then distinguish whether test samples are from the same distribution as the reference data. This kind of detectors is trained to learn a new representation [Liu *et al.*, 2021; Liznerski *et al.*, 2020] that enhances the dissimilarity between embeddings of normal and anomalous samples, thereby improving the detectability of anomalies. While the one-class assumption is vulnerable since real datasets often contain multiple inliers [Xu *et al.*, 2023].

Generative models are another standard procedure in detecting outliers. They learn by reconstructing normal samples, which leads to poor reconstruction of anomalies in the target data due to their distinct reconstruction error. This is attributed to the fact that the reference data for the anomaly detection task exclusively consists of normal samples. Popular frameworks such as autoencoder (AE) [Chen *et al.*, 2017], generative adversarial networks [Di Mattia *et al.*, 2019] are widely used. Nevertheless, such GAN-based models like [Zenati *et al.*, 2018] still struggle with distinguishing multiple normal samples, and are prone to occurring model-collapse during training. Additionally, AE frameworks exhibit limitations handling the noisy or high-dimensional sparse data.

2.2 Single Cell RNA-Seq Data Subtyping

Single-cell subtyping aims to identify distinct subtypes within the same cell type, e.g., tumors, which is similar to a broader task called cell annotation. The essence of two tasks lies in learning representations that effectively capture the heterogeneity among distinct types to the fullest extent. Cell annotation simply follows three steps [Pliner *et al.*, 2019]: learning a compact representation by projecting cells to a lower-dimensional space, mapping similar cells to groups in the low-dimensional representation (typically via clustering), and finally characterizing the differences in gene expression among the cell groups. ACE [Lu *et al.*, 2021] uses AE to generate low-dimensional representations and considers the intrinsic dependencies among genes. Kratos [Zhou *et al.*, 2022] fuses the dimension reduction and clustering cells to optimize jointly. scTAG [Yu *et al.*, 2022] simultaneously identifies cells clusters and learns topological representations between cells. scPOT [Zhai *et al.*, 2023] achieves annotating seen cell types and novel cell type clustering simultaneously. For subtype detection, SCEVAN [De Falco *et al.*, 2023] and CopyKAT [Gao *et al.*, 2021] are designed specifically for tumors since they used the prior information, CAMLU [Li *et al.*, 2022] exhibits a higher degree of generality via the employment of AE, albeit its performance is suboptimal.

Detecting subtypes solely involves assigning labels to distinct types of samples, while cell annotation demands the utilization of prior information to assign specific labels to the cells. However, compared to cell annotation, subtype detec-

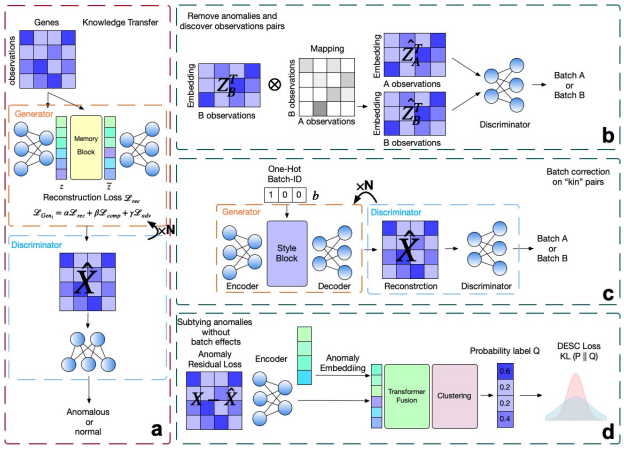

Figure 1: Illustration of ACsleuth. The overall model consists of GAN-based anomaly detection and batch effect correction modules and DESC clustering module for fine-grained anomalous cell detection.

tion necessitates the acquisition of more robust representations. Given that the samples from different subtypes fall under the same subcategory, the inherent differences between them tend to be subtler. Therefore, fine-grained subtype detection relies heavily on distinctive representations to achieve accurate classification. In summary, both anomaly detection and fine-grained detection necessitate learning precise representations, so that it is straightforward that we can propose a pipeline comprising anomaly detection and fine-grained subtype detection via the identical objective. Notably, to the best of our knowledge, none of the previous works are designed for a comprehensive workflow which contains such subtasks including anomaly detection and anomalies detailed multi-classification.

3 Methods

We first give an overview of subtyping cells workflow, which can be divided into two main parts: anomaly detection and subtype clustering. We start with training the detector unsupervisedly, as the reference dataset contains only normal cells. The target dataset contains two kinds of labels: the two-class one for distinguishing between normal and anomalous cells, and the detailed one for categorizing subtypes of cells. Specifically, we exclusively leverage the subtyping module for anomalies discovered by the detector, which we refer to as "pre-anomalies". That means they may include a few "fake anomalies". ACsleuth consists of three modules, each dedicated to anomaly detection, batch effect correction and subtyping.

3.1 Detecting Anomaly Cells

ACsleuth detects anomalies based on learning the reconstruction of normal samples in the reference datasets through the generative adversarial networks (GAN). As the GAN model is specifically trained to reconstruct normal samples, anomalous cells in target datasets are more likely to hold larger reconstruction errors, which are treated as an assessment criterion.

Model Training. Our GAN module (referred as *module I*) consists of a generator and a discriminator. The generator can be subdivided into three distinct components: an encoder, a decoder and a memory block. To define the gene expression pattern, we employ the mathematical model similar to [Hornung *et al.*, 2016]. Let S^k denote the group of cells in the same batch k. Then, we propose an assumption as follows.

Assumption 3.1. *The gene expression vector $\boldsymbol{x}_i \in S^k \cap \mathbb{R}^{N_{gene}}$ of cell i can be represented as:*

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

where $\boldsymbol{x}_i^*$ denotes the biological factor, $\boldsymbol{b}_i^k$ denotes batch-specific factor. $\boldsymbol{x}_i^* \sim P_{\boldsymbol{x}^*}$, $\boldsymbol{b}_i^k \sim P_{\boldsymbol{b}^k}$, and random noise $\boldsymbol{\epsilon}_i \sim N(0, \boldsymbol{\sigma}_i^2)$.

Thus, we have the embeddings of the cell i with ACsleuth's encoder (MLP backbone, $G_E : \mathbb{R}^{N_{gene}} \to \mathbb{R}^p$) as:

$$\boldsymbol{z}_i = G_E(\boldsymbol{x}_i) \tag{2}$$

The memory block is fundamentally an embedding queue $\boldsymbol{Q} \in \mathbb{R}^{N_{mem} \times p}$ filled with $\boldsymbol{z}$, where N_{mem} is the number of in-memory embeddings. It provides an attention-based means for reconstructing the embedding as $\widetilde{\boldsymbol{z}}_i \in \mathbb{R}^p$:

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

where τ is the temperature hyperparameter. During the entire training procedure, $\boldsymbol{Q}$ is dynamically updated by enqueuing the most recently reconstructed $\widetilde{\boldsymbol{z}}$ and dequeuing the oldest ones, thereby striking a balance between preserving learnt features and adapting to new samples also effectively mitigating the risks of mode collapse. Subsequently, the decoder (MLP backbone, $G_D : \mathbb{R}^p \to \mathbb{R}^{N_{gene}}$) reconstructs the gene expression vectors with $\widetilde{\boldsymbol{z}}_i$ as

$$\widehat{\boldsymbol{x}}_i = G_D(\widetilde{\boldsymbol{z}}_i) \tag{4}$$

The discriminator $D : \mathbb{R}^{N_{gene}} \to \mathbb{R}^k$ is trained to distinguish whether $\boldsymbol{x}$ and $\widehat{\boldsymbol{x}}$ is real or generated. Therefore, loss functions of the generator and the discriminator for anomaly detection is defined as:

$$\begin{aligned} \mathcal{L}_{G_1} &= \alpha \mathcal{L}_{\text{rec}} + \beta \mathcal{L}_{\text{adv}} \\ &= \alpha \mathbb{E}\left[\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|_1\right] - \beta \mathbb{E}\left[D(\widehat{\boldsymbol{x}})\right] \end{aligned} \tag{5}$$

$$\mathcal{L}_{D_1} = \mathbb{E}\left[D(\widehat{\boldsymbol{x}})\right] - \mathbb{E}\left[D(\boldsymbol{x})\right] + \lambda \mathbb{E}\left[\left(\|\nabla_{\widetilde{\boldsymbol{x}}} D(\widetilde{\boldsymbol{x}})\|_2 - 1\right)^2\right] \tag{6}$$

where $\widetilde{\boldsymbol{x}} = \epsilon \widehat{\boldsymbol{x}}_i + (1-\epsilon)\boldsymbol{x}$, $\epsilon \in (0,1)$. $\mathcal{L}_{\text{rec}}$ is defined as the data reconstruction loss. $\mathcal{L}_{\text{adv}}$ denotes the adversarial loss. $\alpha, \beta, \lambda \geq 0$ represent the weights of three loss functions. Significantly, the discriminator's loss function contains of a gradient penalty factor, which promotes the stability of the adversarial training and reduces the risk of mode collapse [Gulrajani *et al.*, 2017].

Anomaly Inference. Let $\boldsymbol{\delta}_m^x := \boldsymbol{x}_m - \widehat{\boldsymbol{x}}_m$, $\boldsymbol{\delta}_n^y := \boldsymbol{y}_n - \widehat{\boldsymbol{y}}_n$ denote the reconstruction errors of normal and anomaly samples. Motivated by the main idea that the reconstruction errors of anomaly cells are significantly larger than normal cells, we hope to maximize the discrepancy of $\boldsymbol{\delta}_m^x$ and

$\boldsymbol{\delta}_n^y$. Here, we describe our objective with MMD (Maximum Mean Discrepancy), a non-parametric metric to quantify the difference between two probability distributions in a reproducing kernel Hilbert space (RKHS) [Kolouri *et al.*, 2016; Gretton *et al.*, 2012]. And the squared MMD between two sets of samples $\boldsymbol{x}_m \sim p$ and $\boldsymbol{y}_n \sim q$ can be derived as [Gretton *et al.*, 2012]:

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

where k is a postive definite kernel. Therefore, the objective for detecting anomalous and normal cells can be expressed as follows:

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

In fact, it indicates we need to optimize a mapping function $f_s : \mathbb{R}^p \rightarrow \{0, 1\}$ that partitions unlabeled total samples $\boldsymbol{\delta}_{m+n}$ into $\boldsymbol{\delta}_m^x$ and $\boldsymbol{\delta}_n^y$, aiming to minimize $\mathcal{L}_p$. Therefore, we propose Theorem 3.1 for transforming (8), and the proof is provided in Appendix A.1.

Theorem 3.1. *Define $s_i \coloneqq f_s(\boldsymbol{\delta}_i)$. The optimization* (8) *have an equivalent form:*

$$\min_{f_s} \quad \mathcal{L}_p = -\sum_{i}^{m+n} \sum_{j \neq i}^{m+n} k(\boldsymbol{\delta}_i, \boldsymbol{\delta}_j) \gamma(s_i, s_j) \tag{9}$$

where

$$\gamma(s_i, s_j) = \begin{cases} \frac{1}{m(m-1)}, & s_i = s_j = 0 \\ \frac{1}{n(n-1)}, & s_i = s_j = 1 \\ \frac{-1}{mn}, & s_i \neq s_j \end{cases} \tag{10}$$

If $s_i = 1$, sample i will be classified as anomaly; otherwise, it will be classified as normal.

For detecting anomalies, We aim to learn a predictor $f_p : \mathbb{R}^p \rightarrow [0, 1]$ to provide anomaly scores $p_i \coloneqq f_p(\boldsymbol{\delta}_i)$. However, $\gamma(s_i, s_j) : \{0, 1\} \times \{0, 1\} \rightarrow \mathbb{R}$ is discrete and doesn't exist gradients. A natural idea is to extend γ to the continuous scenario, resulting in the function $\gamma_c(p_i, p_j) : (0, 1) \times (0, 1) \rightarrow \mathbb{R}$. Here, we propose a potential γ_c based on Theorem 3.2, and the analytic properties of γ_c everywhere indicate its strong smoothness, making it amenable to gradient descent optimization. The proof of Theorem 3.2 is available at Appendix A.2.

Theorem 3.2. *For any given m, n, there always exists γ_c that extends γ to the continuous scenario and 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.

A potential γ_c can be defined as follows:

$$\begin{aligned} \gamma_c(p_i, p_j) \coloneqq & \frac{\sin \pi p_i \sin \pi p_j}{\pi^2} \left(\frac{[m(m-1)]^{-1}}{p_i p_j} - \frac{(mn)^{-1}}{(p_i - 1) p_j} \right. \\ & \left. - \frac{(mn)^{-1}}{p_i (p_j - 1)} + \frac{[n(n-1)]^{-1}}{(p_i - 1)(p_j - 1)} \right) \end{aligned} \tag{11}$$

Subsequently, the predictor can be optimized with the following loss function:

$$\mathcal{L}_p = -\sum_{i} \sum_{j \neq i} k(\boldsymbol{\delta}_i, \boldsymbol{\delta}_j) \gamma_c(p_i, p_j) \tag{12}$$

Ultimately, by minimizing (12) through updating predictor with gradient descent, we can achieve our objective (8) and obtain anomaly scores.

Cross-domain Anomaly Detection. Compared with other methods, ACsleuth not only measures the difference between anomalous and normal samples in a more effective manner, enhancing the accuracy of anomaly detection, but also introduces transferability into this process. Specifically, in the context of large sample sizes, even in the presence of batch effects or domain biases between the target and reference datasets, we can still ensure that the accuracy of prediction remains unaffected. From a mathematical perspective, ACsleuth satisfies Theorem 3.3, which indicates the cross-domain transfer error can be bounded by sample size n. In practice, when we minimize the objective (8), that is, when maximizing this batch-effect-independent $MMD^2\left(P_{\delta^{y*}}, P_{\delta^{x*}}\right)$. Therefore, Theorem 3.3 can indicate inherent transferability to realize cross-domain detection. The proof of Theorem 3.3 is provided in Appendix A.3.

Theorem 3.3. *We assume the proportion of normal samples $\boldsymbol{x}_m$ to anomaly samples $\boldsymbol{y}_n$ satisfies $m/n = C \geq 1$, where C is a constant but not large enough. If Assumption 3.1 is valid and MMD is induced by linear kernels, for any samples $\boldsymbol{x}_m, \boldsymbol{y}_n$ there exist the biological information reconstruction errors $\boldsymbol{\delta}_n^{y*}, \boldsymbol{\delta}_m^{x*}$ that are independent of batch effects, such that their distributions $P_{\boldsymbol{\delta}^{y*}}, P_{\boldsymbol{\delta}^{x*}}$ satisfy:*

$$\begin{aligned} \mathbb{P}\left(\left|MMD^2(\boldsymbol{\delta}_m^x, \boldsymbol{\delta}_n^y) - MMD^2(P_{\boldsymbol{\delta}^{x*}}, P_{\boldsymbol{\delta}^{y*}})\right| \geq \varepsilon\right) \leq \\ \alpha \exp\left(-\frac{\beta C}{1+C} n \varepsilon^2\right) \end{aligned} \tag{13}$$

where ε denotes transfer error, α, β are constants.

Based on Theorem 3.3, we can reconstruct normal samples in the reference scRNA-seq dataset and simply apply it on the target scATAC-seq dataset. This cross-domain task shows ACsleuth is capable of learning a robust representation of the entire cell thereby exhibiting strong generalization performance.

3.2 Multi Sample Batch Effect Correction

The identification and removal of confounding anomalies from target datasets empower ACsleuth to align multiple target datasets onto the feature space of the reference dataset, thus fulfilling the task of multi-sample batch correction for single-cell data. As illustrated in figure 1, this task contains two subtasks: (1) matching each normal sample in the target datasets with its most analogous counterpart in the reference dataset. A pair of such samples is termed a "kin" pair, indicating that the two paired samples are more likely to share identical biological contents. (2) learning a "style-divergence" matrix to represent the batch effects between target datasets and the reference dataset. This matrix can then be applied to map

target datasets to the reference data space in a "style-transfer" manner.

Formally, we first define the batch effect in the context of gene expression analysis as:

$$x_{ij} = x_{ij}^{'} + b_j + \epsilon_{ij} \tag{14}$$

where x_{ij} denotes the gene expression values for $j-th$ cell in the batch i, b_{ij} is the batch effects of the batch i and the term ϵ_{ij} $N(0,1)$ represents random noise, which is uneffected by batch effects.

For the first subtask, we introduce a GAN module (referred as *module II*) that takes as input the embeddings of normal samples from both the reference and target datasets, which are generated by the encoder within *module I*. The generator of this module is trained to learn a non-negative mapping matrix for generating samples in target datasets through those in the reference dataset, while the discriminator corresponding to learning to distinguish the authentic and generated samples. In detail, let N_T and N_R denote the number of samples in the target and reference dataset, then define $Z_T \in \mathbb{R}^{N_T \times p}$ as the embeddings of multi-sample target datasets, $Z_R \in \mathbb{R}^{N_R \times p}$ the embeddings of the reference dataset, $M \in \mathbb{R}^{N_T \times N_R}$ the trainable non-negative mapping matrix. The generated sample represents as:

$$\widehat{Z}_T = ReLU(M)Z_R \tag{15}$$

The rectified linear unit (ReLU) function serves as the non-negative constraint on M. Furthermore, we indicate the loss functions for the generator and the discriminator of *module II* as:

$$\mathcal{L}_{\text{Gen}_2} = \alpha\mathbb{E}\|Z_T - \widehat{Z}_T\|_1 - \beta\mathbb{E}[D(\widehat{Z}_T)] \tag{16}$$

$$\mathcal{L}_{D_2} = D(\widehat{Z}_T) - \mathbb{E}\left[D(Z_T)\right] + \lambda\mathbb{E}\left[(\|\nabla D(\widetilde{Z}_T)\|_2 - 1)^2\right] \tag{17}$$

where $\widetilde{Z} = \epsilon\widehat{Z} + (1-\epsilon)\,Z,\ \epsilon \in (0,1).\alpha, \beta, \lambda \geq 0$ represent the weights of the loss terms. Upon the completion of the adversarial training of *module II*, the column index of the maximum value on the i^{th} row of matrix M corresponds to the index of the reference sample that forms a "kin" pair with the i^{th} sample in the target dataset. We propose a hypothesis that samples belonging to the same "kin" pair share analogous biological content, allowing us to approximate the reference sample by eliminating the "style-divergence" from the target sample. Therefore, for the second subtask, a GAN for "style-transfer" (referred as *module III*) is employed to learn the "style-divergences", or namely batch effects, between pairs of the target and reference datasets as rows of a trainable matrix $S \in \mathbb{R}^{N_{batch} \times p}$ in the latent embedding space. In detail, for an sample i in the target datasets, the encoder of the generator within *module III* maps its raw gene expression vector x_i to $z_i \in \mathbb{R}^p$ in the latent embedding space. This latent representation is then employed to estimate the embeddings of its "kin" sample j as follows:

$$z_i = f_{MLP}(x_i, W_5), \widehat{z}_j = z_i - S^T b_i \tag{18}$$

where $b_i \in \mathbb{R}^{N_{batch}}$ denotes i's one-hot encoded batch-identity vector for selecting the corresponding "style-divergence" row from S. In sequence, the decoder of the generator maps $\widehat{z}_j$ back to the original reference data space as $\widehat{x}_j$. The discriminator of *module III* is simply trained to distinguish the x_j and $\widehat{x}_j$. We indicate the loss functions for the generator and the discriminator of *module III* as:

DATA	Type	Genes	Cells	Anomaly Ratio
PBMCs	scRNA-seq	32738	3684	13.14%
			3253	12.73%
Cancer	scRNA-seq	33538	7721	50.03%
			4950	58.12%
cSCC	scRNA-seq	32738	6181	37.58%
TME	scATAC-seq	23127	2968	60.24%

Table 1: Information of Datasets.

$$\mathcal{L}_{Gen_3} = \alpha\mathbb{E}||x_R - \widehat{x}_R||_1 - \beta\mathbb{E}[D(\widehat{x}_R)] \tag{19}$$

$$\mathcal{L}_{D_3} = \mathbb{E}\left[D\left(\widehat{x}_R\right)\right] - \mathbb{E}\left[D\left(x_R\right)\right] + \lambda\mathbb{E}[(||\nabla D(\widetilde{x}_R)||_2 - 1)^2] \tag{20}$$

where α, β, λ and $\widetilde{x}_R$ have the same definitions as their counterparts in *module II*. Finally, samples in multiple target datasets can be batch-corrected and aligned in the common reference data space by passing through the trained generator within *module III*.

3.3 Fine-grained Anomalous Cell Detection

In this section, we introduce how ACsleuth subtypes anomaly cells detected by preceding modules. Since *module III* can align anomalies identified by module I across multiple target datasets in the common reference space, thus significantly mitigating the confounding batch variations and facilitating the anomaly subtyping task. Specifically, we combine the embeddings and reconstruction loss generated by *module I* to distinguish between various novel cell subtypes and address the variations among them. We first give some notations: for an anomalous cell i, ${x_i}^g$ denotes cell's batch-corrected gene expression vector, and $\widehat{x}_i^g$ the gene expression vector reconstructed by *module I*, respectively. ${r_i}^g$ represents the reconstruction loss of gene expressions. z_i and ζ_i represent the embeddings of x_i and r_i, which are also generated by the encoder within *module I*. Above terms are calculated as follows:

$${r_i}^g = {x_i}^g - \widehat{x}_i^g, r_i = {r_i}^g \tag{21}$$

$$z_i = f_{MLP}({x_i}^g, W_1), {z_i}^* = TF([z_i||\zeta_i], W_{tf}), \quad where \zeta_i = f_{MLP}({r_i}^g, W_1) \tag{22}$$

where ${z_i}^*$ represents the fused embeddings and reconstruction loss of anomalous cell i by a transformer fusion (TF) block. We then leverage a discriminatively boosted clustering algorithm, DESC [Li *et al.*, 2020], to cluster anomalies according to their z^*. In detail, DESC applies a Cauchy kernel to compute the soft assignment score of an anomaly cell i to a cluster j as:

$$q_{i,j} = \frac{\left(1 + \left\|z_i^* - \mu_j\right\|^2 / v\right)^{-1}}{\sum_{j'}\left(1 + \left\|z_i^* - \mu_{j'}\right\|^2 / v\right)^{-1}} \tag{23}$$

where $q_{i,j}$ represents the probability that cell i belongs to cluster j, μ_j the centroid of cluster j, v the degree of freedom of the Cauchy kernel. The clustering loss function is a

KL-divergence $\mathcal{L}$ calculated on q and an auxiliary target distribution p, defined as:

$$p_{i,j} = \frac{q_{i,j}^2 / \sum_i q_{i,j}}{\sum_j \left(q_{i,j}^2 / \sum_i q_{i,j}\right)} \tag{24}$$

$$\mathcal{L} = \sum_i \sum_j p_{i,j} log\left(\frac{p_{i,j}}{q_{i,j}}\right) \tag{25}$$

Note that anomalies with a high-confident assignment are overweighted in p. Experimentally, the iterative updating of TF weights W_{TF} and cluster centroid μ with the objective of minimizing $\mathcal{L}$ drives q to p, resulting in a gradual transformation of harder-to-cluster embeddings z^* into easier ones. The self-paced and iterative DESC persists until the change in the hard assignments of anomalous cells falls below a predefined threshold or reaches a predefined number of iterations. The subtype labels of anomalies can be readily obtained from their final hard cluster assignments. Additionally, the number of subtype labels is assumed to beknown or automatically inferred during the clustering. In detail,

4 Experiments

4.1 Experimental Settings

Dataset. Our experiments contain anomaly cells' detection and anomaly cell clustering. Furthermore, each task splits to two parts. For the former, we use datasets simply called PBMCs, Cancer, cSCC and TME, including scRNA-seq and scATAC-seq, as we have both intra-data detection and cross-data detection. In detail, we specifically treat B cells and natural killer (NK) cells as anomalies in PBMCs. For the latter, we use Cancer and cSCC for the task of single-batch and concatenate different batches for the multi-batch task. The detailed information is described in Table 1. Each dataset is split according to the number of highly variable genes, including 3000, 6000 and all genes respectively. Besides, we normalize and log the data using the Scanpy [Wolf *et al.*, 2018] package.

Baselines. For the anomaly detection, we compare ACsleuth with four recently state-of-the-art anomaly detection methods in tabular data (SLAD [Xu *et al.*, 2023], ICL [Shenkar and Wolf, 2021], NeuTraL [Qiu *et al.*, 2021], RCA [Liu *et al.*, 2021]) and two classical methods (Scmap [Kiselev *et al.*, 2018] and AE [Sakurada and Yairi, 2014]). The code we use is provided by authors, and each one we use default hyperparameters if no specific instructions are given.

For the fine-grained anomalous cell detection, we compare ACsleuth with the combination of one baseline of anomaly detection and the state-of-the-art method of clustering. The former we choose the SLAD, as it has better performance both in the ability of both detecting anomalies and computational efficiency than other baselines. Then select four recent clustering methods (scTAG [Yu *et al.*, 2022], EDESC [Cai *et al.*, 2022], DFCN [Tu *et al.*, 2021], Leiden [Traag *et al.*, 2019] and K-means [MacQueen and others, 1967]), including deep embedded clustering and scRNA-seq clustering, and one classical clustering method.

Evaluation Metrics. We assess the performance with two widely used evaluation metrics: F1 score for anomaly detection, normalized mutual information (NMI) for cell clustering. For the cell's subtype detection, we simply multiply the F1 score and the NMI to create a new metric, which is used to measure the effect of the whole task. The higher the values of these metrics, the better the detecting and the clustering performance. The reported metrics are averaged results with standard deviations over ten independent runs.

Implementation Details. Our experiments are conducted with a NVIDIA GeForce RTX 3090 GPU and 24GB of memory. (Hyperparameters settings)

4.2 Results

Anomaly detection results. Table 1 illustrates the detection performance in terms of F1 score, of our model ACsleuth and the competing methods. Each dataset's best detector is boldfaced. On the three datasets with different number of highly variable genes, we totally

Anomaly detection Cross-domain results. To evaluate the generalization capacity of ACsleuth in anomaly detection, we trained the model on the PBMCs then tested it on the TME. The former dataset is the scRNA-seq data, while the latter is the scATAC-seq data, both representing different domains for the same cells. Specifically, we screen out genes common to both datasets, then still split them according to the number of highly variable genes. The performance of ACsleuth and baseline methods illustrate in the bottom of Table 1. Results demonstrate that ACsleuth outperforms baseline anomaly detection methods, which means it can construct robust representations for each single-cell.

Anomalies fine-grained detection results. We evaluate ACsleuth by comparing it with simple combinations of anomaly detection methods and clustering methods for the fine-grained anomaly detection task. In this section, we assess the impact of fine-grained detection task in two scenarios: the single-batch representing a common condition and multi-batch where different batches of Cancer datasets are concatenated to emphasize the importance of batch effect correction in scRNA-seq data. As shown in Table 2, we can conclude that for the single-batch task, ACsleuth consistently achieves stable and superior performance, as reflected by the product of F1 scores for detecting anomalies and NMI scores for clustering on most of cases. In the multi-batch task, ACsleuth significantly outperforms baseline methods. Results fully demonstrate that batch effects are intractable for recent methods. ACsleuth addresses this challenge by learning such kind of biases from the normal samples split from anomaly detection, then subsequently correcting them on the detected anomalies before subtyping them. We empirically establish the necessity of the batch correction module for the scRNA-seq data analysis.

Computational Performance.

4.3 Ablation Studies

In this section, we conduct a series of ablation studies to validate how different configurations of each module impact the anomaly subtyping task. We still assess ACsleuth with the Cancer dataset since it contains more genes (features)

DATA	Anomaly	Highly Genes	ACsleuth	SLAD	ICL	NeuTraL	RCA	AE	Scmap
PBMCs	B Cells	3000	**0.833±0.034**	0.347±0.023	0.262±0.029	0.299±0.014	0.267±0.007	0.449±0.015	0.377±0.
		6000	**0.818±0047**	0.279±0.021	0.246±0.024	0.234±0.022	0.266±0.009	0.401±0.008	0.393±0.
		full	**0.785±0.044**	0.149±0.007	0.176±0.014	0.215±0.010	0.266±0.006	0.258±0.006	0.390±0.
	NK Cells	3000	**0.804±0.004**	0.548±0.013	0.307±0.042	0.330±0.025	0.258±0.025	0.462±0.012	0.495±0.
		6000	**0.819±0.022**	0.587±0.021	0.359±0.038	0.351±0.030	0.259±0.024	0.545±0.006	0.615±0.
		full	**0.757±0.035**	0.607±0.015	0.168±0.013	0.294±0.065	0.259±0.024	0.617±0.005	0.589±0.
Cancer	Epithelial & Immune Tumor	3000	**0.926±0.009**	0.881±0.006	0.856±0.006	0.889±0.011	0.818±0.033	0.896±0.002	0.679±0.
		6000	0.832±0.014	**0.862±0.004**	0.844±0.005	0.847±0.006	0.600±0.041	0.853±0.003	0.683±0.
		full	**0.826±0.035**	0.785±0.005	0.572±0.006	0.691±0.040	0.649±0.009	0.717±0.006	0.734±0.
	Epithelial & Stromal Tumor	3000	0.860±0.049	0.888±0.014	0.899±0.018	**0.913±0.008**	0.677±0.036	0.746±0.005	0.688±0.
		6000	0.824±0.007	0.828±0.014	0.856±0.014	**0.892±0.008**	0.543±0.025	0.681±0.006	0.735±0.
		full	**0.717±0.059**	0.638±0.005	0.556±0.006	0.686±0.024	0.471±0.026	0.571±0.003	0.683±0.
TME	Tumor(pre)	3000	0.0±0.0	0.582±0.011	0.791±0.050	0.578±0.016	0.493±0.030	0.488±0.001	0.752±0.
		6000	0.0±0.0	0.590±0.017	0.728±0.028	0.591±0.013	0.511±0.038	0.509±0.002	0.733±0.
		full	0.0±0.0	0.640±0.021	0.756±0.012	0.638±0.017	0.538±0.032	0.532±0.002	0.529±0.
cSCC	Diff & Basal & Cyc Tumor	3000	0.0±0.0	0.486±0.014	0.173±0.025	0.085±0.042	0.0±0.0	0.484±0.004	0.0±0.
		6000	0.0±0.0	0.503±0.007	0.154±0.020	0.123±0.030	0.0±0.0	0.491±0.006	0.0±0.
		full	0.0±0.0	0.510±0.010	0.155±0.052	0.140±0.031	0.0±0.0	0.561±0.005	0.0±0.
PBMC-TME	Tumor(pre)	3000		0.570±0.021	0.583±0.032	0.645±0.044	0.±0.	0.524±0.002	0.±0.
		6000		0.573±0.034	0.629±0.044	0.637±0.070	0.±0.	0.533±0.001	0.±0.
		full		0.562±0.019	0.637±0.031	0.664±0.054	0.±0.	0.524±0.001	0.±0.

Table 2: Average F1 score with standard deviation for anomaly detection on single-cell transcriptomics datasets.

DATA	Anomaly	Highly Genes	ACsleuth	DFCN	EDESC	scTAG	Leiden	K-Means
Cancer	Epithelial & Immune Tumor	3000	**0.926±0.009**	0.881±0.006	0.856±0.006	0.889±0.011	0.818±0.033	0.896±0.002
		6000	0.832±0.014	**0.862±0.004**	0.844±0.005	0.847±0.006	0.600±0.041	0.853±0.003
		full	**0.826±0.035**	0.785±0.005	0.572±0.006	0.691±0.040	0.649±0.009	0.717±0.006
	Epithelial & Stromal Tumor	3000	0.860±0.049	0.888±0.014	0.899±0.018	**0.913±0.008**	0.677±0.036	0.746±0.005
		6000	0.824±0.007	0.828±0.014	0.856±0.014	**0.892±0.008**	0.543±0.025	0.681±0.006
		full	**0.717±0.059**	0.638±0.005	0.556±0.006	0.686±0.024	0.471±0.026	0.571±0.003

Table 3: Average F1*NMI score with standard deviation for fine-grained anomaly detection on single-cell transcriptomics datasets.

and cells (samples). We present experiments sequentially for anomaly detection, batch effect correction and anomaly subtyping as outlined below.

Effect of memory bank for anomaly detection. As introduced in section 3.1, we employ a memory block to enhance the training of the anomaly detector. In particular, we evaluate ACsleuth without the memory block in the anomaly detection module, and results are illustrated in Figure 2. Upon the removal of the memory block, the anomaly detection module easily collapses, significantly impairing affect the subsequent subtyping task.

Essence of the batch correction module. We have discussed that batch effect correction is imperative for the scRNA-seq data analysis. It is straightforward to question whether the subtyping results will be worse if we directly subtype anomalies samples after detecting them, following the same settings as baseline methods. As shown in figure 2, the direct connection of two tasks is considerably adverse compared with the original ACsleuth.

Validity of the reconstruction error in the fine-grained detection task. At last, we carry out an ablation study for the input of the fine-grained anomaly detection module. The reconstruction error generated from anomaly detection module (*Module I*) serves as anomalous scores leveraged to identify anomalies in *Module I*, encompassing similarity information among them. If we merely consider the original gene expression matrix as the input of subtype clustering, there will be an underutilization of information regarding anomalous cells, as empirically demonstrated in figure 2.

5 Conclusion

In this paper, we innovatively propose ACsleuth, a comprehensive workflow that sequentially integrates anomaly detection with fine-grained anomaly detection, which further contains a batch correction module to alleviate bias inherent in the raw data. More precisely, we first employ a GAN-based model to detect anomalies according to the reconstruction error, then capture batch effects from filtered normal samples and correct them on the detected anomalies. We also prove that the domain adaptation module is effective for cross-domain tasks. Finally, we utilize the DESC clustering algorithm to fine-grained detecting anomalous cells. ACsleuth outperforms state-of-the-art approaches in anomaly detection methods for tabular data and their simple combinations with deep clustering algorithms. Empirically results strongly improve ACsleuth' superiority, demonstrating its generalizability and robustness. We'd like to further explore the potential of the domain adapyative fine-grained anomaly detection workflow for tabular data generalized to another fields.

Acknowledgments

References

[Aggarwal, 2017] Charu C Aggarwal. *An introduction to outlier analysis*. Springer, 2017.

[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.

[Amodio *et al.*, 2019] Matthew Amodio, David Van Dijk, Krishnan Srinivasan, William S Chen, Hussein Mohsen, Kevin R Moon, Allison Campbell, Yujiao Zhao, Xiaomei Wang, Manjunatha Venkataswamy, et al. Exploring single-cell data with deep multitasking neural networks. *Nature methods*, 16(11):1139–1145, 2019.

[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.

[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.*, 2017] Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM International Conference on Data Mining (SDM)*, pages 90–98, 2017.

[Chen *et al.*, 2020] Liang Chen, Weinan Wang, Yuyao Zhai, and Minghua Deng. Deep soft k-means clustering with self-training for single-cell rna sequence data. *NAR Genomics and Bioinformatics*, 2(2), 2020.

[Cheng *et al.*, 2023] Furui Cheng, Mark S Keller, Huamin Qu, Nils Gehlenborg, and Qianwen Wang. Polyphony: an interactive transfer learning framework for single-cell data analysis. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):591–601, 2023.

[De Falco *et al.*, 2023] Antonio De Falco, Francesca Caruso, Xiao-Dong Su, Antonio Iavarone, and Michele Ceccarelli. A variational algorithm to detect the clonal copy number substructure of tumors from scrna-seq data. *Nature Communications*, 14(1):1074, 2023.

[Di Mattia *et al.*, 2019] Federico Di Mattia, Paolo Galeone, Michele De Simoni, and Emanuele Ghelfi. A survey on gans for anomaly detection. *arXiv preprint arXiv:1906.11632*, 2019.

[Gao *et al.*, 2021] Ruli Gao, Shanshan Bai, Ying C Henderson, Yiyun Lin, Aislyn Schalck, Yun Yan, Tapsi Kumar, Min Hu, Emi Sei, Alexander Davis, et al. Delineating copy number and clonal substructure in human tumors from single-cell transcriptomes. *Nature biotechnology*, 39(5):599–608, 2021.

[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.

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

[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.

[Hu *et al.*, 2020] Jian Hu, Xiangjie Li, Gang Hu, Yafei Lyu, and Mingyao Li. Iterative transfer learning with neural network for clustering and cell type classification in single-cell rna-seq analysis. *Nature Machine Intelligence*, 2(10):1–12, 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.

[Kolouri *et al.*, 2016] Soheil Kolouri, Yang Zou, and Gustavo K Rohde. Sliced wasserstein kernels for probability distributions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5258–5267, 2016.

[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 Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1505–1511. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track.

[Liznerski *et al.*, 2020] Philipp Liznerski, Lukas Ruff, Robert A Vandermeulen, Billy Joe Franks, Marius Kloft, and Klaus-Robert Müller. Explainable deep one-class classification. *arXiv preprint arXiv:2007.01760*, 2020.

[Lotfollahi *et al.*, 2022] Mohammad Lotfollahi, Mohsen Naghipourfar, Malte D Luecken, Matin Khajavi, Maren Büttner, Marco Wagenstetter, Žiga Avsec, Adam Gayoso, Nir Yosef, Marta Interlandi, et al. Mapping single-cell data to reference atlases by transfer learning. *Nature biotechnology*, 40(1):121–130, 2022.

[Lu *et al.*, 2021] Yang Young Lu, C Yu Timothy, Giancarlo Bonora, and William Stafford Noble. Ace: Explaining cluster from an adversarial perspective. In *International Conference on Machine Learning*, pages 7156–7167. PMLR, 2021.

[MacQueen and others, 1967] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[Pliner *et al.*, 2019] Hannah A Pliner, Jay Shendure, and Cole Trapnell. Supervised classification enables rapid annotation of cell atlases. *Nature methods*, 16(10):983–986, 2019.

[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.

[Sakurada and Yairi, 2014] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, MLSDA'14, pages 4–11, New York, NY, USA, 2014. Association for Computing Machinery.

[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.

[Stuart *et al.*, 2021] Tim Stuart, Avi Srivastava, Shaista Madad, Caleb A. Lareau, and Rahul Satija. Single-cell chromatin state analysis with signac. *Nature Methods*, 18(11):1333–1341, 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.

[Wang and Deng, 2018] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[Wang *et al.*, 2022] Hai-Yun Wang, Jian-Ping Zhao, Chun-Hou Zheng, and Yan-Sen Su. scCNC: a method based on capsule network for clustering scRNA-seq data. *Bioinformatics*, 38(15):3703–3709, 06 2022.

[Wolf *et al.*, 2018] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19:1–5, 2018.

[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.

[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.

[Zenati *et al.*, 2018] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*, 2018.

[Zhai *et al.*, 2023] Yuyao Zhai, Liang Chen, and Minghua Deng. Realistic cell type annotation and discovery for single-cell rna-seq data. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 4967–4974. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Main Track.

[Zhou *et al.*, 2022] Zihan Zhou, Zijia Du, and Somali Chaterji. Kratos: Context-aware cell type classification and interpretation using joint dimensionality reduction and clustering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2616–2625, 2022.

A Proofs

A.1 Proof of Theorem 3.1

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

$$\begin{aligned} MMD^2\left(\boldsymbol{\delta}_m^x, \boldsymbol{\delta}_n^y\right) &= \frac{1}{m(m-1)} \sum_i^m \sum_{j \neq i}^m k(\boldsymbol{\delta}_i^x, \boldsymbol{\delta}_j^x) \\ &+ \frac{1}{n(n-1)} \sum_i^n \sum_{j \neq i}^n k(\boldsymbol{\delta}_i^y, \boldsymbol{\delta}_j^y) - \frac{2}{mn} \sum_i^m \sum_j^n k(\boldsymbol{\delta}_i^x, \boldsymbol{\delta}_j^y) \\ &= \sum_i^{m+n} \sum_{j \neq i}^{m+n} k(\boldsymbol{\delta}_i, \boldsymbol{\delta}_j) \gamma(s_i, s_j) \end{aligned} \tag{26}$$

where $\boldsymbol{\delta}_i$ denotes unlabeled total samples, and the adjustment coefficients γ are defined as:

$$\gamma(s_i, s_j) = \begin{cases} \frac{1}{m(m-1)}, & s_i = s_j = 0 \\ \frac{1}{n(n-1)}, & s_i = s_j = 1 \\ \frac{-1}{mn}, & s_i \neq s_j \end{cases} \tag{27}$$

If $s_i = 1$, sample i will be classified as anomaly; otherwise, it will be classified as normal. By the way, Appendix A.1 has been proved. □

A.2 Proof of Theorem 3.2

Proof. According to (10), we define the two-dimensional sequence $\{\gamma_{mn}\}$ as:

$$\begin{gathered} \gamma_{00} = \frac{1}{m(m-1)}, \quad \gamma_{01} = \gamma_{10} = \frac{-1}{mn} \gamma_{11} = \frac{1}{n(n-1)} \\ \forall i, j \geq 2, \quad \gamma_{ij} = 0 \end{gathered} \tag{28}$$

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

$$H(x, y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \gamma_{ij} x^i y^j \tag{29}$$

where $(x, y) \in \mathbb{D} \coloneqq [0, 1] \times [0, 1]$.

According to [Bradshaw and Vignat, 2023; Amdeberhan *et al.*, 2012], the extension of Ramanujan's master theorem in the k-dimensional case have been proposed as Lemma A.1.

Lemma A.1. *If a complex-valued function $f(x_1, \cdots, x_k)$ has an expansion:*

$$f(x_1, \cdots, x_k) = \sum_{n_1, \cdots, n_k}^{\infty} g(n_1, \cdots, n_k) \prod_{i=1}^{k} \frac{(-1)^{n_i}}{n_i!} x_i^{n_i} \tag{30}$$

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:

$$\begin{aligned} &\mathcal{M}[f(x_1, \cdots, x_k)](s_1, \cdots, s_k) \\ &\coloneqq \int_{\mathbb{R}_+^k} \prod_{i=1}^{k} x_i^{s_i - 1} f(x_1, \cdots, x_k) dx_1 \cdots dx_k \\ &= \prod_{i=1}^{k} \Gamma(s_i) g(-s_1, \cdots, -s_k) \end{aligned} \tag{31}$$

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

Note that $H(-x, -y)$ satisfies:

$$H(-x, -y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \gamma_{ij} \Gamma(i+1) \Gamma(j+1) \frac{(-1)^i (-1)^j}{i! j!} x^i y^j \tag{32}$$

By Lemma A.1, the k-dimensional Mellin transform follows:

$$\begin{aligned} &\mathcal{M}[H(-x, -y)](s, t) \\ &\coloneqq \int_{\mathbb{D}} x^{s-1} y^{t-1} H(-x, -y) dx dy \\ &= \Gamma(s) \Gamma(t) \Gamma(1-s) \Gamma(1-t) \gamma_c(-s, -t) \\ &= \frac{\pi^2}{\sin \pi s \sin \pi t} \gamma_c(-s, -t) \end{aligned} \tag{33}$$

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

$$\begin{aligned} &\frac{\pi^2}{\sin \pi s \sin \pi t} \gamma_c(-s, -t) \\ &= \int_{\mathbb{D}} x^{s-1} y^{t-1} H(-x, -y) dx dy \\ &= \int_{\mathbb{D}} \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \gamma_{ij} x^{i+s-1} y^{j+t-1} dx dy \\ &= \int_{\mathbb{D}} (\gamma_{00} x^{s-1} y^{t-1} + \gamma_{10} x^s y^{t-1} + \gamma_{01} x^{s-1} y^t + \gamma_{11} x^s y^t) dx dy \\ &= \frac{[m(m-1)]^{-1}}{st} + \frac{(mn)^{-1}}{(s+1)t} + \frac{(mn)^{-1}}{s(t+1)} + \frac{[n(n-1)]^{-1}}{(s+1)(t+1)} \end{aligned} \tag{34}$$

By replacing $-s$ and $-t$, the original theorem is thereby proven. □

A.3 Proof of Theorem 3.3

Proof. Let the reference samples belong to batch k and the target samples belong to batch k'. If Assumption 3.1 holds, the reference samples $\boldsymbol{\zeta}_l$ satisfy:

$$\begin{cases} \boldsymbol{\zeta}_l = \boldsymbol{\zeta}_l^* + \boldsymbol{b}_l^k + \boldsymbol{\epsilon}_l \\ \widehat{\boldsymbol{\zeta}}_l \coloneqq G(\boldsymbol{\zeta}_l) = \widehat{\boldsymbol{\zeta}}_l^* + \boldsymbol{b}_l^k \end{cases} \tag{35}$$

where $\widehat{\boldsymbol{\zeta}}_m^* \sim P_{\widehat{\boldsymbol{\zeta}}^*}, \boldsymbol{b}_l^k \sim P_{\boldsymbol{b}^k}$, and l is the reference sample size. Because the generator doesn't learn the distribution of random noise, $\widehat{\boldsymbol{\zeta}}_l$ excludes $\boldsymbol{\epsilon}_l$.

Considering that the target samples $\widehat{\boldsymbol{y}}_n, \widehat{\boldsymbol{x}}_m$ are reconstructed with the reference information,

$$\begin{cases} \boldsymbol{x}_m = \boldsymbol{x}_m^* + \boldsymbol{b}_m^{k'} + \boldsymbol{\epsilon}_m \\ \widehat{\boldsymbol{x}}_m = \widehat{\boldsymbol{\zeta}}_m^* + \boldsymbol{b}_m^k \\ \boldsymbol{y}_n = \boldsymbol{y}_n^* + \boldsymbol{b}_n^{k'} + \boldsymbol{\epsilon}_n \\ \widehat{\boldsymbol{y}}_n = \widehat{\boldsymbol{\zeta}}_n^* + \boldsymbol{b}_n^k \end{cases} \tag{36}$$

where $\widehat{\boldsymbol{\zeta}}_m^*, \widehat{\boldsymbol{\zeta}}_n^* \overset{i.i.d.}{\sim} P_{\widehat{\boldsymbol{\zeta}}^*}$, $\boldsymbol{b}_m^k, \boldsymbol{b}_n^k \overset{i.i.d.}{\sim} P_{\boldsymbol{b}^k}$ and $\boldsymbol{b}_m^{k'}, \boldsymbol{b}_n^{k'} \overset{i.i.d.}{\sim} P_{\boldsymbol{b}^{k'}}$. Subsequently, the reconstruction errors satisfy:

$$\begin{cases} \boldsymbol{\delta}_m^x = \boldsymbol{x}_m - \widehat{\boldsymbol{x}}_m = \boldsymbol{x}_m^* - \widehat{\boldsymbol{\zeta}}_m^* + \boldsymbol{b}_m^{k'} - \boldsymbol{b}_m^k + \boldsymbol{\epsilon}_m \\ \boldsymbol{\delta}_n^y = \boldsymbol{y}_n - \widehat{\boldsymbol{y}}_n = \boldsymbol{y}_n^* - \widehat{\boldsymbol{\zeta}}_n^* + \boldsymbol{b}_n^{k'} - \boldsymbol{b}_n^k + \boldsymbol{\epsilon}_n \end{cases} \tag{37}$$

For a more concise representation, we define:

$$\begin{cases} \boldsymbol{\delta}_m^{x*} = \boldsymbol{x}_m^* - \widehat{\boldsymbol{\zeta}}_m^*, \quad \boldsymbol{\delta}_m^b = \boldsymbol{b}_m^{k'} - \boldsymbol{b}_m^k + \boldsymbol{\epsilon}_m \\ \boldsymbol{\delta}_m^{y*} = \boldsymbol{y}_n^* - \widehat{\boldsymbol{\zeta}}_n^*, \quad \boldsymbol{\delta}_n^b = \boldsymbol{b}_n^{k'} - \boldsymbol{b}_n^k + \boldsymbol{\epsilon}_n \end{cases} \tag{38}$$

Under these symbol representations, if MMD is induced by linear kernel, the kernel is expandable as follows:

$$\begin{aligned} k(\boldsymbol{\delta}_i^x, \boldsymbol{\delta}_j^x) &= k(\boldsymbol{\delta}_i^{x*}, \boldsymbol{\delta}_j^{x*}) + k(\boldsymbol{\delta}_i^b, \boldsymbol{\delta}_j^b) + k(\boldsymbol{\delta}_i^{x*}, \boldsymbol{\delta}_j^b) + k(\boldsymbol{\delta}_i^b, \boldsymbol{\delta}_j^{x*}) \\ k(\boldsymbol{\delta}_i^y, \boldsymbol{\delta}_j^y) &= k(\boldsymbol{\delta}_i^{y*}, \boldsymbol{\delta}_j^{y*}) + k(\boldsymbol{\delta}_i^b, \boldsymbol{\delta}_j^b) + k(\boldsymbol{\delta}_i^{y*}, \boldsymbol{\delta}_j^b) + k(\boldsymbol{\delta}_i^b, \boldsymbol{\delta}_j^{y*}) \\ k(\boldsymbol{\delta}_i^x, \boldsymbol{\delta}_j^y) &= k(\boldsymbol{\delta}_i^{x*}, \boldsymbol{\delta}_j^{y*}) + k(\boldsymbol{\delta}_i^b, \boldsymbol{\delta}_j^b) + k(\boldsymbol{\delta}_i^{x*}, \boldsymbol{\delta}_j^b) + k(\boldsymbol{\delta}_i^b, \boldsymbol{\delta}_j^{y*}) \end{aligned} \tag{39}$$

Subsequently, considering (26), therefore it follows that:

$$\begin{aligned} MMD^2(\boldsymbol{\delta}_m^x, \boldsymbol{\delta}_n^y) =& \frac{1}{m(m-1)} \sum_i^m \sum_{j \neq i}^m k(\boldsymbol{\delta}_i^x, \boldsymbol{\delta}_j^x) + \frac{1}{n(n-1)} \sum_i^n \sum_{j \neq i}^n k(\boldsymbol{\delta} \\ &- \frac{2}{mn} \sum_i^m \sum_j^n k(\boldsymbol{\delta}_i^x, \boldsymbol{\delta}_j^y) \\ =& MMD^2(\boldsymbol{\delta}_m^{x*}, \boldsymbol{\delta}_n^{y*}) + MMD^2(\boldsymbol{\delta}_m^b, \boldsymbol{\delta}_n^b) + 2R_{mn}^x + 2 \end{aligned} \tag{40}$$

where the remainder term R_{mn}^x, R_{mn}^y are defined as:

$$\begin{aligned} R_{mn}^x &\coloneqq \frac{1}{m(m-1)} \sum_i^m \sum_{j \neq i}^m \boldsymbol{\delta}_i^{b^T} \boldsymbol{\delta}_j^{x*} - \frac{1}{n^2} \sum_i^n \sum_j^n \boldsymbol{\delta}_i^{b^T} \boldsymbol{\delta}_j^{x*} \\ R_{mn}^y &\coloneqq \frac{1}{n(n-1)} \sum_i^m \sum_{j \neq i}^m \boldsymbol{\delta}_i^{b^T} \boldsymbol{\delta}_j^{y*} - \frac{1}{m^2} \sum_i^m \sum_j^m \boldsymbol{\delta}_i^{b^T} \boldsymbol{\delta}_j^{y*} \end{aligned} \tag{41}$$

Before analyzing the transfer error of $MMD^2(\boldsymbol{\delta}_m^x, \boldsymbol{\delta}_n^y)$, let's first discuss the convergence of each term in (40). For the first and second terms, we analyze them using the following Lemma A.2 [Gretton *et al.*, 2012].

Lemma A.2. *Assume* $0 \leq k(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq K$. *Then:*

$$\mathbb{P}\left(\left|MMD^2(\boldsymbol{X}, \boldsymbol{Y}) - MMD^2(p, q)\right| \geq \varepsilon\right) \leq 2\exp\left(\frac{-\varepsilon^2 mn}{8K^2(m+n)}\right) \tag{42}$$

where $\boldsymbol{x}_i \sim p, \boldsymbol{y}_j \sim q$.

Then, we obtain:

$$\mathbb{P}\left(\left|MMD^2(\boldsymbol{\delta}_m^{x*}, \boldsymbol{\delta}_n^{y*}) - MMD^2(P_{\boldsymbol{\delta}^{x*}}, P_{\boldsymbol{\delta}^{y*}})\right| \geq \varepsilon\right) \leq 2\exp\left(\frac{-Cn\varepsilon^2}{8(1+C)K_+^{x\,2}}\right) \tag{43}$$

$$\mathbb{P}\left(\left|MMD^2(\boldsymbol{\delta}_m^{b}, \boldsymbol{\delta}_n^{b}) - 0\right| \geq \varepsilon\right) \leq 2\exp\left(\frac{-Cn\varepsilon^2}{8(1+C)K_+^{b\,2}}\right) \tag{44}$$

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

For the third term R_{mn}^x in (40), we employ the following Lemma A.3 to discuss the convergence, and the proof is available at Appendix A.4.

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

$$\mathbb{P}\left(\left|\sum_{i=1}^{k} x_i\right| \geq \varepsilon\right) \leq \mathbb{P}\left(\sum_{i=1}^{k} |x_i| \geq \varepsilon\right) \leq \sum_{i=1}^{k} \mathbb{P}\left(|x_i| \geq \frac{\varepsilon}{k}\right) \tag{45}$$

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

Considering that $\boldsymbol{\delta}^b$ and $\boldsymbol{\delta}^{x*}$ are mutually independent, we define the random variable $\xi \coloneqq {\boldsymbol{\delta}^b}^T \boldsymbol{\delta}^{x*} \in \mathbb{R}$. According to Lemma A.3 and Hoeffding's Inequality [Hoeffding, 1994], R_{mn}^x can be bounded as follows:

$$\begin{aligned}
\mathbb{P}(\left|R_{mn}^x\right| \geq \varepsilon) &= \mathbb{P}\left(\left|\frac{1}{m(m-1)}\sum_{i=1}^{m(m-1)} \xi_i - \frac{1}{n^2}\sum_{j=1}^{n^2} \xi_j\right| \geq \varepsilon\right) \\
&= \mathbb{P}\left(\left|\frac{1}{m(m-1)}\sum_{i=1}^{m(m-1)} \xi_i - \mathbb{E}(\xi) + \mathbb{E}(\xi) - \frac{1}{n^2}\sum_{j=1}^{n^2} \xi_j\right| \geq \varepsilon\right) \\
&\leq \mathbb{P}\left(\left|\frac{1}{m(m-1)}\sum_{i=1}^{m(m-1)} \xi_i - \mathbb{E}(\xi)\right| \geq \frac{\varepsilon}{2}\right) + \mathbb{P}\left(\left|\frac{1}{n^2}\sum_{j=1}^{n^2} \xi_j - \mathbb{E}(\xi)\right| \geq \frac{\varepsilon}{2}\right) \\
&\leq 2\exp\left(\frac{-2m(m-1)(\varepsilon/2)^2}{K_+^{\xi\,2}}\right) + 2\exp\left(\frac{-2n^2(\varepsilon/2)^2}{K_+^{\xi\,2}}\right) \\
&\leq 4\exp\left(\frac{-m(m-1)\varepsilon^2}{2K_+^{\xi\,2}}\right)
\end{aligned} \tag{46}$$

where $K_+^\xi \coloneqq \sup_{i,j}(\xi_i - \xi_j)$

Similarly, for the forth term R_{mn}^y in (40), we define the random variable $\theta \coloneqq {\boldsymbol{\delta}^b}^T \boldsymbol{\delta}^{y*} \in \mathbb{R}$, and $K_+^\theta \coloneqq \sup_{i,j}(\theta_i - \theta_j)$. Thus, R_{mn}^y can be bounded as follows:

$$\mathbb{P}(\left|R_{mn}^y\right| \geq \varepsilon) \leq 4\exp\left(\frac{-m^2\varepsilon^2}{2K_+^{\theta\,2}}\right) \tag{47}$$

Combined (43), (45) and (46), we also employ Lemma A.3 to obtain:

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

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

$$\frac{Cn}{1+C} < m(m-1) < m^2 \tag{48}$$

Finally, we can obtain:

$$\mathbb{P}\left(\left|MMD^2(\boldsymbol{\delta}_m^{x}, \boldsymbol{\delta}_n^{y}) - MMD^2(P_{\boldsymbol{\delta}^{x*}}, P_{\boldsymbol{\delta}^{y*}})\right| \geq \varepsilon\right) \leq 12\exp\left(\frac{-Cn}{128(1+C}\right. \tag{49}$$

If $\alpha \coloneqq 12, \beta \coloneqq (128K_+^2)^{-1}$, the original theorem will be proven. □

A.4 Proof of Lemma A.3

Proof. On one hand, according to the triangle inequality, we have:

$$\left|\sum_{i=1}^{k} x_i\right| \leq \sum_{i=1}^{k} |x_i| \tag{50}$$

which also indicates

$$\left\{\left|\sum_{i=1}^{k} x_i\right| \geq \varepsilon\right\} \subset \left\{\sum_{i=1}^{k} |x_i| \geq \varepsilon\right\} \tag{51}$$

On the other hand, the following relationship always holds:

$$\left\{\sum_{i=1}^{k} |x_i| \geq \varepsilon\right\} \subset \bigcup_{i=1}^{k} \left\{|x_i| \geq \frac{\varepsilon}{k}\right\} \tag{52}$$

Thus, we have:

$$\mathbb{P}\left(\left|\sum_{i=1}^{k} x_i\right| \geq \varepsilon\right) \leq \mathbb{P}\left(\sum_{i=1}^{k} |x_i| \geq \varepsilon\right) \leq \sum_{i=1}^{k} \mathbb{P}\left(|x_i| \geq \frac{\varepsilon}{k}\right) \tag{53}$$

□

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