LAKEHEAD UNIVERSITY

# A Hybrid Framework for Weak Signal Learning in Breast Cancer Prediction Using Metabolomics Data

by

Jiahui Fang

Student ID: 1273946

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE FACULTY OF GRADUATE STUDIES
OF LAKEHEAD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

Master OF Computer Science

April 2026

## Thesis Committee

---

Committee Member (Internal)

Dr. M. Mazhar Rathore

*(Assistant Professor, Department of Computer Science, Lakehead University, Thunder Bay, Ontario, Canada.)*

---

Committee Member (External)

Dr. Yong Deng

*(Assistant Professor, Software Engineering, Lakehead University, Thunder Bay, Ontario, Canada.)*

---

Supervisor

Dr. Abedalrhman Alkhateeb

*(Assistant Professor, Department of Computer Science, Lakehead University, Thunder Bay, Ontario, Canada.)*

# *Abstract*

Clinical MS-based metabolomics prediction in small cohorts is often constrained by weak class separation, class imbalance, and heterogeneous sample reliability. Under these conditions, predictive performance is limited not by a single factor, but by the combined effects of unstable feature structure, limited minority class support, and unequal learning difficulty across samples. Existing methods have addressed some of these challenges separately, but a unified framework for stable learning under weak signal conditions remains insufficiently developed.

This thesis studies weak signal clinical metabolomics prediction as a structured learning problem rather than a standard supervised classification task. To address this setting, a unified and fold-disciplined framework is developed that integrates transformer representation learning, conditional generative adversarial network (cGAN) augmentation, and curriculum learning (CL) within stratified cross-validation (CV). The framework is designed to provide a more stable representation space, strengthen minority class support during training, and organize training in a way that better reflects variation in sample reliability.

The proposed framework is evaluated on two breast cancer-related metabolomics datasets with different signal conditions. ST004145 is used as the primary weak signal dataset, while ST000355 is used as a strong signal stability-check dataset. On ST004145, the full hybrid model achieved the highest mean Area Under the ROC Curve (AUC) among the compared methods ($0.6794 \pm 0.0871$). Ablation analysis further indicated that both cGAN minority support and CL difficulty-aware training contributed to the final performance pattern. On ST000355, performance differences between models were much smaller, although the proposed model remained highly competitive, with an AUC of $0.9896 \pm 0.0195$.

These findings suggest that the value of the proposed framework is most evident under weak signal conditions, where predictive robustness depends on addressing multiple interacting sources of instability within a single training design. Therefore, this thesis contributes a more structured methodological perspective on weak signal clinical metabolomics prediction and supports the usefulness of a unified, fold-disciplined learning framework in small, class imbalanced clinical cohorts.

# *Acknowledgements*

I would like to thank my supervisor, Dr. Abedalrhman Alkhateeb, for his support, guidance, and encouragement. I am deeply grateful for his patience, understanding, and the time he devoted to this work. Throughout this work, we had many meetings, both in person and on Zoom, on weekdays as well as weekends. He was always willing to make time to discuss my work and offer helpful advice. Even when I was not fully prepared for meetings, he was understanding and accommodating. I feel very fortunate to have such a supportive supervisor, who was always respectful of my progress and pace.

I am also grateful to my thesis committee members, Dr. M. Mazhar Rathore and Dr. Yong Deng, who have shown their continued interest in my work and offered encouragement. Moreover, I would like to acknowledge the Department of Computer Science and the Faculty of Graduate Studies for providing a supportive research environment.

Finally, I am grateful to my family. Although my parents are not nearby, their support has given me the strength to overcome challenges. I am especially grateful to my daughters for their support.

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **DNA** | **D**eoxyribo**n**ucleic **A**cid |
| **RNA** | **R**ibo**n**ucleic **A**cid |
| **NMR** | **N**uclear **M**agnetic **R**esonance |
| **MS** | **M**ass **S**pectrometry |
| **SNR** | **S**ignal-to-**N**oise **R**atio |
| **GAN** | **G**enerative **A**dversarial **N**etwork |
| **cGAN** | **C**onditional **G**enerative **A**dversarial **N**etwork |
| **CL** | **C**urriculum **L**earning |
| **CV** | **C**ross-**V**alidation |
| **LOD** | **L**imit-**o**f-**D**etection |
| **ML** | **M**achine **L**earning |
| **DL** | **D**eep **L**earning |
| **SMOTE** | **S**ynthetic **M**inority **O**ver-sampling **T**echnique |
| **PCA** | **P**rincipal **C**omponent **A**nalysis |
| **FDR** | **F**alse **D**iscovery **R**ate |
| **JSON** | **J**ava**S**cript **O**bject **N**otation |
| **LR** | **L**ogistic **R**egression |
| **ROC** | **R**eceiver **O**perating **C**haracteristic |
| **AUC** | **A**rea **U**nder the ROC **C**urve |
| **PR** | **P**recision–**R**ecall |
| **PR-AUC** | **P**recision–**R**ecall **A**rea **U**nder the **C**urve |
| **AP** | **A**verage **P**recision |
| **TPR** | **T**rue **P**ositive **R**ate |
| **FPR** | **F**alse **P**ositive **R**ate |
| **GELU** | **G**aussian **E**rror **L**inear **U**nit |

| | |
|---|---|
| **SVM** | **S**upport **V**ector **M**achine |
| **LightGBM** | **Light** **G**radient **B**oosting **M**achine |
| **OOF** | **O**ut-**o**f-**F**old |

# Chapter 1

# Introduction

## 1.1 Metabolomics and Disease Phenotype

Metabolism refers to the set of biochemical reactions that sustain life, including anabolism and catabolism [1]. Metabolites are small-molecule products or intermediates of metabolic reactions that reflect the combined effects of biological regulation and environmental exposures. Metabolomics is the analysis of small-molecule metabolites in biological systems under specific conditions [2]. In this study, metabolomic profiles are regarded as integrated molecular readouts of the physiological state and thus serve as input features for classification models [3].

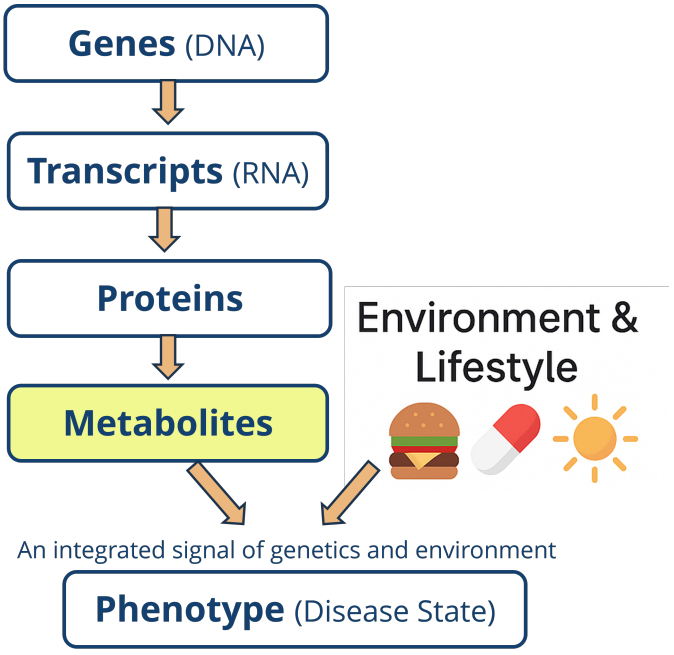


Figure 1.1: From Genes to Phenotypes

Genes encoded in deoxyribonucleic acid (DNA) are transcribed into ribonucleic acid (RNA), followed by proteins, metabolites, and ultimately phenotype, as illustrated in Figure 1.1. Unlike genomic and transcriptomic data, metabolomic measurements reflect the biochemical results of cellular activity [4]. Because metabolite levels are shaped by both genetic and environmental factors, such as diet, lifestyle, and exposure to toxins, metabolite profiles are often closer to the current physiological state [5]. However, their use for clinically meaningful prediction depends on measurements that are reliable and sufficiently informative [6].

## 1.2 Measurement Platforms: NMR vs. MS

Metabolomics research mainly relies on two measurement platforms: nuclear magnetic resonance (NMR) spectroscopy, which probes atomic nuclei in a magnetic field through absorption of electromagnetic radiation at specific frequencies [7], and mass spectrometry (MS), which analyzes ionized species according to their mass-to-charge ratios [8]. The choice of platform strongly affects measurement sensitivity and quantification reliability, and also shapes the structure and noise characteristics of the resulting data [9, 10].

NMR is widely used as a stable quantitative platform in metabolomics due to its non-destructive nature, high reproducibility, and relatively simple sample preparation [10]. However, the sensitivity of NMR is limited. NMR is primarily sensitive to metabolites in the micromolar range ($\mu$M) [6, 10]. This sensitivity limit is critical when used in the prediction of breast cancer because some metabolites associated with the disease can vary on the nanomolar (nM) scale, which is below the sensitivity limit of NMR [10]. This may lead to less informative features for predictive modeling.

To address the sensitivity bottleneck, this study uses MS for nanomolar-level metabolite detection. This is particularly important in the context of breast cancer prediction, where phenotype-specific information is often found in low abundance variations [6]. However, this sensitivity presents particular structural challenges as a result of the physics of the measurements. In other words, MS improves detectability but does not necessarily yield reliable measurements because signals near the detection limit are more likely to be noisy, missing, or suppressed by ion competition. MS detection is based on ionization efficiency, a competitive process in which high-abundance molecules can suppress the signal of lower-abundance compounds (ion suppression). As a result, low-abundance metabolites may fall below the detection limit, producing weak or missing signals [11, 12]. Furthermore, the aggregation of distinct acquisition modes (e.g., positive and negative ionization polarities) introduces platform-dependent heterogeneity [9]. Consequently, the resulting datasets are often characterized by high sparsity and

signal instability [6]. These properties are commonly observed in MS data, creating the specific weak-signal challenges that are formally described in the following section.

## 1.3 Problem Statement

The main bottleneck in translating MS-based metabolomics into clinical utility is learning stable predictive patterns in weak signal settings [3, 6]. In this thesis, weak signal mainly refers to a weakly observed association between the phenotype and the measured metabolome (small effect size) in heterogeneous clinical cohorts, which is further compounded by measurement noise and missingness in MS data. Operationally,weak-signal settings are characterized by three properties: (i) small observed effect sizes, often reflected in few features remaining significant after multiple-testing correction; (ii) substantial technical noise and structured missingness that increase variance and reduce effective Signal-to-Noise Ratio (SNR); and (iii) limited class separability in small-cohort settings.

Throughout the remainder of the thesis, we use this weak signal definition as a consistent basis for robustness analysis, representation learning, imbalance-aware training, and difficulty-aware model selection. MS measurements can exhibit ionization-related biases and platform-dependent heterogeneity, which lead to sparse and unstable feature matrices in clinical cohorts, as discussed in Section 1.2. In many cohorts, these effects can be comparable to phenotype-related variation, resulting in datasets with low effective signal-to-noise ratios, structured missingness, and substantial measurement variability [11–13]. Consequently, clinically realistic MS datasets are often weak-signal and noisy, while strong-signal datasets exist but are relatively rare [13].

These properties are especially significant for prediction. Platform and batch effects can induce correlated intensity shifts across large portions of the feature set, creating noisy log-scale patterns that may be learned as predictive structure even though they do not reflect true biology [13, 14]. Missingness is often structured. Non-detection can reflect detection limits, peak calling, or model-specific acquisition biases rather than biological absence [12, 15]. Low effective SNR further reduces separability because informative signals can be suppressed by conservative replacement under non-detection assumptions while inflating measurement variation [12, 16]. As a result, the observed feature matrix may contain structured distortion in addition to weak class separation, making predictive learning sensitive to measurement noise, preprocessing decisions, and data partitioning [13, 17].

Overall, these characteristics make weak-signal clinical MS-metabolomics learning difficult not only because class separation is limited, but also because training is affected by instability in the observed feature space, minority class fragility, and variation in sample reliability. As a result, the challenge is not simply to apply an existing predictive model to processed metabolomic data, but to identify what methodological gaps remain under this weak-signal setting and how these gaps guide the design of the present thesis.

## 1.4 Research Gap

Despite continued progress in metabolomics based prediction, an important methodological gap remains in weak signal prediction using clinical MS data. Existing studies have advanced prediction methods, preprocessing practice, and machine learning (ML) applications in this area, but a unified approach to stable learning under noisy and heterogeneous signal conditions remains underdeveloped [6, 18].

First, there remains a need for more stable sample level representation under weak signal. When the informative structure is weak and sensitive to measurement noise, missingness, or preprocessing variation, the processed feature space may not provide a sufficiently robust basis for downstream classification [15, 19]. In this setting, the issue is not only that individual features may be weakly informative but also that the overall predictive structure may vary across folds and across samples. Because the learned representation is already unstable under weak signal, limited informative signal from the minority class becomes insufficient to support stable boundary formation. This gap motivates the development of more stable representations [20, 21].

Building on the need for stable representation, there remains a need for stronger minority class support during model development. In imbalanced small-cohort settings, limited minority support can weaken the learning decision boundary and increase instability across folds [22, 23]. When the learned representation is already unstable under weak signal, limited minority evidence within each training fold becomes even less sufficient to support stable model fitting and evaluation. This gap motivates the development of stronger minority support during training [22, 24].

However, improved representation and minority class support are not sufficient to resolve the variation in sample level learning difficulty. In such data, some samples are more ambiguous or less reliable than others, but standard supervised training generally treats all cases as equally informative under a uniform training schedule [25, 26]. This creates a mismatch between heterogeneous sample reliability and the way learning is typically organized. This gap motivates the introduction of difficulty aware training [25].

Together, these gaps indicate that the central challenge is not simply to improve prediction accuracy through isolated techniques, but to develop a unified methodological response to this setting. More specifically, what remains insufficiently developed is a fold-disciplined framework that jointly addresses representation instability, minority class fragility, and heterogeneous sample difficulty [18]. In this sense, the three components are not independent additions, but sequential responses to interconnected challenges. This is the central gap addressed by this thesis and directly motivates the research objectives defined in the following section.

## 1.5 Research Objectives

Given the research gaps identified above and the weak signal reliability challenges, this thesis defines the following research objectives. These objectives are organized according to the complementary methodological roles of the proposed components, rather than their execution order within the training pipeline.

**Objective 1: Stable Feature Representation under Weak Signal** Develop a transformer encoder that maps the full feature vector into a compact embedding to capture global structure at the representation level and support more robust downstream learning in low effective SNR settings.

**Objective 2: Minority Class Stabilization under Class Imbalance** Evaluate whether cGAN oversampling restricted to training folds can reduce minority class scarcity and support more stable learning under class imbalance in small clinical cohorts.

**Objective 3: Difficulty-Aware Training under Heterogeneous Reliability** Design a CL strategy that breaks away from the uniform reliability assumption by starting with lower-difficulty samples and progressively expanding to harder cases, preventing early training from being dominated by difficult or noisier cases.

**Objective 4: Cross-Dataset Assessment of Generalization** Assess the framework's generalizability through performance comparisons between strong signal and weak signal datasets.

## 1.6 Contributions

This thesis develops a fold-disciplined and difficulty-aware learning framework for weak signal MS-based metabolomics prediction, with contributions in transformer-based embedding refinement, cGAN oversampling, and CL based on sample difficulty.

First, this thesis introduces a transformer embedding refinement stage for sample level representation learning from processed metabolomic input. Its role is to reduce the direct dependence on unstable raw features before downstream prediction under weak signal conditions.

Second, this framework integrates generative oversampling within the training portion of each CV fold. This improves support for the minority class while maintaining fold discipline and reducing leakage risk during model development.

Third, this thesis introduces a difficulty-aware training mechanism to account for heterogeneous sample reliability. Rather than treating all samples as equally informative, the framework adjusts the training emphasis using fold-local difficulty signals.

Finally, this thesis adopts a fold-disciplined evaluation framework combining stratified CV, leakage prevention. This ensures that the methodological gains are interpreted in a controlled protocol.

# Chapter 2

# Background and Preliminaries

## 2.1 Challenges in Clinical MS-Metabolomics

This thesis treats MS-based metabolomics as an input for the supervised prediction of clinical endpoints [4, 6]. Although metabolites reflect physiology more directly than upstream omics [4], clinical MS-metabolomics remains technically noisy and analytically unstable. These limitations become more consequential in small cohorts, where estimates are inherently less stable and biological effects are harder to distinguish from measurement variation [13, 14]. This section outlines the main factors that make clinical MS-metabolomics a difficult setting for supervised prediction.

The first challenge lies in the clinical endpoint itself. In metabolomics studies, the results are often operational rather than direct molecular readouts [27, 28], because they depend on diagnostic criteria, response thresholds, timing windows, and treatment protocols [27, 29]. As a result, the supervised target may not correspond perfectly to the underlying biological state of interest.

A second challenge lies on the input side of the learning problem. The model does not receive clean biological variables directly, but a sample-by-feature matrix produced through peak detection, alignment, and quantification [30, 31]. Consequently, the observed features do not necessarily represent only biologically meaningful metabolites. They may also reflect adducts, isotopes, fragments, and related compound spectra generated through the measurement and processing pipeline [32]. Thus, uncertainty in clinical MS-metabolomics arises not only from the endpoint but also from the way the input matrix is produced.

This input-side complexity is further amplified by measurement unreliability and cohort heterogeneity. In clinical MS-metabolomics, biological effects can be difficult to distinguish once technical variation reduces the effective SNR [6, 14]. Measurement reliability may vary across samples because of handling, run order, and batch effects [13, 14], and across features because of detection limits, integration error, and alignment uncertainty [30, 31]. These factors make the observed matrix less reliable even before missingness is considered.

Missingness introduces an additional complication. In clinical MS-metabolomics, missing values are often structured and concentration-dependent rather than random, with low-abundance features more likely to remain undetected [12]. Under this interpretation, non-detection is linked to the measurement process itself, because very low intensities may reflect the detection limit rather than label-dependent biological variation [12, 15]. In small cohorts, such structured missingness further reduces the stability, reliability, and interpretability of the input matrix and increases sensitivity to data partitioning.

Endpoint uncertainty, measurement-space complexity, measurement unreliability, and structured missingness are not only measurement challenges, but also factors that affect the reliability and stability of the supervised learning problem constructed from the observed matrix.

## 2.2 Learning under Weak Signal and Class Imbalance

The challenges outlined in Sections 1.3 and 2.1 enter supervised learning through both the observed input matrix and the clinical endpoint. Under weak signal, the main difficulty lies not only in limited class separation, but in unstable predictive structure across folds and across samples. Class imbalance, in which the classes are not equally represented and one class has fewer samples than the other, further increases the uncertainty in minority class learning.

In small-cohort clinical metabolomics, training can become sensitive to fold composition [33, 34]. Technical variation and concentration-dependent non-detection may appear predictive in one fold but fail to persist across others [12, 13]. CV does not eliminate split-dependent variability, but evaluation across multiple folds can make such variability more visible than reliance on a single split [35, 36]. Therefore, under weak signal, learning becomes more dependent on cohort partitioning than in stronger signal settings.

In breast cancer prediction, the minority class often corresponds to the clinically decisive group, but is represented by fewer samples in each fold [37]. This makes minority class estimates less stable across folds [36]. Imbalance handling is therefore part of the learning

setup, but under weak signal it should be treated as a structural learning difficulty rather than a purely numerical class count problem.

Under weak signal, representation also becomes a central part of the learning problem. Learning must operate on processed MS measurements whose predictive structure may not remain stable across folds [30, 32]. Treating these features as fully independent predictors can make the learned decision boundary more sensitive to fold-specific artifacts and technical variation [11, 13]. Therefore, a useful representation in this setting should reduce the sensitivity to unstable measurements while preserving the predictive structure across data partitions. Sample difficulty is therefore not uniform across cases: some are more weakly measured, more ambiguous, or more unstable than others. Thus, under weak signal and class imbalance, the central learning difficulty lies not only in limited class separation but also in the instability of the learned predictive structure across folds and across samples.

## 2.3 Model Evaluation Protocol

Under the weak signal conditions outlined in Sections 2.1 and 2.2 and the small-cohort setting of clinical metabolomics, evaluation protocol is a study-wide methodological requirement rather than a reporting detail. Performance estimates are sensitive to split composition, data leakage, and class imbalance [33, 38]. Therefore, this section defines the fold-disciplined protocol used for data partitioning, fold-local processing, and performance assessment.

### 2.3.1 Stratified Cross-Validation

In small clinical cohorts, a single random split can produce unstable results because minority class representation can vary substantially across partitions. This problem is more serious when the minority class is limited and clinically important. In this thesis, stratified CV is used to preserve overall class proportion across folds as closely as possible [38]. Although stratification cannot eliminate split-dependent variability, it reduces the distortion introduced by uneven class allocation. Therefore, CV is more informative than a single partition in weak-signal settings because it makes fold-to-fold variation visible under the same cohort constraints.

### 2.3.2 Data Leakage Prevention

Data leakage arises when data-adaptive processing is performed before fold separation. In weak signal, small-cohort settings, even limited validation information can bias performance estimates. Therefore, operations whose parameters are learned from the data must be restricted to the training portion of each CV fold and applied to validation without re-estimation [33, 38]. This principle is most directly relevant to procedures whose parameters or structure are estimated from the data for model development and evaluation. Treating such steps as fold-local preserves the independence of validation data and supports fair comparison across models [33, 38].

Consistent with the missingness interpretation in Section 2.1, low-intensity replacement is treated in this thesis as a predefined dataset preparation rule under a concentration-dependent non-detection interpretation [16, 17]. In the present workflow, this replacement is applied to the full data matrix before cross-validation, with missing values assigned a small constant equal to half of the minimum positive value in the dataset [16, 17]. By contrast, subsequent preprocessing steps whose parameters are estimated from the data distribution for model development remain fold-dependent.

### 2.3.3 Performance Metrics for Imbalanced Prediction

In imbalanced clinical prediction, evaluation should reflect performance beyond overall correctness [37, 39]. Performance is therefore assessed across folds rather than from a single partition, so that split-dependent variation remains visible [38]. This is particularly important in the weak signal metabolomics setting. Performance was evaluated using AUC as the primary metric, together with PR-AUC, F1-score, precision, and recall as complementary measures. Accuracy was also recorded as a descriptive metric, but was not treated as a primary criterion because it may be less informative under class imbalance.

#### 2.3.3.1 Primary Metric

The primary metric used in this study is the area under the Receiver Operating Characteristic (ROC) curve denoted as AUC. The AUC evaluates the model's ability to rank positive samples above negative samples across classification thresholds and is less dependent on a specific decision cutoff than accuracy. Formally,

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}(t)\, d\mathrm{FPR}(t), \tag{2.1}$$

where TPR($t$) and FPR($t$) denote the true positive rate and false positive rate at threshold $t$, respectively.

#### 2.3.3.2 Complementary Metrics

Because the datasets considered in this study are moderately imbalanced, AUC was complemented by additional metrics that reflect classification behavior from different perspectives. PR-AUC was included to provide a precision–recall view of performance under class imbalance. Precision reflects the reliability of positive predictions, recall reflects the ability to recover positive samples, F1-score summarizes their balance in a single measure, and accuracy provides an overall proportion of correct predictions. Their standard definitions are given as follows:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{2.2}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{2.3}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{2.4}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2.5}$$

where $TP$, $TN$, $FP$, and $FN$ denote true positives, true negatives, false positives, and false negatives, respectively. In addition, PR-AUC also referred to here as average precision (AP), was used to summarize performance across recall levels. Unlike ROC-AUC, which evaluates ranking ability over the true positive rate and false positive rate, PR-AUC focuses on the trade-off between precision and recall under class imbalance. Formally, AP is computed as:

$$\text{AP} = \sum_i (R_i - R_{i-1}) P_i, \tag{2.6}$$

where $P_i$ and $R_i$ denote the precision and recall at the $i$-th threshold, respectively. All metrics were computed across stratified CV folds and reported as mean ± standard deviation.

## 2.4 Representation Learning under Weak Signal

As discussed in Sections 2.1 and 2.2, MS-metabolomics prediction is difficult not only because class separation can be limited, but also because the processed feature space

is unstable and imperfectly observed [12, 13]. Under such conditions, useful predictive information may be weak, distributed across multiple features, and not reliably captured by any single feature. Predictions under these conditions cannot be based directly on raw feature values alone. When features are treated as independent predictors, the learned structure becomes more sensitive to unstable measurements, more likely to reflect split-specific or noisy patterns, and less stable across folds [38]. Therefore, a representation-learning stage is introduced to construct a more stable latent representation before downstream prediction [20].

### 2.4.1 Transformer Embedding Refinement

Among candidate representation mechanisms, a transformer-based embedding refinement stage is introduced here for sample-level representation learning from processed metabolomic input. Its role is not to construct a full token-interaction model over long sequences, but to provide an encoder-compatible nonlinear refinement step on a compact sample-level embedding before downstream prediction. This design is motivated by the weak-signal setting, where relying directly on unstable individual features may make downstream prediction more sensitive to noise and fold-specific variation [21, 40].

Let a metabolomic sample be represented by a feature vector

$$\mathbf{x} = [x_1, x_2, \ldots, x_p]^\top \in \mathbb{R}^p, \tag{2.7}$$

where $p$ denotes the number of measured variables.In the present formulation, the processed feature vector is first mapped into a $d$-dimensional latent embedding,

$$z^{(0)} = \phi(\mathbf{x}) \in \mathbb{R}^d, \tag{2.8}$$

where $\phi(\cdot)$ denotes the input projection module and $z^{(0)}$ is the initial sample-level representation. For compatibility with the encoder architecture, this representation is rewritten as a length-1 encoder-compatible sequence,

$$Z^{(0)} \in \mathbb{R}^{1\times d}, \qquad Z^{(0)} = \text{reshape}\left(z^{(0)}\right), \tag{2.9}$$

and then refined by the encoder as

$$Z^{(1)} = \text{Encoder}\left(Z^{(0)} + a\right), \tag{2.10}$$

where $a \in \mathbb{R}^{1\times d}$ denotes a learned additive embedding parameter. In this formulation, the encoder is used primarily as a structured nonlinear refinement block on the compact

embedding, rather than as a long-sequence interaction mechanism. Within each encoder block, the nonlinear transformation is implemented in the feed-forward network using the Gaussian Error Linear Unit (GELU). The feed-forward layer is defined as

$$\mathrm{FFN}(x) = W_2\,\mathrm{GELU}(W_1 x + b_1) + b_2, \tag{2.11}$$

where $W_1$, $W_2$, $b_1$, and $b_2$ are learnable parameters. Here, GELU is defined as

$$\mathrm{GELU}(u) = u\,\Phi(u), \tag{2.12}$$

where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution. The refined latent representation is then obtained by removing the length-1 sequence dimension,

$$z = \mathrm{squeeze}\Big(Z^{(1)}\Big) \in \mathbb{R}^d, \tag{2.13}$$

which is used for downstream prediction.

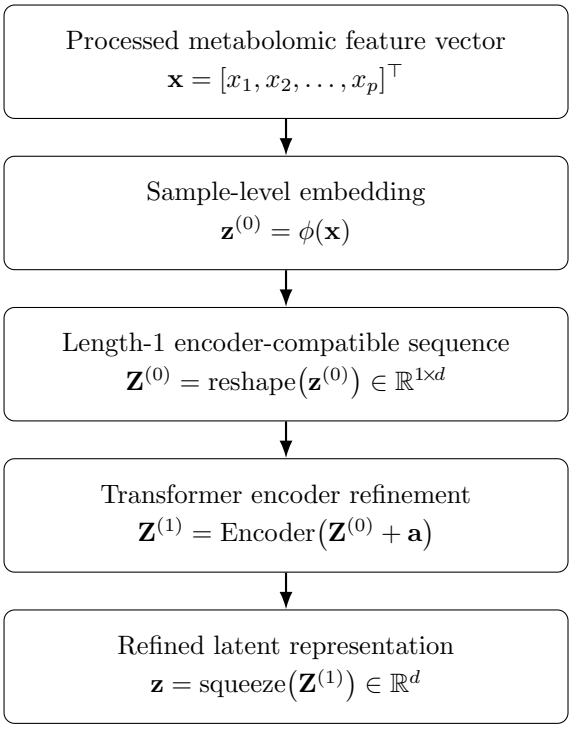


Figure 2.1: Sample-level representation pipeline

A conceptual view of the sample-level representation pipeline is shown in Figure 2.1. This discussion is conceptual only and does not aim to specify a full feature-interaction model. Instead, it motivates a transformer-based embedding refinement stage between the processed input space and prediction [20, 21].

## 2.5 Generative Oversampling for Imbalanced Data

Weak signal clinical prediction involves more than unequal class counts and also affects how well the minority class is supported [23, 41]. In small cohorts, limited minority class samples may be insufficient to stabilize the learned decision structure in a noisy and heterogeneous feature space [38]. For this reason, imbalance handling in the present setting cannot be treated as a purely numerical class-frequency problem. Accordingly, the augmentation mechanism should remain fold-local, preserve class identity, and strengthen minority class support instead of simple duplication or interpolation. This motivates the use of a conditional generative augmentation strategy that does not introduce information from validation data.

### 2.5.1 Conditional GAN Augmentation

To satisfy these requirements, a cGAN augmentation mechanism is introduced here. When minority observations are sparse or unevenly distributed, simple duplication or interpolation may increase sample count without adequately improving class-conditional coverage [22, 42]. Thus, a conditional generative model is used here because it can generate synthetic samples with explicit class identity while learning from the current training fold [43, 44]. The cGAN augmentation pipeline consists of two stages: conditional adversarial training within the current training fold, followed by minority class sample generation using the trained generator for fold-local augmentation.

Let $\mathbf{x} \in \mathbb{R}^p$ denote a metabolomic feature vector and let $y \in \{0, 1\}$ denote the class label. In a conditional adversarial framework, the generator receives a noise vector $\mathbf{z}$ together with $y$ and produces a synthetic sample

$$\tilde{\mathbf{x}} = G(\mathbf{z}, y), \tag{2.14}$$

conditioned on the class label $y$. The discriminator then receives both the sample and its associated label as a sample-label pair $(\mathbf{x}, y)$ and attempts to distinguish real from generated instances, which yields the objective

$$\min_G \max_D \; \mathbb{E}_{(\mathbf{x},y)\sim p_{\text{data}}}[\log D(\mathbf{x}, y)] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}},\, y\sim p_y}\left[\log\big(1 - D(G(\mathbf{z}, y), y)\big)\right]. \tag{2.15}$$

Generation is used specifically to enrich the minority class within the current training fold. Let $y_{\text{gen}}$ denote the minority-class condition used for synthetic generation. After fold-local conditional adversarial training, the trained generator $\hat{G}$ is used to produce

synthetic minority class samples as

$$\tilde{\mathbf{x}}_i = \hat{G}(\mathbf{z}_i, y_{\text{gen}}), \qquad \mathbf{z}_i \sim p_{\mathbf{z}}, \qquad i = 1, \ldots, N_{\text{syn}}, \tag{2.16}$$

which form the synthetic minority set

$$\tilde{\mathcal{D}}_{\text{syn}} = \{(\tilde{\mathbf{x}}_i, y_{\text{gen}})\}_{i=1}^{N_{\text{syn}}}. \tag{2.17}$$

The augmented training fold is then constructed by merging the original training fold with the generated minority samples,

$$\mathcal{D}_{\text{train}}^{\text{aug}} = \mathcal{D}_{\text{train}} \cup \tilde{\mathcal{D}}_{\text{syn}}. \tag{2.18}$$

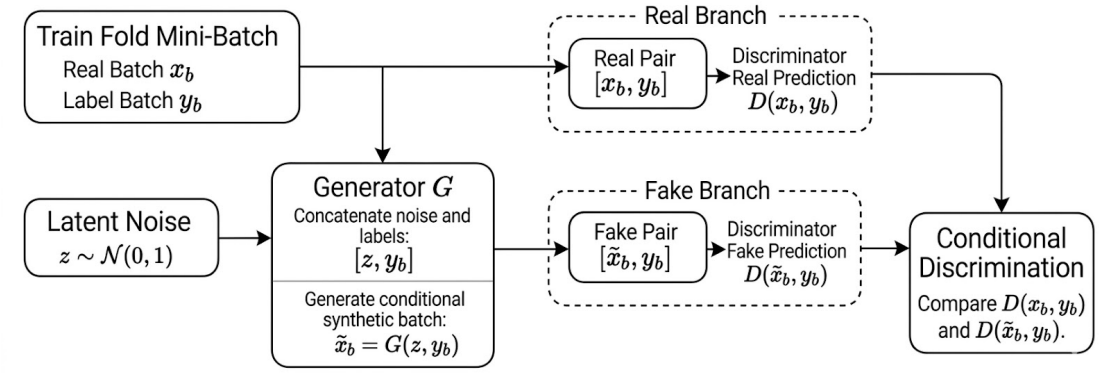


Figure 2.2: Conditional Discrimination in cGAN

Conditional adversarial training compares real and generated sample-label pairs under the same class condition, as shown in Figure 2.2.

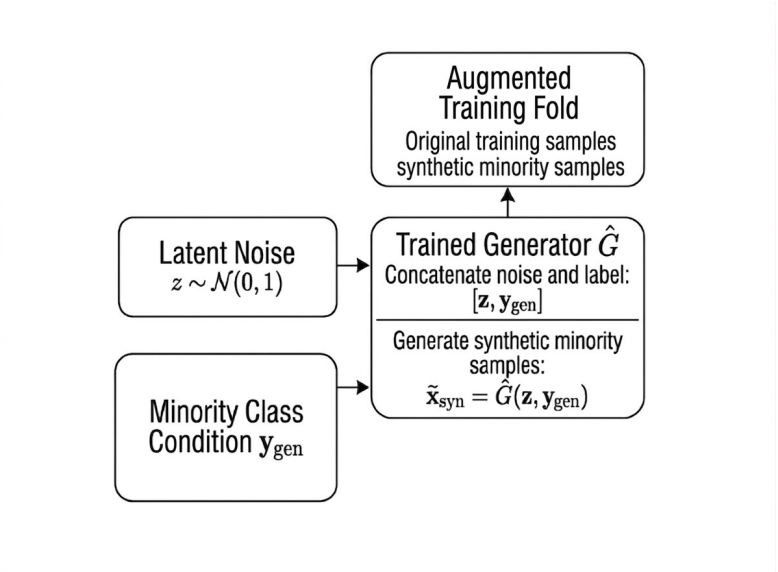


Figure 2.3: Class-Conditional Minority Augmentation

Class-conditional synthetic minority samples are generated by the trained generator and combined with the original training data to form an augmented training fold, as illustrated in Figure 2.3.

## 2.6 Curriculum Learning for Weak Signal Training

The weak signal setting described in Sections 2.1 – 2.2 also creates an additional challenge for model training. Because samples may differ in estimated learning difficulty, a staged learning strategy becomes appropriate, in which training examples are introduced progressively according to difficulty rather than being treated uniformly throughout training [25, 26]. CL provides a general framework for this idea [25]. In this work, this idea is implemented through difficulty aware staged learning.

### 2.6.1 Difficulty Aware Learning

Difficulty aware learning expresses curriculum learning in terms of estimated sample difficulty [26]. Let $(\mathbf{x}_i, y_i)$ denote the $i$-th training sample, and let $s_i$ denote a scalar score representing its estimated learning difficulty. In general terms, $s_i$ may be derived from quantities such as prediction uncertainty, classification confidence, instance hardness, distance from typical training patterns, or related indicators of how difficult a sample is to learn under the current representation [26, 45]. In this view, training samples do not need to enter learning in the same way or at the same stage. Instead, samples can be

ranked according to the estimated difficulty, so that training begins with an easier subset of the current training fold and progressively expands in stages until the full training fold is used [25, 46].

A general formulation can be written as:

$$s_{(1)} \leq s_{(2)} \leq \cdots \leq s_{(N)}, \tag{2.19}$$

where the training samples are ranked from lower to higher estimated difficulty. At training stage $t$, the subset $D^{(t)}$ used for learning is defined as:

$$\mathcal{D}^{(t)} = \{(\mathbf{x}_{(i)}, y_{(i)})\}_{i=1}^{m_t}, \tag{2.20}$$

where $m_t$ denotes the number of samples admitted at stage $t$, with later stages using progressively larger subsets.

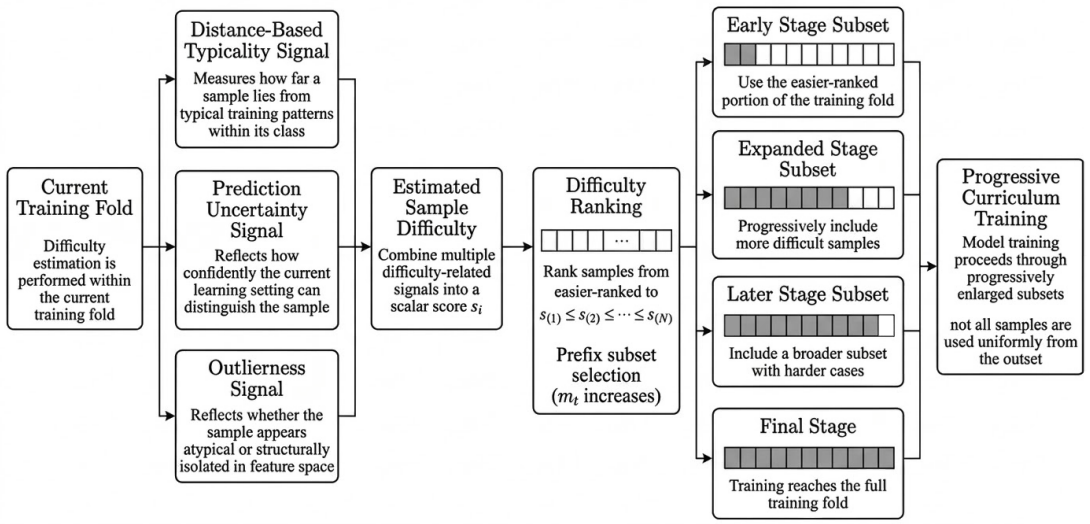


Figure 2.4: Difficulty Aware Curriculum Learning Pipeline

Fold-local difficulty signals are integrated into a scalar difficulty score $s_i$, after which the samples are ranked from easier to harder as in (2.19), and progressively larger subsets $D^{(t)}$ are formed across stages as in (2.20) until the complete training fold is included, as shown in Figure 2.4.

## 2.7 Chapter Summary

This chapter framed weak signal clinical MS-metabolomics as a structured learning problem in which input instability, small cohort imbalance, and sample level reliability variation jointly affect predictive modeling. It also established the fold-disciplined

evaluation setting used throughout the thesis and motivated three corresponding design directions, namely representation refinement, fold-local minority support, and difficulty aware training. These preliminaries provide the basis for the next chapter, which examines the related literature and clarifies why a unified methodological response remains needed.

# Chapter 3

# Related Work

## 3.1 Literature Review

Clinical metabolomics is increasingly used in disease characterization, biomarker discovery, and precision medicine, but prior work has also emphasized that predictive modeling in this setting strongly depends on data quality and preprocessing. Long et al. (2020) reviewed its translational role, while Liebal et al. (2020) highlighted the dependence of ML performance on preprocessing and data handling choices [47, 48]. Reinhold et al. (2019) showed that the processed feature space can be shaped by pre-analytic handling and preprocessing choices, while Do et al. (2018) demonstrated that missingness is a structural rather than marginal issue in such data [15, 49]. Abram et al. (2022) further showed that preprocessing choices can materially affect downstream deep learning (DL) performance [19]. Together, these studies suggest that predictive performance reflects both model choice and the way preprocessing shapes the feature space, making representation learning relevant as an approach to reorganizing feature information.

Vaswani et al. (2017) introduced the transformer architecture and demonstrated a new neural modeling framework for learning input representations without relying on recurrence or convolution [40]. More recent work has continued to examine representation learning for structured and tabular data. Badaro et al. (2023) reviewed transformer-based models and applications for tabular data, while Jiang et al. (2025) provided a broader recent survey of representation learning for tabular data [50, 51]. These studies support the broader methodological idea that learned representations can reorganize feature information before prediction. However, in this thesis, the transformer is not used as a feature-token interaction model. Instead, it is used as an encoder-based refinement block over a compact sample-level embedding.

Imbalanced classification refers to settings in supervised learning where one class is represented by substantially fewer training examples than another. A widely used response is the Synthetic Minority Oversampling Technique (SMOTE), introduced by Chawla et al. (2002), which generates synthetic minority samples by interpolating between existing observations [42]. Krawczyk (2016) argued that class proportions alone do not determine learning performance on imbalanced data. Factors such as class overlap, sample sparsity, and the structure of minority regions also matter [22]. Accordingly, the issue extends beyond class proportion adjustment alone and includes the need for sufficiently stable minority support during training.

Although interpolation-based oversampling can partially address this, its ability to represent a more complex class structure remains limited [23]. This limitation points to augmentation strategies that go beyond local interpolation. One such direction is generative modeling, which seeks to learn a data distribution. Goodfellow et al. (2014) introduced GANs, where a generator produces synthetic samples and a discriminator attempts to distinguish them from real data [43]. More recent work has extended this direction to tabular augmentation. Zhao et al. (2022) proposed a conditional GAN framework for tabular data synthesis, while Eom and Byeon (2023) compared CGAN/CTGAN-based oversampling with traditional oversampling methods for imbalanced classification [52, 53]. Unlike SMOTE, which generates minority samples by interpolation between observed cases, cGAN-based augmentation is designed to learn a class-conditional distribution and generate new samples from it. In this sense, SMOTE mainly fills local gaps between minority observations, while cGAN augmentation is intended to model the minority class more explicitly. This makes cGANs better aligned with the objective of this framework.

CL addresses the order in which training samples are introduced during optimization. Bengio et al. (2009) introduced curriculum learning as an easy to hard training strategy. Kumar et al. (2010) extended this idea through self-paced learning, in which samples are progressively introduced according to the current state of the model [25, 46]. More recent work has continued to develop curriculum-based learning for imbalance-aware settings. Escudero-Viñolo and López-Cifuentes (2022) proposed class-wise curriculum learning for class imbalance problems, while Chaudhry et al. (2025) introduced a dynamic data distribution-based curriculum learning approach that incorporates self-paced learning [54, 55]. CL organizes sample exposure progressively so that more stable patterns can be learned before harder cases are introduced. Representation learning and data augmentation focus on feature structure and minority support, while CL controls how training information is introduced over time.

## 3.2 Gap in the Existing Literature

The literature reviewed above points to three relevant methodological directions: representation learning for improved feature structure, augmentation strategies for stronger minority support, and curriculum training for heterogeneous sample difficulty. However, these directions have mostly been developed separately, with prior work typically emphasizing one component at a time rather than their joint use within a unified framework.

Table 3.1: Summary of Prior Studies and Limitations

| **Prior study** | **Representation learning** | **cGAN augmentation** | **Curriculum learning** |
|---|---|---|---|
| Badaro et al. (2023) [50] | ✓ | × | × |
| Jiang et al. (2025) [51] | ✓ | × | × |
| Zhao et al. (2022) [52] | × | ✓ | × |
| Eom and Byeon (2023) [53] | × | ✓ | × |
| Escudero-Viñolo et al. (2022) [54] | × | × | ✓ |
| Chaudhry et al. (2025) [55] | × | × | ✓ |

The relevant literature is summarized in Table 3.1. Prior studies provide partial methodological support across representation learning, minority class support, and CL, but these directions remain largely explored in isolation rather than integrated within a unified prediction framework. In addition, limited prior work has directly addressed weak-signal clinical MS-metabolomics prediction as a combined methodological problem.

# Chapter 4

# Materials and Methods

This chapter describes how the study was conducted, including data construction, preprocessing, and the evaluation protocol. Then it presents the main components of the proposed framework and their integration into a unified prediction pipeline. Finally, implementation details and evaluation metrics are provided to support the empirical analysis presented in the following chapters.

## 4.1 Overall Analytical Workflow

This study presents a structured workflow that covers data construction, fold-disciplined model development, and evaluation under different signal conditions. It evaluates predictive performance, stability, and robustness under heterogeneous signal conditions. The analytical workflow adopted in this study is presented in Figure 4.1. Raw data are first extracted and harmonized to construct dataset-specific analytic matrices. The data is then partitioned using stratified CV. Within each training fold, minority-class augmentation is applied when enabled, followed by fold-local preprocessing, transformer-based representation learning, and curriculum-guided training. The trained model is then evaluated on the corresponding validation fold without fitting on validation data. This workflow is repeated across all folds and applied consistently to each dataset, so that the framework can be evaluated in the weak signal setting and checked for stable performance under stronger signal conditions.

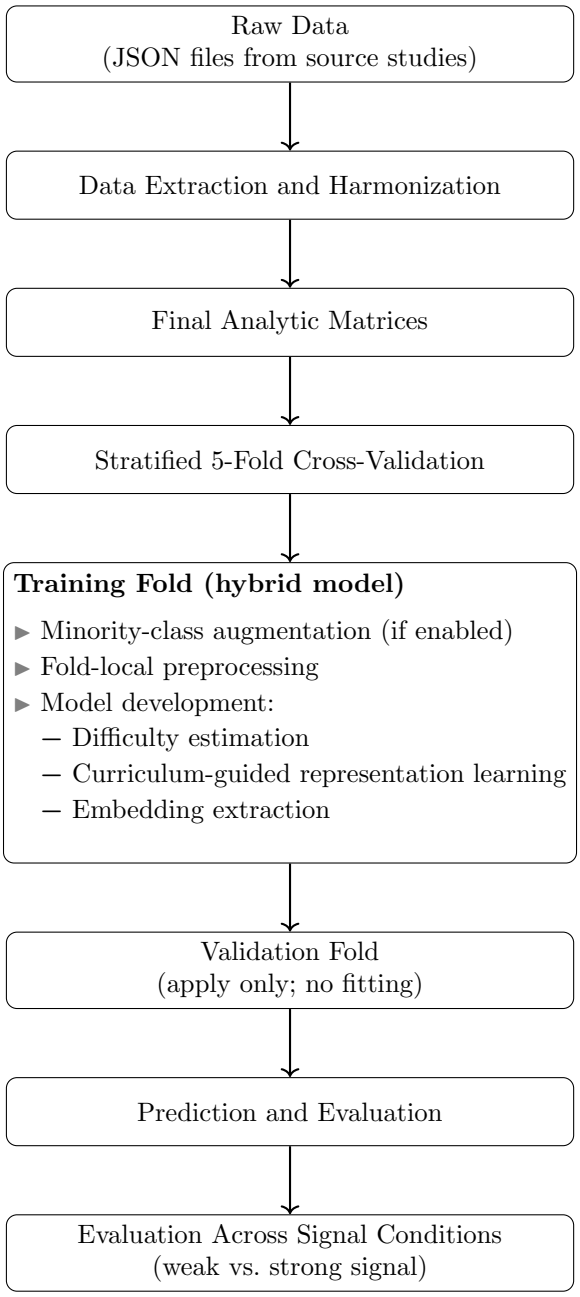


Figure 4.1: Overall Analytical Workflow of the Proposed Framework

## 4.2 Materials

### 4.2.1 Data Source Overview and Dataset Roles

Two human plasma case–control metabolomics datasets are used in this study, both obtained from Metabolomics Workbench (NIH Common Fund) [56]. Both datasets are related to breast cancer and are used to define a supervised case-control prediction setting. Rather than being treated as independent experiments, they are jointly analyzed to evaluate the proposed framework under different signal conditions. ST004145 [57]

serves as the weak signal dataset and the primary analytical setting. ST000355 [58], in contrast, is included as a strong signal dataset to assess whether the framework remains stable. The roles of the two datasets are summarized in Table 4.1.

Table 4.1: Dataset Roles

| Dataset | Signal Condition | Role in the Study | Primary Purpose |
|---|---|---|---|
| ST004145 | Weak signal | Primary dataset | Evaluate robustness |
| ST000355 | Strong signal | Stability-check dataset | Assess stability |

### 4.2.2 ST004145 Weak Signal Dataset

ST004145 is the primary dataset used in this study. Its final analytic matrix contains 232 samples, including 91 cases and 141 controls, with 12 retained features for modeling. A moderate class imbalance is shown in Figure 4.2.

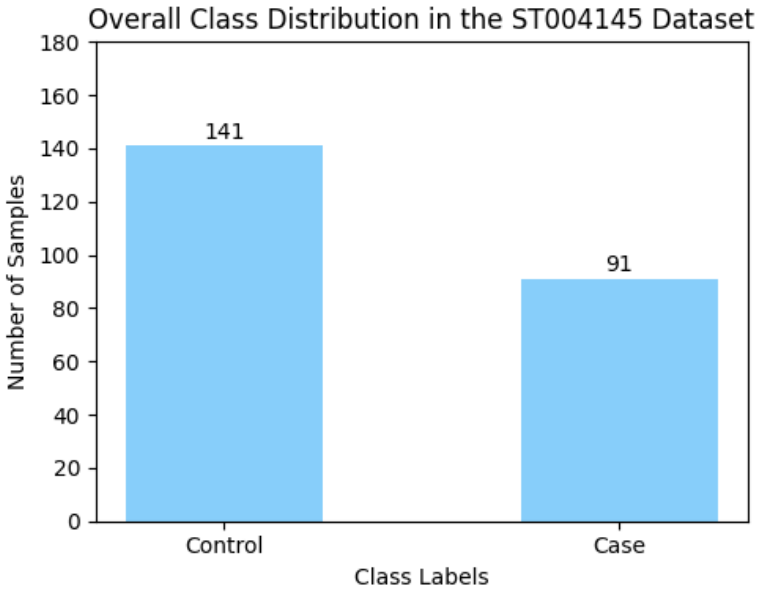


Figure 4.2: Class Distribution of ST004145

Only one of the 12 selected features reaches the significance threshold, defined here as a False Discovery Rate (FDR)-adjusted p-value below 0.05 using the Benjamini-Hochberg procedure [59], indicating an extremely weak univariate signal, as shown in Figure 4.3. Figure 4.4 further shows substantial overlap between cases and controls, with no clear linear separation in the low-dimensional projection obtained using Principal Component Analysis (PCA) [60]. Together, these results suggest that ST004145 provides only limited class-related structure for prediction.

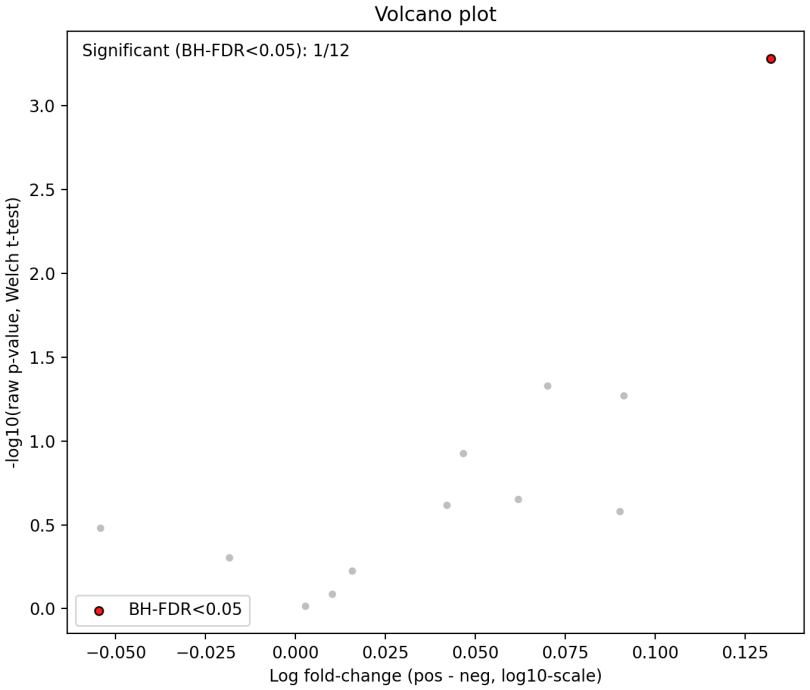


Figure 4.3: ST004145 Volcano Plot

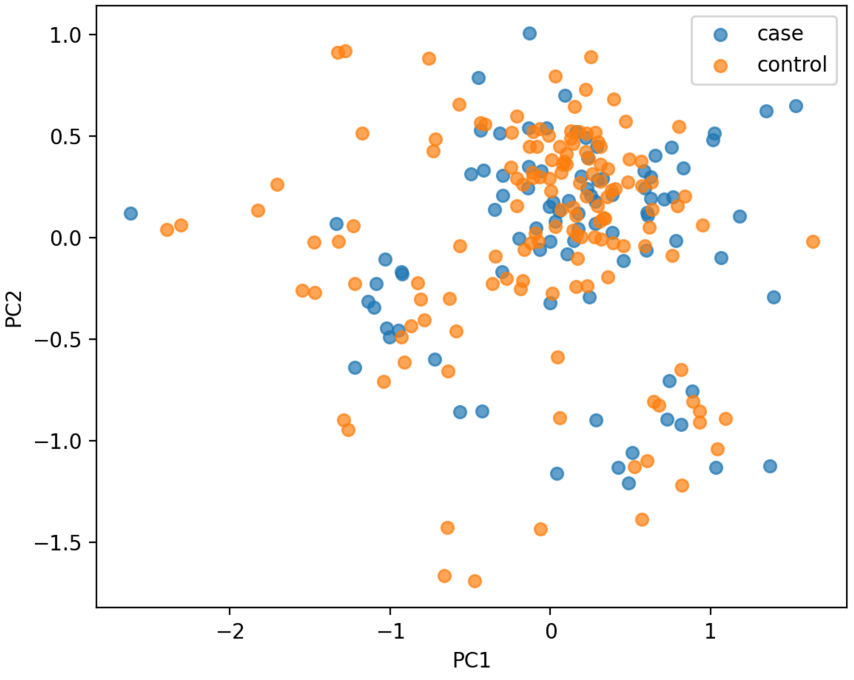


Figure 4.4: ST004145 PCA

### 4.2.3 ST000355 Strong Signal Dataset

ST000355 is the strong signal dataset used in this study. Its final analytic matrix contains 211 samples, including 76 cases and 135 controls, with 227 selected features for modeling. A moderate class imbalance is shown in Figure 4.5.

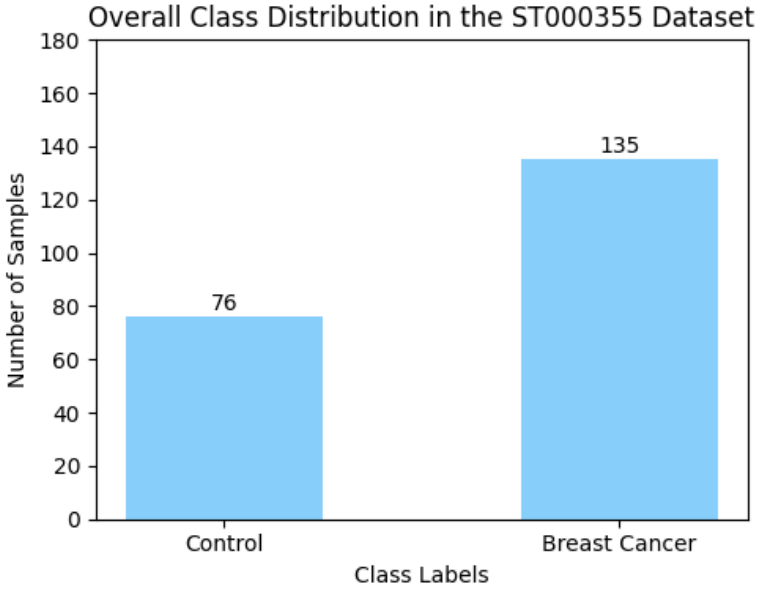


Figure 4.5: Class Distribution of ST000355

90 of the 227 selected features reach the significance threshold, defined here as an FDR-adjusted p-value below 0.05 using the Benjamini–Hochberg procedure [59], indicating a strong univariate discriminative signal, as shown in Figure 4.6. Figure 4.7 further shows clear separation between cases and controls in the low-dimensional projection. Together, these results suggest that ST000355 provides much stronger discriminative structure for prediction than ST004145.

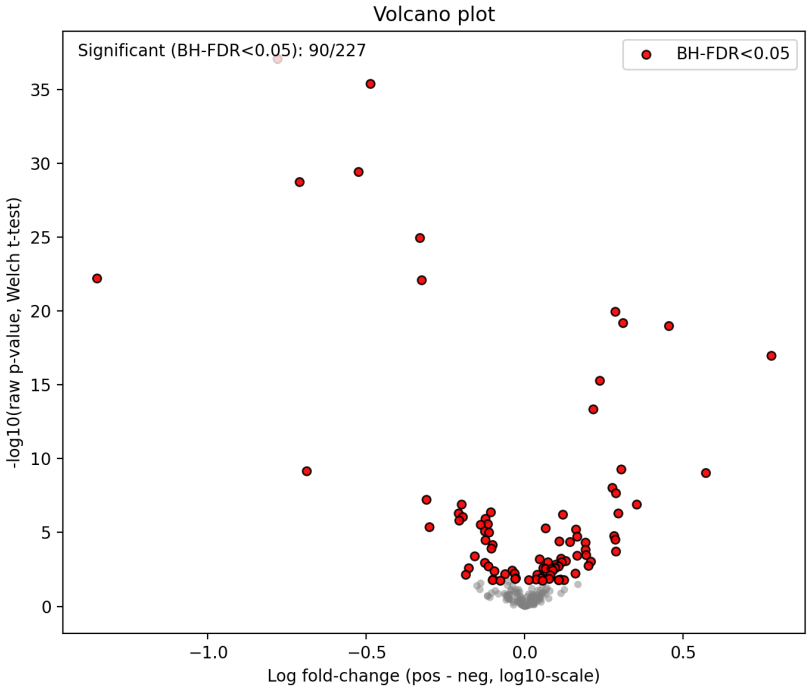


Figure 4.6: ST000355 Volcano Plot

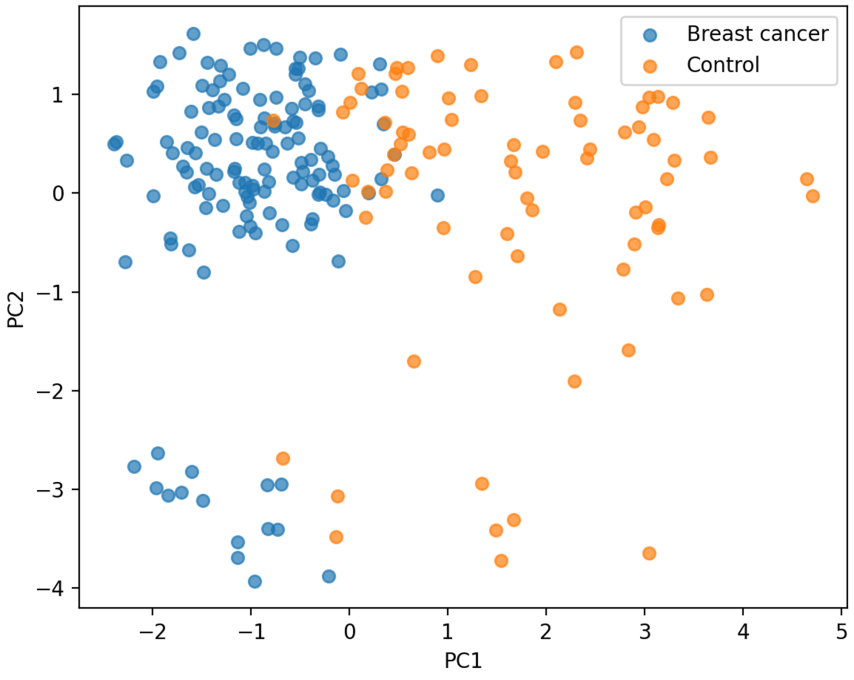


Figure 4.7: ST000355 PCA

## 4.3 Data Preparation

This section describes the data preparation procedures used to produce a consistent analytic dataset before model fitting.

### 4.3.1 Data Extraction and Harmonization

For each dataset, two raw JavaScript Object Notation (JSON) files are processed using a common extraction and harmonization procedure to construct a sample-by-feature matrix. This procedure includes extraction of sample-level metadata and metabolite measurements, alignment of sample identifiers across files, consolidation of features into a unified feature set, and removal of inconsistent or duplicated entries. Variable source formats are standardized during parsing. To avoid unintended sample loss, samples with all missing values in a given mode are retained during matrix construction. After extraction, the two JSON derived matrices are aligned on their shared sample set and merged into a single matrix. Although the source data differ between datasets, both are organized into the same final matrix format.

### 4.3.2 Feature Cleaning, Transformation, and Missingness Handling

After matrix construction, feature-level cleaning is performed, including removal of invalid or non-informative features where applicable, and log transformation of metabolite intensity values. In metabolomics data, missingness often reflects concentration dependent non-detection rather than random absence. For ST004145, missingness is minimal after harmonization, and no additional imputation is introduced. For ST000355, missing values are handled using a Min/2 replacement rule, in which each missing entry is replaced by half of the minimum observed value for that feature. This treatment is consistent with the interpretation that many missing entries arise from values below the detection limit. The replacement is performed as part of dataset level preparation and does not use class labels.

### 4.3.3 Fold-Local Standardization

Unlike the preprocessing steps described above, standardization is performed separately within each CV fold. For each fold, the transformation is fit on the training data only and then applied to the corresponding validation data. In this way, standardization remains consistent with the leakage prevention protocol adopted throughout this study.

## 4.4 Proposed Framework

This section presents the proposed unified prediction framework. It focuses on how representation learning, conditional generative augmentation, and CL are integrated within a fold-disciplined pipeline.

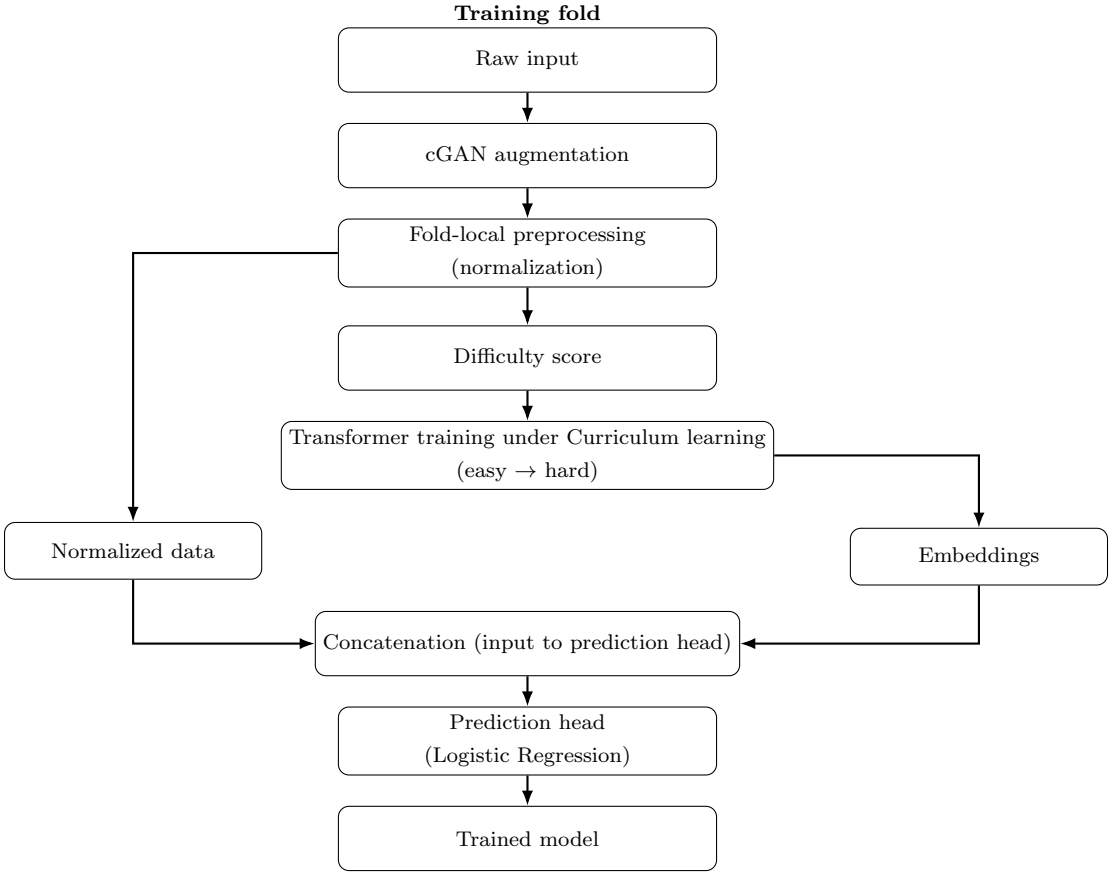


Figure 4.8: Workflow of the Hybrid Framework within the Training Fold

Minority class augmentation is first applied to the raw input, followed by fold-local normalization. Difficulty estimation is performed on the resulting normalized data, after which transformer training under a CL strategy proceeds from easier to harder samples. The trained transformer is subsequently used to extract sample-level embeddings, which are combined with normalized data for prediction, as shown in Figure 4.8.

### 4.4.1 Representation Learning Module

One component of the proposed framework is a transformer-based representation learning module that maps the normalized feature vector of each sample to a latent embedding for prediction. Let $x_i^{\text{norm}} \in \mathbb{R}^p$ denote the normalized feature vector of sample $i$ within

the current fold. The representation module produces a sample-level embedding

$$z_i = f_\theta(x_i^{\text{norm}}), \tag{4.1}$$

where $f_\theta$ denotes the transformer-based encoder and $z_i \in \mathbb{R}^d$ is the learned latent representation, with $d < p$. Rather than replacing the processed feature vector, the learned embedding is combined with the normalized input to form a hybrid representation:

$$h_i = [x_i^{\text{norm}}; z_i], \tag{4.2}$$

where $[\cdot;\cdot]$ denotes feature concatenation. This design preserves information from the processed metabolomic feature space while augmenting it with a learned sample-level representation, thereby avoiding exclusive reliance on either the original normalized input or the latent embedding alone. The final prediction is then obtained by applying the downstream classifier to the hybrid representation:

$$\hat{y}_i = g_\phi(h_i), \tag{4.3}$$

where $g_\phi$ denotes the prediction head (e.g., Logistic Regression (LR)).

### 4.4.2 cGAN for Minority Class Support

Another component of the proposed framework is a conditional generative augmentation module that enriches the minority class within each training fold. Following Section 2.5.1, this subsection describes how cGAN-based minority augmentation is used within the current training fold.

Let $D_{\text{train}}^{(k)}$ denote the training set in the $k$-th CV fold, and let $D_{\text{min}}^{(k)} \subset D_{\text{train}}^{(k)}$ denote its minority class subset. Conditional on the minority class label $y_{\text{min}}$, the generator produces synthetic minority samples as

$$\tilde{x}_j^{(k)} = G_\theta^{(k)}(z_j, y_{\text{min}}), \qquad z_j \sim \mathcal{N}(0,1), \qquad j = 1, \ldots, N_{\text{syn}}^{(k)}. \tag{4.4}$$

The augmented training fold is then constructed by merging the original training data with the generated minority samples:

$$D_{\text{train,aug}}^{(k)} = D_{\text{train}}^{(k)} \cup \left\{ \tilde{x}_j^{(k)} \right\}_{j=1}^{N_{\text{syn}}^{(k)}}. \tag{4.5}$$

The augmented training fold is used for fold-local preprocessing and subsequent model training within the same CV split.

### 4.4.3 CL and Difficulty Estimation

The third component of the proposed framework is a difficulty-aware training module that assigns a fold-local difficulty score to each training sample and applies curriculum-guided training from easier to harder samples.

For each sample $x_i$, the difficulty score is computed from three fold-local quantities: its distance to the nearest class centroid, its prediction uncertainty under a shallow random forest, and an outlier indicator from Isolation Forest. Let

$$\delta_i = \min_{c \in C} \|x_i - \mu_c\|_2 \tag{4.6}$$

denote the Euclidean distance from sample $i$ to the nearest class centroid, where $\mu_c$ is the centroid of class $c$ computed from the training data within the fold. This quantity is normalized as

$$\bar{\delta}_i = \frac{\delta_i}{\max_j \delta_j}. \tag{4.7}$$

Let

$$u_i = 1 - \max_{k \in \{0,1\}} \hat{p}(y = k \mid x_i) \tag{4.8}$$

denote the uncertainty score estimated by the shallow random forest, where larger values indicate lower prediction confidence. Let

$$o_i = \begin{cases} 1, & \text{if sample } i \text{ is identified as an outlier by Isolation Forest,} \\ 0, & \text{otherwise.} \end{cases} \tag{4.9}$$

The final difficulty score is defined as

$$s_i = \frac{\bar{\delta}_i + u_i + o_i}{3}. \tag{4.10}$$

Samples are then ordered from easier to harder according to their difficulty scores:

$$s_{(1)} \leq s_{(2)} \leq \cdots \leq s_{(N)}. \tag{4.11}$$

This ordering is used to guide curriculum-based transformer training on the normalized data, so that earlier stages emphasize easier samples while later stages progressively include harder ones.

### 4.4.4 Integrated Framework

The components described above are jointly implemented within a unified training pipeline for each CV fold, following the fold-level execution order illustrated in Figure 4.8. Within each training fold, minority class augmentation is first applied when enabled to expand the available training data without introducing information from the validation fold. The augmented training data are then normalized within the current fold. Difficulty estimation is subsequently performed on the normalized training data using fold-local quantities. Based on the resulting scores, curriculum-guided training is applied from easier to harder samples. After training, the transformer module is used to extract sample-level embeddings from the normalized input data. The model concatenates the extracted embeddings with the normalized feature vectors and then sends the resulting representation to the prediction head, producing the final trained model for the current fold.

## 4.5 Prediction Setting and Evaluation Protocol

The proposed framework is evaluated as a supervised binary classification model for sample-level clinical prediction. The model outputs a scalar probability score indicating the likelihood that the sample belongs to the positive class. Thus, for sample $i$ with vector $x_i \in \mathbb{R}^p$, the prediction function can be written as

$$f(x_i) \rightarrow \hat{y}_i, \tag{4.12}$$

where $\hat{y}_i \in [0, 1]$ denotes the predicted probability for the positive class.

Model evaluation is conducted using stratified 5-fold CV. No independent hold-out test set is used in this study, since reserving a separate test set would further limit the data available for model development and could increase evaluation variability in a small cohort. The dataset is divided into five folds while preserving class proportions across folds as closely as possible, as illustrated in Figure 4.9. In each iteration, one fold is used for validation and the remaining four folds are used for model development. This process is repeated until each fold has served once as the validation fold, and performance is summarized across folds.

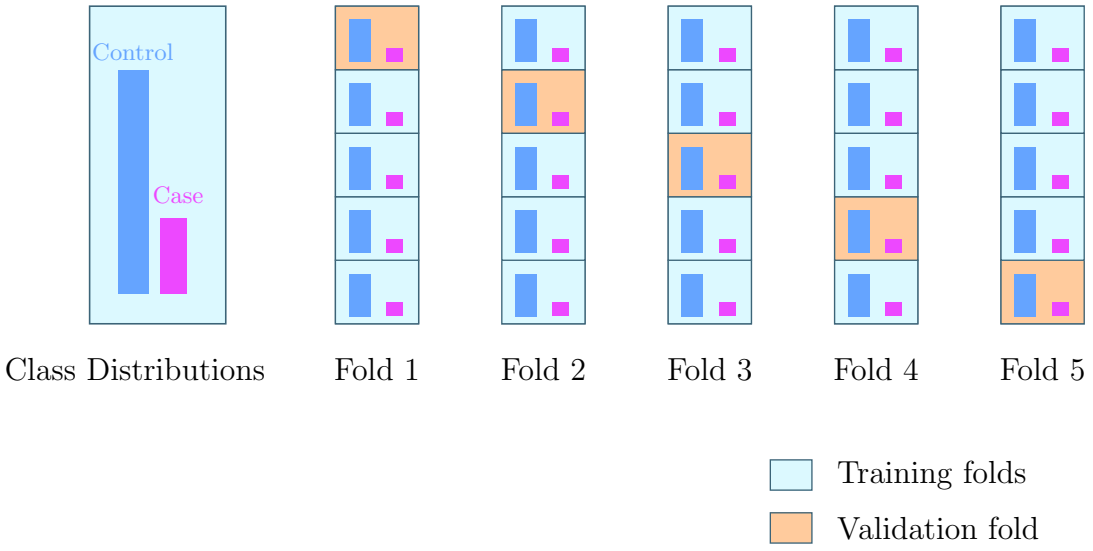


Figure 4.9: Stratified 5-Fold Cross-Validation

All data-dependent operations are restricted to the training portion of each CV fold. Preprocessing steps that require parameter estimation, such as standardization, are fit on the training data and then applied to the corresponding validation fold without re-estimation. cGAN minority class augmentation and CL difficulty estimation are performed within the current training fold only. The validation fold is therefore excluded from model fitting, synthetic sample generation, and difficulty estimation. This fold-local protocol is used throughout the study to reduce information leakage and to support fair comparison across datasets under a consistent evaluation setting.

## 4.6 Implementation Details

This section summarizes the main implementation choices used in the proposed framework. The complete final configuration is reported in Table C.2.

### 4.6.1 Software and Training Configuration

All experiments were implemented in Python using PyTorch for transformer representation learning and cGAN training, together with standard ML libraries for preprocessing, evaluation, and baseline comparisons. The experiments were executed on Google Colab under CPU settings. For ST004145, the transformer representation module used the 12 features as input and mapped each sample to an 8-dimensional latent embedding. The encoder used two attention heads and 1 transformer encoder layer, with a dropout

rate of 0.3. The cGAN module used a latent noise dimension of 16, and the generator and discriminator were both optimized using Adam. Transformer optimization was performed separately using AdamW rather than through a single end-to-end training procedure. In the selected hybrid model, the final prediction head is LR.

### 4.6.2 Reproducibility and Execution Control

All experiments were conducted under a fixed random seed to improve reproducibility across runs. The same stratified 5-fold CV protocol was applied consistently across all models and both datasets to support fair comparison under a shared evaluation design. To reduce information leakage, all fold-dependent operations were restricted to the training portion of each CV split. This included fold-local preprocessing, synthetic sample generation, difficulty estimation, and model fitting. The corresponding validation fold was used only for transformation application and performance assessment without parameter fitting. The implementation also maintained consistent execution control across experiments by using the same fold generation procedure, the same evaluation metrics, and the same comparison protocol for baseline, reduced, and full hybrid models.

# Chapter 5

# Results

## 5.1 Chapter Overview

This chapter reports the results of the proposed framework under two signal conditions. The primary analysis is conducted on ST004145, which serves as the main weak signal dataset. The strong signal dataset ST000355 is then used as a stability check dataset to examine whether the proposed framework remains competitive under a more separable prediction setting. In addition to reporting overall model comparison results, this chapter further reports ablation results for CL and cGAN oversampling, examines the classification behavior of the final selected model, and summarizes the configuration used in the reported results.

## 5.2 Overall Model Comparison on ST004145

The overall performance of the evaluated models on the primary weak signal dataset ST004145 is summarized in Table 5.1. The proposed hybrid framework with both CL and cGAN oversampling achieves the best performance, reaching a mean AUC of 0.6794. Compared with its reduced variants, removing either component leads to lower AUC values, suggesting that both CL and cGAN contribute to performance under weak signal conditions. Although several models achieved comparable accuracy values, model selection was based primarily on mean AUC. `Hybrid_Curriculum_cGAN` is used as the proposed model in the remainder of this chapter. It also shows a balanced set of complementary classification metrics, including accuracy, F1 score, precision, and recall. The following sections examine its component contributions and classification behavior.

Table 5.1: Model Performance on ST004145

| Model | Mean AUC | Std AUC | Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|---|---|---|
| **HybridCurriculum_cGAN** | **0.6794** | **0.0871** | **0.6250** | **0.4727** | **0.5270** | **0.4286** |
| Hybrid_cGAN | 0.6284 | 0.1305 | 0.5819 | 0.4260 | 0.4615 | 0.3956 |
| Baseline_SVM | 0.6138 | 0.0573 | 0.6509 | 0.3520 | 0.6471 | 0.2418 |
| Baseline_LR | 0.6129 | 0.0883 | 0.6293 | 0.1400 | 0.7778 | 0.0769 |
| Hybrid_Curriculum | 0.6001 | 0.0941 | 0.6293 | 0.5376 | 0.5263 | 0.5495 |
| Baseline_LightGBM | 0.5959 | 0.0633 | 0.6466 | 0.4605 | 0.5738 | 0.3846 |

## 5.3 Ablation Analysis on ST004145

The ablation results on ST004145 are summarized in Table 5.2 through comparisons between the full proposed model, selected reduced variants, and a conventional baseline. These comparisons assess the contributions of CL, cGAN minority class support, and the full hybrid framework.

Table 5.2: Component Gains on ST004145

| Base Model | Enhanced Model | Δ Mean AUC |
|---|---|---|
| Hybrid_cGAN (0.6284) | Hybrid_Curriculum_cGAN (0.6794) | **+0.0510** |
| Hybrid_Curriculum (0.6001) | Hybrid_Curriculum_cGAN (0.6794) | **+0.0793** |
| Baseline_SVM (0.6138) | Hybrid_Curriculum_cGAN (0.6794) | **+0.0656** |

First, the contribution of CL was examined under the cGAN-augmented setting by comparing `Hybrid_cGAN` with `Hybrid_Curriculum_cGAN`. Adding CL increased the mean AUC from 0.6284 to 0.6794, corresponding to a positive gain of 0.0510. This suggests that CL improves training effectiveness beyond minority class support.

Second, the contribution of cGAN minority class support was examined under CL by comparing `Hybrid_Curriculum` with `Hybrid_Curriculum_cGAN`. Under this comparison, adding cGAN increased the mean AUC from 0.6001 to 0.6794, corresponding to an absolute gain of 0.0793. The larger gain indicates that minority class support remained an important component of the framework even when CL was already present.

Finally, the overall contribution of the full hybrid framework was assessed relative to a conventional baseline by comparing Baseline SVM with `Hybrid_Curriculum_cGAN`. The proposed model improved the mean AUC from 0.6138 to 0.6794, yielding an absolute gain of 0.0656. Together, these results show that the reported improvement is not limited to a single component, but arises from the combined effect of CL, cGAN support, and the full hybrid design under the fold-disciplined evaluation setting.

## 5.4 Performance Evaluation

The pooled out-of-fold (OOF) ROC curve of the final proposed model on ST004145 is shown in Figure 5.1. The corresponding OOF AUC is 0.6297, indicating that the model retains useful discrimination ability under the weak signal setting. The OOF AUC is not expected to be identical to the mean AUC reported in Table 5.1. The mean AUC is obtained by averaging fold-wise validation AUC values, while the OOF AUC is computed after pooling all out-of-fold predictions across folds into a single evaluation set. Because these two quantities are based on different calculation procedures, the two values are not expected to be identical.

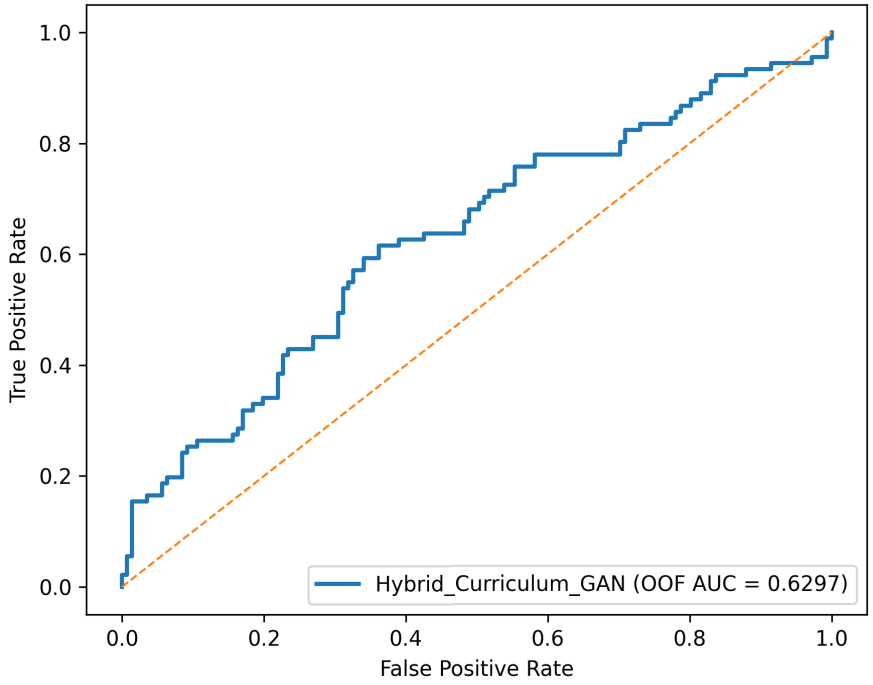


Figure 5.1: ROC Curve of the Proposed Model on ST004145

In addition to the ROC analysis, the precision–recall curve was examined for the selected model on ST004145. This perspective is particularly informative under class imbalance, as it directly reflects the trade-off between positive-class recovery and false positive burden. The proposed model achieved an AP of 0.5536 on ST004145, which is clearly above the positive class prevalence of 0.3922. This indicates that the model captured useful predictive signal beyond random guessing. At the same time, precision decreased as recall increased, showing that recovery of positive samples remained difficult in this weak signal setting, as shown in Figure 5.2.

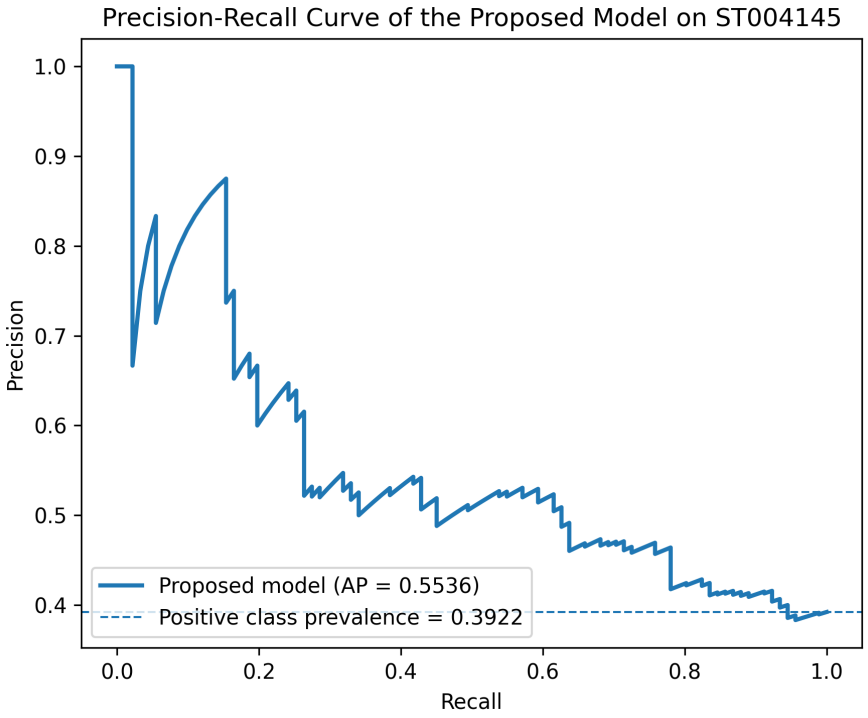


Figure 5.2: PR Curve of the Proposed Model on ST004145

The final selected configuration corresponds to the proposed model, which achieved the best overall performance on ST004145. The detailed hyperparameter settings are provided in Table C.2. This configuration was used for all reported results, including the evaluation on ST000355.

## 5.5 Evaluation on the Stability-Check Dataset ST000355

ST000355 was used as a stability-check dataset to examine whether the proposed framework remains competitive under a strong signal setting. The overall model comparison on ST000355 is summarized in Table 5.3. The proposed model, `Hybrid_Curriculum_cGAN`, achieves the highest mean AUC 0.9896. `Baseline_LR` and `Baseline_SVM` also achieved strong performance, with identical mean AUC values of 0.9881. The proposed model still achieves the top result under the selected configuration as shown in Table **??**, which provides additional support for the stability of the framework.

The pooled OOF ROC curve of the proposed model on ST000355 is shown in Figure 5.3 and indicates strong discrimination. This result is consistent with the overall comparison in Table 5.3. These findings show that the proposed framework remained highly competitive on ST000355.

Table 5.3: Model Performance on ST000355

| Model | Mean AUC | Std AUC | Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|---|---|---|
| **HybridCurriculum_cGAN** | **0.9896** | **0.0195** | **0.9763** | **0.9814** | **0.9851** | **0.9778** |
| Baseline_LR | 0.9881 | 0.0237 | 0.9763 | 0.9818 | 0.9643 | 1.0000 |
| Baseline_SVM | 0.9881 | 0.0237 | 0.9763 | 0.9814 | 0.9851 | 0.9778 |
| Hybrid_cGAN | 0.9867 | 0.0242 | 0.9858 | 0.9888 | 0.9925 | 0.9852 |
| Hybrid_Curriculum | 0.9853 | 0.0237 | 0.9668 | 0.9740 | 0.9776 | 0.9704 |
| Baseline_LightGBM | 0.9807 | 0.0254 | 0.9716 | 0.9779 | 0.9708 | 0.9852 |

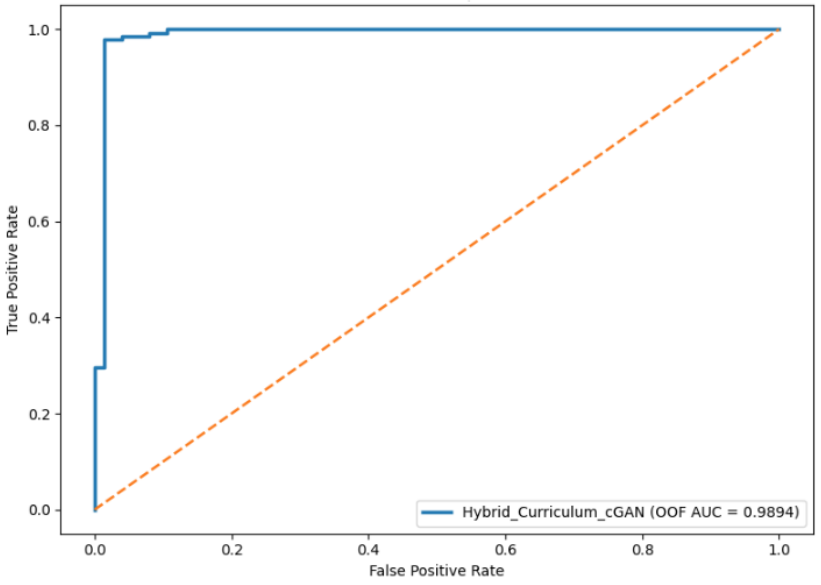


Figure 5.3: ROC Curve of the Proposed Model on ST000355

## 5.6 Cross-Dataset Findings

The cross-dataset performance of the selected model is summarized in Table 5.4. On ST004145, the hybrid framework with CL and cGAN achieved the highest AUC. On ST000355, by contrast, performance differences between models were much smaller, as several methods reached near-ceiling AUC values. Overall, these results suggest that the proposed framework is most beneficial under weak signal conditions, while under stronger signal conditions the prediction task becomes less sensitive to modeling choice. This pattern is consistent with the dataset roles defined in subsection 4.2.1.

Table 5.4: Cross-Dataset Summary of the Main Findings

| Dataset | Selected Model | AUC (mean ± std) |
|---|---|---|
| ST004145 (Weak signal) | Hybrid (Curriculum + cGAN) | 0.6794 ± 0.0871 |
| ST000355 (Strong signal) | Hybrid (Curriculum + cGAN) | 0.9896 ± 0.0195 |

The research objectives defined in Section 1.5 are directly aligned with the corresponding methodological contributions and the empirical evidence reported in Chapter 5, as shown in Table 5.5.

Table 5.5: Alignment of Research Objectives, Contributions, and Empirical Evidence

| **Research Objective** | **Corresponding Contribution** | **Evidence from Chapter 5** | **Supported** |
|---|---|---|---|
| Objective 1: Develop a more stable representation under weak signal conditions | Transformer-based embedding refinement for sample-level representation learning | Overall model comparison on ST004145 | Yes |
| Objective 2: Strengthen minority class support through fold-local cGAN augmentation | Fold-local generative oversampling within the training portion of each CV fold | Ablation analysis comparing models with and without cGAN | Yes |
| Objective 3: Introduce difficulty-aware training through curriculum learning | Difficulty-aware training using fold-local difficulty signals | Ablation analysis comparing models with and without CL | Yes |
| Objective 4: Assess cross-dataset stability under different signal conditions | Fold-disciplined evaluation framework across weak and strong signal datasets | Evaluation on ST000355 and cross-dataset comparison | Yes |

# Chapter 6

# Discussion

## 6.1 Interpretation of Weak Signal Results

The results on weak signal dataset ST004145 suggest that weak signal prediction is constrained by multiple interacting sources of instability rather than by poor separation between the two classes alone. Performance depends not only on whether predictive signal exists, but also on whether the learning process can remain stable under noisy measurements, sparse minority class evidence, and heterogeneous sample reliability. Accordingly, the proposed framework is best understood as a joint response to three related constraints:

- **Feature instability**, arising from noisy and weakly informative metabolomic measurements, which makes learning overly sensitive to raw feature variation across folds.
- **Minority support instability**, arising from limited positive class evidence in small and imbalanced cohorts, which weakens boundary formation and increases fold-to-fold variability.
- **Variation in sample reliability**, arising from heterogeneity in sample informativeness, which makes uniform training schedules less appropriate when some samples are substantially noisier or more ambiguous than others.

Under this interpretation, the advantage of the proposed framework does not lie in any single module alone, but in addressing these three constraints within one fold-disciplined training design. Therefore, the observed improvement should be interpreted as a coordinated response to instability rather than as the isolated effect of any one component.

## 6.2 Cross-Dataset Interpretation

The two datasets do not provide the same type of evidence. ST004145 is the primary weak signal dataset, and carries the main evidential weight for assessing whether the proposed learning design improves robustness. Performance improvement is meaningful because it indicates whether a learning design can remain effective under realistic weak signal constraints. ST000355 serves a different evidential role. Because multiple models reached near-ceiling performance, the room for observable separation was limited. The ST000355 results show that the proposed framework remains competitive under strong signal conditions. Overall, the cross-dataset results suggest that weak signal datasets provide a more revealing setting for evaluating robustness, while strong signal datasets mainly serve as stability checks. For this reason, the main conclusions of this thesis are based on the ST004145 results.

## 6.3 Claim Boundaries and Limitations

The contribution of this thesis should be interpreted within a limited scope. The present results do not show that weak signal prediction has been solved. Predictive discrimination remained moderate, and positive class detection remained limited. The findings therefore support partial improvement under difficult conditions rather than a complete methodological resolution. The results also do not establish clinical readiness. The results also do not demonstrate that the transformer, cGAN, and CL components form a universally optimal combination for clinical metabolomics prediction. Instead, the contribution of the present study is more specific: it supports the value of a unified and fold-disciplined design under small-cohort, weak signal, class-imbalanced conditions.

These boundaries of interpretation are distinct from the empirical and methodological limitations of this study. The empirical coverage remains limited. ST004145 serves as the primary weak signal setting, while ST000355 functions mainly as a strong signal stability check. Although this comparison is informative, it does not replace the need for external validation across broader cohorts, platforms, and disease settings. The findings also remain conditional on the preprocessing assumptions used to construct the analytic matrices. The reported results should therefore be understood in the context of the current preprocessing design, rather than taken as evidence that this missingness treatment is the uniquely correct one.

## 6.4 Future Work

Several future directions follow directly from the limitations. First, broader external validation is needed. Although the proposed framework was evaluated across weak and strong signal datasets, both datasets were derived from the same general application domain. Future work should examine whether the present findings remain stable on additional independent clinical metabolomics cohorts with different cohort structure, measurement characteristics, and preprocessing pipelines. This is necessary to determine whether the observed robustness under the current protocol extends to broader clinical settings.

Second, further work is needed to improve positive class detection. Although the proposed framework improved overall robustness on ST004145, minority class sensitivity remained limited. Future work may therefore examine stronger forms of minority-class support, alternative decision calibration, or training objectives more directly aimed at improving recall without undermining overall stability.

Third, the modeling of heterogeneous sample reliability can be further strengthened. In this study, CL was implemented through fold-local difficulty estimation based on uncertainty, centroid distance, and outlier signals. This design was sufficient to support the current results, but it remains a relatively simple approximation to the broader problem of sample-level reliability variation. More adaptive or data-specific difficulty modeling strategies that better capture heterogeneity remain a direction for future work.

In addition to these specific directions, an open question is whether the central design idea of this thesis remains useful in other small-cohort settings with weak and unstable predictive structure. This should be investigated in future work rather than treated as a general conclusion supported by the current evidence.

# Chapter 7

# Conclusion

This thesis studied weak signal clinical metabolomics prediction in small cohorts under class imbalance and sample-level heterogeneity. Rather than treating these difficulties as separate issues, it approached them as a joint learning problem characterized by unstable feature structure, limited minority class support, and unequal sample reliability. To address this setting, the thesis developed a unified and fold-disciplined framework that integrates transformer representation learning, cGAN augmentation, and CL within stratified CV.

The empirical findings suggest that this integrated design is most informative in the weak signal setting represented by ST004145. On that dataset, the full hybrid framework achieved the strongest overall performance among the compared models, suggesting that combining normalized raw features with learned transformer representations was more effective than relying on either level of representation alone. The ablation results showed that the combination of representation refinement, fold-local minority support and difficulty-aware training was more effective than partial variants evaluated under the same protocol. By contrast, performance differences on ST000355 were much smaller, suggesting that strong signal datasets are less informative for testing methodological robustness. Together, these results indicate that the main evidential weight of this thesis lies in the weak signal setting.

In small-cohort, weak signal, class-imbalanced clinical metabolomics prediction, a unified and fold-disciplined integration of complementary learning mechanisms can improve robustness compared with partial alternatives evaluated under the same setting. Overall, this thesis provides a clearer methodological perspective on weak signal prediction. It shows that representation instability, minority support, and training difficulty should not be handled in isolation, because under weak signal they jointly shape whether predictive structure can be learned in a stable way. The findings suggest that robust learning

in this setting depends not only on model design, but also on whether the learning and evaluation process remains aligned with the underlying sources of instability in the data. In this sense, the thesis contributes not only a predictive framework, but also a more precise way to understand what robust learning requires in this problem setting. This framework may provide a useful methodological basis for future work involving larger cohorts, external validation, and broader weak signal clinical omics settings.

# Appendix A

# Code Snippets

The appendix highlights the main components that correspond directly to the methodology described in Chapter 4, including fold-local preprocessing, difficulty-aware CL, cGAN augmentation, transformer representation learning, hybrid classification, and stratified CV.

## A.1 Difficulty Scoring for Curriculum Learning

Listing A.1 shows the difficulty scoring mechanism used to rank training samples within each fold. The score combines three signals: distance from class centroids, prediction uncertainty from a shallow random forest, and an outlier indicator from isolation forest. The resulting score is used to order samples from easier to harder cases.

```
class DifficultyScorer:
    def __init__(self):
        self.centroids = None
        self.simple_model = None
        self.iso_forest = None

    def fit(self, X, y):
        Xv = X.values if isinstance(X, pd.DataFrame) else X
        yv = y.values if hasattr(y, "values") else y

        self.centroids = {}
        for cls in np.unique(yv):
            mask = yv == cls
            self.centroids[cls] = np.mean(Xv[mask], axis=0)

        self.simple_model = RandomForestClassifier(
            n_estimators=30, max_depth=2, random_state=SEED
```

```
        )
        self.simple_model.fit(Xv, yv)

        self.iso_forest = IsolationForest(contamination=0.1, random_state=SEED)
        self.iso_forest.fit(Xv)
        return self

    def score_difficulty(self, X):
        Xv = X.values if isinstance(X, pd.DataFrame) else X
        n = len(Xv)

        dist = np.zeros(n)
        conf = np.zeros(n)
        outl = np.zeros(n)

        if self.centroids:
            for i, x in enumerate(Xv):
                md = float("inf")
                for _, c in self.centroids.items():
                    md = min(md, np.linalg.norm(x - c))
                dist[i] = md
        if dist.max() > 0:
            dist = dist / dist.max()

        if self.simple_model:
            prob = self.simple_model.predict_proba(Xv)
            conf = 1 - np.max(prob, axis=1)

        if self.iso_forest:
            lab = self.iso_forest.predict(Xv)
            outl = (lab == -1).astype(float)

        return (dist + conf + outl) / 3
```

Listing A.1: Fold-local difficulty scoring used for curriculum learning

## A.2 Curriculum-Guided Transformer Training

The code in Listing A.2 applies CL after fold-local preprocessing. Samples are ranked by estimated difficulty, and progressively larger subsets are introduced across stages. This allows the transformer to begin with easier training cases and then incorporate harder or noisier samples later in the optimization process.

```
class TransformerCurriculumTrainer:
    """
```

```
    IMPORTANT: works on *already preprocessed* data for that fold
    (no internal fit_transform).
    """
    def __init__(self, transformer, curriculum_stages=3, device=None):
        self.transformer = transformer
        self.curriculum_stages = curriculum_stages
        self.device = device or ("cuda" if torch.cuda.is_available() else "cpu")
        self.difficulty_scorer = DifficultyScorer()

    def train_on_preprocessed(self, X_proc_df, y, epochs=30, batch_size=8, lr=0.0005):
        Xp = X_proc_df
        self.difficulty_scorer.fit(Xp, y)
        diff = self.difficulty_scorer.score_difficulty(Xp)
        sorted_idx = np.argsort(diff)

        X_tensor = torch.FloatTensor(Xp.values).to(self.device)
        y_tensor = torch.LongTensor(y.values).to(self.device)

        optimizer = optim.AdamW(self.transformer.parameters(), lr=lr, weight_decay=0.1)
        criterion = nn.CrossEntropyLoss()

        epochs_per_stage = max(1, epochs // self.curriculum_stages)

        for stage in range(self.curriculum_stages):
            frac = 0.3 + 0.7 * (stage / (self.curriculum_stages - 1))
            n = max(1, int(len(sorted_idx) * frac))
            stage_idx = sorted_idx[:n]

            ds = TensorDataset(X_tensor[stage_idx], y_tensor[stage_idx])
            dl = DataLoader(ds, batch_size=batch_size, shuffle=True, drop_last=True)

            for _ in range(epochs_per_stage):
                self.transformer.train()
                for bx, by in dl:
                    optimizer.zero_grad()
                    _, logits = self.transformer(bx)
                    loss = criterion(logits, by)
                    loss.backward()
                    torch.nn.utils.clip_grad_norm_(self.transformer.parameters(), 0.5)
                    optimizer.step()

        return self.transformer
```

Listing A.2: Curriculum-guided training of the transformer encoder

## A.3 cGAN Minority Augmentation

Listing A.3 shows the cGAN augmentation module used within the training fold. The generator and discriminator are trained using fold-local data only, and synthetic minority samples are generated to strengthen minority class support without using validation information.

```
def _cgan_augment_minority_df(
    Xtr_df, ytr,
    random_state=SEED,
    epochs=250,
    batch_size=32,
    z_dim=16,
    target_ratio=1.0,
    lr_g=2e-4,
    lr_d=2e-4,
):
    rng = np.random.RandomState(random_state)
    X = np.asarray(Xtr_df.values, dtype=np.float32)
    y = np.asarray(ytr, dtype=np.int64)

    n_min = int((y == 1).sum())
    n_maj = int((y == 0).sum())
    if n_min == 0 or n_maj == 0 or n_min >= n_maj:
        return Xtr_df, pd.Series(y)

    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
    gan = _TabularCGAN(x_dim=X.shape[1], z_dim=z_dim).to(device)

    X_t = torch.tensor(X, device=device)
    y_t = torch.tensor(y.reshape(-1, 1).astype(np.float32), device=device)

    opt_g = optim.Adam(gan.gen.parameters(), lr=float(lr_g))
    opt_d = optim.Adam(gan.disc.parameters(), lr=float(lr_d))
    bce = nn.BCELoss()

    idx = np.arange(len(X))
    for _ in range(epochs):
        rng.shuffle(idx)
        for s in range(0, len(X), batch_size):
            b = idx[s:s + batch_size]
            xb = X_t[b]
            yb = y_t[b]

            # Train discriminator
            z = torch.randn(len(b), z_dim, device=device)
            fake = gan.g(z, yb).detach()
```

```
            real_pred = gan.d(xb, yb)
            fake_pred = gan.d(fake, yb)
            d_loss = (bce(real_pred, torch.ones_like(real_pred)) +
                      bce(fake_pred, torch.zeros_like(fake_pred))) / 2
            opt_d.zero_grad()
            d_loss.backward()
            opt_d.step()

            # Train generator
            z = torch.randn(len(b), z_dim, device=device)
            gen = gan.g(z, yb)
            pred = gan.d(gen, yb)
            g_loss = bce(pred, torch.ones_like(pred))
            opt_g.zero_grad()
            g_loss.backward()
            opt_g.step()

    target_min = int(np.ceil(float(target_ratio) * n_maj))
    need = max(0, target_min - n_min)
    if need == 0:
        return Xtr_df, pd.Series(y)

    y_gen = np.ones((need, 1), dtype=np.float32)
    z = torch.randn(need, z_dim, device=device)
    with torch.no_grad():
        x_syn = gan.g(z, torch.tensor(y_gen, device=device)).cpu().numpy()

    X_aug = np.vstack([X, x_syn])
    y_aug = np.concatenate([y, np.ones(need, dtype=np.int64)])

    X_aug = pd.DataFrame(X_aug, columns=Xtr_df.columns)
    y_aug = pd.Series(y_aug)
    return X_aug, y_aug
```

Listing A.3: cGAN augmentation

## A.4 Transformer Representation Learning

The model architecture in Listing A.4 defines the transformer encoder used for sample-level representation learning. The processed feature vector is first projected into a compact embedding space, rewritten as a length-1 sequence, refined by a transformer encoder block, and used both as a latent representation and as input to a classifier head.

```
class HormonePathwayTransformer(nn.Module):
    def __init__(self, input_dim=12, embedding_dim=8, num_heads=2,
```

```
                num_layers=1, dropout=0.3):
        super().__init__()
        self.input_projection = nn.Sequential(
            nn.Linear(input_dim, embedding_dim),
            nn.BatchNorm1d(embedding_dim),
            nn.GELU(),
            nn.Dropout(dropout)
        )
        self.pos_encoder = nn.Parameter(torch.randn(1, 1, embedding_dim) * 0.01)

        enc_layer = nn.TransformerEncoderLayer(
            d_model=embedding_dim,
            nhead=num_heads,
            dim_feedforward=embedding_dim * 2,
            dropout=dropout,
            activation="gelu",
            batch_first=True,
            norm_first=True
        )
        self.transformer = nn.TransformerEncoder(
            enc_layer, num_layers=num_layers, enable_nested_tensor=False
        )

        self.classifier = nn.Sequential(
            nn.Linear(embedding_dim, embedding_dim),
            nn.BatchNorm1d(embedding_dim),
            nn.GELU(),
            nn.Dropout(dropout * 0.5),
            nn.Linear(embedding_dim, 2)
        )

    def forward(self, x):
        x = self.input_projection(x)
        x = x.unsqueeze(1) + self.pos_encoder
        x = self.transformer(x).squeeze(1)
        emb = x
        logits = self.classifier(x)
        return emb, logits

    def encode(self, x):
        self.eval()
        with torch.no_grad():
            x = self.input_projection(x)
            x = x.unsqueeze(1) + self.pos_encoder
            x = self.transformer(x).squeeze(1)
        return x.cpu().numpy()
```

Listing A.4: Transformer encoder for sample-level representation learning

## A.5 Hybrid Concatenation Model with LR Head

The proposed model is a hybrid design in which fold-local processed raw features are concatenated with transformer derived embeddings and then passed to a LR head. Listing A.5 shows the corresponding implementation.

```
class SteroidHormoneHybridConcatLR:
    """
    (Processed raw features) + (Transformer embedding) -> LogisticRegression(head)
    """
    def __init__(self, use_curriculum=True, curriculum_stages=3,
                 device=None, lr_params=None, processor_params=None):
        self.use_curriculum = use_curriculum
        self.curriculum_stages = curriculum_stages
        self.device = device or ("cuda" if torch.cuda.is_available() else "cpu")

        pp = processor_params or {}
        self.processor = SteroidHormoneProcessor(**pp)
        self.transformer = HormonePathwayTransformer(input_dim=12).to(self.device)
        self.lr_model = LogisticRegression(
            max_iter=5000, solver="liblinear",
            class_weight="balanced", random_state=SEED
        )

    def fit(self, X_train_raw, y_train):
        _fit_processor_and_transformer(
            self.processor, self.transformer, X_train_raw, y_train,
            use_curriculum=self.use_curriculum,
            curriculum_stages=self.curriculum_stages,
            device=self.device
        )
        Xtr_proc = self.processor.transform(X_train_raw)
        emb_train = self._emb_df_from_proc(Xtr_proc, X_train_raw.index)
        feats = pd.concat(
            [Xtr_proc.reset_index(drop=True),
             emb_train.reset_index(drop=True)],
            axis=1
        )
        self.lr_model.fit(feats, y_train.reset_index(drop=True))
        return self

    def _emb_df_from_proc(self, X_proc_df, index):
        Xt = torch.FloatTensor(X_proc_df.values).to(self.device)
        emb = self.transformer.encode(Xt)
        cols = [f"embedding_{i}" for i in range(emb.shape[1])]
        return pd.DataFrame(emb, columns=cols, index=index)
```

```
    def predict_proba(self, X_raw):
        Xp = self.processor.transform(X_raw)
        emb = self._emb_df_from_proc(Xp, X_raw.index)
        feats = pd.concat(
            [Xp.reset_index(drop=True),
             emb.reset_index(drop=True)],
            axis=1
        )
        return self.lr_model.predict_proba(feats)
```

Listing A.5: Hybrid Concatenation Model with LR Head

## A.6 Stratified CV

The evaluation protocol is implemented in Listing A.6. For each split, sampling is applied only to the training fold, the model is trained on the training portion, and predictions are generated on the untouched validation fold. Fold-wise AUC values are then aggregated into the reported mean and standard deviation.

```
def _cv_driver(
    X, y,
    model_ctor,
    label,
    n_splits=5,
    sampling=None,
    sampling_params=None,
    save_oof=False,
    oof_model_name=None,
    out_dir="oof_outputs"
):
    print(f"\n {label} | {n_splits}-fold CV")
    skf = StratifiedKFold(n_splits=n_splits, shuffle=True, random_state=SEED)

    scores = []
    oof_y_true = []
    oof_y_prob = []

    for tr, va in skf.split(X, y):
        Xtr_raw, Xva_raw = X.iloc[tr], X.iloc[va]
        ytr, yva = y.iloc[tr], y.iloc[va]

        # sampling (TRAIN fold only) to avoid leakage
        Xtr_raw_s, ytr_s = apply_sampling_train_fold_df(
            Xtr_raw, ytr,
            sampling=sampling,
```

```
            random_state=SEED,
            sampling_params=sampling_params
        )

        m = model_ctor()
        m.fit(Xtr_raw_s, ytr_s)
        p = m.predict_proba(Xva_raw)[:, 1]
        scores.append(roc_auc_score(yva, p))

        if save_oof:
            oof_y_true.extend(np.asarray(yva).astype(int).tolist())
            oof_y_prob.extend(np.asarray(p).astype(float).tolist())

    mean_auc = float(np.mean(scores))
    std_auc = float(np.std(scores))
    print(f"{label}: {mean_auc:.4f} {std_auc:.4f}")

    result = {"mean_auc": mean_auc, "std_auc": std_auc}

    if save_oof and len(oof_y_true) > 0:
        model_name = oof_model_name if oof_model_name is not None else label.replace(" "
    , "_")
        metrics = save_oof_outputs(
            model_name=model_name,
            y_true=np.array(oof_y_true),
            y_prob=np.array(oof_y_prob),
            out_dir=out_dir
        )
        result["oof_auc"] = metrics["AUC"]
        result["oof_accuracy"] = metrics["Accuracy"]
        result["oof_precision"] = metrics["Precision"]
        result["oof_recall"] = metrics["Recall"]
        result["oof_f1"] = metrics["F1"]

    return result
```

Listing A.6: Stratified CV

## A.7 GitHub Link

The complete source code and datasets for this thesis are available at:

https://github.com/Jiahui88/Master-s-thesis

# Appendix B

# Software and Packages Used

This research was implemented using the following software tools, libraries, and computational environments:

Table B.1: Software tools, libraries, and computational environments

| **Category** | **Tools and Libraries (with Citations)** |
|---|---|
| Programming Language | Python [61] – Used for model development, data processing, evaluation, and pipeline integration. |
| Deep Learning Framework | PyTorch [62] – Used to implement the transformer-based representation learning module and cGAN, as well as model training and tensor operations. |
| Data Manipulation | NumPy [63] and Pandas [64] – Used for numerical computation, matrix manipulation, tabular data processing, and result aggregation. |
| Machine Learning Utilities | Scikit-learn [65] – Used for stratified cross-validation, preprocessing, baseline models, anomaly detection, and performance evaluation. |
| Baseline Models | Logistic Regression, Support Vector Machine, and Random Forest implemented through Scikit-learn [65] – Used as baseline or comparison models. |

Continued on next page

Table B.1: Software tools, libraries, and computational environments (continued)

| **Category** | **Tools and Libraries (with Citations)** |
|---|---|
| Data Balancing | Imbalanced-learn (SMOTE) [42, 66] – Applied for minority-class oversampling in baseline comparison settings. |
| Gradient Boosting Models | LightGBM [67] and XGBoost [68] – Used as additional comparison models in the classification experiments. |
| Visualization | Matplotlib [69] – Used for plotting ROC curves, precision–recall curves, confusion matrices, and other performance visualizations. |
| Runtime and Utilities | JSON, OS, Glob, Time, Datetime, Shutil, Random, and Warnings (Python standard library) – Used for file handling, reproducibility control, runtime management, and result organization. |
| Development Environment | Jupyter Notebook [70] and Google Colab [71] – Used for experimentation, interactive execution, and workflow prototyping. |

# Appendix C

# Hyperparameters and Model Configuration

This appendix summarizes the important dataset settings and final implementation choices used in the proposed hybrid framework.

Table C.1: Datasets

| **Item** | **ST004145** | **ST000355** |
|---|---|---|
| Role in Thesis | Primary weak signal | Strong signal stability-check |
| Hyperparameter Strategy | Tuned on ST004145 | ST004145 best setting transferred |
| Best Mean AUC | $\mathbf{0.6794 \pm 0.0871}$ | $\mathbf{0.9896 \pm 0.0195}$ |
| Samples | 232 | 211 |
| Class Distribution | 91 cases / 141 controls | 76 cases / 135 controls |
| Selected Features for Modeling | 12 | 227 |

Table C.2: Final Framework and Key Hyperparameter Configuration

| **Parameter** | **ST004145 and ST000355** |
|---|---|
| Final Model | Hybrid_Curriculum_cGAN |
| Preprocessing Principle | Fold-local only |
| Scaler Type | RobustScaler |
| Representation Module | Transformer embedding |
| Embedding Dimension | 8 |
| Attention Heads | 2 |
| Transformer Encoder Layers | 1 |
| Dropout Rate | 0.3 |
| Activation Function | GELU |
| Curriculum Learning | Enabled |
| Curriculum Stages | 2 |
| Minority Class Support | cGAN within training fold |
| cGAN Latent Dimension | 16 |
| cGAN Epochs | 500 |
| cGAN Batch Size | 32 |
| cGAN Learning Rate | 0.0002 |
| Prediction Head | Logistic Regression |
| LR Solver | liblinear |
| LR Penalty | l2 |
| LR $C$ | 0.3 |
| Class Weight | None |
| Max Iterations | 5000 |

# Bibliography


[1] David L. Nelson and Michael M. Cox. *Lehninger Principles of Biochemistry*. Macmillan, 8th edition, 2021. ISBN 9781319228002.

[2] Oliver Fiehn. Metabolomics – the link between genotypes and phenotypes. *Plant Molecular Biology*, 48(1):155–171, 2002. doi: 10.1023/A:1013713905833.

[3] Clary B. Clish. Metabolomics: an emerging but powerful tool for precision medicine. *Molecular Case Studies*, 1(1):a000588, 2015. doi: 10.1101/mcs.a000588.

[4] Gary J Patti, Oscar Yanes, and Gary Siuzdak. Metabolomics: the apogee of the omics trilogy. *Nature reviews Molecular cell biology*, 13(4):263–269, 2012. doi: 10.1038/nrm3314.

[5] Jeremy K. Nicholson and John C. Lindon. Metabonomics. *Nature*, 455(7216): 1054–1056, 2008. doi: 10.1038/4551054a.

[6] Daniel R. Schmidt, Rutulkumar Patel, David G. Kirsch, Caroline A. Lewis, Matthew G. Vander Heiden, and Jason W. Locasale. Metabolomics in cancer research and emerging applications in clinical oncology. *CA: a cancer journal for clinicians*, 71(4):333–358, 2021. doi: 10.3322/caac.21670.

[7] Heidi Goenaga Infante, John Warren, John Chalmers, Geoffrey Dent, Jose Luis Todoli, Joanna Collingwood, Neil Telling, Martin Resano, Andreas Limbeck, Torsten Schoenberger, et al. Glossary of methods and terms used in analytical spectroscopy (iupac recommendations 2019). *Pure and applied chemistry*, 93(6): 647–746, 2021. doi: 10.1515/pac-2019-0203.

[8] John FJ Todd. Recommendations for nomenclature and symbolism for mass spectroscopy (including an appendix of terms used in vacuum technology).(recommendations 1991). *Pure and applied chemistry*, 63(10):1541–1566, 1991. doi: 10.1351/pac199163101541.

[9] Warwick B. Dunn, David I. Broadhurst, Helen J. Atherton, Royston Goodacre, and Julian L. Griffin. Systems level studies of mammalian metabolomes: the roles of

mass spectrometry and nuclear magnetic resonance spectroscopy. *Chemical society reviews*, 40(11):387–426, 2011. doi: 10.1039/b906712b.

[10] Abdul-Hamid Emwas, Raja Roy, Ryan T. McKay, Leonardo Tenori, Edoardo Saccenti, G. A. Nagana Gowda, Daniel Raftery, Fatimah Alahmari, Lukasz Jaremko, Mariusz Jaremko, and David S. Wishart. NMR spectroscopy for metabolomics research. *Metabolites*, 9(7):123, 2019. doi: 10.3390/metabo9070123.

[11] Thomas M. Annesley. Ion suppression in mass spectrometry. *Clinical Chemistry*, 49(7):1041–1044, 2003. doi: 10.1373/49.7.1041.

[12] Olga Hrydziuszko and Mark R. Viant. Missing values in mass spectrometry based metabolomics: an undervalued step in the data processing pipeline. *Metabolomics*, 8(Suppl 1):161–174, 2012. doi: 10.1007/s11306-011-0366-4.

[13] Martin Rusilowicz, Michael Dickinson, Adrian Charlton, Simon O'keefe, and Julie Wilson. A batch correction method for liquid chromatography–mass spectrometry data that does not depend on quality control samples. *Metabolomics*, 12(3):56, 2016. doi: 10.1007/s11306-016-0972-2.

[14] Warwick B. Dunn, David Broadhurst, Paul Begley, Eva Zelena, Sheila Francis-McIntyre, Nicola Anderson, Mary Brown, Joshua D. Knowles, Anthony Halsall, John N. Haselden, Andrew Nicholls, Ian D. Wilson, Douglas B. Kell, and Royston Goodacre. Procedures for large-scale metabolic profiling of serum and plasma using gas chromatography and liquid chromatography coupled to mass spectrometry. *Nature Protocols*, 6(7):1060–1083, 2011. doi: 10.1038/nprot.2011.335.

[15] Kieu Trinh Do, Simone Wahl, Johannes Raffler, Sophie Molnos, Michael Laimighofer, Jerzy Adamski, Karsten Suhre, Konstantin Strauch, Annette Peters, Christian Gieger, Claudia Langenberg, Isobel D. Stewart, Fabian J. Theis, Harald Grallert, Gabi Kastenmüller, and Jan Krumsiek. Characterization of missing values in untargeted ms-based metabolomics data and evaluation of missing data handling strategies. *Metabolomics*, 14(10):128, 2018. doi: 10.1007/s11306-018-1420-2.

[16] Runmin Wei, Jingye Wang, Mingming Su, Erik Jia, Shaoqiu Chen, Tianlu Chen, and Yan Ni. Missing value imputation approach for mass spectrometry-based metabolomics data. *Scientific Reports*, 8(1):663, 2018. doi: 10.1038/s41598-017-19120-0.

[17] Riccardo Di Guida, Jasper Engel, J. William Allwood, Ralf J. M. Weber, Matthew R. Jones, Ute Sommer, Mark R. Viant, and Warwick B. Dunn. Non-targeted uhplc-ms metabolomic data processing methods: A comparative investigation of normalisation, missing value imputation, transformation and scaling. *Metabolomics*, 12(5):93, 2016. doi: 10.1007/s11306-016-1030-9.

[18] Jinhua Chi, Jingmin Shu, Ming Li, Rekha Mudappathi, Yan Jin, Freeman Lewis, Alexandria Boon, Xiaoyan Qin, Li Liu, and Haiwei Gu. Artificial intelligence in metabolomics: A current review. *TrAC Trends in Analytical Chemistry*, 178:117852, 2024. doi: 10.1016/j.trac.2024.117852.

[19] Florian Abram, Marcus Jünger, Anne Mattes, Jessica Wörmann, Sascha Rohn, and Riyas Ahamed Vettukattil. A comprehensive evaluation of metabolomics data preprocessing methods for deep learning. *Metabolites*, 12(3):248, 2022. doi: 10.3390/metabo12030248.

[20] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE transactions on neural networks and learning systems*, 35(6):7499–7519, 2022. doi: 10.1109/TNNLS.2022.3229.161.

[21] Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020. doi: 10.48550/arXiv.2012.06678.

[22] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016. doi: 10.1007/s13748-016-0094-0.

[23] Alberto Fernández, Salvador García, Francisco Herrera, and Nitesh V. Chawla. Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, 61:863–905, 2018. doi: 10.1613/jair.1.11192.

[24] Francisco Traquete, Marta Sousa Silva, and António E. N. Ferreira. Enhancing supervised analysis of imbalanced untargeted metabolomics datasets using a cwgan-gp framework for data augmentation. *Computers in Biology and Medicine*, 184:109414, 2025. doi: 10.1016/j.compbiomed.2024.109414.

[25] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009. doi: 10.1145/1553374.1553380.

[26] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576, 2021. doi: 10.1109/TPAMI.2021.3069908.

[27] International Conference on Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human Use. ICH E9: Statistical principles for clinical trials. ICH Harmonised Tripartite Guideline, Step 5, 1998. URL https://www.ema.europa.eu/en/documents/scientific-guideline/ich-e-9-statistical-principles-clinical-trials-step-5_en.pdf.

[28] U.S. Food and Drug Administration. Multiple endpoints in clinical trials guidance for industry, 2022. URL https://www.fda.gov/media/162416/download.

[29] Elizabeth A. Eisenhauer, Patrick Therasse, Jan Bogaerts, Lawrence H. Schwartz, Daniel Sargent, Robert Ford, Janet Dancey, Susan Arbuck, Susan Gwyther, Margaret Mooney, Lawrence Rubinstein, Lalitha Shankar, Laura Dodd, Richard Kaplan, Denis Lacombe, and Jaap Verweij. New response evaluation criteria in solid tumours: Revised RECIST guideline (version 1.1). *European Journal of Cancer*, 45 (2):228–247, 2009. doi: 10.1016/j.ejca.2008.10.026.

[30] Colin A. Smith, Elizabeth J. Want, Gary O'Maille, Ruben Abagyan, and Gary Siuzdak. XCMS: Processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Analytical Chemistry*, 78 (3):779–787, 2006. doi: 10.1021/ac051437y.

[31] Ralf Tautenhahn, Christoph Böttcher, and Steffen Neumann. Highly sensitive feature detection for high resolution LC/MS. *BMC Bioinformatics*, 9(1):504, 2008. doi: 10.1186/1471-2105-9-504.

[32] Carsten Kuhl, Ralf Tautenhahn, Christoph Böttcher, Tony R. Larson, and Steffen Neumann. CAMERA: An integrated strategy for compound spectra extraction and annotation of liquid chromatography/mass spectrometry data sets. *Analytical Chemistry*, 84(1):283–289, 2012. doi: 10.1021/ac202450g.

[33] Amit Moscovich and Saharon Rosset. On the cross-validation bias due to unsupervised preprocessing. *Journal of the Royal Statistical Society: Series B Statistical Methodology*, 84(4):1474–1502, 2022. doi: 10.1111/rssb.12537.

[34] Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7(1):91, 2006. doi: 10.1186/1471-2105-7-91.

[35] Claude Nadeau and Yoshua Bengio. Inference for the generalization error. *Advances in Neural Information Processing Systems*, 12, 1999.

[36] Yoshua Bengio and Yves Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research*, 5(Sep):1089–1105, 2004.

[37] Haibo He and Eduardo A. Garcia. Learning from imbalanced data. In *IEEE Transactions on Knowledge and Data Engineering*, volume 21, pages 1263–1284. Ieee, 2009. doi: 10.1109/TKDE.2008.239.

[38] Damjan Krstajic, Ljubomir J. Buturovic, Dermot E. Leahy, and Simon Thomas. Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of Cheminformatics*, 6(1):10, 2014. doi: 10.1186/1758-2946-6-10.

[39] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015. doi: 10.1371/journal.pone.0118432.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. doi: 10.48550/arXiv.1706.03762.

[41] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009. doi: 10.1109/TKDE.2008.239.

[42] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[43] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. doi: 10.1145/3422622.

[44] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. doi: doi.org/10.48550/arXiv.1411.1784.

[45] Tianyi Zhou, Shengjie Wang, and Jeff A. Bilmes. Curriculum learning by dynamic instance hardness. *Advances in neural information processing systems*, 33:8602–8613, 2020.

[46] M Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23, 2010.

[47] Nigel P. Long, Tran D. Nghi, Yun-Pyo Kang, Nguyen Hoang Anh, Hyeonuk Kim, and Sang Won Kwon. Toward a standardized strategy of clinical metabolomics for the advancement of precision medicine. *Metabolites*, 10(2):51, 2020. doi: 10.3390/metabo10020051.

[48] Ulf W. Liebal, Anh N. Phan, Manogna Sudhakar, Karthik Raman, and Lars M. Blank. Machine learning applications for mass spectrometry-based metabolomics. *Metabolites*, 10(6):243, 2020. doi: 10.3390/metabo10060243.

[49] Dominik Reinhold, Harrison Pielke-Lombardo, Sean Jacobson, Debashis Ghosh, and Katerina Kechris. Pre-analytic considerations for mass spectrometry-based untargeted metabolomics data. In Angelo D'Alessandro, editor, *High-Throughput Metabolomics: Methods and Protocols*, volume 1978 of *Methods in Molecular Biology*, pages 323–340. Springer, 2019. doi: 10.1007/978-1-4939-9236-2_20.

[50] Gilbert Badaro, Mohammed Saeed, and Paolo Papotti. Transformers for tabular data representation: A survey of models and applications. *Transactions of the Association for Computational Linguistics*, 11:227–249, 2023. doi: 10.1162/tacl_a_00544.

[51] Jun-Peng Jiang, Si-Yang Liu, Hao-Run Cai, Qile Zhou, and Han-Jia Ye. Representation learning for tabular data: A comprehensive survey. *arXiv preprint arXiv:2504.16109*, 2025. doi: 10.48550/arXiv.2504.16109.

[52] Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y. Chen. Ctab-gan+: Enhancing tabular data synthesis. *arXiv preprint arXiv:2204.00401*, 2022. doi: 10.48550/arXiv.2204.00401.

[53] Gayeong Eom and Haewon Byeon. Searching for optimal oversampling to process imbalanced data: Generative adversarial networks and synthetic minority oversampling technique. *Mathematics*, 11(17):3656, 2023. doi: 10.3390/math11173656.

[54] Marcos Escudero-Viñolo and Alejandro López-Cifuentes. Ccl: Class-wise curriculum learning for class imbalance problems. In *2022 IEEE International Conference on Image Processing*. IEEE, 2022.

[55] S. Chaudhry et al. Dynamic data distribution-based curriculum learning. *Information Sciences*, 2025. doi: 10.1016/j.ins.2025.121924.

[56] Manish Sud, Eoin Fahy, Daniel Cotter, Kamran Azam, Indhumathi Vadivelu, Charles Burant, Arthur Edison, Oliver Fiehn, Richard Higashi, Kannan Nair, et al. Metabolomics workbench: An international repository for metabolomics data and metadata, metabolite standards, protocols, tutorials and training, and analysis tools. *Nucleic Acids Research*, 44(D1):D463–D470, 2016. doi: 10.1093/nar/gkv1042.

[57] Metabolomics Workbench. Study st004145: Human plasma metabolomics study, 2025. URL https://www.metabolomicsworkbench.org/data/DRCCMetadata.php?Mode=Study&StudyID=ST004145. NIH Common Fund National Metabolomics Data Repository.

[58] Metabolomics Workbench. Study st000355: Human plasma metabolomics study, 2025. URL https://www.metabolomicsworkbench.org/data/DRCCMetadata.php?Mode=Study&StudyID=ST000355. NIH Common Fund National Metabolomics Data Repository.

[59] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.

[60] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[61] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, 2009. ISBN 9781441412690.

[62] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

[63] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, et al. Array programming with numpy. *Nature*, 585(7825): 357–362, 2020. doi: 10.1038/s41586-020-2649-2.

[64] Wes McKinney et al. Data structures for statistical computing in python. *scipy*, 445(1):51–56, 2010.

[65] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[66] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

[67] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[68] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016. doi: 10.1145/2939672.2939785.

[69] John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

[70] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E. Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B. Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter notebooks – a publishing format for reproducible computational workflows. In Fernando Loizides and Birgit Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90. IOS Press, 2016. doi: 10.3233/978-1-61499-649-1-87.

[71] Google. Google colaboratory, 2026. URL https://colab.google/. Accessed: 2026-04-11.