ACTIVE LEARNING FOR DRUG BLOOD-BRAIN BARRIER PERMEABILITY PREDICTION

by
AHMED MOHAMED MAHMOUD ELMOSELHY SALEM

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of Master of Science

Sabancı University
October 2024

ABSTRACT

ACTIVE LEARNING FOR DRUG BLOOD-BRAIN BARRIER PERMEABILITY PREDICTION

AHMED MOHAMED MAHMOUD ELMOSELHY SALEM

Data Science M.S. THESIS, October 2024

Thesis Supervisor: Assoc. Prof. ÖZNUR TAŞTAN OKAN

Keywords: Active Learning, Dynamic Sampling, Scaffold Splitting, Molecular Scaffolds, Blood-Brain Barrier, QSAR

The blood-brain barrier (BBB) is a highly selective, semipermeable border that regulates the transfer of chemicals between the circulatory and central nervous systems (CNS). Assessing whether a compound can permeate the BBB is critical in drug development for treating CNS disorders, as it determines the compound's ability to reach targets within the brain. The chemical space is vast, and traditional methods for measuring a chemical compound's BBB permeability are time-consuming and costly. However, with the availability of open datasets for compounds with experimentally verified permeability assessments, several machine learning (ML) models have been proposed to accelerate BBB permeability prediction. A large pool of labeled examples is necessary for a machine learning model to learn BBB permeability status in a supervised setting. Yet, the size of labeled datasets remains far from comprehensive when compared to the immense chemical space, limiting the effectiveness of traditional supervised passive learning procedures. The active learning (AL) framework offers an alternative. Active learners iteratively achieve high-accuracy classifiers with fewer label requests compared to passive learning by strategically selecting which examples to label in each iteration. In this thesis,

we explored various AL strategies for predicting the BBB permeability of chemical compounds and compared their effects on the performance of machine learning models. Specifically, we examined the following sampling strategies: random sampling, uncertainty-based sampling, and dissimilarity-based sampling. Additionally, we proposed and implemented two novel AL methods: explore-intensify and round-robin cycle switching. We also performed a comparative analysis of all the AL methods against passive learning in two separate setups: one based on a label-stratified splitting technique and another based on splitting the data by the molecular scaffolds of the chemical compounds, which is a more challenging evaluation setup. Our results show that the scaffold-splitting setup resulted in lower performance compared to the label-stratified setup across both passive and active learning paradigms. Furthermore, our experiments revealed that the active learning approaches we implemented matched the performance of passive learning in nearly every performance metric we tested, typically after labeling only 10-65% of the data, depending on the specific metric. Moreover, the results of our proposed active learning methods demonstrated that the round-robin cycle switching strategy outperformed other active learning strategies in the stratified-split setup. This highlights the potential of dynamic AL methods to efficiently reduce the need for large labeled datasets while maintaining high performance in predicting BBB permeability.

ÖZET

İLAÇLARIN KAN-BEYIN BARIYERI GEÇIRGENLIĞINI TAHMIN ETMEDE AKTIF ÖĞRENME

AHMED MOHAMED MAHMOUD ELMOSELHY SALEM

Veri Bilimi Yüksek Lisans TEZİ, Ekim 2024

Tez Danışmanı: Assoc. Prof. ÖZNUR TAŞTAN OKAN

Anahtar Kelimeler: Aktif Öğrenme, Dinamik Örnekleme, İskele Ayrımı, Moleküler İskeleler, Kan-Beyin Bariyeri, QSAR

Kan-beyin bariyeri (BBB), dolaşım sistemi ile merkezi sinir sistemi (CNS) arasındaki kimyasal transferi düzenleyen, oldukça seçici, yarı geçirgen bir sınırdır. Bir bileşiğin BBB'yi geçip geçemeyeceğini değerlendirmek, beynin içindeki hedeflere ulaşma yeteneğini belirlediği için CNS bozukluklarının tedavisinde ilaç geliştirme açısından kritik öneme sahiptir. Kimyasal uzay çok geniştir ve kimyasal bir bileşiğin BBB geçirgenliğini ölçmek için kullanılan geleneksel yöntemler zaman alıcı ve maliyetlidir. Bununla birlikte, deneysel olarak doğrulanmış geçirgenlik değerlendirmelerine sahip bileşikler için açık veri kümelerinin bulunması sayesinde, BBB geçirgenliği tahminini hızlandırmak için çeşitli makine öğrenimi (ML) modelleri önerilmiştir. Bir makine öğrenimi modelinin, denetimli bir ortamda BBB geçirgenlik durumunu öğrenebilmesi için büyük bir etiketlenmiş örnek havuzuna ihtiyaç vardır. Ancak, etiketlenmiş veri setlerinin boyutu, devasa kimyasal uzay karşısında hala kapsamlı olmaktan uzaktır ve bu durum, geleneksel denetimli pasif öğrenme prosedürlerinin etkinliğini sınırlar. Aktif öğrenme (AL) çerçevesi bu duruma bir alternatif sunar. Aktif öğrenme modelleri, her iterasyonda hangi örneklerin etiketleneceğini stratejik olarak seçerek, pasif öğrenmeye kıyasla daha az etiket talebiyle yüksek doğruluklu sınıflandırıcılar elde eder. Bu tezde, kimyasal bileşiklerin BBB geçirgenliğini tahmin etmek için çeşitli AL stratejilerini inceledik ve bunların makine öğrenimi modellerinin performansı üzerindeki etkilerini karşılaştırdık. Özellikle şu örnekleme stratejilerini inceledik: rastgele örnekleme, belirsizlik temelli örnekleme ve benzemezlik temelli örnekleme. Ek olarak, keşif-yoğunlaştırma ve döngüsel yuvarlak-robin adında iki yeni AL yön-

temini önerdik ve uyguladık. Ayrıca, tüm AL yöntemlerini pasif öğrenme yöntemleriyle iki ayrı kurulumda karşılaştırmalı olarak analiz ettik: biri verilerin etiket sınıfına göre katmanlı olarak ayrıldığı bir kurulum, diğeri ise kimyasal bileşiklerin moleküler iskeletlerine dayalı olarak ayrıldığı ve daha zor bir değerlendirme ortamı sunan bir kurulum. Sonuçlarımız, iskelet ayrımı kurulumunun, hem pasif hem de aktif öğrenme paradigmalarında, etiket katmanlı ayrım kurulumuna göre daha düşük performansla sonuçlandığını göstermektedir. Ayrıca, deneylerimiz, uyguladığımız aktif öğrenme yaklaşımlarının, test edilen hemen hemen her performans metriğinde, genellikle yalnızca verilerin %10-65'i etiketlendikten sonra pasif öğrenme performansıyla eşleştiğini ortaya koymuştur. Dahası, önerdiğimiz aktif öğrenme yöntemlerinin sonuçları, döngüsel yuvarlak-robin stratejisinin, katmanlı ayrım kurulumunda diğer aktif öğrenme stratejilerinden daha iyi performans gösterdiğini göstermiştir. Bu, dinamik AL yöntemlerinin, büyük etiketlenmiş veri setlerine duyulan ihtiyacı etkin bir şekilde azaltırken BBB geçirgenliğini tahmin etmede yüksek performansı koruma potansiyelini vurgulamaktadır.

ACKNOWLEDGMENTS

I would like to begin by thanking Allah for granting me the strength, patience, and guidance to complete this research work. Without His blessings and mercy, none of this would have been possible.

I would like to express my deepest gratitude to my advisor, Dr. Öznur Taşţan, whose guidance, advice, insights, and unwavering support have been the cornerstone of this journey. Her understanding, consideration, and motivation have not only made this thesis possible but have also profoundly shaped my academic and personal growth. Öznur Hoca, thank you for believing in me and inspiring me throughout this journey. The valuable advice you have kindly shared will stay with me always.

I am also grateful to the jury members, Dr. Gülden Olgun and Dr. Nur Mustafaoğlu Varol, for their time, valuable feedback, and insightful suggestions, which significantly improved this research.

I extend my sincere appreciation to my professors, Dr. Albert Levi, Dr. Berrin Yanıkoğlu, Dr. Duygu Karaoğlan Altop, Dr. Onur Varol, and Dr. Yuki Kaneko, whose courses and mentorship have enriched my knowledge and shaped the direction of this thesis.

I sincerely thank Sabancı University for providing the resources and supportive environment that facilitated my academic journey. Special thanks to the FENS Graduate Student Office for their dedication and support, as well as the administrative staff whose efforts ensured a smooth and enriching experience.

To my dear friends, Ahmed Fouad, Ahmmad Saleh, Anes Abdennebi, Dr. Aydın Gerek, Emre Yavaş, Hazem Nomer, Dr. Khaled El-Wazan, Dr. Khalid Ibrahim, Dr. Mohamed Mousa, Mohammed El-Gamal, Moustafa Omar, and Dr. Selim Tanrıseven, your support and encouragement have been invaluable.

To my brothers, and to my parents and family, your love and unwavering belief in me have been my anchor through this challenging yet rewarding journey.

Each one of you has had a positive impact on my life and this work in ways that words cannot fully capture. I am truly thankful for your presence and support during this remarkable journey.

Dedication

This thesis is dedicated to my family, whose constant support and encouragement have been my strength throughout this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

This chapter introduces the research problem that this thesis tackles, shows its impact, and presents an overview of the prerequisite knowledge needed to understand the present thesis.

1.1 Problem Description

Predicting the permeability of BBB for a chemical compound is one of the most critical stages in the drug discovery pipeline. This becomes even more essential when developing drugs that target the CNS or treat CNS-related diseases. The traditional method for predicting the BBB permeability for a chemical compound involves experimental techniques such as in vitro and in vivo testing. However, these experimental techniques are costly and time-consuming. As a result, computational approaches, particularly machine learning models, have gained significant attention. These models can predict BBB permeability by analyzing molecular descriptors, chemical fingerprints, or other features derived from the chemical structure of a compound. Using large data sets and sophisticated algorithms, machine learning models can provide faster and cost-effective, helping optimize the drug discovery process. Moreover, integration of active learning and advanced molecular representations, such as embeddings, can further enhance the accuracy and generalizability of these predictions, mainly when dealing with diverse chemical scaffolds.

1.2 Background

This section introduces the necessary background for this work. This includes concepts from chem-informatics, chemistry, and active machine learning.

1.2.1 Blood-Brain Barrier Permeability

The Blood-Brain Barrier (BBB) is one of the most complex barriers in the human body. It is a membrane between the blood vessels in the brain and the other brain tissues. A longitudinal view of the BBB is shown in figure 1.1. The BBB regulates the movement of molecules in and out of the brain. One of the most essential functions of the BBB is protecting the brain from harmful toxic compounds from sneaking into the brain tissues. However, this functionality comes with a price. The strict selectivity of the BBB does not always work in favor of the body's needs. It follows strict rules when determining whether or not to allow compounds to pass through it into the brain. These rules can, sometimes, be an obstacle in the way of a drug targeting an area inside the brain. Thus preventing the drug from reaching its target, hindering, or even eliminating the effect of that drug. Understanding how the BBB decides on which to allow and which not to is a long-time research problem that scientists have been working on for decades (Cornelissen et al., 2023). Biologists and medicinal chemists have achieved significant advancements in this area, later joined by computational chemists employing their computational skills to accelerate the work of biologists in wet labs.

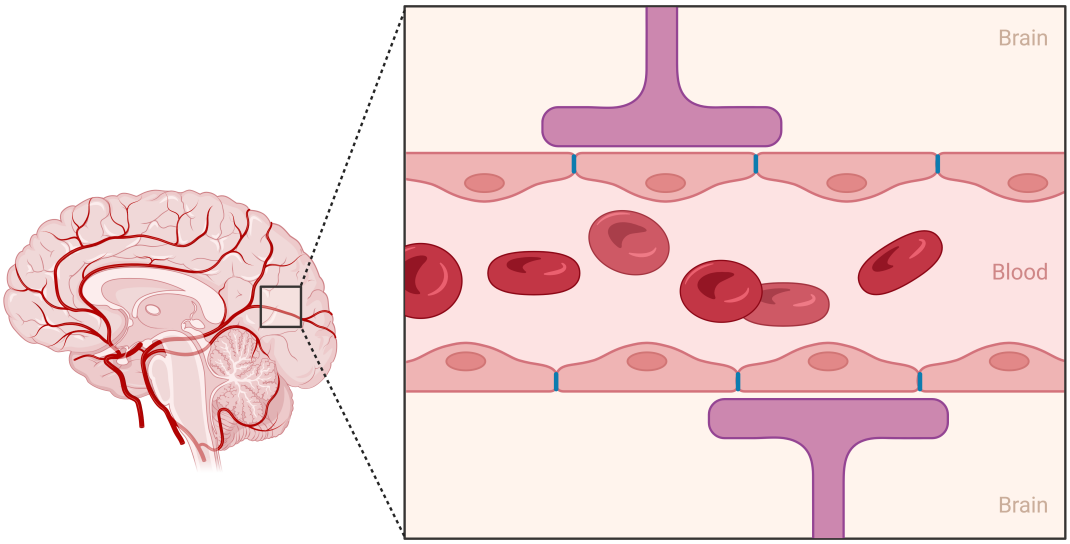
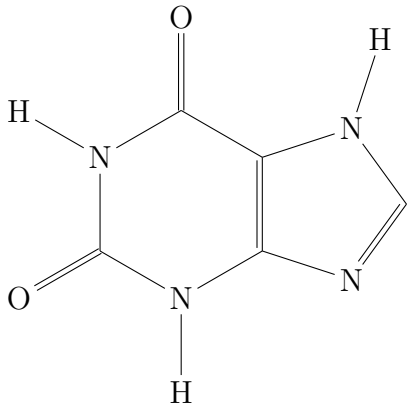


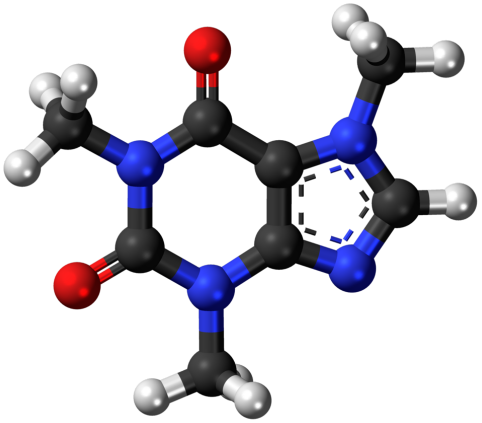
Figure 1.1 A longitudinal illustration of the BBB.

1.2.2 Computational Representations of Molecules

Molecules are typically represented by their structural and chemical formulas, which convey essential information about atomic arrangements and connections. For example, Figure 1.2 illustrates the 2D and 3D structures of caffeine, with its chemical formula $C_8H_{10}N_4O_2$. While these visualizations are intuitive for chemists, computational tasks require molecules to be translated into machine-readable formats.



(a) 2D Representation of Caffeine



(b) 3D Representation of Caffeine

Figure 1.2 2D (A) and 3D (B) representations of caffeine, illustrating its molecular structure.

Among the many available representations, SMILES (Simplified Molecular Input Line Entry System) is one of the most widely used. Introduced in 1988 (Weininger, 1988), SMILES encodes chemical structures into linear textual strings, enabling efficient storage, retrieval, and processing of molecular data. Despite its utility, early versions of SMILES were non-unique, allowing multiple valid strings to represent the same molecule. This ambiguity created challenges in machine learning, where consistent and high-quality data representations are critical. To address this, canonical SMILES were developed (Weininger et al., 1989), ensuring that each molecule has a unique representation. Additionally, alternative formats like SELFIES (Self-Referencing Embedded Strings) have emerged, offering syntactic validity and robustness against errors.

Table 1.1 provides examples of SMILES and SELFIES representations for caffeine. These formats are vital for integrating chemical data into computational pipelines, enabling downstream tasks such as property prediction, molecular generation, and virtual screening. In this thesis, we focus on key molecular representations relevant to the methodologies employed in our study.

Table 1.1 SMILES and SELFIES Representations of Caffeine

SMILES string of Caffeine:
`CN1C=NC2=C1C(=O)N(C(=O)N2C)C`
SELFIES string of Caffeine:
`[C][N][C][=Branch1][C][=O][C][=C][Branch1]` `[#Branch1][N][=C][N][Ring1][Branch1][C][N]` `[Branch1][C][C][C][Ring1][N][=O]`

1.2.2.1 Self-Referencing Embedded Strings (SELFIES)

SELFIES, which stands for Self-Referencing Embedded Strings, is a newer textual representation of molecules that has gained popularity in recent years (Krenn et al., 2020). SELFIES can be considered an improved and more robust alternative to SMILES. While SMILES offers advantages, one of its key limitations is that it does not guarantee the generation of valid molecular structures (i.e., chemically feasible molecules). SELFIES addresses this issue by ensuring that every generated string corresponds to a valid molecule, making it particularly attractive for cheminformatics applications. Building upon the SELFIES representation, embedding techniques have been developed to convert these strings into numerical vectors suitable for machine learning models. One notable approach is SELFormer (Yüksel et al., 2023), a transformer-based chemical language model that utilizes SELFIES as input to learn flexible and high-quality molecular representations. In this thesis, we used SELFIES embeddings generated via SELFormer to represent molecular structures numerically. This approach leverages the syntactic validity guaranteed by SELFIES and the advanced representation learning capabilities of transformer architectures, facilitating more accurate predictions in our machine learning models.

1.2.2.2 Extended-Connectivity Fingerprints (ECFP)

Extended-Connectivity Fingerprints (ECFP) are another widely used molecular representation, particularly suited for machine learning tasks. ECFP captures important local substructures of molecules—such as groups of atoms and their connections—essentially creating a "snapshot" of a molecule's core building blocks (Rogers & Hahn, 2010). This representation focuses on the structural features that are most relevant for predicting molecular properties.

The captured substructures are then transformed into a fixed-length binary vector, composed of 0s and 1s, that can be directly processed by machine learning models. Due to their ability to encode critical molecular features effectively, ECFPs are extensively used for tasks such as property prediction and virtual screening. Their utility and relevance are key reasons for their inclusion in this thesis.

1.2.3 Active Learning

Active learning is a powerful approach within the field of machine learning. In this approach, the learning algorithm selects the most informative data points to be labeled by an oracle rather than passively receiving a randomly sampled training set (Settles, 2009). By carefully choosing which data to learn from, active learning aims to develop accurate models with substantially less data than traditional supervised learning methods (figure 1.3).

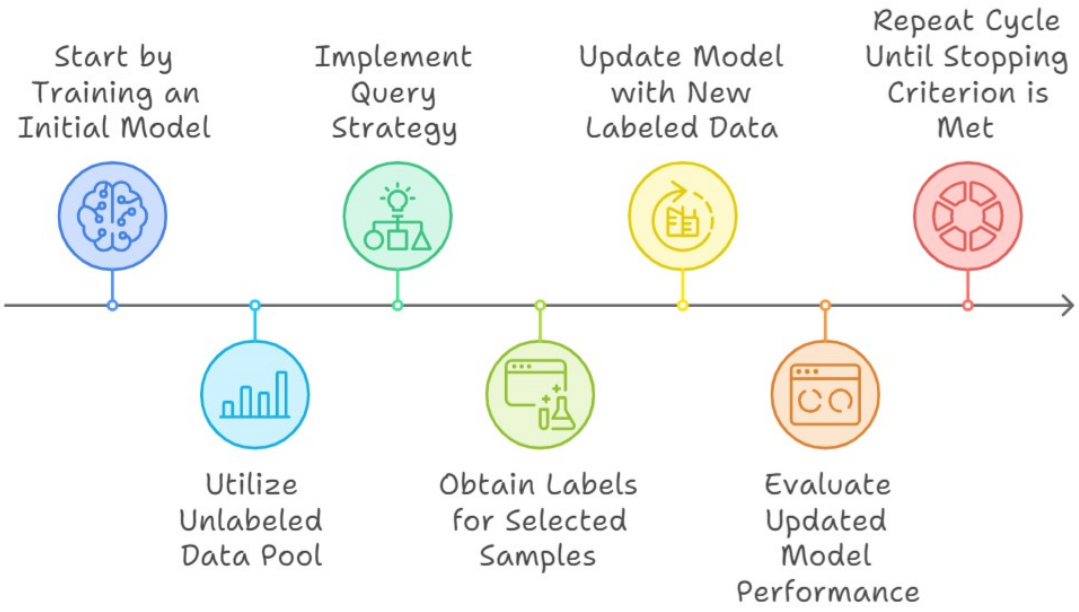


Figure 1.3 An Illustration for the general active learning cycle

Figure 1.3 presents a general overview of the iterative active learning process, which is highly relevant to machine learning tasks where labeled data is scarce, expensive, or difficult to obtain. The active learning paradigm strategically selects the most informative samples from an unlabeled data pool for labeling, allowing the model to improve its performance with minimal labeling effort. Each stage of the active learning workflow depicted in the figure is described below:

- **Start by Training an Initial Model:** The active learning cycle begins by training a machine learning model on an initial labeled data set.
- **Utilize Unlabeled Data Pool:** A key component of active learning is the presence of a large pool of unlabeled data from which the algorithm selects new data points for labeling. This pool remains unlabeled during each iteration, and only specific samples that are believed to be most informative are selected to be labeled.

- **Implement Query Strategy:** The query strategy is a core element of the active learning process. It determines how data points are selected from the unlabeled pool for labeling. Common query strategies include uncertainty sampling, where the model selects samples it is least confident about, and diversity-based sampling, where diverse molecular structures are chosen to enrich the training data. This ensures that each newly labeled sample maximally contributes to improving the model's performance.

- **Obtain Labels for Selected Samples:** Once the query strategy identifies informative samples, they are sent to an oracle—usually a human expert or an experimental process—for labeling. In the case of BBB permeability prediction, this step would involve conducting laboratory experiments to determine whether selected molecules can cross the blood-brain barrier.

- **Update Model with New Labeled Data:** After obtaining the labels, the model is updated/retrained using both the newly labeled data and the previously labeled data. This step ensures that the models progressively improve by incorporating the new knowledge gained from the labeled samples.

- **Evaluate Updated Model Performance:** The performance of the updated model is evaluated to assess its improvement compared to previous iterations. This evaluation is based on standard machine learning metrics such as accuracy, ROC-AUC, or F1-score.

- **Repeat Cycle Until Stopping Criteria Are Met:** The active learning cycle is repeated until predefined stopping criteria are met. These criteria may include achieving a certain performance threshold, exhausting a labeling budget, or detecting that further labeled data no longer significantly improves model performance.

The active learning iterative paradigm aims to minimize labeling efforts while maximizing the model's generalization capabilities. It also enables efficient model improvement by focusing labeling efforts on the most informative samples, thus reducing the overall experimental burden.

The following section provides an overview of key active learning sampling strategies: random sampling, uncertainty sampling, and dissimilarity sampling, all of which fall under the category of pool-based active learning.

1.2.3.1 Pool-based Active Learning

In a pool-based active learning framework, we assume that we have access to a small pool of labeled data D_L and a large pool of unlabeled data D_U. The learner first trains a model on the labeled data D_L. Then, it uses a query strategy to select the most informative instances from the unlabeled pool D_U and requests their labels from an oracle. The newly labeled data points are then added to D_L, and the process repeats until a stopping criterion is met. The stopping criterion can be until you run out of the labeling budget or until you reach a predefined performance threshold.

1.2.3.2 Random Sampling

Random sampling is the most basic method of selecting data points to be labeled. Data points are randomly selected from the pool of unlabeled data D_U and labeled. This approach serves as a computationally inexpensive baseline to evaluate the effectiveness of more advanced sampling strategies.

While random sampling is straightforward, its main drawback is that it does not prioritize informativeness or diversity. As a result, it may require a larger number of labeled samples to achieve the same performance as smarter strategies. However, it remains a critical baseline due to its simplicity and widespread applicability.

1.2.3.3 Uncertainty Sampling

The uncertainty sampling strategy relies on the assumption that the most informative data point for the model is the one that it is most uncertain about. There are many available measures to quantify the uncertainty of the model. In our implementation, we used Shannon entropy.

Given a model that predicts a probability distribution $p(y \mid x)$ over the possible labels y for a data point x, the Shannon entropy $H(x)$ can be used as a measure of uncertainty. Shannon entropy is defined as:

$$H(x) = -\sum_{i=1}^{C} p(y_i \mid x) \log p(y_i \mid x) \tag{1.1}$$

where:

- C is the number of possible classes,

- $p(y_i \mid x)$ is the predicted probability for class y_i.

For binary classification, let $p = p(y = 1 \mid x)$ be the predicted probability of class 1. The Shannon entropy $H(x)$ can then be simplified to:

$$H(x) = -p \log p - (1-p) \log(1-p) \tag{1.2}$$

The data point x^* that the model is most uncertain about, and hence considered most informative, is the one that maximizes the Shannon entropy:

$$x^* = \arg\max_x H(x) \tag{1.3}$$

1.2.3.4 Dissimilarity Sampling

Dissimilarity sampling focuses on selecting data points from the unlabeled pool D_U that are most different from the labeled data D_L. The underlying assumption is that data points which are dissimilar to the already labeled set are likely to add diversity and provide new information for the model.

To quantify dissimilarity, we used the **cosine distance** metric. The cosine distance measures how different two vectors are in terms of their orientation in the feature space. Given two vectors $\mathbf{u}$ and $\mathbf{v}$, the cosine similarity is defined as:

$$\text{cosine similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \tag{1.4}$$

The cosine distance is then calculated as:

$$\text{cosine distance}(\mathbf{u}, \mathbf{v}) = 1 - \text{cosine similarity}(\mathbf{u}, \mathbf{v}) \tag{1.5}$$

For dissimilarity sampling, we identify the unlabeled data points that are farthest from the already labeled data points. Specifically, we used the **maximum of the minimum distances** strategy. For each data point $x \in D_U$, the dissimilarity score is computed as the minimum distance between x and every data point in D_L:

$$d(x) = \min_{x' \in D_L} \text{cosine distance}(x, x') \tag{1.6}$$

The next data point to label, x^*, is the one with the highest minimum distance:

$$x^* = \arg\max_{x \in D_U} d(x) \tag{1.7}$$

This strategy ensures that the selected data point is as dissimilar as possible from the labeled data, encouraging exploration of diverse regions in the feature space. By selecting dissimilar points, the model can improve its understanding of underrepresented areas in the dataset and potentially reduce bias in its predictions.

2. Literature Review

This chapter presents a review of the current literature on the topic of drug BBB permeability prediction and active learning applications in related research areas. We start this chapter by showing the related research in the area of computational methods in drug BBB permeability prediction, followed by a section about two string representation formats of molecules. Then, we end this chapter with a section about the applications of active learning in drug-related research and the usage of active learning in the area of high-throughput experimentation.

2.1 Computational Approaches for Drug Blood-Brain Permeability Prediction

Table 2.1 and the mind map in figure 2.1 depict the landscape of the topic in the literature. It illustrates the key components and research areas within the topic. The mind map provides a comprehensive overview of the interconnected factors involved in drug BBB permeability prediction. The regions explained in the mind map are crucial to understanding the methodologies and tools employed in this area of research. The first branch, "Chemical Features Used" highlights the importance of molecular descriptors, such as the ECFP Morgan Fingerprints and the RDKit's descriptors, in transforming chemical structures into data that can be used for predictions. The "Methods of Handling Imbalanced Data" branch emphasizes the various techniques, such as Synthetic Minority Over-sampling (SMOTE) and class weighting, used to address the common issue of class imbalance in these datasets, ensuring that models are not biased towards one class. Another essential branch is the "Chemical String Representations", which focuses on how compounds are encoded as SMILES or SELFIES. Finally, the "Algorithms Used" branch touches on the different machine learning and deep learning approaches employed for drug

BBB permeability prediction. This includes traditional ML models as well as more advanced deep-learning techniques.

Figure 2.1 Mind map illustrating the key research areas and methodologies in drug BBB permeability prediction. The branches represent critical components such as chemical features, available datasets, methods for handling imbalanced data, chemical string representations, and algorithms used in predictive modeling.

Table 2.1 Summary of recent drug BBB permeability prediction literature

Reference	**Features Used**	**Methods Used**	**Handling Imbalance Data**
Miao et al. (2019)	Molecular Descriptors	Feed-forward NN	SMOTE
Alsenan et al. (2021)	6,394 molecular descriptors	KPCA, enhanced feed-forward ANN, and CNN.	Oversampling techniques
Wu et al. (2021)	A group contribution method that analyzed 52 molecular groups	ANN	N/A
Shaker et al. (2021)	Dragon Molecular Descriptors	LightGBM	N/A
Cherian Parakkal et al. (2022)	SMILES	MLP and CNN	N/A
Singh et al. (2023)	SELFIES-based	ANN and LSTM	N/A
Liang et al. (2024)	RDKit2D, RDKit3D, ECFP4, RDKit-ECFP4, molecular graphs (for GCN)	XGBoost, ExtraTree, SVM, AdaBoost, DNN, GCN	Generating multiple non-canonical SMILES
van Tilborg & Grisoni (2024)	ECFP	Deep active learning (graph-Based, ECFP-based models)	N/A

In a recent study, Liang et al. (Liang et al., 2024), used six different classification prediction algorithms with four different molecular feature representations to present 25 models for the prediction of BBB permeability prediction of molecules. The six algorithms are five molecular descriptor-based, and one based on GCN. For the four different molecular feature representations, they used RDKit2D, RDKit3D, ECFP4, and RDKit+ECFP4. That constitutes 24 models with each algorithm trained 4 times each time with one of the above-mentioned 4 molecular feature representations, plus one model based only on GCN with features being extracted from the molecular graph itself.

Several articles, mainly by Banerjee and Roy (Banerjee & Roy, 2024), have been

published that centered on a couple of concepts: quantitative Read-Across Structure-Activity Relationship (q-RASAR) and classification-based RASAR (c-RASAR) that are aimed at incorporating read-across with QSAR for the prediction of chemical properties. However, this trend has not received wide attention from other researchers in the field.

2.2 SMILES, Canonical SMILES, and SELFIES

Multiple text-based representation schemes have been proposed in the literature. David Weininger introduced early attempts with SMILES back in 1988 (Weininger, 1988). The following year, 1989, came with an improvement that was based on the initial idea by introducing the canonical form of SMILES, which allowed every molecule to be uniquely represented by a canonical SMILES string (Weininger et al., 1989). Recently, in 2020, a new string-based representation method was introduced, SELFIES (Krenn et al., 2020). As the paper claims, it guarantees 100% robustness.

Most available molecular datasets predominantly consist of non-canonical SMILES strings. Our analysis reveals that approaches to handling these datasets generally fall into three categories: directly using non-canonical SMILES strings as input for feature generation tools such as Mordred and RDKit, converting non-canonical SMILES to their canonical counterparts first, or, conversely, embracing the non-uniqueness of the dataset. The latter is achieved by utilizing non-canonical SMILES in a data augmentation strategy that generates multiple non-canonical SMILES strings for the same molecule. This technique enhances dataset variability and is exemplified in the studies by (Liang et al., 2024) and (Tang et al., 2022).

2.3 Active Learning in Related Research Areas

Indeed, active learning, thanks to its effectiveness, has been used in all sorts of areas, ranging from text classification (Tong & Koller, 2001) to virtual screening (Czarnecki et al., 2015). Next, we show one of the most related work to our proposed methods. Then, in other rest of this section, we show AL applications in some drug-related

research areas.

The authors in (Donmez et al., 2007) propose the DUAL algorithm, a dynamic active learning technique that combines density-weighted uncertainty sampling and standard uncertainty sampling. The algorithm adaptively switches between the strategies based on the estimated residual classification error, aiming to optimize performance across different data labeling phases. Empirical results in the paper demonstrate that DUAL outperforms static strategies on various datasets by effectively balancing density and uncertainty during the active learning process.

2.3.1 Active Learning in Drug-Related Research

This thesis focuses on predicting BBB permeability, a critical molecular property in drug discovery. To provide context, this section explores the application of AL across various stages of the drug discovery process, highlighting its utility in tasks such as compound–target interaction prediction, virtual screening, and molecular optimization.

The versatility of AL is evident across multiple drug-related research areas. In the broader domain of biological networks, Sverchkov and Craven (Sverchkov & Craven, 2017) demonstrated how AL integrates data modeling, hypothesis generation, and experimental validation to refine knowledge of complex systems. By employing strategies such as entropy-based approaches and the maximum difference criterion, they showed how AL can prioritize experiments that enhance understanding of gene regulatory and metabolic networks, as well as uncover causal relationships within these systems.

Focusing specifically on drug discovery, Reker (Reker, 2019) highlighted AL's potential to improve machine learning model performance through efficient data selection. Similarly, Ding et al. (Ding et al., 2021) illustrated the data efficiency of AL in predicting blood drug levels while maintaining robust predictive accuracy. These studies emphasize the importance of AL in handling data scarcity—a common challenge in drug-related research.

Tilborg and Grisoni (van Tilborg & Grisoni, 2024) further extended the discussion to low-data drug discovery scenarios, testing six acquisition functions in active deep learning. Their findings underscored the consistent superiority of models based on ECFP over graph-based models, demonstrating AL's effectiveness in leveraging limited data environments.

Finally, Wang et al. (Wang et al., 2024) reviewed the application of AL across various phases of the drug discovery pipeline, showing how it can help overcome data limitations and optimize workflows from early-stage screening to late-stage validation.

- **Compounds-Target Interaction Prediction:** AL helps address challenges in compound-target interaction (CTI) prediction by creating balanced training datasets. Reker et al. improved CTI models by iteratively selecting uncertain molecule-target pairs, enriching training data, while Naik et al. used AL to predict condition-target phenotypes efficiently with minimal experimental data.

- **Virtual Screening:** AL combined with ML enhances virtual screening efficiency. Cao et al. integrated pre-trained molecular representations to screen ultra-large libraries, identifying 60% of top compounds by sampling only 0.6% of the data. Similarly, Zhou et al. used the OpenVS platform to screen billions of compounds, achieving high hit rates in fewer iterations, showcasing scalability and efficiency.

- **Molecular Generation and Optimization:** AL improves molecular generation and optimization by guiding generative models and refining molecule selection. Iovanac et al. used AL with generative models to produce better molecules iteratively, while Bengio et al. integrated AL into GFlowNet to explore chemical space and enhance molecular diversity and binding affinity predictions.

- **Synergistic Drug Discovery:** AL accelerates synergistic drug discovery by navigating combinatorial drug spaces efficiently. Wang et al. incorporated cellular features like gene expression to boost prediction accuracy, identifying 60% of synergistic drug pairs by testing only 10% of the combinations. Their dynamic AL strategy balanced exploration and exploitation to improve detection and reduce experimental workload.

- **Molecular Properties Prediction:** Recent research has focused on improving molecular property prediction through AL techniques to reduce annotation costs. Tyger, a task-type-generic framework, learns a chemically-meaningful embedding space for active selection across various task types, outperforming existing methods (Zhou et al., 2022). PREVAIL, a pre-trained variational adversarial approach, selects informative initial datasets and adapts to both molecular distribution and prediction task information, demonstrating superior performance in benchmark experiments (Li et al., 2022). Another algo-

rithm combines a local model of interatomic interactions with active learning to optimize training set selection, addressing issues of large training set requirements and outlier errors in previous methods (Gubaev et al., 2018). These approaches aim to enhance the efficiency and accuracy of molecular property prediction by strategically selecting the most informative samples for annotation, potentially accelerating drug discovery processes while reducing experimental costs. Our presented work falls under this category.

2.3.2 Active Learning in High-throughput Experimentation

High-throughput Experimentation (HTE) typically involves the screening of a vast number of samples to identify optimal candidates. This process is often computationally and resource-intensive, requiring significant time and effort to generate and evaluate large datasets. Active learning offers a strategic solution by intelligently selecting experiments or samples for evaluation, thereby minimizing the number of tests needed while achieving comparable or superior outcomes.

In this context, Graff et al. (Graff et al., 2021) highlight the transformative potential of AL in optimizing HTE workflows. The study introduces a Bayesian optimization framework leveraging surrogate models—Random Forests, feed-forward neural networks, and message-passing neural networks—to effectively prioritize candidate compounds. Remarkably, the proposed approach recovers over 90% of top-performing compounds from chemical libraries containing millions of molecules while evaluating less than 3% of the total dataset. By employing acquisition strategies such as upper confidence bound and greedy metrics, the framework achieves a 40-fold reduction in the number of evaluations compared to traditional exhaustive screening methods. These findings underscore the utility of AL in significantly reducing resource demands while maintaining high accuracy, making it an invaluable tool for accelerating HTE processes across diverse scientific domains.

AL approaches have also demonstrated substantial promise in improving High-throughput Screening (HTS) efficiency across various fields. For instance, Chen et al. (Chen et al., 2020) developed a model combining categorical matrix completion with AL to guide HTS experiments evaluating chemical compound effects on protein localization. Their method emphasizes exploration over exploitation and incorporates margin sampling for uncertainty estimation, enabling more informed decision-making. Similarly, Kumar et al. (Kumar et al., 2019) proposed an AL algorithm based on Gaussian Processes for high-throughput plant phenotyping. Their

work showcases the ability to efficiently sample the most informative data points in large agricultural fields, achieving superior performance compared to exhaustive coverage methods.

Furthermore, Grave et al. (De Grave et al., 2008) introduced the concept of active k-optimization for approximating the k best instances with respect to an unknown function. They developed a Gaussian process-based algorithm to address this challenge, applying it to structure-activity relationship prediction. These studies collectively highlight the versatility and potential of AL techniques to enhance the efficiency, scalability, and precision of HTS across a wide range of scientific disciplines.

3. Methods and Experiments

In this chapter, we describe the dataset used in this thesis and the main flow of the experiments, highlighting their core stages.

We start with section 3.1 of this chapter by describing the primary dataset that is used throughout this work. Specifically, in subsection 3.1.1, we show the steps we conducted to clean the dataset and preprocess it. A preprocessing process specific to SMILES representation is explained in subsection 3.2.1. We then describe, in section 3.2, the featurization approaches that we followed to prepare the data for training the machine learning models. The two types of data splittings that are used in the experiments are shown in section 3.3. In section 3.4, we list the performance metrics that we used in evaluating the machine learning models throughout this work. In section 3.5, we mention some details about the hyper-parameters optimization process that we used to obtain the best possible parameters for the XGBoost models. In section 3.6, we show the high-level details of the passive learning models that we built and used in this thesis. Lastly, we highlight the details of the active learning models that we developed in section 3.7 showing an overview of the general idea of our proposed methods in subsections 3.7.1 and 3.7.2.

3.1 Dataset

The dataset utilized in this thesis comprises SMILES strings representing molecules, each accompanied by a binary label indicating whether the molecule is BBB permeable (classified as a positive data point, denoted as BBB+) or non-permeable (classified as a negative data point, denoted as BBB-).

The dataset, originally introduced by (Martins et al., 2012), was later included in the MoleculeNet benchmark (Wu et al., 2018). We accessed and downloaded the dataset

from the URL provided on the MoleculeNet website. Initially, the dataset consisted of 2,053 samples (molecules) represented in a non-canonical SMILES format. Table 3.1 provides detailed information about this initial version of the dataset.

Table 3.1 The original distribution of dataset used in this thesis

Total number of molecules	2053
Number of molecules with BBB+ label	1570
Number of molecules with BBB- label	483

3.1.1 Data Cleaning and Pre-processing

The initial stage of data cleaning involved verifying the validity of the SMILES notations. Using the RDKit library (Landrumet al. , 2006), we checked all the SMILES strings and found that 2,039 out of 2,053 were valid, resulting in the removal of 14 invalid SMILES strings during this initial quality control step. This process ensured a dataset containing 2,039 valid SMILES strings for further analysis.

After conducting preliminary experiments with this version of the dataset and examining it for duplicates, we identified an issue with the SMILES representation. Specifically, the type of SMILES notations in the dataset proved unsuitable for subsequent stages, particularly feature generation. During molecular representation using Mordred molecular descriptors (Moriwaki et al., 2018), we discovered groups of data points that shared identical Mordred feature representations despite having different SMILES strings. These data points corresponded to the same molecule but were represented by different non-canonical SMILES notations.

Further investigation revealed that the SMILES strings in the MoleculeNet BBB dataset were not in a unique canonical form. To address this, we transformed each non-canonical SMILES string into its canonical form using the RDKit library. This canonicalization process inevitably reduced the dataset size, as multiple non-canonical SMILES strings representing the same molecule were consolidated into a single canonical SMILES string.

Table 3.2 summarizes the characteristics of this canonicalized version of the dataset, while Figure 3.1 outlines the data cleaning and pre-processing steps.

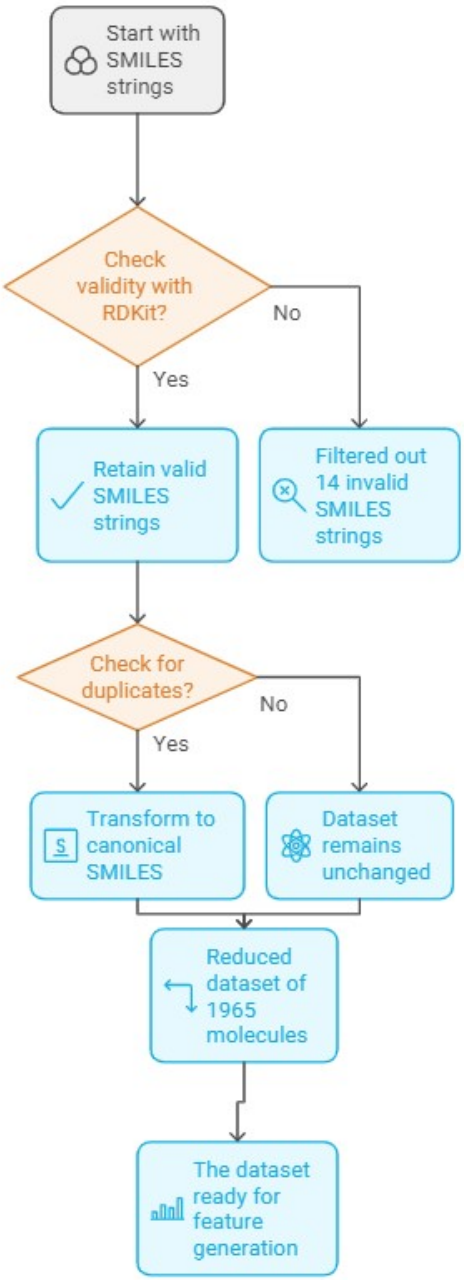


Figure 3.1 Data cleaning and pre-processing pipeline

3.2 Molecular Features and Representations

To enable machine learning models to process molecular data effectively, we needed to prepare numerical representations of the molecules. There are many methods for generating molecular representations, and in this work, we explored several ap-

proaches. Below, we describe the steps we took to preprocess the data and create features suitable for machine learning.

Initially, we represented the molecules using SMILES, a widely used notation that captures the connectivity between atoms in a molecule. However, the dataset we retrieved contained non-canonical SMILES, which posed a problem. We observed that these non-canonical representations led to duplicates, meaning the same molecule appeared multiple times in different forms. This redundancy could confuse the machine learning models and reduce the efficiency of our training process.

To resolve this issue, we converted the non-canonical SMILES into their canonical form using the RDKit library. Canonical SMILES ensures a unique representation for each molecule, removing duplicates from the dataset. After this preprocessing step, the size of our dataset was reduced from 2,053 molecules to 1,965 molecules, providing a cleaner and more reliable dataset for further analysis.

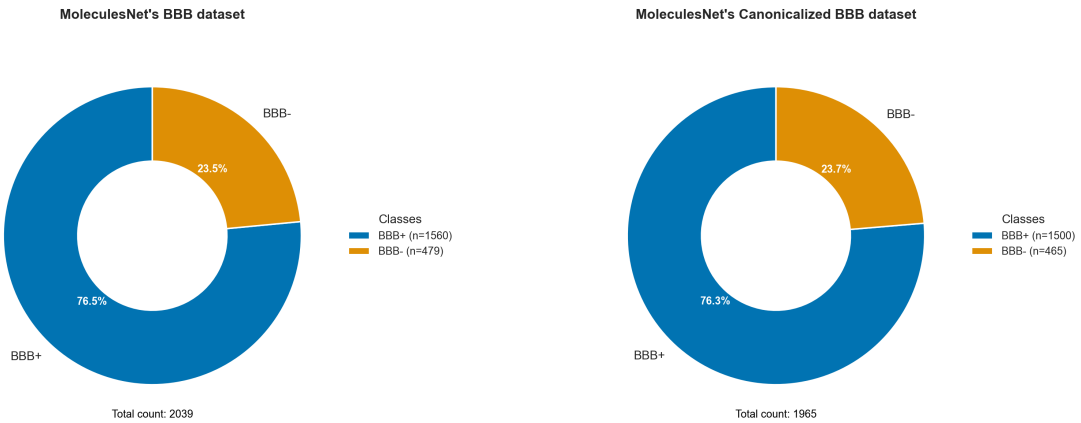
Since SMILES is a textual representation and not inherently numerical, we needed to transform it into a format that machine learning models could interpret. For this, we used ECFP, a widely accepted molecular representation in cheminformatics. Using the RDKit library, we computed ECFP fingerprints for all molecules in the dataset. These fingerprints are binary vectors that indicate the presence or absence of specific molecular substructures, making them particularly useful for capturing chemical information in a way that is amenable to machine learning.

In addition to SMILES, we explored using SELFIES as an alternative molecular representation. SELFIES has recently gained attention as a robust method for encoding molecular structures. Unlike SMILES, SELFIES is designed to be error-resistant and offers greater flexibility while retaining the essential benefits of SMILES. By incorporating SELFIES, we aimed to investigate its potential as a more advanced molecular representation in our experiments.

3.2.1 SMILES Strings Canonicalization

To prepare the dataset for machine learning, we performed SMILES string canonicalization, which converts non-canonical SMILES strings into their unique, canonical representations (Weininger et al., 1989). This step was critical for ensuring that each chemical compound in the dataset was represented uniquely, avoiding redundancy that could arise from multiple representations of the same molecule (Deng et al., 2023).

In our work, we applied this canonicalization process to all SMILES strings in the dataset before feature generation. By doing so, we ensured that the dataset was consistent and free from duplicates, a necessary condition for effective machine learning workflows. The Python library RDKit (Landrumet al. , 2006) was used for this purpose, as it provides a reliable implementation of SMILES canonicalization. Interestingly, while canonicalization reduced the total number of molecules in the dataset, it did not alter the class distribution significantly. Figure 3.2 illustrates the class distribution before and after canonicalization, highlighting that the ratio between classes remained consistent. This result confirmed that the canonicalization process preserved the dataset's overall balance, maintaining its suitability for machine learning tasks.



(a) The non-canonicalized dataset (b) The canonicalized dataset

Figure 3.2 Class distribution in BBB MoleculeNet dataset before and after SMILES canonicalization.

Table 3.2 The dataset used in this thesis **after** performing SMILES canonicalization

	MoleculeNet's BBB
Total number of molecules	1965
Number of molecules with BBB+ label	1500
Number of molecules with BBB- label	465

3.3 Dataset Splitting

When deciding to train a machine learning model on a data set, that data set must be split into at least two subsets. That is, training and testing. The training set can be further split into training and validation. The way data split is performed has been shown to have a significant impact on model performance (Birba, 2020). The well-known and most straightforward splitting strategy is random splitting, and an enhanced version of it is stratified sampling, where you stratify based on a given column in the dataset to ensure an almost equal ratio of a certain feature in all the splits to avoid sampling bias. Often, the stratification is done on the class label column, especially when the data set is imbalanced, which is the case with the dataset that we worked with; thus, stratification based on the label was the first strategy that we followed to split the data.

3.3.1 Label-Stratified Splitting

In this splitting strategy, we split the data according to the class label (BBB+ or BBB-) to ensure that each split (training, validation, or testing) has almost the same positive (BBB+) to negative (BBB-) sample ratio.

3.3.2 Scaffold-based Splitting

When it comes to molecular datasets, some special characteristics of molecules make the well-known random splitting strategy not the best splitter ever. That is because random splitting—in this case—does not reflect the real-world scenario where the model is expected to be tested on molecules with totally different structures than those in the training set. For the reasons mentioned above and more, scaffold-based splitting is believed to be preferred as a splitting strategy for molecular datasets (Deng et al., 2023).

Naturally, molecular scaffolds can be thought of as an intuitive method to group chemical compounds based on their structures. A molecular scaffold represents the core structure of a chemical compound. Many scaffold schemes have been proposed

in the literature; one of the widely used ones, especially in the computational chemistry realm, is known as Bemis and Murcko (BM) scaffolds (Bemis & Murcko, 1996). In scaffold splitting, molecules are grouped according to their core scaffold (structure), and then each set of molecules in the same scaffold group (i.e., sharing the same core scaffold) is assigned to a certain split. This makes scaffold-based splits a more challenging and realistic scenario because molecules in the test set have unseen scaffolds during the model training cycle.

3.4 Metrics

In this section, we mention the metrics that we used to evaluate individual machine learning models that we built and the metrics we used during the comparison phase among them.

3.4.1 Machine Learning Models Performance Metrics

As the dataset with which we worked is imbalanced, relying on simple metrics such as accuracy will be misleading. Instead, metrics like ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) and Average Precision (AP) are commonly used to measure better the performance of machine learning models trained on imbalanced datasets.

The AP score is particularly effective for imbalanced datasets, providing a single scalar value summarizing the precision-recall curve. It calculates a weighted average of precision values at different recall thresholds, with the weights determined by the change in recall compared to the previous threshold. This captures how well the model balances precision and recall across various decision boundaries, making it a robust metric for this study (Equation 3.1).

$$\text{AP} = \sum_{n} (R_n - R_{n-1}) P_n \tag{3.1}$$

Where:

- P_n is the precision at the n-th threshold.

- R_n is the recall at the n-th threshold.

Additionally, as suggested by a recent systematic study (Deng et al., 2023), metrics such as Negative Predictive Value (NPV) (Equation 3.2) and Positive Predictive Value (PPV) (Equation 3.3) are included for a more comprehensive evaluation. These metrics offer additional insights into the model's ability to identify true negatives and true positives, respectively,

$$\text{NPV} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Negatives}} \tag{3.2}$$

$$\text{PPV} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3.3}$$

For the purpose of comparing the results in the next chapter, we relied on AP as the primary metric due to its effectiveness in handling imbalanced data. Unlike ROC-AUC, which measures overall classifier performance and can sometimes mask poor performance on specific classes due to class imbalance, AP focuses on the model's ability to identify positive instances (BBB+ in our case). AP is useful for evaluating performance in imbalanced datasets because it captures the precision-recall trade-off across thresholds. While other metrics, including Accuracy, Recall (TPR), F1 Score, Specificity (TNR), False Positive Rate (FPR), and False Negative Rate (FNR), are also reported later for completeness, AP remains the focus here due to its particular relevance for imbalanced datasets.

3.4.2 Metrics for Performance Comparison

After conducting the experiments and calculating the performance metrics—where AP was selected as the primary metric as detailed in Section 3.4.1 each experiment was repeated 20 times with different random seeds to ensure robustness. This included evaluations of both passive learning models and all active learning strategies under stratified and scaffold-splitting setups. The results from these 20 repetitions were used to construct Win/Tie/Loss (W/T/L) tables, enabling pairwise comparisons of model performance. The results are presented in tabular form, where each

entry contains the count of wins, ties, and losses for one strategy compared to another. These counts are taken across all experimental repetitions and specific points (25%, 50%, 75%, and 100%). These percentages (i.e., 25%, 50%, 75%, and 100%) refer to the proportion of labeled training data used at different stages of the active learning process.

Types of W/T/L Tables:

- **Passive vs. Active Learning:** Passive learning models (stratified and scaffold splits) were compared against all active learning strategies.
- **Active Learning Pairwise:** Pairwise comparisons were conducted among all active learning strategies.
- **Binary Tables:** Simplified versions of the tables as they condense the comparisons to show the majority winner in each case.

3.5 Hyper-parameters Optimization

Bayesian hyperparameter optimization using the Hyperopt python library (Bergstra et al., 2015) (with the tree-structured Parzen Estimator algorithm) has been used to obtain the best hyperparameters for the passive learning-based models. The decision to opt for Hyperopt rather than resorting to random search or grid search is based on the fact that Hyperopt would result in better performance, be more efficient, and less time-consuming, which stems from its theoretical basis of how it works, and also has been supported by empirical studies, specifically for XGBoost (Putatunda & Rama, 2018). The initial hyperparameters and their range of values were adopted from (Boldini et al., 2023) and adjusted after many trials to achieve better performance tailored to the data set specifications.

3.6 Passive Learning Models

In this thesis, we define passive learning models as XGBoost machine learning models that are trained in a single step, where the entire training dataset is provided to the model at once without employing any special data sampling strategy. These models will serve as a baseline for comparison against models using active learning techniques.

To represent the molecular data, we utilize two distinct feature encoding methods. The first is ECFP based on Morgan fingerprints, a widely adopted approach for capturing molecular substructures through circular fingerprinting. The second encoding method involves embeddings generated by the SELFormer model, which was pre-trained on SELFIES, a robust representation of chemical compounds (Yüksel et al., 2023). By using both traditional and deep learning-based molecular representations, we aim to assess their respective contributions to model performance in a passive learning training setup.

3.7 Active Learning Models

In this work, we employed a variety of active learning paradigms with different sampling strategies. We experimented with random sampling, uncertainty-based sampling, dissimilarity-based sampling, and, finally, our proposed methods.

The subsection below explains the proposed method and its two versions. But before diving deep into the details of the sampling strategies, we should mention the configuration of the experiments that are common to all the active learning strategies.

To mimic the real-world scenario where the active learning training paradigms are usually applied, we start the training process with very tiny labeled samples of the dataset. In this work, we start each active learning training loop with an initial training set of size 4 samples: 2 samples are randomly selected from the BBB+ class and 2 samples are randomly selected from the BBB- class. We aimed at starting with an equal number of BBB+ and BBB- samples, even if it means oversampling the minority class because this will ensure that the model does not become biased

early on. Then, for each iteration, we select a batch of samples size 5. The selection of the samples is determined by the sampling strategy employed by the running active learning strategy. Table 3.3 shows the common key parameters for the active learning setup across all the active learning strategies that we employed in this work.

Table 3.3 Key parameters for the active learning setup

Parameter	**Value**
Initial Training Set	4 samples (2 BBB+ and 2 BBB-)
Batch Size per Iteration	5 samples

3.7.1 First Proposed Method: Explore-Intensify

The strategy we propose differs from the existing ones in that it employs variable mechanism(s) during the active learning loop. Contrary to previous approaches, this strategy does not have one fixed sampling strategy from the start of the active learning loop until the end of it. This approach allows flexibility during the active learning loop.

The approach starts the sampling process by following an exploratory fashion and focusing primarily on exploring the chemical space. More precisely, the sub-space of the chemical space that the dataset in hand spans. At the beginning of the active learning loop, the active learner will select data points to be labeled solely based on the diversity factor. Then, at a later stage the active learner will switch to considering the areas in the chemical space where the model is most uncertain about, i.e., the active learner will rely on uncertainty. This approach, which we call *explore first, intensify later*, introduces a parameter that determines the transition point or the balance between exploration and intensification, enabling a variety of strategy configurations.

Thus, our proposed method introduces a dynamic sampling strategy throughout the entire active learning loop, as opposed to a static, single-strategy approach from start to finish.

This dynamic switching is in the form of an "explore-intensify" approach. This approach splits the sampling process into two distinct phases. In the first initial phase, which constitutes the first X% of the dataset, the focus is on employing a diversity sampling strategy. This strategy aims to enhance the representativeness of the initial training set by selecting a broad group of diverse samples. Following

this exploratory phase, the method transitions to intensification, which is the second phase, wherein the uncertainty sampling strategy is employed. This phase focuses on refining the model's decision boundaries by prioritizing samples where the model shows the highest uncertainty.

3.7.2 Second Proposed Method: Round Robin Cycle Switching

In this strategy, we implement a dynamic sampling approach where different sampling strategies are alternated during the active learning iterations using a round-robin scheduling mechanism. Specifically, we cycle through dissimilarity sampling, uncertainty sampling, and random sampling in a sequential order. This approach aims to leverage the strengths of each sampling method: dissimilarity for diversity, uncertainty for model refinement, and random sampling for exploration.

To ensure a systematic and balanced data exploration, the switching point is set at every 250 data points. This means that after every 250 samples are added to the training set, the strategy switches to the next in the cycle. The 250 data points value represents a key parameter of the strategy, which can be adjusted based on the dataset size, problem domain, or desired level of balance between the sampling strategies. For instance, a smaller switching point might lead to more frequent alternation between strategies, while a larger one allows for a longer focus on each method before switching.

By combining the complementary strengths of the three sampling strategies through this structured cycle, the *Round Robin Cycle Switching* approach seeks to achieve both broad exploration and focused refinement during the active learning loop. This method, along with the *Explore-Intensify* strategy, is evaluated to assess the efficacy of passive and active learning approaches under different data-splitting strategies. The subsequent chapter presents the results and discusses their implications.

4. Results and Discussion

This chapter presents the key findings from the dataset analysis, as well as the results obtained from training various XGBoost-based models under both passive and active learning frameworks. The active learning models were assessed using five distinct sampling methods, providing a comprehensive evaluation of their performance. A detailed comparison is then presented, contrasting the results of these active learning methods with those of the passive learning models across two data splitting setups: label-stratified splitting and scaffold-based splitting. Finally, the chapter concludes with a discussion of the observed results, focusing on their relevance to model performance and the effectiveness of the implemented learning strategies.

4.1 Challenges of Non-Canonical SMILES Representations in the MoleculeNet BBB Dataset

Many open-source molecular datasets, including those in SMILES format, are often represented as non-canonical SMILES strings. These non-canonical representations can introduce several challenges to machine learning models when used without standardization into canonical SMILES. One major issue is duplication or redundancy. Multiple non-canonical SMILES strings can correspond to a single unique molecular compound, which is typically represented by its canonical SMILES form. This redundancy effectively results in the same data point being represented multiple times within the dataset, leading to known challenges in machine learning training processes, such as biased training or overfitting. An even more critical problem arises when non-canonical SMILES strings corresponding to the same canonical SMILES are assigned contradictory labels. Such inconsistencies introduce confusion to the model, as it encounters what appears to be a single data point with conflicting labels. This inconsistency can significantly degrade model performance by impair-

ing its ability to learn meaningful patterns. Our analysis of the MoleculeNet BBB dataset (Wu et al., 2018) revealed the presence of such cases. A representative example is illustrated in figure 4.1, while nine additional similar cases are detailed in section A.2 of the Appendix.

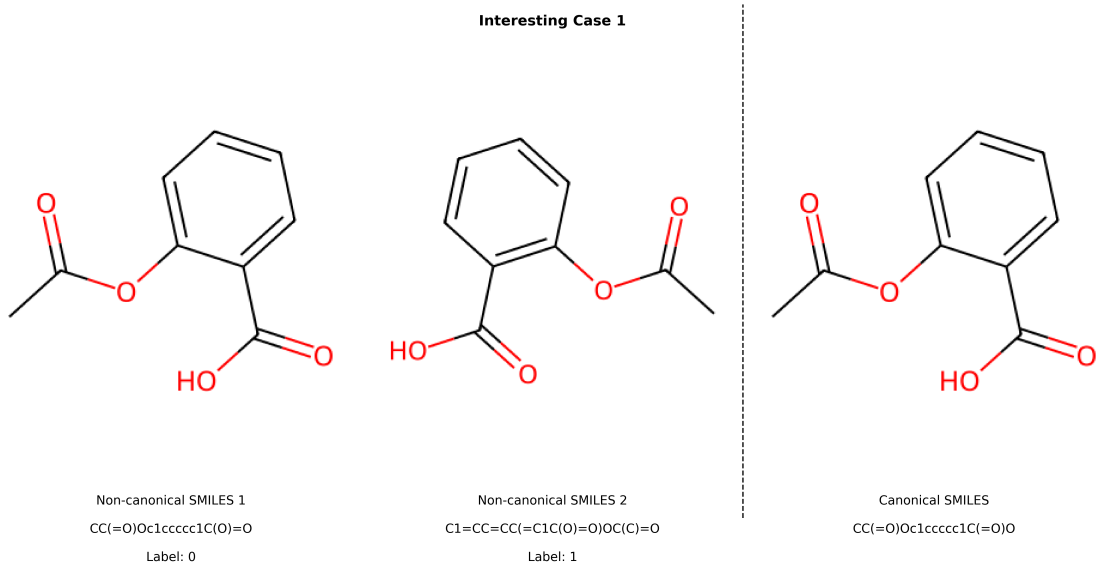


Figure 4.1 An example of two non-canonical SMILES strings with differing labels mapping to the same canonical SMILES string.

4.2 Molecular Scaffold Analysis

To understand the structural characteristics of the molecules in the BBB MoleculeNet dataset, we performed a comprehensive scaffold analysis on the BBB dataset. This analysis focused on the molecular scaffolds that exist within the molecules, allowing us to gain deeper insights into their structural characteristics. Molecular scaffolds, which represent the core structures of chemical compounds, play a crucial role in determining their physicochemical properties and, consequently, their ability to permeate the BBB. We utilized the BM scaffold generation method using RDKit to extract the molecular scaffolds from each compound in the dataset. This method removes all side chain atoms, leaving only the core rings and linker atoms between them.

During the analysis of scaffold groupings in MoleculeNet's BBB dataset, we observed that certain scaffold groups were highly enriched in BBB+ compounds. This enrichment highlights structural features that may contribute significantly to BBB

permeability. Figure 4.2 illustrates an example of such scaffolds. Additional examples and a more detailed discussion are provided in Section A.3 of the Appendix.

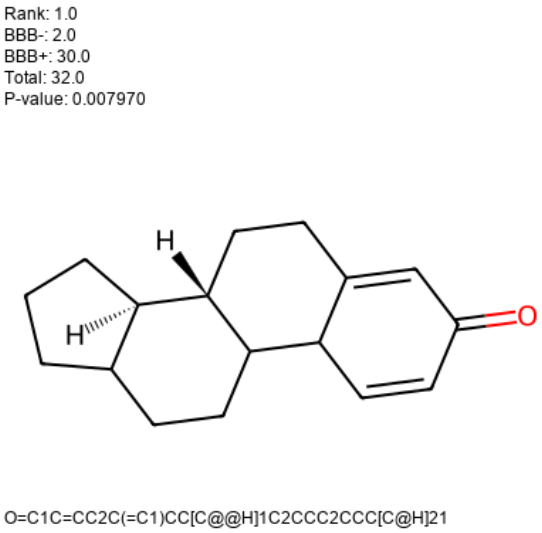


Figure 4.2 An example of a molecular scaffold where its group is highly enriched with BBB+ compounds

4.3 Dataset Splitting

All experiments in this thesis were conducted using two parallel dataset splitting setups:

- A setup based on a **stratified** splitting strategy, which ensures that the ratio of the labels (BBB+/BBB-) remains consistent across the training, validation, and test sets.
- A setup based on the molecular **scaffolds** of the molecules in the dataset, where the splitting is guided by the structural cores (scaffolds) of the molecules.

In both splitting mechanisms, we adopted an 80/10/10 scheme. Specifically, 80% of the data was allocated for training, 10% for validation, and 10% for testing. This ensures that the training set is representative of the entire dataset while leaving sufficient data for unbiased validation and testing.

It is important to note that, for the active learning strategies implemented in this

work, the reference to 100% of the training data corresponds to the 80% training split of the original dataset. Thus, all active learning iterations operate within this 80% subset, progressively utilizing the available labeled training data.

4.3.1 Label-Stratified Splitting

Table 4.1 presents the distribution of the data and the BBB+:BBB- ratios across the training, validation, and testing splits when using the stratified splitting strategy. Since stratified splitting ensures that the ratio of the target classes (BBB+ and BBB-) remains consistent across all subsets, the class proportions in the training, validation, and testing splits are nearly identical to those in the entire dataset.

Table 4.1 Data distribution across splits using **stratified** splitting strategy over the class label (BBB+ or BBB-)

	Training	**Validation**	**Testing**	**All dataset**
BBB+	1200 (76.38 %)	150 (76.14 %)	150 (76.14 %)	1500 (76.34 %)
BBB-	371 (23.62 %)	47 (23.86 %)	47 (23.86 %)	465 (23.66 %)
	1571	197	197	1965

4.3.2 Scaffold-based Splitting

Table 4.2 presents the distribution of the data and the BBB+:BBB- ratios across the training, validation, and testing splits when using the scaffold splitting strategy. Unlike stratified splitting, scaffold-based splitting groups molecules based on their Bemis-Murcko scaffolds—the core structures of the molecules—rather than directly balancing class proportions. Consequently, the BBB+:BBB- ratios in each split are not guaranteed to match the overall dataset ratio.

As shown in the table, the class ratios in the training and validation splits closely align with the overall dataset distribution. However, the test split exhibits a deviation of approximately 10% from the dataset's overall BBB+:BBB- ratio. This deviation reflects the inherent variability of scaffold splitting, which prioritizes structural diversity over label balance. Despite this, the scaffold-based splitting setup provides a more realistic evaluation scenario, as it mimics real-world challenges where test data often contains novel molecular scaffolds unseen during training. Furthermore,

the number of unique molecular scaffolds in each split provides additional insight into the structural diversity introduced by the scaffold splitting strategy. Specifically, the scaffold splitting resulted in 855 unique scaffolds in the training set, 136 in the validation set, and 166 in the testing set.

Table 4.2 Data distribution across splits using **scaffold** splitting strategy

	Training	**Validation**	**Testing**	**All dataset**
BBB+	1219 (77.54 %)	149 (76.02 %)	132 (67.01 %)	1500 (76.34 %)
BBB-	353 (22.46 %)	47 (23.98 %)	65 (32.99 %)	465 (23.66 %)
	1572	196	197	1965

4.4 Passive Learning-based Models

This section details the two passive learning XGBoost models trained on all the training data. We have employed two main setups: label-stratified splitting setup and scaffold-based splitting setup . For each splitting setup, we used two different setups as well: one using the XGBoost model utilizing ECFP and the other using the XGBoost model utilizing SELFIES's embeddings. To ensure the robustness and fairness of our results, we conducted multiple experiments, each repeated twenty times with different random seeds. This approach allowed us to account for variability and avoid biases arising from lucky splits or easy data. The use of multiple random seeds also enabled us to report the average performance and standard deviation, providing a comprehensive understanding of the models' behavior.

Next, we show the results of the passive learning models for these setups. In tables 4.3 and 4.4, bold values indicate better performance for each metric. Metrics where a higher value is better include ROC AUC, Average Precision, Accuracy, Precision, Recall, F1 Score, NPV, and Specificity. False Positive Rate (FPR) and False Negative Rate (FNR) are metrics where lower values are better, with standard deviations ($\pm$) also reported.

Table 4.3 Performance comparison between ECFP and SELFIES' embeddings on the **label-stratified** split.

Label-Stratified Split		
Metric	**ECFP**	**SELFIES' embeddings**
ROC AUC	**0.8922** ± 0.0290	0.8315 ± 0.0290
Average Precision	**0.9561** ± 0.0144	0.9257 ± 0.0176
Accuracy	**0.8574** ± 0.0229	0.8157 ± 0.0325
Precision (PPV)	**0.9138** ± 0.0198	0.8834 ± 0.0196
Recall (TPR)	**0.8977** ± 0.0194	0.8740 ± 0.0457
F1 Score	**0.9055** ± 0.0152	0.8779 ± 0.0242
NPV	**0.6914** ± 0.0464	0.6217 ± 0.0795
Specificity (TNR)	**0.7287** ± 0.0672	0.6298 ± 0.0768
FPR	**0.2713** ± 0.0672	0.3702 ± 0.0768
FNR	**0.1023** ± 0.0194	0.1260 ± 0.0457

In the label-stratified split case (check table 4.3), ECFP outperformed SELFIES' embeddings across all metrics, including ROC AUC, Average Precision, Accuracy, Precision, Recall, F1 Score, and Specificity. ECFP also had lower FPR and FNR, indicating more consistent and reliable performance.

Table 4.4 Performance comparison between ECFP and SELFIES' embeddings on the **scaffold-based** split.

Scaffold-Based Split		
Metric	**ECFP**	**SELFIES' embeddings**
ROC AUC	**0.8759** ± 0.0138	0.7977 ± 0.0107
Average Precision	**0.9161** ± 0.0116	0.8477 ± 0.0210
Accuracy	**0.8381** ± 0.0248	0.7954 ± 0.0153
Precision (PPV)	**0.8722** ± 0.0156	0.8406 ± 0.0098
Recall (TPR)	**0.8890** ± 0.0391	0.8576 ± 0.0303
F1 Score	**0.8800** ± 0.0203	0.8487 ± 0.0139
NPV	**0.7708** ± 0.0617	0.7013 ± 0.0386
Specificity (TNR)	**0.7346** ± 0.0401	0.6692 ± 0.0306
FPR	**0.2654** ± 0.0401	0.3308 ± 0.0306
FNR	**0.1110** ± 0.0391	0.1424 ± 0.0303

ECFP consistently outperformed SELFIES' embeddings in the scaffold-based split across all metrics, including ROC AUC, Average Precision, Accuracy, Precision, Recall, F1 Score, NPV, and Specificity. ECFP also achieved lower values for the

FPR and FNR, further emphasizing its robustness in this more challenging and realistic evaluation scenario (check table 4.4).

Our experimental results (as can be observed from table 4.3 and 4.4) consistently demonstrated that the ECFP-based representation outperformed the SELFIES-based representation across all metrics. This consistent outperformance of ECFP was a key factor in our decision to continue using ECFP as the primary molecular feature representation in our work.

An interesting observation from our experiments' results is that models trained using ECFP consistently outperform those utilizing SELFIES embeddings. We hypothesize that this performance difference is due to the nature of ECFP as a feature representation method. ECFP, being a structural fingerprint, focuses heavily on the core chemical structure of molecules, capturing crucial details about molecular fragments and their connectivity. This structural information encoded in the ECFP bit vector appears to be particularly advantageous in the context of scaffold splitting, where the division of molecules based on their underlying scaffolds is crucial for generalization. Hence, ECFP provides more discriminative power in distinguishing scaffold groups, leading to better model performance in these cases.

4.5 Active Learning-based Models

Here, we show the active learning strategies applied for the label-stratified and the scaffold-splitted setups.

In this work, we experimented with the following sampling (querying) strategies:

- Random Sampling
- Uncertainty Sampling
- Dissimilarity Sampling
- Proposed Scheduled Strategies

4.5.1 Random Sampling

We implemented random sampling as a baseline strategy in our active learning framework. This approach is a crucial benchmark against which we can compare more sophisticated sampling methods. In our implementation, we employed a straightforward random selection process. At each iteration of the active learning loop, we randomly chose -without replacement- a batch of molecules from the unlabeled pool. This batch is labeled and then added to the previously labeled training set, and then the model is re-trained on the updated version of the training set. Figure 4.3 shows an overview of the random sampling active learning strategy. In the figure, k represents the batch size, when k is 1, only one molecule is randomly selected at each iteration in the active learning loop.

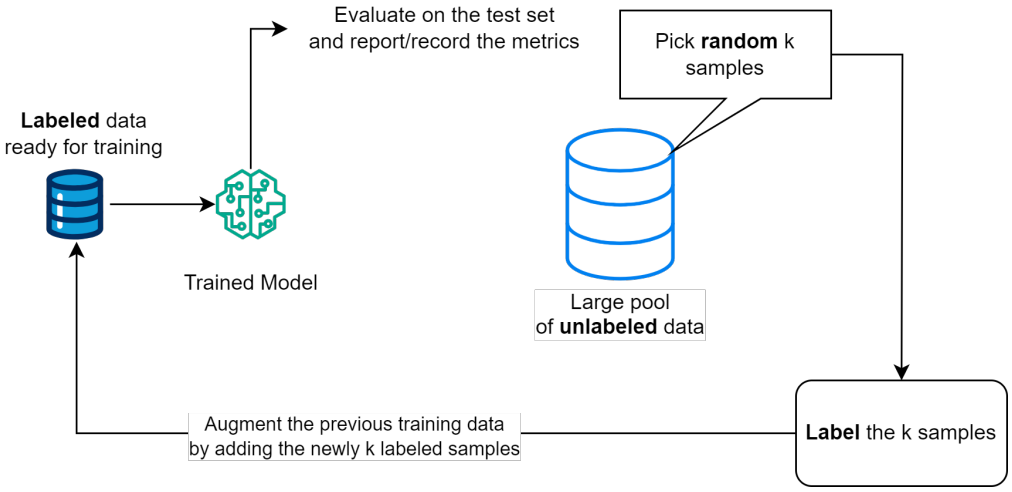


Figure 4.3 Overview of the random sampling active learning strategy

4.5.2 Uncertainty Sampling

We implemented a sampling strategy based on the model's uncertainty. This strategy aims to identify and prioritize the labeling of molecules about which the current model is most uncertain. In active learning, uncertainty is used as a signal to focus the model's attention on data points where it is least confident, thereby improving its performance more efficiently. We quantified the uncertainty of the model's predictions for each unlabeled molecule by measuring the entropy of the prediction probabilities. Entropy is a measure of uncertainty that indicates how spread out the predicted probabilities are.

At each iteration of the active learning loop, we selected the k most uncertain samples by ranking the unlabeled molecules based on their entropy scores in descending

order. These samples were labeled and added to the training dataset, augmenting it with data points that are likely to improve the model's performance. The process of uncertainty sampling is illustrated in Figure 4.4. The diagram demonstrates the workflow, starting from the model's predictions on the unlabeled pool, computing prediction uncertainty, and selecting the top k most uncertain samples for labeling. This iterative approach enables the model to learn more effectively from challenging examples.

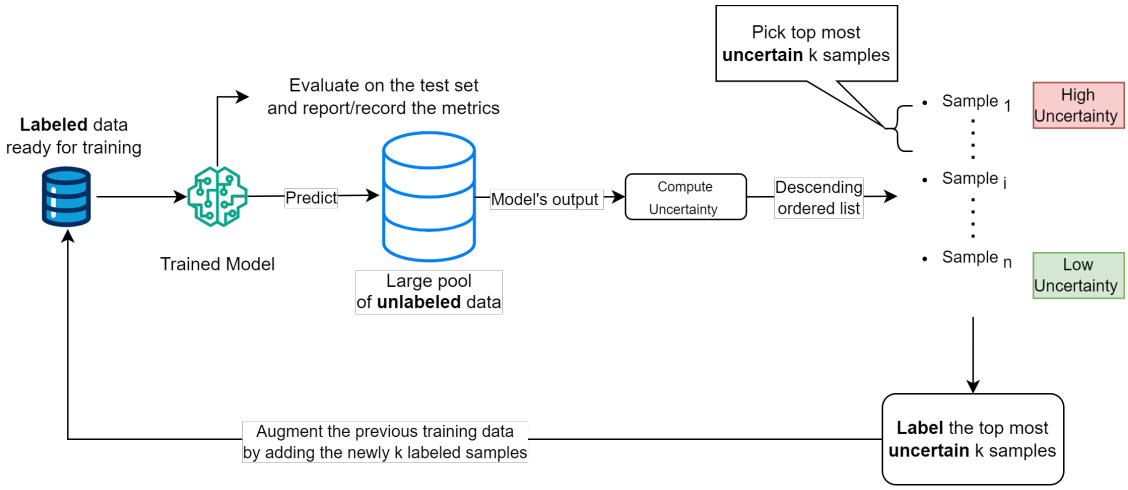


Figure 4.4 Workflow of the uncertainty sampling strategy in active learning. The model predicts outputs for the unlabeled pool, calculates prediction uncertainty (e.g., using entropy), selects the k most uncertain samples, and augments the training dataset with their labeled counterparts.

4.5.3 Dissimilarity Sampling

In this sampling strategy, we implemented a dissimilarity-based approach to select the potentially most informative molecules for labeling. The method aimed to explore the chemical space efficiently by prioritizing compounds structurally different from those already in the labeled set. We calculated the dissimilarity between molecules using the cosine distance metric applied to their ECFP fingerprints. This metric was chosen for its effectiveness in capturing structural differences between molecules represented as high-dimensional vectors. At each iteration of the active learning loop, we employed a greedy batch selection process. For each molecule in the unlabeled pool, we computed its minimum cosine distance to the set of labeled molecules. We then selected the molecule with the maximum of these minimum distances, choosing the compound most dissimilar to any in the labeled set.

Mathematically, for each unlabeled molecule u in the pool, we computed its dissimilarity score $D(u)$ as follows:

$$D(u) = \max_{u \in U} \min_{l \in L} (1 - \cos(u, l))$$

where U is the set of unlabeled molecules, L is the set of labeled molecules, and $\cos(u, l)$ is the cosine similarity between molecules u and l.

Our results showed that dissimilarity sampling performed well in the early stages of the active learning process. This suggests that the dissimilarity-based strategy effectively identified diverse and informative molecular scaffolds. However, we observed that the performance gains from dissimilarity sampling tended to plateau in later iterations. This may be because, as the labeled set grows, it becomes increasingly difficult to find highly dissimilar molecules. At this stage, other sampling strategies, such as uncertainty sampling, began to show comparative advantages.

The process of dissimilarity sampling is illustrated in Figure 4.5. The diagram demonstrates the steps involved, starting from the pool of labeled and unlabeled molecules, through the computation of dissimilarity scores using cosine distance, to the selection of the most dissimilar molecules for labeling. This iterative procedure effectively expands the training dataset with structurally diverse compounds.

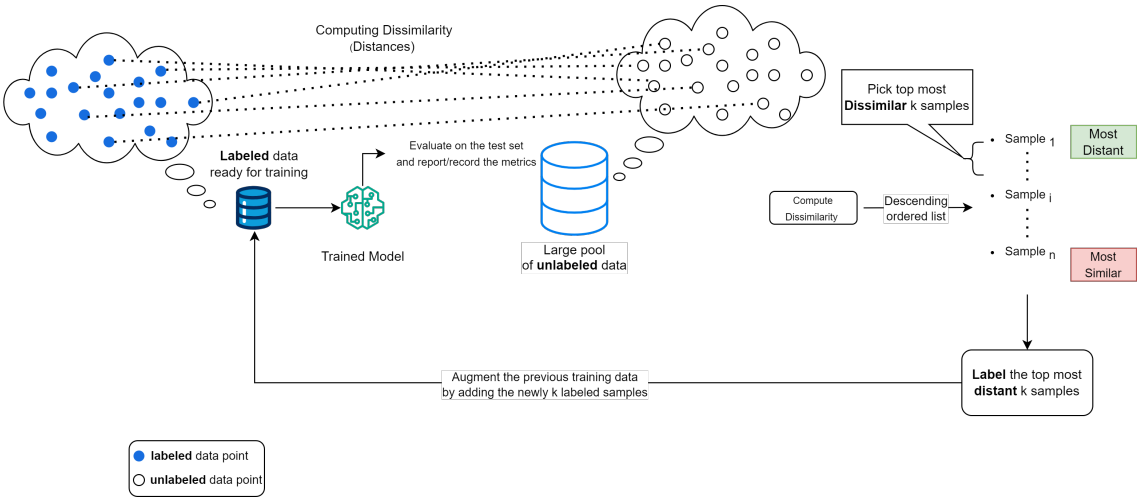


Figure 4.5 Workflow of the dissimilarity sampling strategy in active learning. The process involves computing the cosine distance between unlabeled and labeled molecules based on ECFP fingerprints, selecting the most dissimilar molecules, and augmenting the labeled dataset with these newly labeled samples.

4.5.4 Scheduled Strategy

In this strategy, a combination of sampling strategies is used within the active learning loop, with transitions between them scheduled dynamically. This approach aims to balance the benefits of exploration, uncertainty-based refinement, and randomness in a structured manner.

The first mode is the Explore-Intensify strategy. This begins with an exploratory phase, where diversity sampling is employed to maximize the representativeness of the initial training set by selecting a wide range of chemically diverse molecules. This phase allows the active learner to explore the chemical sub-space spanned by the dataset comprehensively. After this initial exploratory phase, which spans a predefined proportion of the dataset, the strategy transitions to the intensification phase. In this phase, uncertainty sampling is prioritized to refine the model by focusing on regions in the chemical space where the model exhibits the highest uncertainty. This phased approach dynamically adjusts the sampling strategy based on the progress of the active learning loop, ensuring both broad exploration and targeted refinement.

The second mode is Round-Robin Scheduling, where we alternate between a list of sampling strategies in a cyclical manner. Specifically, strategies such as dissimilarity sampling, uncertainty sampling, and random sampling are applied sequentially. Once the list of strategies is exhausted, the process returns to the first strategy in the list, continuing the cycle. The alternation can be configured in several ways, with one straightforward method being to switch strategies after a fixed number of iterations. For instance, we might set the switching point at every 250 samples added to the training set. This cyclical switching ensures that each strategy contributes to the active learning process, leveraging the strengths of diversity, model uncertainty, and randomness in a balanced way.

The scheduled strategy provides two distinct modes, each offering a flexible framework for active learning. The Explore-Intensify mode is designed to transition from an initial exploratory phase, where diversity sampling ensures broad coverage of the chemical space, to a refinement phase, where uncertainty sampling focuses on regions of high model uncertainty. In contrast, the Round-Robin Scheduling mode alternates between multiple sampling strategies in a cyclical manner, ensuring balanced contributions from dissimilarity sampling, uncertainty sampling, and random sampling.

4.6 Comparisons of Active Learning and Passive Learning Models

This section presents the experimental results obtained from two primary setups: stratified splitting and scaffold splitting. Within each setup, we evaluated the performance of XGBoost models trained using two paradigms: passive learning and active learning.

For passive learning, we trained a single XGBoost model in a one-shot manner on a fixed static training dataset without iterative interaction. In contrast, for active learning, we trained multiple XGBoost models iteratively, utilizing the five active learning strategies outlined in the previous section.

As previously mentioned, scaffold splitting presents a significant challenge for ML models due to the increased structural diversity in the test set compared to the training set. This challenge is reflected in the generally lower performance of ML models under the scaffold splitting setup compared to the stratified splitting setup.

To ensure robustness in our comparisons, we conducted each experiment 20 times using different random seeds for both the passive learning model and all active learning methods. From these repeated experiments, we constructed win/tie/loss (w/t/l) tables to compare the performance of each active learning method against the passive learning model, as well as against the other active learning methods.

We begin by presenting the results of the comparisons between the active learning methods and the passive learning model. Then, we discuss the results of the best-performing active learning method in the label-stratified setup, and we conclude this chapter by describing the results of the rest of the active learning methods, which their results are included in the appendix.

Table 4.5 presents the win/tie/loss (w/t/l) results for all active learning methods compared to the passive learning model under the label-stratified splitting setup. To simplify the comparison, Table 4.6 provides a binary conversion of Table 4.5, where each cell is assigned a value of one if the corresponding active learning method outperformed the passive learning model in more than 10 out of the 20 randomized experiments and zero otherwise.

Table 4.5 Win/Tie/Loss counts for split strategy **"stratified"** against **"passive learning"** baseline at specified percentages of labeled training data

Sampling Strategy	**25%**	**50%**	**75%**	**100%**
dissimilarity	1/0/19	10/0/10	13/0/7	18/0/2
uncertainty	3/0/17	10/0/10	15/0/5	16/0/4
random	4/0/16	10/0/10	15/0/5	17/0/3
rr_cycle_switching_50	0/0/20	8/0/12	15/0/5	19/0/1
explore_intensify_0.1	2/0/18	8/0/12	13/0/7	15/0/5
explore_intensify_0.2	5/0/15	9/0/11	15/0/5	17/0/3
explore_intensify_0.3	1/0/19	10/0/10	14/0/6	15/0/5
explore_intensify_0.4	1/0/19	11/0/9	13/0/7	16/0/4
explore_intensify_0.5	1/0/19	10/0/10	12/0/8	16/0/4
explore_intensify_0.6	1/0/19	10/0/10	15/0/5	15/0/5
explore_intensify_0.7	1/0/19	10/0/10	15/0/5	15/0/5
explore_intensify_0.8	1/0/19	10/0/10	13/0/7	15/0/5
explore_intensify_0.9	1/0/19	10/0/10	13/0/7	16/0/4

Table 4.6 Binary performance of sampling strategies against **"passive learning"** in split strategy **"stratified"** at specified percentages of labeled training data

Sampling Strategy	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	1	1
uncertainty	0	0	1	1
random	0	0	1	1
rr_cycle_switching_50	0	0	1	1
explore_intensify_0.1	0	0	1	1
explore_intensify_0.2	0	0	1	1
explore_intensify_0.3	0	0	1	1
explore_intensify_0.4	0	1	1	1
explore_intensify_0.5	0	0	1	1
explore_intensify_0.6	0	0	1	1
explore_intensify_0.7	0	0	1	1
explore_intensify_0.8	0	0	1	1
explore_intensify_0.9	0	0	1	1

This binary representation highlights that all active learning methods outperformed the passive learning model when 75% of the training data was labeled. Further-

more, all active learning methods consistently outperformed the passive learning model with 100% of the training data labeled. Notably, the "explore_intensify_0.4" method achieved superior performance with only 50% of the training data labeled.

Tables 4.7 and 4.8 present corresponding results for the molecular scaffold splitting setup. In this case, all active learning methods outperformed the passive learning model with 100% of the training data labeled. Additionally, the "random", "explore_intensify_0.3", "explore_intensify_0.4", "explore_intensify_0.5", and "explore_intensify_0.7" methods achieved better performance with 75% of the training data labeled. Furthermore, methods such as "dissimilarity", "explore_intensify_0.5", "explore_intensify_0.6", "explore_intensify_0.7", "explore_intensify_0.8", and "explore_intensify_0.9" outperformed the passive learning model with only 50% of the training data labeled.

Table 4.7 Win/Tie/Loss counts for split strategy **"scaffold"** against **"passive learning"** baseline at specified percentages of labeled training data

Sampling Strategy	**25%**	**50%**	**75%**	**100%**
dissimilarity	0/0/20	11/0/9	8/0/12	20/0/0
uncertainty	2/0/18	2/0/18	5/0/15	16/0/4
random	2/0/18	8/0/12	17/0/3	20/0/0
rr_cycle_switching_50	3/0/17	3/0/17	10/0/10	11/0/9
explore_intensify_0.1	0/0/20	0/0/20	6/0/14	17/0/3
explore_intensify_0.2	2/0/18	2/0/18	5/0/15	15/0/5
explore_intensify_0.3	0/0/20	4/0/16	11/0/9	17/0/3
explore_intensify_0.4	0/0/20	1/0/19	11/0/9	14/0/6
explore_intensify_0.5	0/0/20	11/0/9	12/0/8	13/0/7
explore_intensify_0.6	0/0/20	11/0/9	7/0/13	16/0/4
explore_intensify_0.7	0/0/20	11/0/9	11/0/9	12/0/8
explore_intensify_0.8	0/0/20	11/0/9	8/0/12	19/0/1
explore_intensify_0.9	0/0/20	11/0/9	8/0/12	18/0/2

Table 4.8 Binary performance of sampling strategies against **"passive learning"** in split strategy **"scaffold"** at specified percentages of labeled training data

Sampling Strategy	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	1	0	1
uncertainty	0	0	0	1
random	0	0	1	1
rr_cycle_switching_50	0	0	0	1
explore_intensify_0.1	0	0	0	1
explore_intensify_0.2	0	0	0	1
explore_intensify_0.3	0	0	1	1
explore_intensify_0.4	0	0	1	1
explore_intensify_0.5	0	1	1	1
explore_intensify_0.6	0	1	0	1
explore_intensify_0.7	0	1	1	1
explore_intensify_0.8	0	1	0	1
explore_intensify_0.9	0	1	0	1

The best-performing active learning method in the label-stratified setup was the "round-robin cycle switching" method. As shown in the last two columns (75% and 100%) of Tables 4.9 and 4.10, the "rr_cycle_switching_50" method consistently outperformed all other active learning methods. We hypothesize that this superior performance stems from its dynamic strategy, which alternates between different sampling strategies during the active learning loop, allowing it to adapt effectively to the data.

Table 4.9 Win/Tie/Loss counts for **"rr_cycle_switching_50"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	10/0/10	9/0/11	11/0/9	12/0/8
uncertainty	13/0/7	11/0/9	11/0/9	11/0/9
random	8/0/12	7/0/13	10/0/10	12/0/8
explore_intensify_0.1	7/0/13	12/0/8	10/0/10	13/0/7
explore_intensify_0.2	5/0/15	9/0/11	13/0/7	10/0/10
explore_intensify_0.3	10/0/10	10/0/10	12/0/8	17/0/3
explore_intensify_0.4	10/0/10	10/0/10	10/0/10	12/0/8
explore_intensify_0.5	10/0/10	9/0/11	11/0/9	13/0/7
explore_intensify_0.6	10/0/10	9/0/11	8/0/12	12/0/8
explore_intensify_0.7	10/0/10	9/0/11	12/0/8	11/0/9
explore_intensify_0.8	10/0/10	9/0/11	11/0/9	14/0/6
explore_intensify_0.9	10/0/10	9/0/11	11/0/9	15/0/5

Table 4.10 Binary performance for **"rr_cycle_switching_50"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	1	1
uncertainty	1	1	1	1
random	0	0	0	1
explore_intensify_0.1	0	1	0	1
explore_intensify_0.2	0	0	1	0
explore_intensify_0.3	0	0	1	1
explore_intensify_0.4	0	0	0	1
explore_intensify_0.5	0	0	1	1
explore_intensify_0.6	0	0	0	1
explore_intensify_0.7	0	0	1	1
explore_intensify_0.8	0	0	1	1
explore_intensify_0.9	0	0	1	1

Figure 4.6 illustrates the performance of the proposed round-robin cycle switching method, which alternates between three distinct sampling strategies: dissimilarity sampling (peach background), uncertainty sampling (green background), and random sampling (light purple background). The method operates cyclically, repeating these phases throughout the active learning process. The top panel of the figure tracks the imbalance ratio across varying percentages of labeled training data, while the bottom panel highlights the performance metric (average precision).

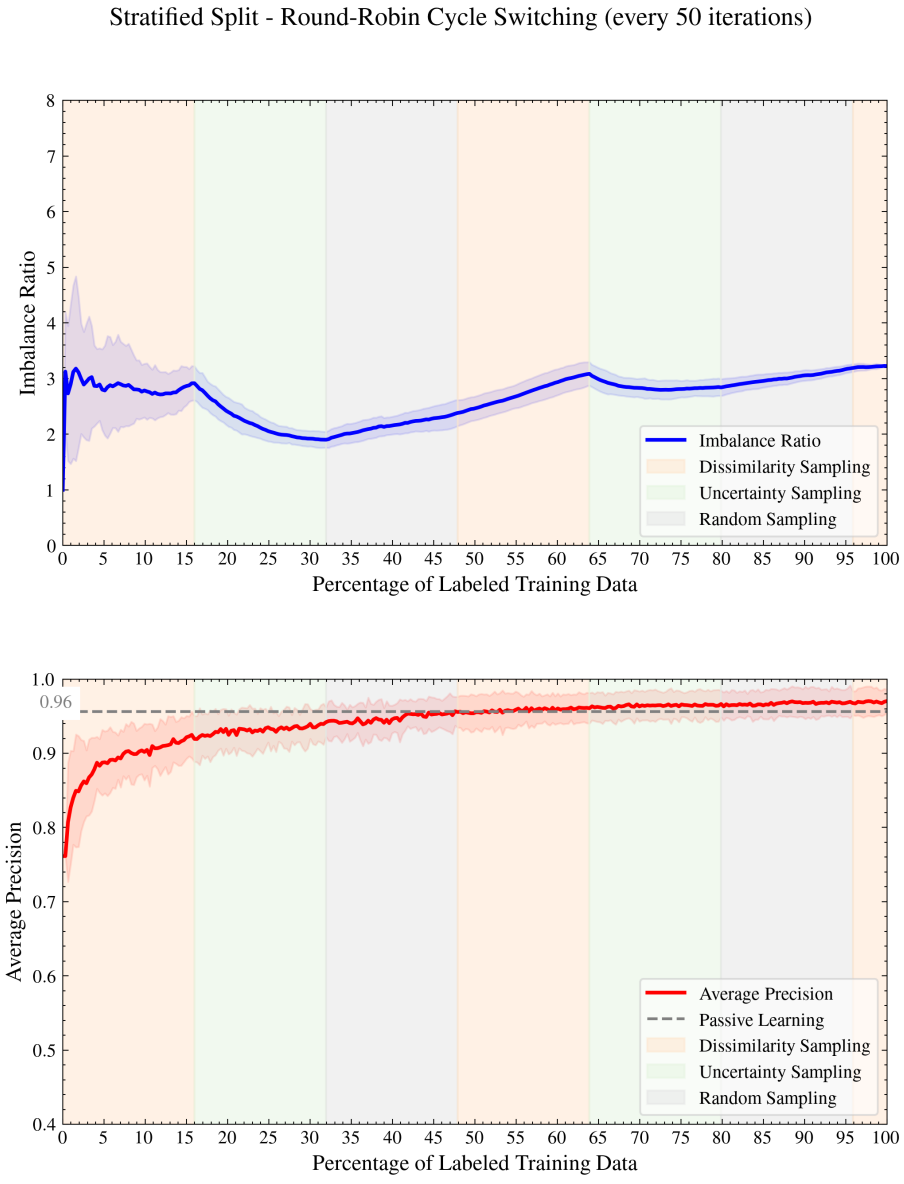


Figure 4.6 Round-Robin Cycle Switching method in the stratified data splitting setup

From the bottom panel of the figure, we observe that the round-robin cycle switching method initially improves rapidly, with the average precision rising from approximately 0.75 to 0.90 within the first 10% of labeled data. This indicates significant

early performance gains, particularly during the dissimilarity sampling phase. As more data is labeled, the average precision continues to improve steadily. The curve approaches the passive learning baseline (dashed line at 0.96) around the 40–50% labeled data mark, where the two performances nearly coincide. Beyond this point, the round-robin cycle switching method consistently exceeds the passive learning performance for the remaining training data.

The method's superiority is especially clear as the percentage of labeled training data approaches 100%, where the average precision of the round-robin cycle switching method remains marginally higher than the passive learning baseline. This is evident from the red curve slightly exceeding the dashed line in the figure's latter stages.

We conducted experiments for all active learning methods in both stratified and scaffold setups. The detailed results are provided in Chapter B of the appendix. The win/tie/loss (w/t/l) tables for these methods are presented in tables B.1 to B.48. Additionally, figures B.1 to B.4 illustrate the performance of each active learning method as the labeled training set size increases. In all these figures, the black dashed horizontal line represents the passive learning model's performance for comparison.

Figures B.1 and B.3 highlight the exploration and exploitation phases with distinct background shading. For example, in Figure B.1, panel (a), the Explore_Intensify_0.1 method switches from the exploration phase (shaded in light beige, dissimilarity sampling) to the exploitation phase (shaded in pale mint green, uncertainty sampling) after 10% of the training data is labeled. Similarly, figures B.4, panel (d), and B.2, panel (d), illustrate the cyclic alternation of the three basic sampling strategies (dissimilarity, uncertainty, and random) employed by the round-robin cycle switching method, with their corresponding background colors highlighting the transitions.

5. Limitation, Future Work, and Conclusion

In this chapter, we provide the limitations, future work, and conclusion of this thesis.

5.1 Limitation

In this section, we present the limitations or threats to the validity of this work. In the next subsections, we show the limitations with respect to dataset bias, molecular representation methods, and biological mechanism.

5.1.1 Dataset Bias

The dataset that is used in this work has an inherent bias, and that is a challenge common to most available datasets in this domain. The MoleculeNet BBB dataset used in this thesis has a considerable imbalance. This imbalance likely stems from sampling bias in the field, where positive cases may be over-represented. We hypothesize that this bias stems from the historical focus of researchers on identifying compounds that successfully penetrate the BBB. This emphasis on positive cases may have led to an over-representation of BBB+ molecules in the literature and, consequently, in curated datasets. While this approach has been valuable for identifying potential CNS-active drugs, it has inadvertently created a skewed representation of the chemical space with respect to BBB permeability. Such bias can lead to models that are less accurate in predicting BBB- compounds, potentially limiting their real-world applicability.

5.1.2 Molecular Representation

While we employed one of the most widely used molecular representation techniques (ECFP) and evaluated it against SELFIES-based embeddings, we believe that investigating more advanced approaches, such as graph neural networks or transformer-based models, could lead to better understanding and improved predictions.

5.1.3 Biological Mechanism

In this thesis, we primarily focused on predictive performance and did not explore in depth the biological mechanisms underlying BBB permeability. Integrating more mechanistic insights in future work could enhance the interpretability and reliability of the models.

5.2 Future Work

There are many ways to extend this work, specifically in the directions of both active learning strategies and drug BBB molecular property prediction. Instead of relying solely on ECFP or SELFIES embeddings, it would be advantageous to incorporate diverse molecular representations. For instance, integrating pharmacokinetic data, molecular dynamics simulations, and biochemical interaction profiles could provide a more holistic view This multidimensional strategy would more accurately mirror the real-world processes of BBB penetration. On the active learning front, new sampling strategies can be employed. We briefly describe a few of them below:

We plan to try more variations of the parameters of the proposed active learning strategies. Specifically, we can explore more switching points in the explore_intensify paradigm and different switching points and active learning strategies ordering in the Round-Robin Cycle Switching paradigm. A variation of the dynamic switching strategy that is switching after a predefined x number of iterations of the model not improving its performance on a held-out set. An alternative strategy is to leverage both molecular feature representations used in this work by employing an ensemble of models. Unlike existing techniques, this approach involves training two models in parallel: one using ECFP and the other using SELFIES embeddings. The sampling strategy would then focus on selecting data points where the two models disagree. This method allows for analyzing molecules from different perspectives; ECFP captures the structural components of molecules, while SELFIES embeddings emphasize their sequential and symbolic representation.

5.3 Conclusion Remarks

Predicting the permeability of chemical compounds through the BBB is a critical step in the development of drugs designed to treat CNS disorders. Given that the amount of labeled data available for chemicals with experimentally verified BBB permeability is far from comprehensive, there is a need for more efficient and intelligent methods to maximize the utility of labeling efforts by biologists and chemists in wet labs. In this thesis, we demonstrated that adopting an AL framework for the problem of BBB permeability prediction is both effective and efficient. Our results show that active learning approaches achieved the performance of passive learning models after utilizing only 10%-65% of the labeled data, depending on the specific performance metric. This highlights the efficiency of active learning in reducing labeling costs while maintaining high model performance. We specifically explored and compared multiple sampling strategies, including random sampling, uncertainty-based sampling, and dissimilarity-based sampling. Additionally, we introduced two novel active learning strategies: explore-intensify and round-robin cycle switching. Our experiments revealed that the round-robin cycle switching strategy consistently outperformed other active learning strategies in the stratified-split setup, emphasizing its potential for dynamic and adaptive data selection. Furthermore, we evaluated the impact of different data splitting techniques, including label-stratified splitting and scaffold-based splitting. The scaffold-based splitting, a more challenging setup, resulted in lower performance for both passive and active learning paradigms, underscoring its utility as a rigorous evaluation benchmark. This finding also points to the need for deeper molecular scaffold analyses, potentially involving domain experts, to uncover insights into how molecular scaffolds influence BBB permeability. Such analyses could inform the development of more interpretable and robust ML models for BBB permeability prediction.

BIBLIOGRAPHY

Alsenan, S., Al-Turaiki, I., & Hafez, A. (2020). A recurrent neural network model to predict blood–brain barrier permeability. *Computational Biology and Chemistry*, *89*, 107377.

Alsenan, S., Al-Turaiki, I., & Hafez, A. (2021). A deep learning approach to predict blood-brain barrier permeability. *PeerJ Computer Science*, *7*, e515.

Banerjee, A. & Roy, K. (2024). How to correctly develop q-rasar models for predictive cheminformatics.

Bemis, G. W. & Murcko, M. A. (1996). The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, *39*(15), 2887–2893.

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., & Cox, D. D. (2015). Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, *8*(1), 014008.

Birba, D. (2020). A comparative study of data splitting algorithms for machine learning model selection (2020).

Boldini, D., Grisoni, F., Kuhn, D., Friedrich, L., & Sieber, S. A. (2023). Practical guidelines for the use of gradient boosting for molecular property prediction. *Journal of Cheminformatics*, *15*(1), 73.

Chen, J., Hou, J., & Wong, K.-C. (2020). Categorical matrix completion with active learning for high-throughput screening. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *18*(6), 2261–2270.

Cherian Parakkal, S., Datta, R., & Das, D. (2022). Deepbbbp: high accuracy blood-brain-barrier permeability prediction with a mixed deep learning model. *Molecular Informatics*, *41*(10), 2100315.

Cornelissen, F. M., Markert, G., Deutsch, G., Antonara, M., Faaij, N., Bartelink, I., Noske, D., Vandertop, W. P., Bender, A., & Westerman, B. A. (2023). Explaining blood–brain barrier permeability of small molecules by integrated analysis of different transport mechanisms. *Journal of Medicinal Chemistry*, *66*(11), 7253–7267.

Czarnecki, W. M., Jastrzebski, S., Sieradzki, I., & Podlewska, S. (2015). Active learning of compounds activity–towards scientifically sound simulation of drug candidates identification. In *Proceedings of MLLS: 2nd Workshop on Machine Learning in Life Sciences*, (pp. 40–51).

De Grave, K., Ramon, J., & De Raedt, L. (2008). Active learning for high throughput screening. In *Discovery Science: 11th International Conference, DS 2008, Budapest, Hungary, October 13-16, 2008. Proceedings 11*, (pp. 185–196). Springer.

Deng, J., Yang, Z., Wang, H., Ojima, I., Samaras, D., & Wang, F. (2023). A systematic study of key elements underlying molecular property prediction. *Nature Communications*, *14*(1), 6395.

Ding, X., Cui, R., Yu, J., Liu, T., Zhu, T., Wang, D., Chang, J., Fan, Z., Liu, X., Chen, K., et al. (2021). Active learning for drug design: a case study on the plasma exposure of orally administered drugs. *Journal of Medicinal Chemistry*, *64*(22), 16838–16853.

Donmez, P., Carbonell, J. G., & Bennett, P. N. (2007). Dual strategy active learning. In *Machine Learning: ECML 2007: 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007. Proceedings 18*, (pp. 116–127). Springer.

Graff, D. E., Shakhnovich, E. I., & Coley, C. W. (2021). Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chemical science*, *12*(22), 7866–7881.

Gubaev, K., Podryabinkin, E. V., & Shapeev, A. V. (2018). Machine learning of molecular properties: Locality and active learning. *The Journal of chemical physics*, *148*(24).

Huang, E. T., Yang, J.-S., Liao, K. Y., Tseng, W. C., Lee, C., Gill, M., Compas, C., See, S., & Tsai, F.-J. (2024). Predicting blood–brain barrier permeability of molecules with a large language model and machine learning. *Scientific Reports*, *14*(1), 15844.

Krenn, M., Häse, F., Nigam, A., Friederich, P., & Aspuru-Guzik, A. (2020). Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, *1*(4), 045024.

Kumar, S., Luo, W., Kantor, G., & Sycara, K. (2019). Active learning with gaussian processes for high throughput phenotyping. *arXiv preprint arXiv:1901.06803*.

Landrum, G. et al. (2006). Rdkit: Open-source cheminformatics.

Li, C., Feng, J., Liu, S., & Yao, J. (2022). A novel molecular representation learning for molecular property prediction with a multiple smiles-based augmentation. *Computational Intelligence and Neuroscience*, *2022*(1), 8464452.

Li, L., Xiao, Y., Ma, D., & Zheng, K. (2022). Prevail: Pre-trained variational adversarial active learning for molecular property prediction. In *2022 IEEE 8th International Conference on Cloud Computing and Intelligent Systems (CCIS)*, (pp. 143–149). IEEE.

Liang, L., Liu, Z., Yang, X., Zhang, Y., Liu, H., & Chen, Y. (2024). Prediction of blood-brain barrier permeability using machine learning approaches based on various molecular representation. *Molecular Informatics*, e202300327.

Martins, I. F., Teixeira, A. L., Pinheiro, L., & Falcao, A. O. (2012). A Bayesian Approach to in Silico Blood-Brain Barrier Penetration Modeling. *Journal of Chemical Information and Modeling*, *52*(6), 1686–1697.

Mazumdar, B., Sarma, P. K. D., Mahanta, H. J., & Sastry, G. N. (2023). Machine learning based dynamic consensus model for predicting blood-brain barrier permeability. *Computers in Biology and Medicine*, *160*, 106984.

Meng, F., Xi, Y., Huang, J., & Ayers, P. W. (2021). A curated diverse molecular database of blood-brain barrier permeability with chemical descriptors. *Scientific Data*, *8*(1), 289.

Miao, R., Xia, L.-Y., Chen, H.-H., Huang, H.-H., & Liang, Y. (2019). Improved classification of blood-brain-barrier drugs using deep learning. *Scientific reports*, *9*(1), 8802.

Moriwaki, H., Tian, Y.-S., Kawashita, N., & Takagi, T. (2018). Mordred: a molecular descriptor calculator. *Journal of Cheminformatics*, *10*(1), 4.

Putatunda, S. & Rama, K. (2018). A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of xgboost. In *Proceedings of the 2018 international conference on signal processing and machine learning*, (pp. 6–10).

Reker, D. (2019). Practical considerations for active machine learning in drug discovery. *Drug Discovery Today: Technologies*, *32*, 73–79.

Rogers, D. & Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of chemical information and modeling*, *50*(5), 742–754.

Settles, B. (2009). Active learning literature survey.

Shaker, B., Yu, M.-S., Song, J. S., Ahn, S., Ryu, J. Y., Oh, K.-S., & Na, D. (2021). Lightbbb: computational prediction model of blood–brain-barrier penetration based on lightgbm. *Bioinformatics*, *37*(8), 1135–1139.

Singh, R., Ghosh, P., Ganeshpurkar, A., Anand, A., Swetha, R., Singh, R. B., Kumar, D., Singh, S. K., & Kumar, A. (2023). Natural-language processing (nlp) based feature extraction technique in deep-learning model to predict the blood-brain-barrier permeability of molecules. *Molecular Informatics*, *42*(10), 2200271.

Sverchkov, Y. & Craven, M. (2017). A review of active learning approaches to experimental design for uncovering biological networks. *PLoS computational biology*, *13*(6), e1005466.

Tang, Q., Nie, F., Zhao, Q., & Chen, W. (2022). A merged molecular representation deep learning method for blood–brain barrier permeability prediction. *Briefings in Bioinformatics*, *23*(5), bbac357.

Tong, S. & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of machine learning research*, *2*(Nov), 45–66.

van Tilborg, D. & Grisoni, F. (2024). Traversing chemical space with active deep learning: A computational framework for low-data drug discovery.

Wang, L., Zhou, Z., Yang, X., Shi, S., Zeng, X., & Cao, D. (2024). The present

state and challenges of active learning in drug discovery. *Drug Discovery Today*, 103985.

Wang, Z., Yang, H., Wu, Z., Wang, T., Li, W., Tang, Y., & Liu, G. (2018). In silico prediction of blood–brain barrier permeability of compounds by machine learning and resampling methods. *ChemMedChem*, *13*(20), 2189–2201.

Weininger, D. (1988). Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, *28*(1), 31–36.

Weininger, D., Weininger, A., & Weininger, J. L. (1989). Smiles. 2. algorithm for generation of unique smiles notation. *Journal of chemical information and computer sciences*, *29*(2), 97–101.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., & Pande, V. (2018). MoleculeNet: A benchmark for molecular machine learning. *Chemical Science*, *9*(2), 513–530.

Wu, Z., Xian, Z., Ma, W., Liu, Q., Huang, X., Xiong, B., He, S., & Zhang, W. (2021). Artificial neural network approach for predicting blood brain barrier permeability based on a group contribution method. *Computer methods and programs in biomedicine*, *200*, 105943.

Yüksel, A., Ulusoy, E., Ünlü, A., & Doğan, T. (2023). Selformer: molecular representation learning via selfies language models. *Machine Learning: Science and Technology*, *4*(2), 025035.

Zhang, D., Xiao, J., Zhou, N., Zheng, M., Luo, X., Jiang, H., & Chen, K. (2015). A genetic algorithm based support vector machine model for blood-brain barrier penetration prediction. *BioMed research international*, *2015*(1), 292683.

Zhou, K., Wang, K., Feng, J., Tang, J., Xu, T., & Wang, X. (2022). Tyger: Task-type-generic active learning for molecular property prediction. *arXiv preprint arXiv:2205.11279*.

APPENDIX A Additional Figures

A.1 Multiple non-canonical SMILES mapping to unique SMILES

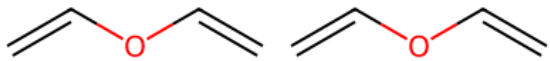


Figure A.1 Mapping non-canonical to canonical_ 0 for two non-canonical SMILES strings mapping to a unique canonical SMILES string

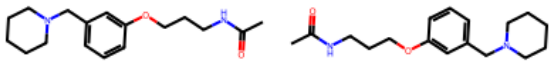


Figure A.2 Mapping non-canonical to canonical_ 1 for two non-canonical SMILES strings mapping to a unique canonical SMILES string

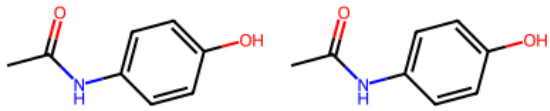


Figure A.3 Mapping non-canonical to canonical_ 2 for two non-canonical SMILES strings mapping to a unique canonical SMILES string

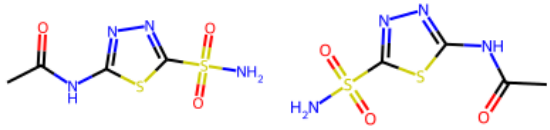


Figure A.4 Mapping non-canonical to canonical_ 3 for two non-canonical SMILES strings mapping to a unique canonical SMILES string

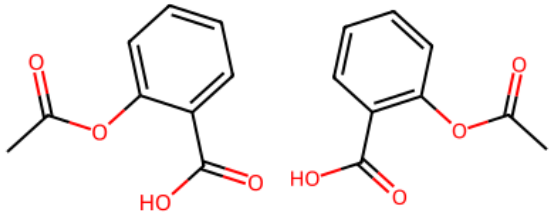


Figure A.5 Mapping non-canonical to canonical_ 4 for two non-canonical SMILES strings mapping to a unique canonical SMILES string

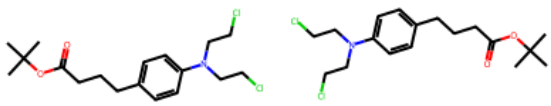


Figure A.6 Mapping non-canonical to canonical_ 5 for two non-canonical SMILES strings mapping to a unique canonical SMILES string

A.2 Interesting cases

The figures below are continuations of the figure 4.1

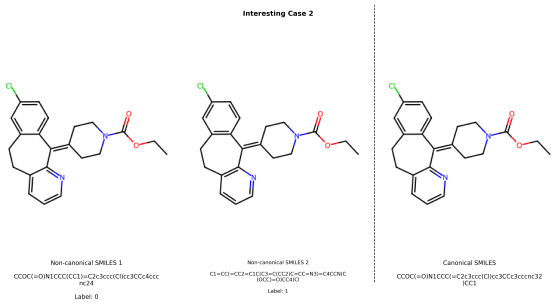


Figure A.7 Interesting case number 2 for two non-canonical SMILES strings with different labels mapping to a unique canonical SMILES string

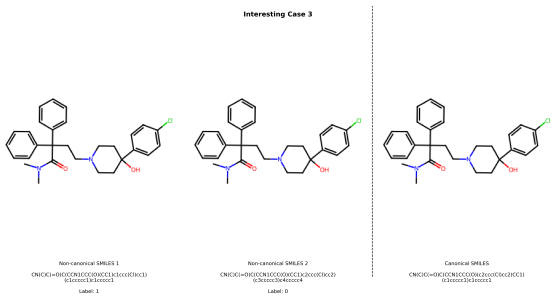


Figure A.8 Interesting case number 3 for two non-canonical SMILES strings with different labels mapping to a unique canonical SMILES string

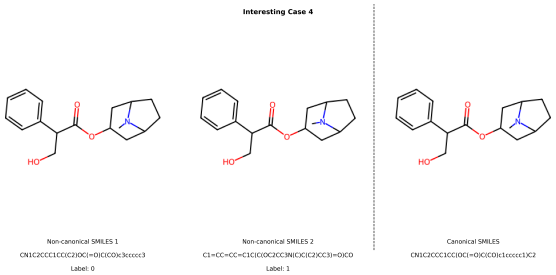


Figure A.9 Interesting case number 4 for two non-canonical SMILES strings with different labels mapping to a unique canonical SMILES string

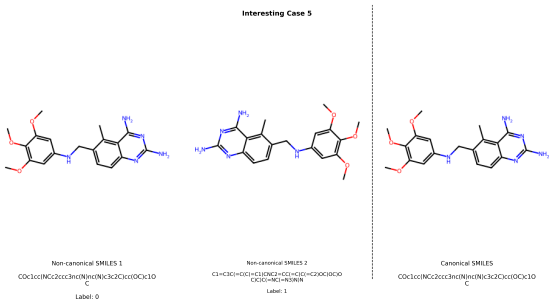


Figure A.10 Interesting case number 5 for two non-canonical SMILES strings with different labels mapping to a unique canonical SMILES string

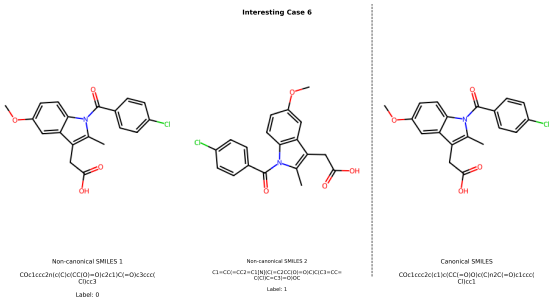


Figure A.11 Interesting case number 6 for two non-canonical SMILES strings with different labels mapping to a unique canonical SMILES string

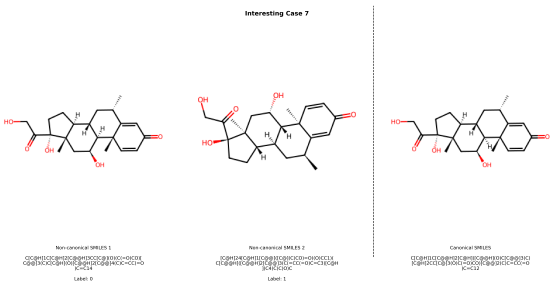


Figure A.12 Interesting case number 7 for two non-canonical SMILES strings with different labels mapping to a unique canonical SMILES string

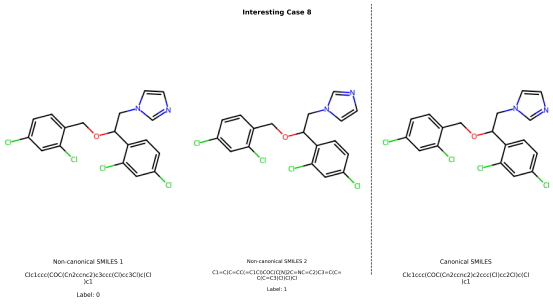


Figure A.13 Interesting case number 8 for two non-canonical SMILES strings with different labels mapping to a unique canonical SMILES string

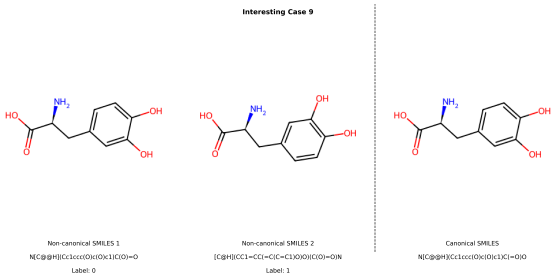


Figure A.14 Interesting case number 9 for two non-canonical SMILES strings with different labels mapping to a unique canonical SMILES string

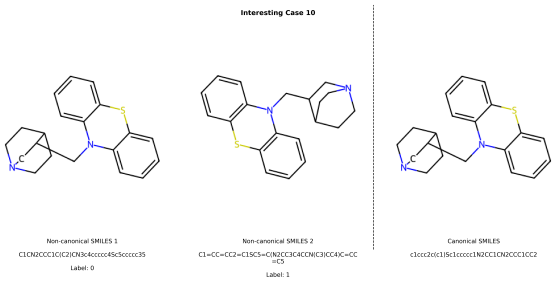


Figure A.15 Interesting case number 10 for two non-canonical SMILES strings with different labels mapping to a unique canonical SMILES string

A.3 Enrichment of some Scaffold Groups

The figures below are continuations of the figure 4.2

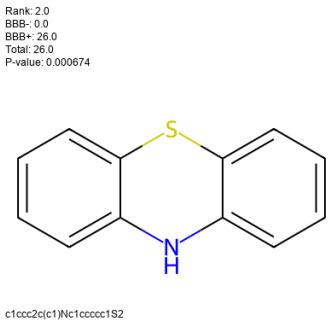


Figure A.16 Scaffold enrichment number 2

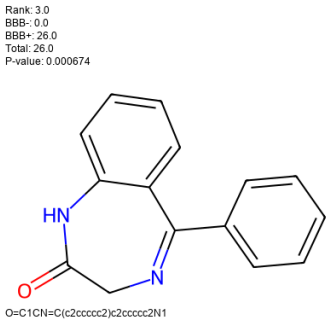


Figure A.17 Scaffold enrichment number 3

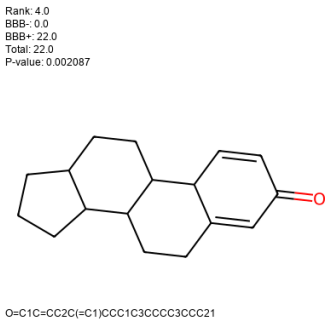


Figure A.18 Scaffold enrichment number 4

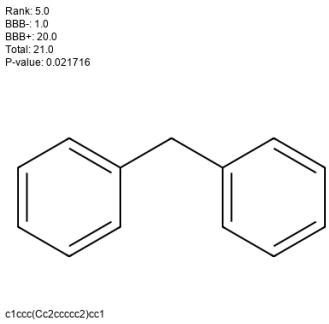


Figure A.19 Scaffold enrichment number 5

APPENDIX B Additional Experimental Results

B.1 Stratified-splitting

Table B.1 Win/Tie/Loss counts for **"dissimilarity"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
uncertainty	14/0/6	9/0/11	10/0/10	8/0/12
random	7/0/13	9/0/11	8/0/12	10/0/10
rr_cycle_switching_50	10/0/10	11/0/9	9/0/11	8/0/12
explore_intensify_0.1	7/0/13	10/0/10	11/0/9	11/0/9
explore_intensify_0.2	6/0/14	12/0/8	13/0/7	7/0/13
explore_intensify_0.3	0/20/0	8/0/12	12/0/8	11/0/9
explore_intensify_0.4	0/20/0	7/0/13	12/0/8	10/0/10
explore_intensify_0.5	0/20/0	0/20/0	9/0/11	12/0/8
explore_intensify_0.6	0/20/0	0/20/0	6/0/14	12/0/8
explore_intensify_0.7	0/20/0	0/20/0	10/0/10	11/0/9
explore_intensify_0.8	0/20/0	0/20/0	0/20/0	12/0/8
explore_intensify_0.9	0/20/0	0/20/0	0/20/0	11/0/9

Table B.2 Binary performance for **"dissimilarity"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
uncertainty	1	0	0	0
random	0	0	0	0
rr_cycle_switching_50	0	1	0	0
explore_intensify_0.1	0	0	1	1
explore_intensify_0.2	0	1	1	0
explore_intensify_0.3	0	0	1	1
explore_intensify_0.4	0	0	1	0
explore_intensify_0.5	0	0	0	1
explore_intensify_0.6	0	0	0	1
explore_intensify_0.7	0	0	0	1
explore_intensify_0.8	0	0	0	1
explore_intensify_0.9	0	0	0	1

Table B.3 Win/Tie/Loss counts for **"uncertainty"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	6/0/14	11/0/9	10/0/10	12/0/8
random	5/0/15	11/0/9	9/0/11	10/0/10
rr_cycle_switching_50	7/0/13	9/0/11	9/0/11	9/0/11
explore_intensify_0.1	3/0/17	12/0/8	9/0/11	6/11/3
explore_intensify_0.2	3/0/17	10/0/10	13/0/7	1/12/7
explore_intensify_0.3	6/0/14	11/0/9	10/0/10	12/6/2
explore_intensify_0.4	6/0/14	11/0/9	12/0/8	7/7/6
explore_intensify_0.5	6/0/14	11/0/9	10/0/10	9/7/4
explore_intensify_0.6	6/0/14	11/0/9	7/0/13	8/8/4
explore_intensify_0.7	6/0/14	11/0/9	9/0/11	7/9/4
explore_intensify_0.8	6/0/14	11/0/9	10/0/10	10/5/5
explore_intensify_0.9	6/0/14	11/0/9	10/0/10	11/1/8

Table B.4 Binary performance for **"uncertainty"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	1	0	1
random	0	1	0	0
rr_cycle_switching_50	0	0	0	0
explore_intensify_0.1	0	1	0	0
explore_intensify_0.2	0	0	1	0
explore_intensify_0.3	0	1	0	1
explore_intensify_0.4	0	1	1	0
explore_intensify_0.5	0	1	0	0
explore_intensify_0.6	0	1	0	0
explore_intensify_0.7	0	1	0	0
explore_intensify_0.8	0	1	0	0
explore_intensify_0.9	0	1	0	1

Table B.5 Win/Tie/Loss counts for **"random"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	13/0/7	11/0/9	12/0/8	10/0/10
uncertainty	15/0/5	9/0/11	11/0/9	10/0/10
rr_cycle_switching_50	12/0/8	13/0/7	10/0/10	8/0/12
explore_intensify_0.1	9/0/11	12/0/8	11/0/9	11/0/9
explore_intensify_0.2	8/0/12	13/0/7	11/0/9	9/0/11
explore_intensify_0.3	13/0/7	12/0/8	12/0/8	11/0/9
explore_intensify_0.4	13/0/7	11/0/9	12/0/8	9/0/11
explore_intensify_0.5	13/0/7	11/0/9	10/0/10	10/0/10
explore_intensify_0.6	13/0/7	11/0/9	10/0/10	8/0/12
explore_intensify_0.7	13/0/7	11/0/9	12/0/8	9/0/11
explore_intensify_0.8	13/0/7	11/0/9	12/0/8	11/0/9
explore_intensify_0.9	13/0/7	11/0/9	12/0/8	9/0/11

Table B.6 Binary performance for **"random"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	1	1	1	0
uncertainty	1	0	1	0
rr_cycle_switching_50	1	1	0	0
explore_intensify_0.1	0	1	1	1
explore_intensify_0.2	0	1	1	0
explore_intensify_0.3	1	1	1	1
explore_intensify_0.4	1	1	1	0
explore_intensify_0.5	1	1	0	0
explore_intensify_0.6	1	1	0	0
explore_intensify_0.7	1	1	1	0
explore_intensify_0.8	1	1	1	1
explore_intensify_0.9	1	1	1	0

Table B.7 Win/Tie/Loss counts for **"explore_intensify_0.1"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	13/0/7	10/0/10	9/0/11	9/0/11
uncertainty	17/0/3	8/0/12	11/0/9	3/11/6
random	11/0/9	8/0/12	9/0/11	9/0/11
rr_cycle_switching_50	13/0/7	8/0/12	10/0/10	7/0/13
explore_intensify_0.2	11/0/9	8/0/12	10/0/10	1/10/9
explore_intensify_0.3	13/0/7	7/0/13	11/0/9	11/4/5
explore_intensify_0.4	13/0/7	10/0/10	10/0/10	6/6/8
explore_intensify_0.5	13/0/7	10/0/10	10/0/10	6/9/5
explore_intensify_0.6	13/0/7	10/0/10	7/0/13	7/8/5
explore_intensify_0.7	13/0/7	10/0/10	8/0/12	4/8/8
explore_intensify_0.8	13/0/7	10/0/10	9/0/11	7/7/6
explore_intensify_0.9	13/0/7	10/0/10	9/0/11	10/2/8

Table B.8 Binary performance for **"explore_intensify_0.1"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	1	0	0	0
uncertainty	1	0	1	0
random	1	0	0	0
rr_cycle_switching_50	1	0	0	0
explore_intensify_0.2	1	0	0	0
explore_intensify_0.3	1	0	1	1
explore_intensify_0.4	1	0	0	0
explore_intensify_0.5	1	0	0	0
explore_intensify_0.6	1	0	0	0
explore_intensify_0.7	1	0	0	0
explore_intensify_0.8	1	0	0	0
explore_intensify_0.9	1	0	0	0

Table B.9 Win/Tie/Loss counts for **"explore_intensify_0.2"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	14/0/6	8/0/12	7/0/13	13/0/7
uncertainty	17/0/3	10/0/10	7/0/13	7/12/1
random	12/0/8	7/0/13	9/0/11	11/0/9
rr_cycle_switching_50	15/0/5	11/0/9	7/0/13	10/0/10
explore_intensify_0.1	9/0/11	12/0/8	10/0/10	9/10/1
explore_intensify_0.3	14/0/6	10/0/10	8/0/12	13/6/1
explore_intensify_0.4	14/0/6	10/0/10	8/0/12	8/7/5
explore_intensify_0.5	14/0/6	8/0/12	6/0/14	10/9/1
explore_intensify_0.6	14/0/6	8/0/12	6/0/14	9/8/3
explore_intensify_0.7	14/0/6	8/0/12	8/0/12	9/7/4
explore_intensify_0.8	14/0/6	8/0/12	7/0/13	13/3/4
explore_intensify_0.9	14/0/6	8/0/12	7/0/13	12/1/7

Table B.10 Binary performance for **"explore_intensify_0.2"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	1	0	0	1
uncertainty	1	0	0	0
random	1	0	0	1
rr_cycle_switching_50	1	1	0	0
explore_intensify_0.1	0	1	0	0
explore_intensify_0.3	1	0	0	1
explore_intensify_0.4	1	0	0	0
explore_intensify_0.5	1	0	0	0
explore_intensify_0.6	1	0	0	0
explore_intensify_0.7	1	0	0	0
explore_intensify_0.8	1	0	0	1
explore_intensify_0.9	1	0	0	1

Table B.11 Win/Tie/Loss counts for **"explore_intensify_0.3"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0/20/0	12/0/8	8/0/12	9/0/11
uncertainty	14/0/6	9/0/11	10/0/10	2/6/12
random	7/0/13	8/0/12	8/0/12	9/0/11
rr_cycle_switching_50	10/0/10	10/0/10	8/0/12	3/0/17
explore_intensify_0.1	7/0/13	13/0/7	9/0/11	5/4/11
explore_intensify_0.2	6/0/14	10/0/10	12/0/8	1/6/13
explore_intensify_0.4	0/20/0	11/0/9	12/0/8	4/6/10
explore_intensify_0.5	0/20/0	12/0/8	9/0/11	7/3/10
explore_intensify_0.6	0/20/0	12/0/8	4/0/16	5/7/8
explore_intensify_0.7	0/20/0	12/0/8	10/0/10	6/6/8
explore_intensify_0.8	0/20/0	12/0/8	8/0/12	9/2/9
explore_intensify_0.9	0/20/0	12/0/8	8/0/12	11/1/8

Table B.12 Binary performance for **"explore_intensify_0.3"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	1	0	0
uncertainty	1	0	0	0
random	0	0	0	0
rr_cycle_switching_50	0	0	0	0
explore_intensify_0.1	0	1	0	0
explore_intensify_0.2	0	0	1	0
explore_intensify_0.4	0	1	1	0
explore_intensify_0.5	0	1	0	0
explore_intensify_0.6	0	1	0	0
explore_intensify_0.7	0	1	0	0
explore_intensify_0.8	0	1	0	0
explore_intensify_0.9	0	1	0	1

Table B.13 Win/Tie/Loss counts for **"explore_intensify_0.4"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0/20/0	13/0/7	8/0/12	10/0/10
uncertainty	14/0/6	9/0/11	8/0/12	6/7/7
random	7/0/13	9/0/11	8/0/12	11/0/9
rr_cycle_switching_50	10/0/10	10/0/10	10/0/10	8/0/12
explore_intensify_0.1	7/0/13	10/0/10	10/0/10	8/6/6
explore_intensify_0.2	6/0/14	10/0/10	12/0/8	5/7/8
explore_intensify_0.3	0/20/0	9/0/11	8/0/12	10/6/4
explore_intensify_0.5	0/20/0	13/0/7	11/0/9	10/5/5
explore_intensify_0.6	0/20/0	13/0/7	6/0/14	9/7/4
explore_intensify_0.7	0/20/0	13/0/7	9/0/11	8/6/6
explore_intensify_0.8	0/20/0	13/0/7	8/0/12	12/4/4
explore_intensify_0.9	0/20/0	13/0/7	8/0/12	12/0/8

Table B.14 Binary performance for **"explore_intensify_0.4"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	1	0	0
uncertainty	1	0	0	0
random	0	0	0	1
rr_cycle_switching_50	0	0	0	0
explore_intensify_0.1	0	0	0	0
explore_intensify_0.2	0	0	1	0
explore_intensify_0.3	0	0	0	0
explore_intensify_0.5	0	1	1	0
explore_intensify_0.6	0	1	0	0
explore_intensify_0.7	0	1	0	0
explore_intensify_0.8	0	1	0	1
explore_intensify_0.9	0	1	0	1

Table B.15 Win/Tie/Loss counts for **"explore_intensify_0.5"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0/20/0	0/20/0	11/0/9	8/0/12
uncertainty	14/0/6	9/0/11	10/0/10	4/7/9
random	7/0/13	9/0/11	10/0/10	10/0/10
rr_cycle_switching_50	10/0/10	11/0/9	9/0/11	7/0/13
explore_intensify_0.1	7/0/13	10/0/10	10/0/10	5/9/6
explore_intensify_0.2	6/0/14	12/0/8	14/0/6	1/9/10
explore_intensify_0.3	0/20/0	8/0/12	11/0/9	10/3/7
explore_intensify_0.4	0/20/0	7/0/13	9/0/11	5/5/10
explore_intensify_0.6	0/20/0	0/20/0	6/0/14	6/10/4
explore_intensify_0.7	0/20/0	0/20/0	8/0/12	4/9/7
explore_intensify_0.8	0/20/0	0/20/0	11/0/9	9/6/5
explore_intensify_0.9	0/20/0	0/20/0	11/0/9	7/4/9

Table B.16 Binary performance for **"explore_intensify_0.5"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	1	0
uncertainty	1	0	0	0
random	0	0	0	0
rr_cycle_switching_50	0	1	0	0
explore_intensify_0.1	0	0	0	0
explore_intensify_0.2	0	1	1	0
explore_intensify_0.3	0	0	1	0
explore_intensify_0.4	0	0	0	0
explore_intensify_0.6	0	0	0	0
explore_intensify_0.7	0	0	0	0
explore_intensify_0.8	0	0	1	0
explore_intensify_0.9	0	0	1	0

Table B.17 Win/Tie/Loss counts for **"explore_intensify_0.6"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0/20/0	0/20/0	14/0/6	8/0/12
uncertainty	14/0/6	9/0/11	13/0/7	4/8/8
random	7/0/13	9/0/11	10/0/10	12/0/8
rr_cycle_switching_50	10/0/10	11/0/9	12/0/8	8/0/12
explore_intensify_0.1	7/0/13	10/0/10	13/0/7	5/8/7
explore_intensify_0.2	6/0/14	12/0/8	14/0/6	3/8/9
explore_intensify_0.3	0/20/0	8/0/12	16/0/4	8/7/5
explore_intensify_0.4	0/20/0	7/0/13	14/0/6	4/7/9
explore_intensify_0.5	0/20/0	0/20/0	14/0/6	4/10/6
explore_intensify_0.7	0/20/0	0/20/0	11/0/9	3/10/7
explore_intensify_0.8	0/20/0	0/20/0	14/0/6	9/5/6
explore_intensify_0.9	0/20/0	0/20/0	14/0/6	10/2/8

Table B.18 Binary performance for **"explore_intensify_0.6"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	1	0
uncertainty	1	0	1	0
random	0	0	0	1
rr_cycle_switching_50	0	1	1	0
explore_intensify_0.1	0	0	1	0
explore_intensify_0.2	0	1	1	0
explore_intensify_0.3	0	0	1	0
explore_intensify_0.4	0	0	1	0
explore_intensify_0.5	0	0	1	0
explore_intensify_0.7	0	0	1	0
explore_intensify_0.8	0	0	1	0
explore_intensify_0.9	0	0	1	0

Table B.19 Win/Tie/Loss counts for **"explore_intensify_0.7"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0/20/0	0/20/0	10/0/10	9/0/11
uncertainty	14/0/6	9/0/11	11/0/9	4/9/7
random	7/0/13	9/0/11	8/0/12	11/0/9
rr_cycle_switching_50	10/0/10	11/0/9	8/0/12	9/0/11
explore_intensify_0.1	7/0/13	10/0/10	12/0/8	8/8/4
explore_intensify_0.2	6/0/14	12/0/8	12/0/8	4/7/9
explore_intensify_0.3	0/20/0	8/0/12	10/0/10	8/6/6
explore_intensify_0.4	0/20/0	7/0/13	11/0/9	6/6/8
explore_intensify_0.5	0/20/0	0/20/0	12/0/8	7/9/4
explore_intensify_0.6	0/20/0	0/20/0	9/0/11	7/10/3
explore_intensify_0.8	0/20/0	0/20/0	10/0/10	9/7/4
explore_intensify_0.9	0/20/0	0/20/0	10/0/10	9/3/8

Table B.20 Binary performance for **"explore_intensify_0.7"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	0	0
uncertainty	1	0	1	0
random	0	0	0	1
rr_cycle_switching_50	0	1	0	0
explore_intensify_0.1	0	0	1	0
explore_intensify_0.2	0	1	1	0
explore_intensify_0.3	0	0	0	0
explore_intensify_0.4	0	0	1	0
explore_intensify_0.5	0	0	1	0
explore_intensify_0.6	0	0	0	0
explore_intensify_0.8	0	0	0	0
explore_intensify_0.9	0	0	0	0

Table B.21 Win/Tie/Loss counts for **"explore_intensify_0.8"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0/20/0	0/20/0	0/20/0	8/0/12
uncertainty	14/0/6	9/0/11	10/0/10	5/5/10
random	7/0/13	9/0/11	8/0/12	9/0/11
rr_cycle_switching_50	10/0/10	11/0/9	9/0/11	6/0/14
explore_intensify_0.1	7/0/13	10/0/10	11/0/9	6/7/7
explore_intensify_0.2	6/0/14	12/0/8	13/0/7	4/3/13
explore_intensify_0.3	0/20/0	8/0/12	12/0/8	9/2/9
explore_intensify_0.4	0/20/0	7/0/13	12/0/8	4/4/12
explore_intensify_0.5	0/20/0	0/20/0	9/0/11	5/6/9
explore_intensify_0.6	0/20/0	0/20/0	6/0/14	6/5/9
explore_intensify_0.7	0/20/0	0/20/0	10/0/10	4/7/9
explore_intensify_0.9	0/20/0	0/20/0	0/20/0	8/2/10

Table B.22 Binary performance for **"explore_intensify_0.8"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

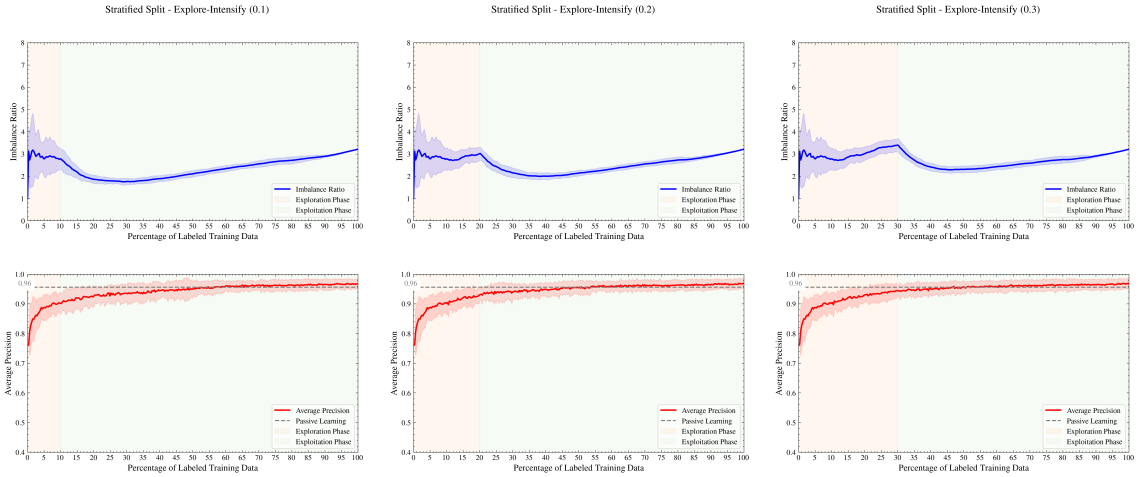
Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	0	0
uncertainty	1	0	0	0
random	0	0	0	0
rr_cycle_switching_50	0	1	0	0
explore_intensify_0.1	0	0	1	0
explore_intensify_0.2	0	1	1	0
explore_intensify_0.3	0	0	1	0
explore_intensify_0.4	0	0	1	0
explore_intensify_0.5	0	0	0	0
explore_intensify_0.6	0	0	0	0
explore_intensify_0.7	0	0	0	0
explore_intensify_0.9	0	0	0	0

Table B.23 Win/Tie/Loss counts for **"explore_intensify_0.9"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0/20/0	0/20/0	0/20/0	9/0/11
uncertainty	14/0/6	9/0/11	10/0/10	8/1/11
random	7/0/13	9/0/11	8/0/12	11/0/9
rr_cycle_switching_50	10/0/10	11/0/9	9/0/11	5/0/15
explore_intensify_0.1	7/0/13	10/0/10	11/0/9	8/2/10
explore_intensify_0.2	6/0/14	12/0/8	13/0/7	7/1/12
explore_intensify_0.3	0/20/0	8/0/12	12/0/8	8/1/11
explore_intensify_0.4	0/20/0	7/0/13	12/0/8	8/0/12
explore_intensify_0.5	0/20/0	0/20/0	9/0/11	9/4/7
explore_intensify_0.6	0/20/0	0/20/0	6/0/14	8/2/10
explore_intensify_0.7	0/20/0	0/20/0	10/0/10	8/3/9
explore_intensify_0.8	0/20/0	0/20/0	0/20/0	10/2/8

Table B.24 Binary performance for **"explore_intensify_0.9"** compared against other strategies in split strategy **"stratified"** at specified percentages of labeled training data

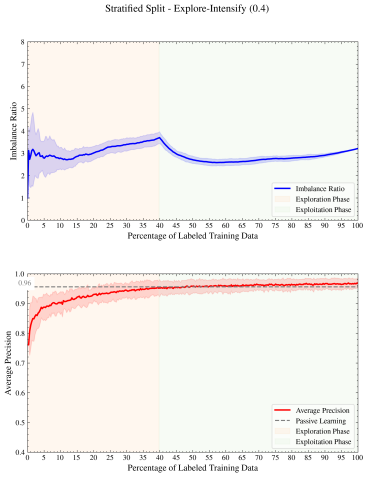
Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	0	0
uncertainty	1	0	0	0
random	0	0	0	1
rr_cycle_switching_50	0	1	0	0
explore_intensify_0.1	0	0	1	0
explore_intensify_0.2	0	1	1	0
explore_intensify_0.3	0	0	1	0
explore_intensify_0.4	0	0	1	0
explore_intensify_0.5	0	0	0	0
explore_intensify_0.6	0	0	0	0
explore_intensify_0.7	0	0	0	0
explore_intensify_0.8	0	0	0	0



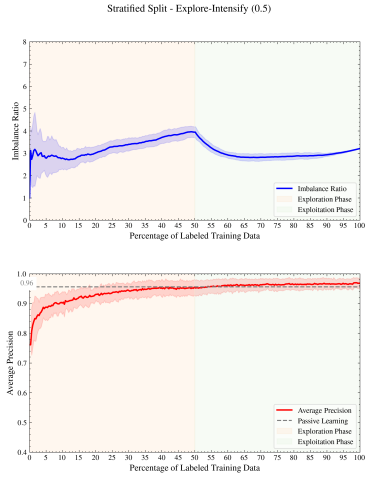
(a) Explore Intensify 0.1

(b) Explore Intensify 0.2

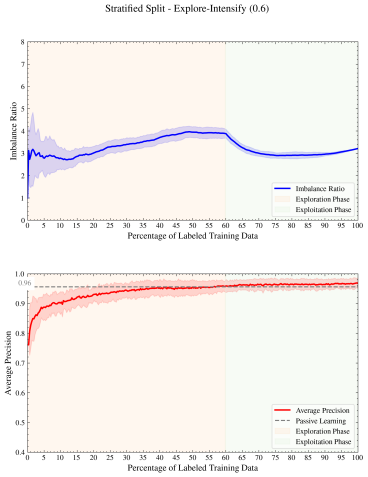
(c) Explore Intensify 0.3



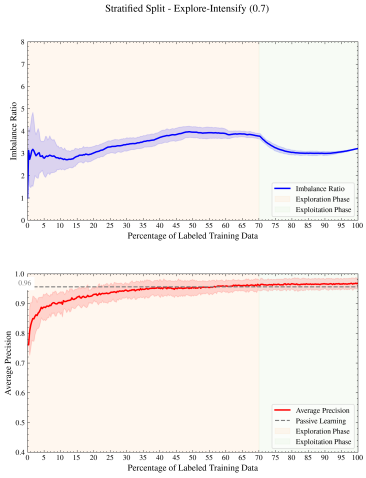
(d) Explore Intensify 0.4



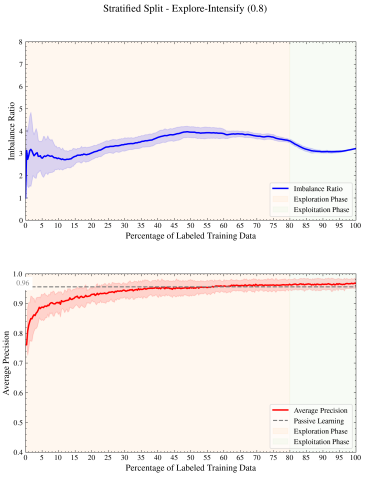
(e) Explore Intensify 0.5



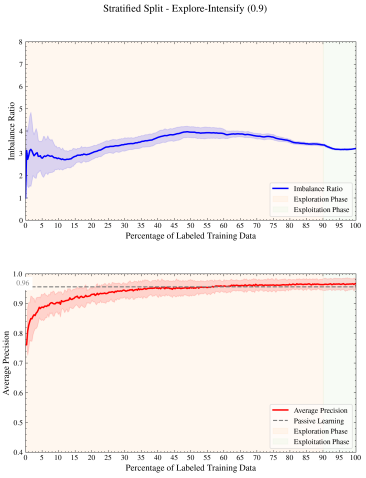
(f) Explore Intensify 0.6



(g) Explore Intensify 0.7

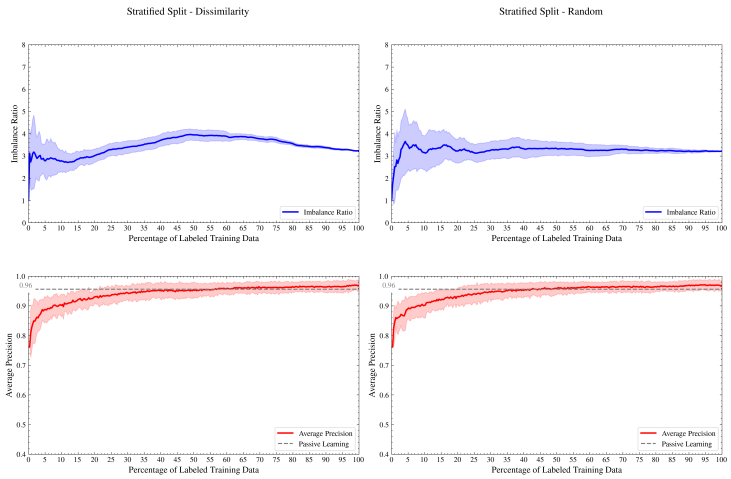


(h) Explore Intensify 0.8

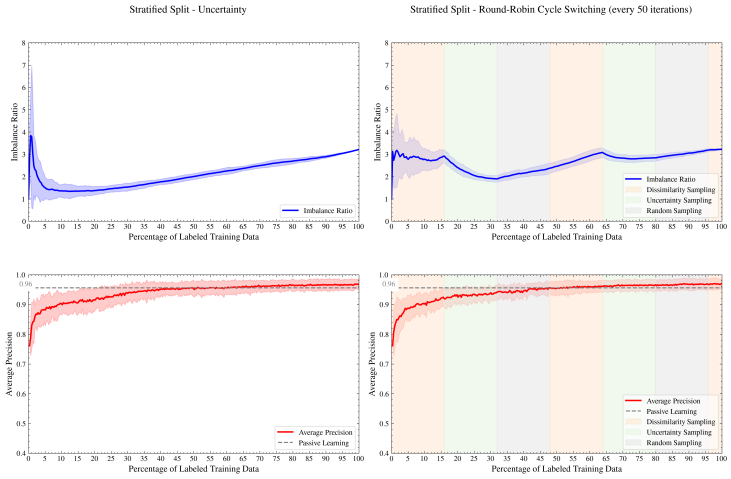


(i) Explore Intensify 0.9

Figure B.1 Explore Intensify strategies (Stratified-split)



(a) Dissimilarity Sampling (b) Random Sampling



(c) Uncertainty Sampling (d) RR Cycle Switching

Figure B.2 Random, Uncertainty, Dissimilarity sampling and RR Cycle Switching (Stratified-split)

B.2 Scaffold-splitting

Table B.25 Win/Tie/Loss counts for **"dissimilarity"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
uncertainty	14/0/6	15/0/5	9/0/11	17/0/3
random	7/0/13	10/0/10	5/0/15	1/0/19
rr_cycle_switching_50	12/0/8	13/0/7	7/0/13	14/0/6
explore_intensify_0.1	14/0/6	19/0/1	10/0/10	16/0/4
explore_intensify_0.2	10/0/10	14/0/6	13/0/7	15/0/5
explore_intensify_0.3	0/20/0	15/0/5	9/0/11	14/0/6
explore_intensify_0.4	0/20/0	18/0/2	10/0/10	19/0/1
explore_intensify_0.5	0/20/0	0/20/0	8/0/12	18/0/2
explore_intensify_0.6	0/20/0	0/20/0	11/0/9	12/0/8
explore_intensify_0.7	0/20/0	0/20/0	7/0/13	15/0/5
explore_intensify_0.8	0/20/0	0/20/0	0/20/0	15/0/5
explore_intensify_0.9	0/20/0	0/20/0	0/20/0	4/0/16

Table B.26 Binary performance for **"dissimilarity"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
uncertainty	1	1	0	1
random	0	0	0	0
rr_cycle_switching_50	1	1	0	1
explore_intensify_0.1	1	1	0	1
explore_intensify_0.2	0	1	1	1
explore_intensify_0.3	0	1	0	1
explore_intensify_0.4	0	1	0	1
explore_intensify_0.5	0	0	0	1
explore_intensify_0.6	0	0	1	1
explore_intensify_0.7	0	0	0	1
explore_intensify_0.8	0	0	0	1
explore_intensify_0.9	0	0	0	0

Table B.27 Win/Tie/Loss counts for **"uncertainty"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	6/0/14	5/0/15	11/0/9	3/0/17
random	5/0/15	7/0/13	5/0/15	1/0/19
rr_cycle_switching_50	10/0/10	8/0/12	4/0/16	11/0/9
explore_intensify_0.1	9/0/11	11/0/9	8/0/12	7/4/9
explore_intensify_0.2	10/0/10	10/0/10	9/0/11	8/4/8
explore_intensify_0.3	6/0/14	9/0/11	7/0/13	8/2/10
explore_intensify_0.4	6/0/14	9/0/11	7/0/13	10/2/8
explore_intensify_0.5	6/0/14	5/0/15	4/0/16	13/2/5
explore_intensify_0.6	6/0/14	5/0/15	6/0/14	10/0/10
explore_intensify_0.7	6/0/14	5/0/15	5/0/15	14/0/6
explore_intensify_0.8	6/0/14	5/0/15	11/0/9	12/0/8
explore_intensify_0.9	6/0/14	5/0/15	11/0/9	4/0/16

Table B.28 Binary performance for **"uncertainty"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	1	0
random	0	0	0	0
rr_cycle_switching_50	0	0	0	1
explore_intensify_0.1	0	1	0	0
explore_intensify_0.2	0	0	0	0
explore_intensify_0.3	0	0	0	0
explore_intensify_0.4	0	0	0	0
explore_intensify_0.5	0	0	0	1
explore_intensify_0.6	0	0	0	0
explore_intensify_0.7	0	0	0	1
explore_intensify_0.8	0	0	1	1
explore_intensify_0.9	0	0	1	0

Table B.29 Win/Tie/Loss counts for **"random"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	13/0/7	10/0/10	15/0/5	19/0/1
uncertainty	15/0/5	13/0/7	15/0/5	19/0/1
rr_cycle_switching_50	16/0/4	15/0/5	15/0/5	19/0/1
explore_intensify_0.1	16/0/4	17/0/3	16/0/4	18/0/2
explore_intensify_0.2	13/0/7	15/0/5	16/0/4	16/0/4
explore_intensify_0.3	13/0/7	12/0/8	16/0/4	16/0/4
explore_intensify_0.4	13/0/7	18/0/2	12/0/8	19/0/1
explore_intensify_0.5	13/0/7	10/0/10	15/0/5	20/0/0
explore_intensify_0.6	13/0/7	10/0/10	15/0/5	15/0/5
explore_intensify_0.7	13/0/7	10/0/10	15/0/5	19/0/1
explore_intensify_0.8	13/0/7	10/0/10	15/0/5	20/0/0
explore_intensify_0.9	13/0/7	10/0/10	15/0/5	11/0/9

Table B.30 Binary performance for **"random"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	1	0	1	1
uncertainty	1	1	1	1
rr_cycle_switching_50	1	1	1	1
explore_intensify_0.1	1	1	1	1
explore_intensify_0.2	1	1	1	1
explore_intensify_0.3	1	1	1	1
explore_intensify_0.4	1	1	1	1
explore_intensify_0.5	1	0	1	1
explore_intensify_0.6	1	0	1	1
explore_intensify_0.7	1	0	1	1
explore_intensify_0.8	1	0	1	1
explore_intensify_0.9	1	0	1	1

Table B.31 Win/Tie/Loss counts for **"explore_intensify_0.1"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	6/0/14	1/0/19	10/0/10	4/0/16
uncertainty	11/0/9	9/0/11	12/0/8	9/4/7
random	4/0/16	3/0/17	4/0/16	2/0/18
rr_cycle_switching_50	9/0/11	8/0/12	7/0/13	12/0/8
explore_intensify_0.2	8/0/12	8/0/12	13/0/7	7/4/9
explore_intensify_0.3	6/0/14	6/0/14	9/0/11	11/0/9
explore_intensify_0.4	6/0/14	11/0/9	7/0/13	13/1/6
explore_intensify_0.5	6/0/14	1/0/19	9/0/11	12/3/5
explore_intensify_0.6	6/0/14	1/0/19	7/0/13	11/0/9
explore_intensify_0.7	6/0/14	1/0/19	7/0/13	14/0/6
explore_intensify_0.8	6/0/14	1/0/19	10/0/10	13/0/7
explore_intensify_0.9	6/0/14	1/0/19	10/0/10	5/0/15

Table B.32 Binary performance for **"explore_intensify_0.1"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	0	0
uncertainty	1	0	1	0
random	0	0	0	0
rr_cycle_switching_50	0	0	0	1
explore_intensify_0.2	0	0	1	0
explore_intensify_0.3	0	0	0	1
explore_intensify_0.4	0	1	0	1
explore_intensify_0.5	0	0	0	1
explore_intensify_0.6	0	0	0	1
explore_intensify_0.7	0	0	0	1
explore_intensify_0.8	0	0	0	1
explore_intensify_0.9	0	0	0	0

Table B.33 Win/Tie/Loss counts for **"explore_intensify_0.2"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	10/0/10	6/0/14	7/0/13	5/0/15
uncertainty	10/0/10	10/0/10	11/0/9	8/4/8
random	7/0/13	5/0/15	4/0/16	4/0/16
rr_cycle_switching_50	11/0/9	9/0/11	5/0/15	13/0/7
explore_intensify_0.1	12/0/8	12/0/8	7/0/13	9/4/7
explore_intensify_0.3	10/0/10	9/0/11	6/0/14	9/2/9
explore_intensify_0.4	10/0/10	12/0/8	5/0/15	12/1/7
explore_intensify_0.5	10/0/10	6/0/14	5/0/15	11/3/6
explore_intensify_0.6	10/0/10	6/0/14	8/0/12	10/2/8
explore_intensify_0.7	10/0/10	6/0/14	6/0/14	11/1/8
explore_intensify_0.8	10/0/10	6/0/14	7/0/13	10/0/10
explore_intensify_0.9	10/0/10	6/0/14	7/0/13	6/0/14

Table B.34 Binary performance for **"explore_intensify_0.2"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	0	0
uncertainty	0	0	1	0
random	0	0	0	0
rr_cycle_switching_50	1	0	0	1
explore_intensify_0.1	1	1	0	0
explore_intensify_0.3	0	0	0	0
explore_intensify_0.4	0	1	0	1
explore_intensify_0.5	0	0	0	1
explore_intensify_0.6	0	0	0	0
explore_intensify_0.7	0	0	0	1
explore_intensify_0.8	0	0	0	0
explore_intensify_0.9	0	0	0	0

Table B.35 Win/Tie/Loss counts for **"explore_intensify_0.3"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0/20/0	5/0/15	11/0/9	6/0/14
uncertainty	14/0/6	11/0/9	13/0/7	10/2/8
random	7/0/13	8/0/12	4/0/16	4/0/16
rr_cycle_switching_50	12/0/8	9/0/11	8/0/12	15/0/5
explore_intensify_0.1	14/0/6	14/0/6	11/0/9	9/0/11
explore_intensify_0.2	10/0/10	11/0/9	14/0/6	9/2/9
explore_intensify_0.4	0/20/0	14/0/6	11/0/9	11/4/5
explore_intensify_0.5	0/20/0	5/0/15	10/0/10	14/0/6
explore_intensify_0.6	0/20/0	5/0/15	9/0/11	12/0/8
explore_intensify_0.7	0/20/0	5/0/15	7/0/13	15/0/5
explore_intensify_0.8	0/20/0	5/0/15	11/0/9	14/0/6
explore_intensify_0.9	0/20/0	5/0/15	11/0/9	8/0/12

Table B.36 Binary performance for **"explore_intensify_0.3"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	1	0
uncertainty	1	1	1	0
random	0	0	0	0
rr_cycle_switching_50	1	0	0	1
explore_intensify_0.1	1	1	1	0
explore_intensify_0.2	0	1	1	0
explore_intensify_0.4	0	1	1	1
explore_intensify_0.5	0	0	0	1
explore_intensify_0.6	0	0	0	1
explore_intensify_0.7	0	0	0	1
explore_intensify_0.8	0	0	1	1
explore_intensify_0.9	0	0	1	0

Table B.37 Win/Tie/Loss counts for **"explore_intensify_0.4"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0/20/0	2/0/18	10/0/10	1/0/19
uncertainty	14/0/6	11/0/9	13/0/7	8/2/10
random	7/0/13	2/0/18	8/0/12	1/0/19
rr_cycle_switching_50	12/0/8	6/0/14	8/0/12	10/0/10
explore_intensify_0.1	14/0/6	9/0/11	13/0/7	6/1/13
explore_intensify_0.2	10/0/10	8/0/12	15/0/5	7/1/12
explore_intensify_0.3	0/20/0	6/0/14	9/0/11	5/4/11
explore_intensify_0.5	0/20/0	2/0/18	9/0/11	13/0/7
explore_intensify_0.6	0/20/0	2/0/18	12/0/8	6/1/13
explore_intensify_0.7	0/20/0	2/0/18	9/0/11	10/1/9
explore_intensify_0.8	0/20/0	2/0/18	10/0/10	9/0/11
explore_intensify_0.9	0/20/0	2/0/18	10/0/10	2/0/18

Table B.38 Binary performance for **"explore_intensify_0.4"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	0	0
uncertainty	1	1	1	0
random	0	0	0	0
rr_cycle_switching_50	1	0	0	0
explore_intensify_0.1	1	0	1	0
explore_intensify_0.2	0	0	1	0
explore_intensify_0.3	0	0	0	0
explore_intensify_0.5	0	0	0	1
explore_intensify_0.6	0	0	1	0
explore_intensify_0.7	0	0	0	0
explore_intensify_0.8	0	0	0	0
explore_intensify_0.9	0	0	0	0

Table B.39 Win/Tie/Loss counts for **"explore_intensify_0.5"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0/20/0	0/20/0	12/0/8	2/0/18
uncertainty	14/0/6	15/0/5	16/0/4	5/2/13
random	7/0/13	10/0/10	5/0/15	0/0/20
rr_cycle_switching_50	12/0/8	13/0/7	14/0/6	10/0/10
explore_intensify_0.1	14/0/6	19/0/1	11/0/9	5/3/12
explore_intensify_0.2	10/0/10	14/0/6	15/0/5	6/3/11
explore_intensify_0.3	0/20/0	15/0/5	10/0/10	6/0/14
explore_intensify_0.4	0/20/0	18/0/2	11/0/9	7/0/13
explore_intensify_0.6	0/20/0	0/20/0	10/0/10	5/1/14
explore_intensify_0.7	0/20/0	0/20/0	11/0/9	7/2/11
explore_intensify_0.8	0/20/0	0/20/0	12/0/8	7/1/12
explore_intensify_0.9	0/20/0	0/20/0	12/0/8	4/0/16

Table B.40 Binary performance for **"explore_intensify_0.5"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	1	0
uncertainty	1	1	1	0
random	0	0	0	0
rr_cycle_switching_50	1	1	1	0
explore_intensify_0.1	1	1	1	0
explore_intensify_0.2	0	1	1	0
explore_intensify_0.3	0	1	0	0
explore_intensify_0.4	0	1	1	0
explore_intensify_0.6	0	0	0	0
explore_intensify_0.7	0	0	1	0
explore_intensify_0.8	0	0	1	0
explore_intensify_0.9	0	0	1	0

Table B.41 Win/Tie/Loss counts for **"explore_intensify_0.6"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0/20/0	0/20/0	9/0/11	8/0/12
uncertainty	14/0/6	15/0/5	14/0/6	10/0/10
random	7/0/13	10/0/10	5/0/15	5/0/15
rr_cycle_switching_50	12/0/8	13/0/7	9/0/11	11/0/9
explore_intensify_0.1	14/0/6	19/0/1	13/0/7	9/0/11
explore_intensify_0.2	10/0/10	14/0/6	12/0/8	8/2/10
explore_intensify_0.3	0/20/0	15/0/5	11/0/9	8/0/12
explore_intensify_0.4	0/20/0	18/0/2	8/0/12	13/1/6
explore_intensify_0.5	0/20/0	0/20/0	10/0/10	14/1/5
explore_intensify_0.7	0/20/0	0/20/0	9/0/11	10/1/9
explore_intensify_0.8	0/20/0	0/20/0	9/0/11	12/0/8
explore_intensify_0.9	0/20/0	0/20/0	9/0/11	3/3/14

Table B.42 Binary performance for **"explore_intensify_0.6"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	0	0
uncertainty	1	1	1	0
random	0	0	0	0
rr_cycle_switching_50	1	1	0	1
explore_intensify_0.1	1	1	1	0
explore_intensify_0.2	0	1	1	0
explore_intensify_0.3	0	1	1	0
explore_intensify_0.4	0	1	0	1
explore_intensify_0.5	0	0	0	1
explore_intensify_0.7	0	0	0	0
explore_intensify_0.8	0	0	0	1
explore_intensify_0.9	0	0	0	0

Table B.43 Win/Tie/Loss counts for **"explore_intensify_0.7"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0/20/0	0/20/0	13/0/7	5/0/15
uncertainty	14/0/6	15/0/5	15/0/5	6/0/14
random	7/0/13	10/0/10	5/0/15	1/0/19
rr_cycle_switching_50	12/0/8	13/0/7	10/0/10	8/0/12
explore_intensify_0.1	14/0/6	19/0/1	13/0/7	6/0/14
explore_intensify_0.2	10/0/10	14/0/6	14/0/6	8/1/11
explore_intensify_0.3	0/20/0	15/0/5	13/0/7	5/0/15
explore_intensify_0.4	0/20/0	18/0/2	11/0/9	9/1/10
explore_intensify_0.5	0/20/0	0/20/0	9/0/11	11/2/7
explore_intensify_0.6	0/20/0	0/20/0	11/0/9	9/1/10
explore_intensify_0.8	0/20/0	0/20/0	13/0/7	6/1/13
explore_intensify_0.9	0/20/0	0/20/0	13/0/7	5/0/15

Table B.44 Binary performance for **"explore_intensify_0.7"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	1	0
uncertainty	1	1	1	0
random	0	0	0	0
rr_cycle_switching_50	1	1	0	0
explore_intensify_0.1	1	1	1	0
explore_intensify_0.2	0	1	1	0
explore_intensify_0.3	0	1	1	0
explore_intensify_0.4	0	1	1	0
explore_intensify_0.5	0	0	0	1
explore_intensify_0.6	0	0	1	0
explore_intensify_0.8	0	0	1	0
explore_intensify_0.9	0	0	1	0

Table B.45 Win/Tie/Loss counts for **"explore_intensify_0.8"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0/20/0	0/20/0	0/20/0	5/0/15
uncertainty	14/0/6	15/0/5	9/0/11	8/0/12
random	7/0/13	10/0/10	5/0/15	0/0/20
rr_cycle_switching_50	12/0/8	13/0/7	7/0/13	12/0/8
explore_intensify_0.1	14/0/6	19/0/1	10/0/10	7/0/13
explore_intensify_0.2	10/0/10	14/0/6	13/0/7	10/0/10
explore_intensify_0.3	0/20/0	15/0/5	9/0/11	6/0/14
explore_intensify_0.4	0/20/0	18/0/2	10/0/10	11/0/9
explore_intensify_0.5	0/20/0	0/20/0	8/0/12	12/1/7
explore_intensify_0.6	0/20/0	0/20/0	11/0/9	8/0/12
explore_intensify_0.7	0/20/0	0/20/0	7/0/13	13/1/6
explore_intensify_0.9	0/20/0	0/20/0	0/20/0	3/0/17

Table B.46 Binary performance for **"explore_intensify_0.8"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	0	0
uncertainty	1	1	0	0
random	0	0	0	0
rr_cycle_switching_50	1	1	0	1
explore_intensify_0.1	1	1	0	0
explore_intensify_0.2	0	1	1	0
explore_intensify_0.3	0	1	0	0
explore_intensify_0.4	0	1	0	1
explore_intensify_0.5	0	0	0	1
explore_intensify_0.6	0	0	1	0
explore_intensify_0.7	0	0	0	1
explore_intensify_0.9	0	0	0	0

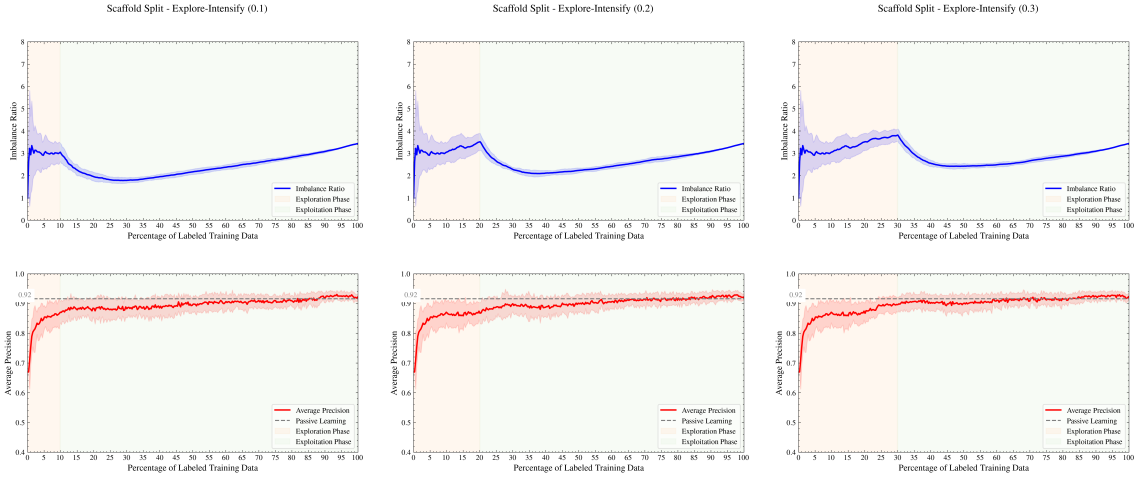
Table B.47 Win/Tie/Loss counts for **"explore_intensify_0.9"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0/20/0	0/20/0	0/20/0	16/0/4
uncertainty	14/0/6	15/0/5	9/0/11	16/0/4
random	7/0/13	10/0/10	5/0/15	9/0/11
rr_cycle_switching_50	12/0/8	13/0/7	7/0/13	19/0/1
explore_intensify_0.1	14/0/6	19/0/1	10/0/10	15/0/5
explore_intensify_0.2	10/0/10	14/0/6	13/0/7	14/0/6
explore_intensify_0.3	0/20/0	15/0/5	9/0/11	12/0/8
explore_intensify_0.4	0/20/0	18/0/2	10/0/10	18/0/2
explore_intensify_0.5	0/20/0	0/20/0	8/0/12	16/0/4
explore_intensify_0.6	0/20/0	0/20/0	11/0/9	14/3/3
explore_intensify_0.7	0/20/0	0/20/0	7/0/13	15/0/5
explore_intensify_0.8	0/20/0	0/20/0	0/20/0	17/0/3

Table B.48 Binary performance for **"explore_intensify_0.9"** compared against other strategies in split strategy **"scaffold"** at specified percentages of labeled training data

Compared Against	**25%**	**50%**	**75%**	**100%**
dissimilarity	0	0	0	1
uncertainty	1	1	0	1
random	0	0	0	0
rr_cycle_switching_50	1	1	0	1
explore_intensify_0.1	1	1	0	1
explore_intensify_0.2	0	1	1	1
explore_intensify_0.3	0	1	0	1
explore_intensify_0.4	0	1	0	1
explore_intensify_0.5	0	0	0	1
explore_intensify_0.6	0	0	1	1
explore_intensify_0.7	0	0	0	1
explore_intensify_0.8	0	0	0	1

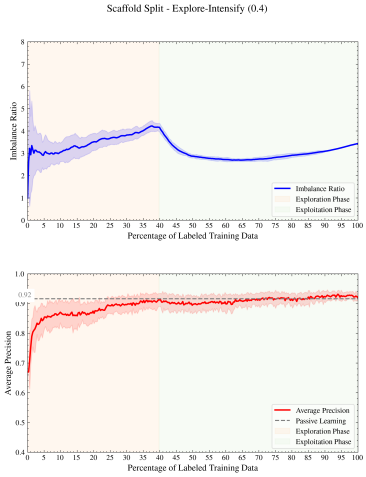
Next, we show the performance figures for all the active learning strategies in the scaffold-split setup.



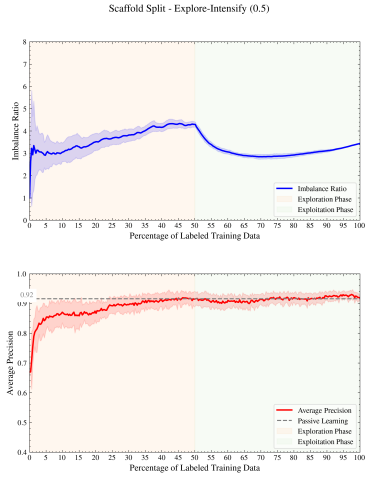
(a) Explore-Intensify 0.1

(b) Explore-Intensify 0.2

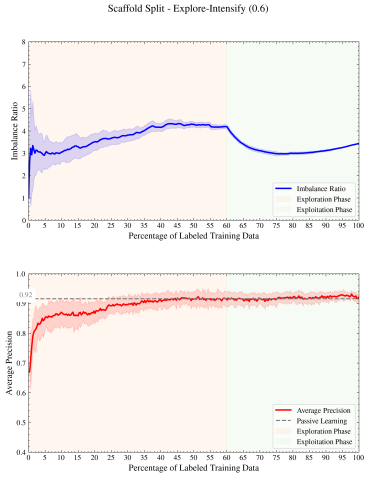
(c) Explore-Intensify 0.3



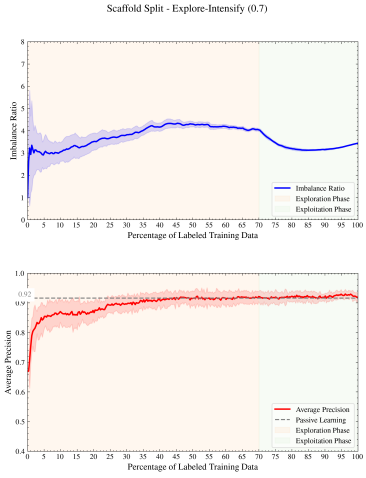
(d) Explore-Intensify 0.4



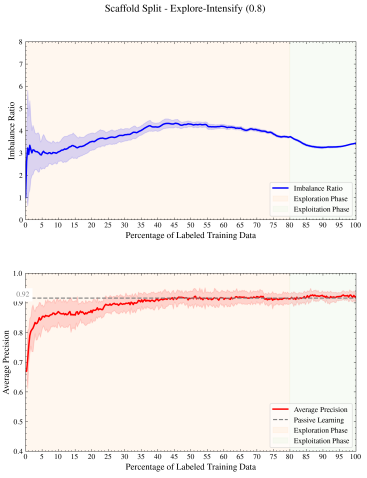
(e) Explore-Intensify 0.5



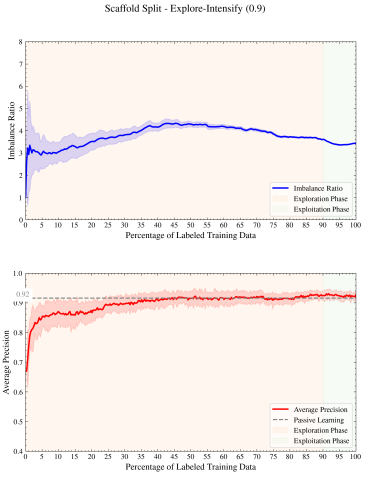
(f) Explore-Intensify 0.6



(g) Explore-Intensify 0.7

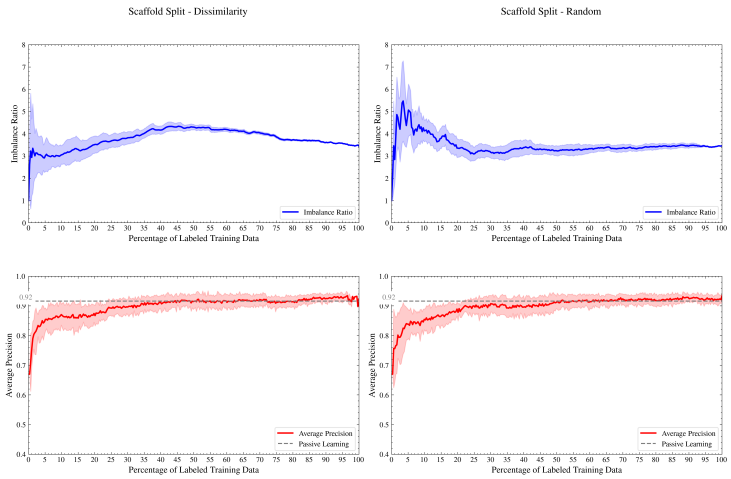


(h) Explore-Intensify 0.8

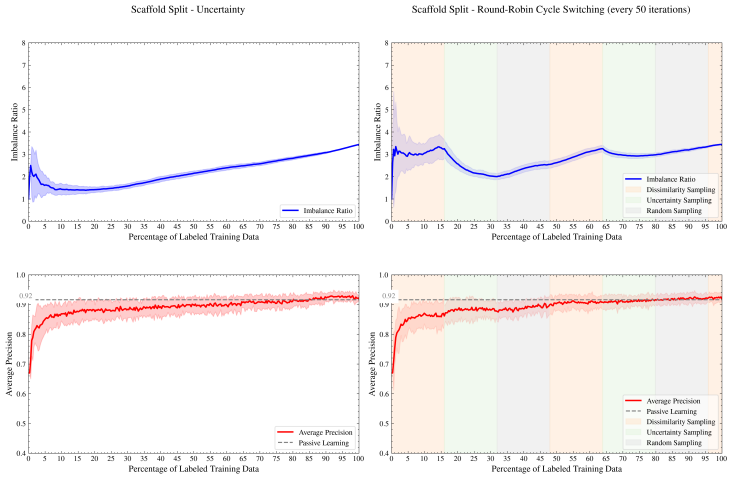


(i) Explore-Intensify 0.9

Figure B.3 Explore Intensify strategies (Scaffold-split)



(a) Dissimilarity Sampling (b) Random Sampling



(c) Uncertainty Sampling (d) RR Cycle Switching

Figure B.4 Random, Uncertainty, Dissimilarity sampling and RR Cycle Switching (Scaffold-split)