# Improving Small Molecules Activity Modelling Capability of Cell Painting Data through Data Augmentation and Effective Representation Learning

by

Son V. Ha

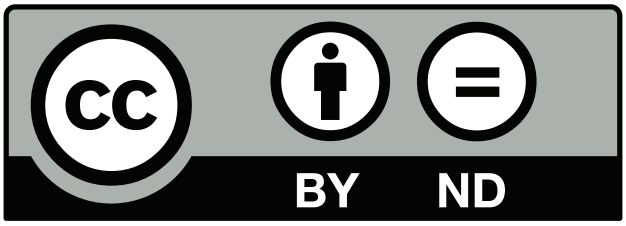
A Dissertation Presented to the

FACULTY OF THE GRADUATE SCHOOL

DEPARTMENT OF CHEMISTRY

JOHANNES GUTENBERG UNIVERSITY MAINZ

September 2024



# Table of Contents

# List of Tables

# List of Figures

# Abstract

This thesis focuses on improving image-based activity modeling, for early-stage drug discovery through data augmentation and representation learning of Cell Painting data. Firstly, a significant contribution is the introduction of the FSL-CP dataset, designed to support few-shot learning (FSL) benchmarking of small-molecule bioactivity prediction using cell microscopy images. This unique dataset simulates scenarios where only limited data is available to predict the activity of compounds in a range of bioassays. Through this dataset we compared several FSL paradigms (singletask, multitask, meta-learning) in a low-data context and study the effectiveness of transfer learning. Notably, prototypical networks and multitask neural networks pre-trained on auxiliary tasks consistently performed well. In addition, meta-learning methods saw diminishing returns with larger support sets, and traditional single-task models showed a marked improvement as support sets expand.

Additionally, this work proposes an application for underused 'low concentration images' in activity modeling. Typically, models are trained on images stained by an optimized protocol (e.g. Cell Painting) after exposure to a fairly high small molecule concentration (referred to as 'image concentration') of $10\mu M$ or higher. Low concentration images (e.g. $0.16\mu M$, $0.8\mu M$, $4\mu M$) tend to yield models with worse performance. In this work, we nevertheless report a practical use for low image concentration data. We propose the combination of well-performing models trained at higher image concentrations, with lower image concentration for inference to identify more potent compounds. We show that this approach improves on the conventional method (directly training a high-potency model) in 65% of assays investigated in terms of AUC-ROC, and 75% of assays in terms of RIPtoP-corrected AUC-PR.

The thesis further investigates cross-modality representation learning of cell painting (CP) and transcriptomics (TX), which are powerful tools in early drug discovery to gain

understanding of the biological effect of compounds on a population of cells post-treatment. While multimodal learning of chemical structure-cell painting, or different omics data has been experimented; a cell painting-bulk transcriptomics multimodal model is still unexplored. In this work, we benchmark two representation learning methods: contrastive learning and bimodal autoencoder. We use the setting of cross modality learning where representation learning is performed with two modalities (CP and TX), but only cell painting is available for new compounds embedding generation and downstream task. This is because for new compounds, we would only have CP data and not TX, due to high data generation cost of the RNA-Seq screen. We show that learned representation improves cluster quality for clustering of CP replicates and different modes of action (MoA). In the supervised bioactivity multitask classification, we demonstrate that CL embedding achieves higher mean AUROC and RIPtoP-AUPRC compared to CP feature across a range of bioactivity tasks. Additionally, we provide a more detailed comparison of feature performance on bioactivity tasks grouped by protein target families. Finally, we show that in the absence of TX features for new compounds, using learned embeddings enhances performance of CP feature on tasks where TX feature excels but CP feature does not.

# Abstract 2

Diese Promotionsarbeit befasst sich mit der Verbesserung des ‘image-based activity modeling’ für die Arzneimittelentdeckung im Frühstadium durch Datenaugmentierung und Repräsentationslernen auf Basis von Cell-Painting-Daten. Ein wichtiger Beitrag ist die Einführung des FSL-CP-Datensatzes, der zur Unterstützung des Benchmarkings von ´Few-shot learning’ (FSL) zur Vorhersage der Bioaktivität kleiner Moleküle anhand von Zellmikroskopie-Bildern entwickelt wurde. Dieser einzigartige Datensatz simuliert Szenarien, in denen nur begrenzte Daten zur Bioaktivitätsvorhersage der Moleküle in einer Reihe von Bioassays zur Verfügung stehen. Anhand dieses Datensatzes haben wir verschiedene FSL-Paradigmen (Singletask, Multitask, Meta-Learning) in einem datenarmen Kontext verglichen und die Wirksamkeit des Transferlernens untersucht. Vor allem ‘Prototypical Network’ und multi-task neural networks, die auf ’Auxiliary Tasks’ vortrainiert wurden, zeigten durchweg gute Leistungen. Die Verbesserung die von Meta-Learning-Methoden erzielt wurde, verringerte sich bei großen ‘Support Sets’. Andererseits zeigten traditionelle Singletask-Modelle eine deutliche Verbesserung mit zunehmender Größe der ‘Support Sets’.

Zusätzlich wird in dieser Promotionsarbeit eine Anwendung für ‘Bilder mit niedriger Konzentration’ die bisher bei der Aktivitätsmodellierung nicht gebräuchlich sind, vorgeschlagen. Normalerweise werden Modelle auf Bildern trainiert, die mit einem optimierten Protokoll (z. B. Cell Painting) gefärbt wurden, nachdem sie einer relativ hohen Konzentration kleiner Moleküle (als ‘Bildkonzentration’ bezeichnet) von $10\mu M$ oder mehr ausgesetzt wurden. Bilder mit niedriger Konzentration (z. B. $0,16\mu M$, $0,8\mu M$, $4\mu M$) führen tendenziell zu Modellen mit schlechterer Leistung. In dieser Arbeit berichten wir jedoch über eine praktische Anwendung für Bilder mit niedriger Konzentration. Wir schlagen vor, gut funktionierende Modelle, die bei höheren Bildkonzentrationen trainiert wurden, mit niedrigeren Bildkonzentrationen für die Inferenz zu kombinieren, um potentere Molekürе zu identifizieren.

Wir zeigen, dass dieser Ansatz die konventionelle Methode (direktes Training eines hochpotenten Modells) in 65% der untersuchten Assays in Bezug auf die AUC-ROC und 75% der Assays in Bezug auf die RIPtoP-korrigierte AUC-PR verbessert.

Schließlich untersucht die Dissertation das 'cross-modality' Repräsentationslernen von Cell Painting (CP) und Transkriptomik (TX), welche leistungsstarke Werkzeuge in der frühen Arzneimittelforschung sind, um die biologische Wirkung von Moleküre auf eine Zellpopulation nach der Behandlung zu verstehen. Während mit multimodale Lernen von chemischen Struktur-Cell Painting, oder verschiedene omics Daten bereits experimentiert wird; sind Cell Painting-Bulk Transcriptomics multimodale Modelle noch weitgehendst unerforscht. In dieser Arbeit vergleichen wir zwei Repräsentationslernmethoden: 'Contrastive Learning' (CL) und 'Bimodal Autoencoder' (BAE). Wir verwenden die Variante des cross-modality Lernens, bei dem das Repräsentationslernen mit zwei Modalitäten (CP und TX) durchgeführt wird, aber nur die CP für die Einbettung neuer Moleküre und die 'Downstream-Tasks' verfügbar ist. Der Grund dafür ist, dass wir für neue Moleküre nur CP-Daten und keine TX-Daten haben, was auf die hohen Kosten der Datengenerierung beim RNA-Seq-Screening zurückzuführen ist. Wir zeigen, dass die erlernte Repräsentation die Clusterqualität für das Clustering von CP-Replikate und verschiedenen 'Mode-of-Actions' (MoA) verbessert. Bei der Supervised-Bioaktivitäts-Multitask-Klassifizierung zeigen wir, dass die CL-Einbettung im Vergleich zu CP bei einer Reihe von Bioaktivitätsaufgaben einen höheren mittleren AUROC und RIPtoP-AUPRC erzielt. Außerdem vergleichen wir detailliert die Merkmalsleistungen bei Bioaktivitätsaufgaben, gruppiert nach Protein-Zielfamilien. Schließlich zeigen wir, dass die Verwendung der gelernten Einbettungen, mangels TX-Daten für neue Moleküle, die Leistung von CP-Daten bei 'Downstream-Tasks' verbessert, bei denen TX-Daten individuell hervorragende Ergebnisse erzielen, CP-Daten jedoch nicht.

# Abstract (Short)

This thesis focuses on improving image-based activity modeling, for early-stage drug discovery through data augmentation and representation learning of Cell Painting data. Firstly, a significant contribution is the introduction of the FSL-CP dataset, designed to support few-shot learning (FSL) benchmarking of small-molecule bioactivity prediction using cell microscopy images. Through this dataset we compared several FSL paradigms in a low-data context and study the effectiveness of transfer learning.

Additionally, this work proposes an application for underused 'low concentration images' in activity modeling. We propose the combination of well-performing models trained at higher image concentrations, with lower image concentration for inference to identify more potent compounds. We show that this approach improves on the conventional method (directly training a high-potency model) in 65% of assays investigated in terms of AUC-ROC, and 75% of assays in terms of RIPtoP-corrected AUC-PR.

The thesis further investigates cross-modality representation learning of cell painting (CP) and transcriptomics (TX), which are powerful tools in early drug discovery to gain understanding of the biological effect of compounds on a population of cells post-treatment. In this work, we benchmark two representation learning methods: contrastive learning and bimodal autoencoder. We use the setting of cross modality learning where representation learning is performed with two modalities (CP and TX), but only cell painting is available for new compounds embedding generation and downstream task. This is because for new compounds, we would only have CP data and not TX, due to high data generation cost of the RNA-Seq screen. We show that learned representation improves cluster quality for clustering of CP replicates and different modes of action (MoA).

# Abstract 2 (Short)

Diese Promotionsarbeit befasst sich mit der Verbesserung des ‘image-based activity modeling’ für die Arzneimittelentdeckung im Frühstadium durch Datenaugmentierung und Repräsentationslernen auf Basis von Cell-Painting-Daten. Ein wichtiger Beitrag ist die Einführung des FSL-CP-Datensatzes, der zur Unterstützung des Benchmarkings von ´Few-shot learning’ (FSL) zur Vorhersage der Bioaktivität kleiner Moleküle anhand von Zellmikroskopie-Bildern entwickelt wurde. Anhand dieses Datensatzes haben wir verschiedene FSL-Paradigmen in einem datenarmen Kontext verglichen und die Wirksamkeit des Transferlernens untersucht.

Zusätzlich wird in dieser Promotionsarbeit eine Anwendung für ‘Bilder mit niedriger Konzentration’ die bisher bei der Aktivitätsmodellierung nicht gebräuchlich sind, vorgeschlagen. Wir schlagen vor, gut funktionierende Modelle, die bei höheren Bildkonzentrationen trainiert wurden, mit niedrigeren Bildkonzentrationen für die Inferenz zu kombinieren, um potentere Moleküre zu identifizieren.

Schließlich untersucht die Dissertation das ‘cross-modality’ Repräsentationslernen von Cell Painting (CP) und Transkriptomik (TX), welche leistungsstarke Werkzeuge in der frühen Arzneimittelforschung sind, um die biologische Wirkung von Moleküre auf eine Zellpopulation nach der Behandlung zu verstehen. In dieser Arbeit vergleichen wir zwei Repräsentationslernmethoden: ‘Contrastive Learning’ (CL) und ‘Bimodal Autoencoder’ (BAE). Wir verwenden die Variante des cross-modality Lernens, bei dem das Repräsentationslernen mit zwei Modalitäten (CP und TX) durchgeführt wird, aber nur die CP für die Einbettung neuer Moleküre und die ‘Downstream-Tasks’ verfügbar ist. Wir zeigen, dass die erlernte Repräsentation die Clusterqualität für das Clustering von CP-Replikate und verschiedenen ‘Mode-of-Actions’ (MoA) verbessert.

# Chapter 1

# Introduction

## 1.1 Cell Painting Protocol

High-throughput Imaging (HTI) had been a powerful tool in drug discovery, contributing to many biological discoveries[1, 2]. It involved capturing the morphological changes of the cells induced by some treatments (e.g. chemical compounds), and quantifying these changes into a large set of numerical features that can characterise samples in a relatively unbiased way. This technique, known as morphological profiling, has proven to be useful for a variety of applications, such as optimizing the diversity of compound libraries [3], determining mechanism of actions of compounds[4, 5, 6], and clustering of genes by their biological functions[7, 8].

Cell Painting, developed at the Broad Institute, is one of such image-based protocol for morphological profiling. The details of the Cell Painting protocol can be found in Bray et al [9]. In short, cells are plated in multi-well plates, perturbed with the treatments to be tested, stained by six fluorescent dyes, fixed, and imaged using a high-throughput microscope. The result is five channel images revealing eight broadly relevant cellular components or organelles, including the Nucleus, Golgi, Mitochondria, Endoplasmic reticulum, Nucleoli, cytoplasmic RNA, F-actin cytoskeleton, and plasma membrane. Then, depending on the applications, one can either directly use the images, or use an automated image analysis software to extract various features (measurements of size, shape, texture, intensity, etc.), creating a **phenotypic profile** for each sample. Cell Painting data has been used for a range

of applications, from predicting mitochondrial toxicity [10, 11], in vitro toxicity[12], clustering of mode of action [13], and hit identification[14]. In addition, Cell Painting can be used in combination with other modalities to enhance prediction such as chemical structure[15] and gene expression data[13].

In this work, chapter 3 uses the Cell Painting 30k compounds data set from the Broad Institute [16] (referred to as 30k Broad dataset), and chapter 4 and 5 uses the internal Cell Painting from Janssen Pharmaceutica N.V. (referred to as Janssen internal dataset). For demonstration, we present sample cell images from both the 30k Broad dataset (Figure 1.1) and Janssen internal dataset (Figure 1.2). We briefly describe the Cell Painting images acquisition process below for the two datasets. Last but not least, it is worth mentioning that at the time this thesis being written, the biggest public Cell Painting data set is JUMP-CP [17], which contains measurements for 120000 compounds. However, the dataset was released too late for this work.

### 1.1.1 Data Acquisition and Processing

#### 30K Broad Dataset

Full details of the assay protocol can be found in Bray et al. [9, 16]. In brief, U2OS cells were plated in 384-well plates, then treated with each of the 30 616 compounds in quadruplicate. Live cell staining was first performed to stain the mitochondria. After incubation, the cells were fixed with formaldehyde, permeabilized with Triton X-100, and stained with the remaining dyes to identify the nucleus (Hoechst), nucleoli and cytoplasmic RNA (SYTO 14), endoplasmic reticulum (concanavalin A), Golgi and plasma membrane (wheat germ agglutinin), and the actin cytoskeleton (phalloidin). Each of the 406 multi-well plates was imaged using an ImageXpress Micro XLS automated microscope (Molecular Devices, Sunnyvale, CA, USA), with 5 fluorescent channels at $\times 20$ magnification, and 6 fields of view (sites) imaged per well. Then a quantitative analysis of the raw images was performed using

a 3-step pipeline workflow created with the modular open-source software CellProfiler[18] for quality control, illumination correction, and features extraction. The extracted features include a broad array of cellular shape and adjacency statistics, as well as intensity and texture statistics that are measured in each channel. The data is publicly available on GigaDB for download (`https://gigadb.org/dataset/100351`).

**Janssen Internal Dataset**

Overall, the protocol is very similar to Bray et al., with small differences which we will mention here. Briefly, U2OS cells were seeded in 1536 well plates (Aurora, Scottsdale, AZ) and allowed to attach for 24 h. Compounds were diluted in DMSO to a final concentration of either $20\mu M$, $10\mu M$, $4\mu M$, $0.8\mu M$, $0.16\mu M$, and the plates were incubated for 24 hours before fixation, permeabilization, and staining. Images of the five fluorescence channels were acquired with a Yokogawa (Tokyo, Japan) CellVoyager 8000 confocal high-content imaging reader. The PerkinElmer (Waltham, MA) Acapella (`https://content.perkinelmer.com/lab-products-and-services/product-support.html`) automated image analysis software was used to extract around 1600 morphological features, including measures of shape, size, texture, intensity, etc., from individual cells. Well-level results were obtained by taking the means and medians over the cells. Results were imported in the Phaedra software (30,31) version 1.0.8 for quality control and technical validation, rejecting wells containing experimental artifacts. In every experiment, 70 reference compounds were included at 4 concentrations, and the consistency with a historical reference of 50 features was verified for selected compound-concentration pairs. (32) Validated feature data were exported and converted to Z-scores by normalization against the DMSO controls on every plate.

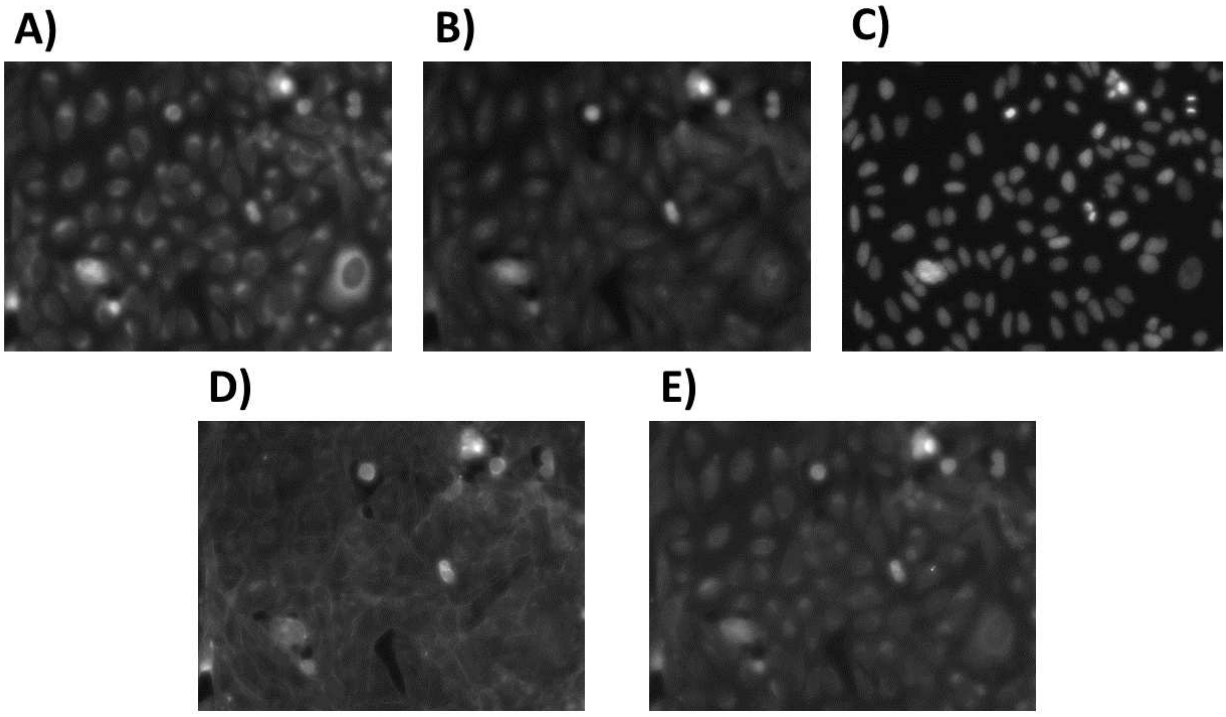



Figure 1.1: **Sample Cell Painting images from file 24294-I23-2.npz, curated from the 30K Broad Institute Dataset.** The file name means plate 24294, well I23, and view 2. Each of these image are from the same view, but with different dyes: a)Mito, b)Ph_Golgi, c)Hoechst, d)ERSytoBleed, e)ERSyto. For modelling, we stack these 5 views to create one 5-channel image.

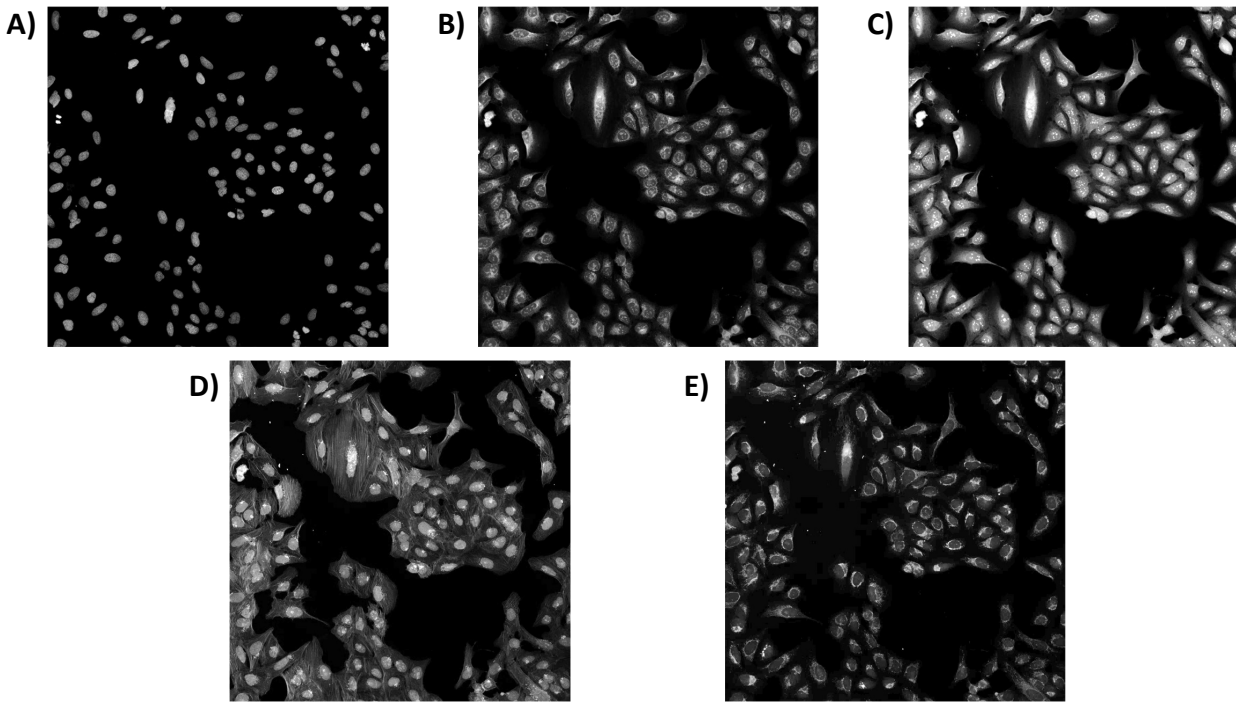



Figure 1.2: **Sample cell painting microscopy images from the Janssen internal dataset.** Five channels of the same view imaged in the Cell Painting protocol. Each highlights a different organelle or cellular component, A) Nucleus, B) Endoplasmic reticulum, C) Nucleoli, cytoplasmic RNA, D) Actin, Golgi, plasma membrane, E) Mitochondria.

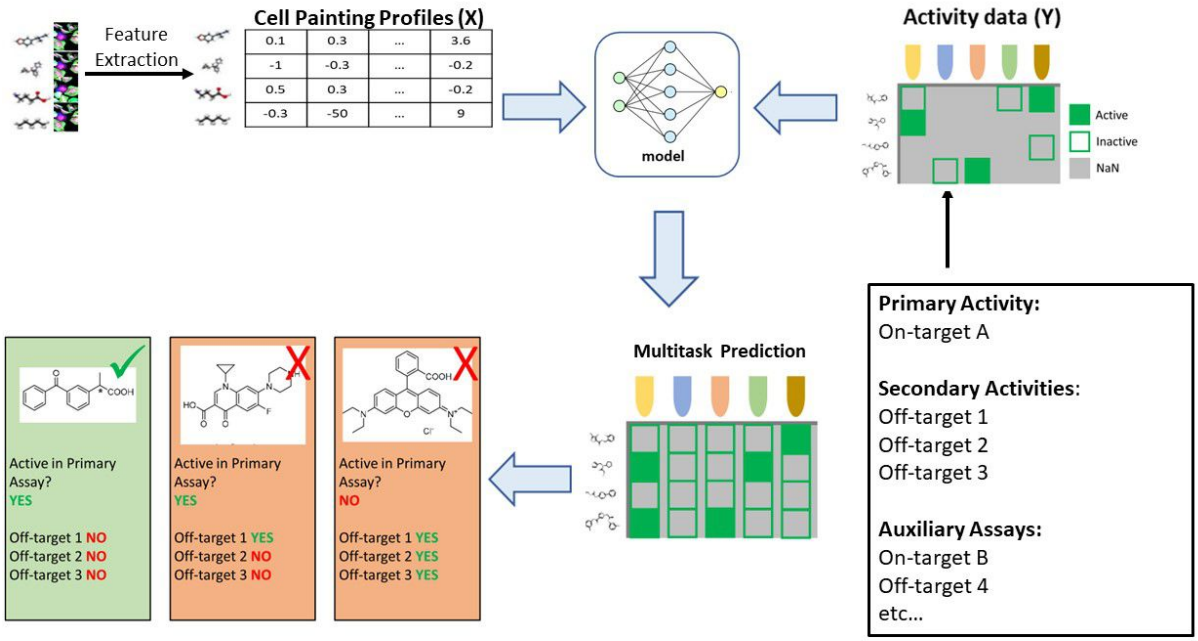



Figure 1.3: **Image-based Activity Modelling.** Using features extracted from Cell Painting images as input, and a sparse activity matrix as labels, the model learns to fill in the activity values for other compounds. These estimated activity values can be used for prioritizing and further optimizing chemical series.

## 1.2 Modelling Small Molecule Activity with Cell Painting

### 1.2.1 Activity Modelling in Early Drug Discovery

In early drug discovery, hit identification is important - the process of finding a starting-point molecule for a drug. The hit should be optimized for not only the primary activity i.e., an assay for a desired biological activity such as inhibition of a known disease target, but also the secondary activities - any additional physicochemical, pharmacokinetic, and toxicological properties. Identified hits will then be sent for further prioritization and optimization. These assays are mainly potency assays, which measures the concentration at which a compound elicits 50% of a desired effects (e.g. inhibition). Potency is measured as IC50, and in line

with previous literature, we use the pIC50 which is the IC50 on logarithmic scale, calculated as -log10(IC50).

One of the popular approaches to identify hits from a large compound library is quantitative structure-activity analysis (QSAR), which refers to any type of model that correlates chemical structure information such as ECFP [19] or SMILES strings [20] with activity data. Using such models, one can quickly screen through a large library and filter out potential hits that satisfies the desired primary and secondary activities for further experimental follow-up.

### 1.2.2 Overcoming Applicability Domain by Replacing Chemistry Input with Cell Painting Data

One downside of QSAR models is that they are **chemically biased**, meaning the prediction obtained for compounds which are substantially chemically different from the training set will be less reliable. This problem is sometimes also referred to as **limited applicability domain**. Every QSAR model has their own applicability domain, which is constrained by the training compounds diversity and hence is not equally reliable for all compounds.

One solution to this issue is using features extracted from cell microscopy images as inputs, rather than chemical descriptors. Previous research has demonstrated that while cell painting-based activity models perform comparably to structure-based models, they also broaden the applicability domain of QSAR models [10], and increase chemical hit diversity over the initial high throughput screening campaigns[21]. This is because structurally different compounds can produce similar phenotype in cell, hence using cell features as input can be seen as a way to scaffold hop into structurally different chemical spaces, that still have the same biological effects.

### 1.2.3 Image-based Activity Modelling

Hereafter, we refer to the activity model utilizing Cell Painting data input as **Image-based Activity Model**. An example involving this model is shown in Figure 2. As the starting point, Cell Painting images or features for compounds in the library are created as model input. For activity labels, data from on-target assays, off-target assays, and optional auxiliary assays from other projects or databases are used. These labels are often binarized, as classification tends to deal with experimental noise better and is generally an easier prediction task than regression. The binarized labels are then combined into a label matrix, where columns represent different assays and rows correspond to compounds in the library. This label matrix is typically sparse with a low percentage of filled entries, especially for new assays in early drug discovery campaigns. The model is then trained to predict the missing activity values for all compounds, which can be leveraged to prioritize and optimize chemical series.

## 1.3 Thesis Structure

This thesis is about our efforts to improve the aforementioned Image-based Activity Model given the data and resources I had during my PhD at Janssen Pharmceutical. The work for this thesis is three-fold. Firstly, in Chapter 2, we seeked to improve the activity prediction performance in terms of model algorithms. By forming the research question into a few-shot learning problem, we curated a public data set from ChEMBL and 30K Broad data to help facilitate comparison of different algorithms in the few-shot bioactivity prediction setting. This data is publicly available to download, along with the codes to generate the data for reproducibility. We compared a variety of singletask, multitask and meta-learning algorithms on said data to gain more insight into: Which algorithm performs the best on average across all assays? What is the behaviours of these algorithms when little data or more data is available? Does transfer learning help improve modelling capabilities?

Secondly, in chapter 3, we aimed to improve model performance for highly imbalanced assays by incorporating Cell Painting data obtained at a lower compound concentration e.g. $0.16\mu M$, $0.8\mu M$, $4\mu M$ (referred to as low-concentration images). Data used in this work is from internal Janssen pharmaceutica. Low concentration images typically tend to yield worse bioactivity predictions than higher concentration images (e.g. $10\mu M$, $20\mu M$), hence are typically not a preferred choice for modelling. Nonetheless, we proposed the combination of well-performing models trained at higher image concentrations, with lower image concentration for inference to identify more potent compounds. We showed that this approach improved on the conventional method (directly training a high-potency model) in 65% of assays investigated in terms of AUC-ROC, and 75% of assays in terms of RIPtoP-corrected AUC-PR. This also motivates the generation of Cell Painting data at different compound concentrations for future research.

Finally, in chapter 4, we tried multimodality representation learning between Cell Painting data and RNA-Seq data to produce new representations that can benefit not only bioactivity prediction, but also other downstream tasks such as Mechanism of Action clustering. In more details, we compared two multimodal representation learning algorithms for Cell Painting and RNA-Seq: Contrastive Learning and Bimodal Autoencoder. We use the setting of cross modality learning where representation learning is performed with two modalities (Cell Painting and RNA-Seq), but only cell painting is available for new compounds embedding generation and downstream task. This is because for new compounds, we would only have Cell Painting data and not RNA-Seq, due to high data generation cost of the RNA-Seq screen. As far we know, there is no public data that contains both Cell Painting data and bulk RNA-Seq for the same cell samples at the scale that the internal Janssen data has. We reported results on a variety of downstream tasks to gain insight on which tasks the learned representation can improve over the original Cell Painting feature.

## 1.4 Related Works

The work of Simm et al. initiated the paradigm of modeling hundreds, and in some cases thousands, of assays simultaneously using features from image-based phenotypic screens, resulting in a dramatic 50- to 250-fold increase in hit rates. Since then, Cell Painting has become the preferred method for such image-based phenotypic screening, largely due to the availability of several public Cell Painting datasets provided by the Broad Institute. A few other notable contributions comes from Hofmarcher et al., who were among the first to benchmark models trained directly on cell images against those using hand-selected features; Herman et al.[10] who used Cell Painting model as an active learning tool to enhance the applicability domain of QSAR models; and Fredin Haslum et al. [22] who trained the Cell Painting model using unrefined single-concentration activity readouts. Last but not least, in terms of few-shot learning works, Stanley et al.[23] created a benchmark for few-shot bioactivity learning which they compared different chemical structure-based models.

Much of the research on Cell Painting images are based on two public datasets: 30K compounds from Bray et al [16] and JUMP-CP[17], both primarily contain images acquired at $10\mu M$ concentration of test compound. Hence, research which involved multiple concentrations for Cell Painting images remains rare, and are mainly from groups that can generate the data themselves[24].

In terms of representation learning of Cell Painting, firstly, using image analysis softwares such as CellProfiler or Acapella to extract expert-chosen features has been the default method since the inception of Cell Painting. With the popularity of deep learning, efforts have been made to learn representations from neural networks unimodally (or pre-train) from cell images using self-supervised methods like DINO, SimCLR and MAE[25, 26]. Additionally, multimodal representation learning for molecules has utilized Cell Painting data as the second modality [27, 28, 29, 30] to introduce biological information into molecular structure-based representations. The aim is not only to enhance performance in modeling tasks such as

QSAR but also to establish a link between chemical structure and biological phenotypes. This can be observed in the downstream tasks introduced by the aforementioned works, such as the mutual retrieval of molecules and their corresponding images.

Finally, for representation learning of RNA-Seq transcriptomics data, to the best of our knowledge, there has not been a public bulk transcriptome dataset at the same scale as the Janssen dataset, hence the lack of literature. The vast majority of literature on representation learning using gene expression data has been using single-cell transcriptome, due to the availability of public data. These methods range from variational autoencoder[31], to transformer-based models [32, 33, 34] based on BERT[35] and GPT[36]. These models have shown self-supervised pretraining on a large single-cell transcriptome corpus (30 to 50 million cells - equivalent to 600 billion to 1 trillion tokens) is an effective strategy to boost modelling accuracy in a variety of downstream tasks. Lastly, Yang et al. used an autoencoder architecture to translate between single-cell RNA-seq and chromatin images[37].

# Chapter 2

# Small Molecule Activity Few-Shot Prediction using Cell Painting Data

## 2.1 Motivation

One way to improve the aforementioned image-based activity model is by choosing the best model. By setting activity prediction as a few-shot learning problem, we establish a rigorous benchmark to compare different modelling paradigms in singletask, multitask, and meta-learning.

### 2.1.1 Few-shot Learning

Few-shot learning is a subfield of machine learning with a focus to effectively train a model with few data available. In the few-shot learning setting, there is no one big dataset $D$ to learn from, but instead many small datasets we called *tasks*, denoted $T$. The aim of few-shot methods is to generalise over new tasks $\{T_u\}_{u=1}^{U} \in D_{test}$ efficiently with only a small number of available datapoints. Each task $T_u$ consists of a *support set* $S$ for learning, and a *query set* $Q$ for evaluation, $T_u = \langle S, Q \rangle$. Typically the size of support set $S$ is very small to reflect the low-data setting.

Few-shot models adapt efficiently to low-data tasks by using an advantage initialisation of its parameters, normally though some sort of *pretraining* on a large data corpus such as a set of auxiliary tasks $\{T_v\}_{v=1}^{V} \in D_{train}$. We expect that knowledge gained from pretraining

can be transferred effectively to new unseen tasks, so that models can quickly learn these new tasks using only little data. This can be compared to, for example, a person who already has prior knowledge of music can pick up a new musical instrument relatively fast with little demonstration.

Most state-of-the-art few-shot learning research came from computer vision and natural language processing domain[38, 39, 36, 40, 41]. In fact, there has also been growing interest in few-shot learning in drug discovery[23], since data scarcity is a common setting for many prediction tasks. Among those, activity prediction is an ideal few-shot learning challenge: there are many related low-data tasks convenient for knowledge transfer between each other. By forming activity prediction with Cell Painting data into a few-shot learning problem, we can leverage existed methods in few-shot learning research to improve the performance of the image-based activity modelling pipeline. (Figure 1.3).

Through this work, we made two contributions:

- A data set for few-shot prediction of small molecule activity using cell microscopy images, which we named FSL-CP. The dataset is curated such that it is easy to experiment few-shot methods on.

- A benchmark of models encompassing different existing singletask, multitask and meta-learning approaches on the dataset. This acts as both a diverse baseline for future algorithms, and a means to study the strength and weaknesses of different modelling paradigms.

## 2.2 Data Curation

The FSL-CP dataset comprises compounds in the intersection of ChEMBL[42] version 31 and the 30K Broad Cell Painting[16] public dataset. We provide an overview of the data construction process below (Figure 2.1), and the exact reproducible source code is available on GitHub, at `https://github.com/czodrowskilab/FSL_CP_DataPrep`.

**Labelling the compounds** For this project we focus on small molecule activity assays (e.g. IC50) available in the ChEMBL database. We query activity data for all the compounds in the original Cell Painting dataset using their InChiKey. We follow a similar data processing strategy as in Hofmarcher et al. [43]. A .db file of the ChEMBL data version 31 is downloaded from `https://chembl.gitbook.io/chembl-interface-documentation/downloads`. We only retrieve data from assays whose ChEMBL standard types are among (pIC50, pEC50, IC50, EC50, -Log EC50, -Log IC50, Potency, GI50, AC50, ED50). For each assay, both the *activity comments* from the experimenter and the *pChEMBL values* (numerical value for activity on a negative logarithmic scale) are retrieved. Duplicate labels are resolved either by averaging if they are pChEMBL values, or majority voting if they are activity comments. The pChEMBL values are restricted to only between 4 and 10, and the activity comments are also chosen to only be spelling variants of 'Active' and 'Inactive'. In more details, these variants are: 'active', 'Active', 'inactive', 'Inactive', 'Not active', 'No activity', 'Not Active (inhibition $< 50\%@10\mu M$ and thus dose-response curve not measured)'. The final modeling *task* is defined as an assay after being binarized, either with a threshold on the pChEMBL value, or based on the activity comments. For the pChEMBL values specifically, we use three thresholds for each assay: 5.5, 6.5, 7.5, which results in three separate modelling tasks. This means an assay can have at most four tasks, three from the pchembl values at thresholds 5.5, 6.5, 7.5, and one from the activity comments. Lastly, we filter out to only allow tasks with at least 10 active and 10 inactive labelled compounds.

**Processing the Cell Painting data** The images, as well as morphological features aggregated at well-level and metadata, can be found at the 'Cell Image Library'. We downloaded the raw images and ran the CellProfiler quality control and illumination scripts provided. Then, the five dye channels are concatenated along the third dimension, converted into 8 bits, have their 0.0028% outlier bits removed, and saved in .npz files. The images are further normalised, randomly cropped, and resize prior to modelling. Further details are covered in the **Benchmark models** section. For the well-level morphological features, we

remove columns that are highly correlated (correlation coefficient $> 0.95$), or have only one value. Finally, we standardize features by removing the mean and scaling to unit variance.

**Features** At the end, each data point of FSL-CP corresponds to one well, represented by six $520 \times 696 \times 5$ images (for six views in a well), and by a feature vector of length 993. We refer to them as CP images and CP features, respectively. SMILES strings and InChiKey are also provided, although for this study we only focus on the cell images and information which comes from them.

**Deep Learning embedding** We also create an 'enhanced' set of CP features by concatenating the original CP features with embeddings from a ResNet50[44] pretrained on ImageNet[45], akin to the method used in Schiff et al.[46]. This embedding provides the input vector with abstract high-level neural-network-based features. For each well, we run the six $520 \times 696 \times 5$ views through the ResNet50 to generate six embeddings, which are then averaged to create one final embedding of length 1000. We tried different variants of ResNet and Inception[47, 48]: ResNet18, ResNet50, ResNet101, ResNet152, inception_resnet_v2, inception_v3. We ended up settling on ResNet50, which yield the best performance on our dataset despite being a simple model. It should be noted that the length of the embedding can be further tuned to boost predictive performance.

ResNet50 and its ImageNet pretrained weights are loaded from the timm[49] library using the command $timm.create\_model(`resnet50`, pretrained = True, in\_chans = 5)$. The weights are then frozen to be used as a feature extractor. When 'pretrained=True', since pretrained weights only have 3 channels (model pretrained on RGB images) and our images have 5 channels, timm applies a custom protocol. For the first layer, the weights are copied 2 times such that the total number of channels are now 6, and then it selects the first 5 channels as weights.

**Pretrain, validation and test splits:** The models are evaluated on 18 tasks which we will call *test set* $D_{test}$. The other 183 tasks, referred to as *auxiliary tasks*, are used for model pre-training. They are randomly splitted into *train set* $D_{train}$ and *validation set* $D_{val}$,

consisting of 161 and 22 tasks, respectively. The test tasks are selected based on the following criteria:

- Tasks in $D_{test}$ does not share the same targets as those in the $D_{train}$ and $D_{val}$, unless a target is unknown (denoted 'unchecked' on ChEMBL). This is to avoid the overlap of very similar tasks during training and inference.

- Test tasks must have over 96 datapoints, to enable model comparison for a range of support set sizes.

- Test tasks must have a ratio of active compound between 0.3 and 0.7, to avoid strongly imbalanced data affecting the model comparison. Some method might be better because it overcomes the data imbalance problem, not the low data problem which is what we focus on.

**Dataset statistics:** FSL-CP contains 201 modelling tasks for 10526 unique compounds, with only 2.58% of the label matrix filled. Number of compounds for each task and their active ratio is visualised in (Figure 2.2A) and (Figure 2.2B), respectively. It is worth noting that the majority of the compounds has 4 replicates, as per the design of the cell painting assay. However, there are cases where there are fewer or more replicates (Figure 2.2C), potentially due to the omission of low-quality images and repeated purchase of some compounds.

## 2.3 Evaluation

**Few-shot prediction:** In order to simulate the low-data setting when evaluating models, we sample from each task in a stratified manner a small number $M$ of datapoints for the support set, and 32 datapoints for the query set. The models are then trained on a binary classification task on the support set, and evaluated on the query set. In literature, this sampled subset is called a $M$-samples 2-shot *episode*, where the 'samples' refer to available

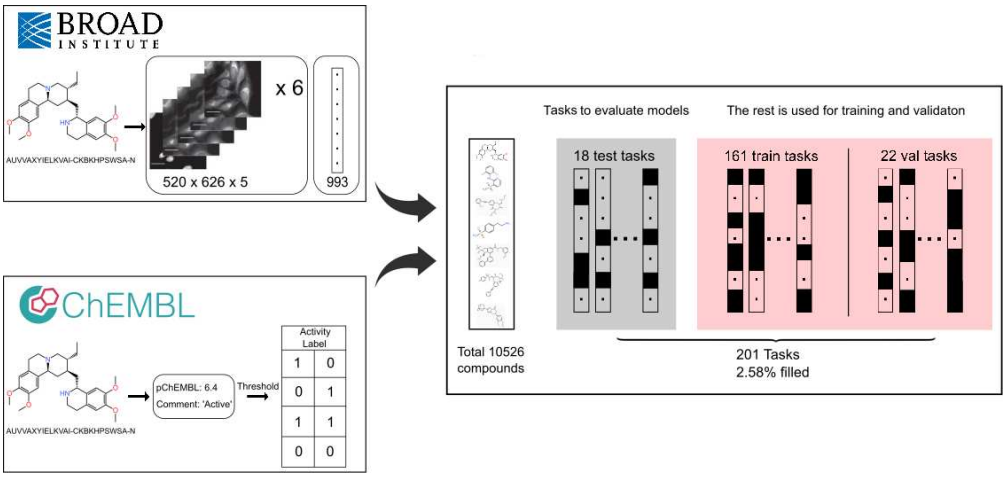



Figure 2.1: **FSL-CP Data Curation and Processing.** Cell painting images and features from CellProfiler[50] comes from Bray et al.[16]. Each well is represented by six $520 \times 696 \times 5$ images, and a feature vector of length 993. Small molecule activity labels are retrieved from assays in ChEMBL31[42]. Then a threshold procedure is applied to binarise the labels, producing different *tasks* from assays. The intersection of the two sources results in 10526 unique compounds, and 201 prediction tasks. 18 tasks are chosen for model evaluation based on a set of criteria, and the rest are for training and validation (referred to as *auxiliary tasks*.)

data (size of support set), and 'shot' refers to the number of classes to predict, which is two for binary classification.

For every test task, we report model performances averaged over 100 episodes in order to eliminate variations from sampling. Additionally, results are recorded over a range of support set sizes $M$: 8, 16, 32, 64, 96, to monitor how well models perform as size of available data increases.

**Metrics:** The results presented and discussed in the result section mainly use area under the receiver operating characteristic curve (AUROC). AUROC comes with many benefits, such as ranking predictions without a decision threshold, meaning predictions can be compared without needing to be rounded to 0 or 1. At the same time, the active ratio of tasks in $D_{test}$ are not too imbalanced that they make AUROC misleading.

In addition, results reported in F1 score, balanced accuracy, Cohen's kappa, and $\Delta$AUPRC [23] can also be found in the Supplementary Information. Since each metrics are formulated differently and focus on different aspects, a more comprehensive comparison can be made

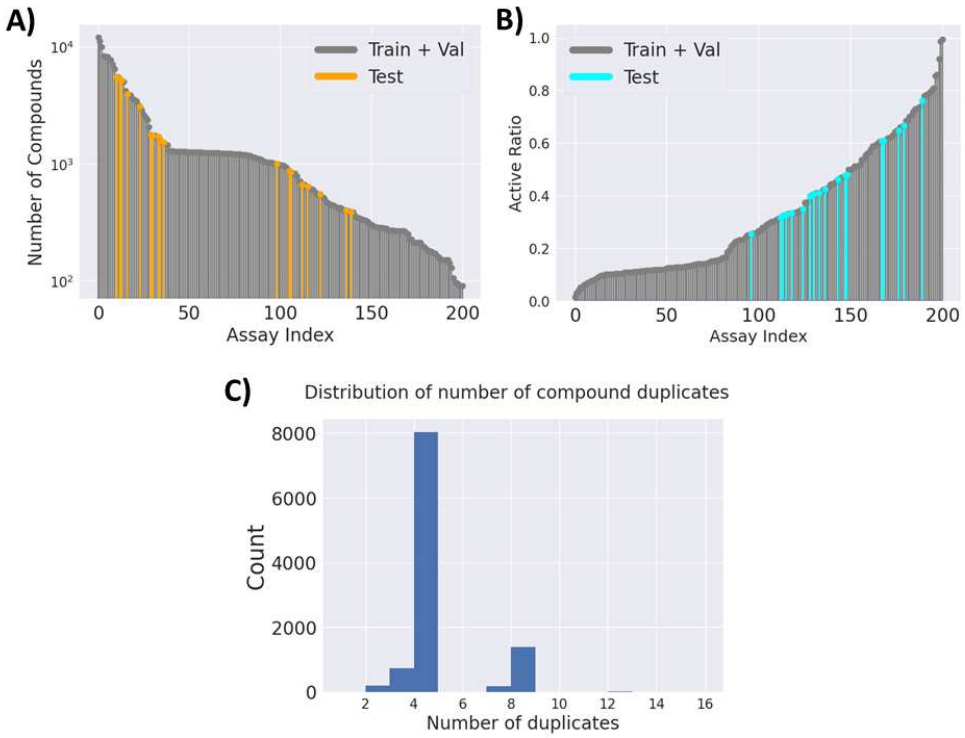



Figure 2.2: **FSL-CP Data Statistics.** (A) Number of compounds for every modelling task. (B) Ratio of active compounds for every modelling task. Test tasks (in turquoise) have their active ratio between 0.3 and 0.7. (C) Distribution of compound duplicates. Most have 4 duplicates as per experimental design, but there can be more or less duplicates, due to omission of low-quality images, or repeated purchases of compounds.

between different models by comparing models across all different metrics.

### AUROC

Area under the receiver operating characteristic curve (AUROC) is a metrics for assessing how good a binary classifier is. Given a binary classifier which outputs a probability between 0 and 1 for a predicted class, one can put a threshold $m$ and define any prediction that is larger than $m$ to be positive, and the other negative. Then, the corresponding true positive rate (TPR) and false positive rate (FPR) can be calculated with TP/(TP + FN) and FP/(FP + TN), respectively (TP: true positive, FN: false negative, FP: false positive, TN: true negative). By varying the threshold $m$, one can obtain different corresponding values of TPR and FPR, which can be plotted against one another to produce the ROC curve. AUROC is the area under said curve. A value of 0.5 means a random model. 1 means perfect correlation between predictions and real values, and -1 means perfect negative correlation between predictions and real values. AUROC is threshold independence, and interpretable: it is the chance of ranking a randomly chosen positive instance higher than a randomly chosen negative inference. However, it can be misleading for highly skewed data.

### F1 Score

F1 score is the harmonic mean of precision and recall. F1 score is a metric to evaluate a binary classifier, where 1 is the best and 0 is the worst. The formula for the F1 score is $\frac{2*precision*recall}{precision+recall}$. F1 does not accept input probability score, hence for models that outputs probability scores, they are binarized at threshold 0.5. F1 is useful because it takes into account both recall and precision equally, while being insensitive to true negatives, which can be beneficial or detrimental based on the positive labels ratio.

### Balanced Accuracy

Balanced accuracy is a metric to evaluate a binary classifier, where 1 is the best and 0 is the worst. It is the arithmetic mean of true positive rate (TPR) and true negative rate (TNR) $(TPR + TNR)/2$, where $TPR = \frac{TP}{TP+FN}$ and $TNR = \frac{TN}{TN+FP}$. Once can also understand balanced accuracy as the arithmetic mean of the recall of each class, as recall for positive labels is TPR, and recall for negative labels is TNR. Because of this, balanced accuracy overcomes data imbalanced by giving equal importance to each class, regardless of its frequency in the dataset. On the other hand, balanced accuracy ignores precision, which depending on the application can be detrimental.

### Cohen's Kappa

Cohen's Kappa [51] measures the level of agreement between two annotators on a classification problem. In our case, the annotators are the true and predicted activity labels. It has the formula: $\kappa = \frac{p_o - p_e}{1-p_e}$, where $p_o$ is the relative observed agreement among raters, and $p_e$ is the hypothetical probability that the two raters agree by chance. In essence, Cohen's Kappa corrects the agreement measure between 2 rates by eliminating agreement from random chance in the calculation. In the case of binary classification, Cohen's Kappa can be obtained from the confusion matrix. The formula for Cohen's Kappa is then:

$\kappa = \frac{2*(TP*TN - FN*FP)}{(TP+FP)*(FP+TN)+(TP+FN)*(FN+TN)}$.

The derivation of the second formula from the first formula can be found in the Supplementary Information.

### ΔAUPRC

Area under the precision-recall curve (AUPRC) is a metric to evaluate a binary classifier, where 1 is perfection, and the random baseline is equal to the ratio of positive labels over all labels. Similar to AUROC, the precision-recall curve shows the trade off between precision and recall for different output thresholds. In our application, we use ΔAUPRC, which is

AUPRC minus the positive ratio, so that the random baseline is at 0. We found that is a bit easier for result interpretation and visualisation. ΔAUPRC is useful when the classes are very imbalanced. In addition, ΔAUPRC does not take into account true negatives, as the term does not appear in the formula for precision (TP/(TP + FP)) and recall (TP/(TP + FN)). This can be useful or detrimental, depending on the application.

### 2.3.1 Benchmark models

In this section, we provide a detailed description of different modelling paradigms for this particular few-shot problem. The code to all of the models and the training/inference scripts can be found at `https://github.com/czodrowskilab/FSL_CP`. In addition, for conciseness, we only include results of the best performing models for each paradigm. The full results for all models we investigated can be found at the above GitHub link, under the folder 'result'.

As a naming convention, models with **_img** are trained directly on the images, **_cp** means they are trained on the original CP features, and **_cp+** means they are trained on the enhanced CP features.

**Singletask models:** Traditionally, modelling of tasks in drug discovery is solely single-task, with models such as random forest or gradient boosting algorithms on top of fingerprints or curated phys-chem properties [52, 53, 54, 19]. In these settings, auxiliary tasks are not used. Here, we mimic the same procedure by assessing the performance of logistic regression (LR), XGBoost, and a singletask fully-connected neural network (FNN), on both the original and enhanced CP features. For each prediction task of each model we run a randomised hyperparameter grid search using the library scikit-learn [55], considering 10 hyperparameter configurations each run. We report results of the two best performing singletask models: LR on enhanced CP features (**logistic_cp+**) and FNN on original CP features (**singletask_cp**).

**Multitask models:** multitask models have been a staple in drug discovery field, being adopted by many academic and industry groups for various prediction tasks [56, 21, 57]. These models consist of multiple 'heads', each specialised on one task, on top of a shared

'trunk'. The trunk aims to learn a common representation across tasks, which allows it to learn knowledge transferable between tasks and improve performance of each one.

For our benchmark, the same FNN model as in the singletask case is used, but with a head of length 183 instead. Pretraining and validation are performed using 183 auxiliary tasks in $D_{train}$ and $D_{val}$ in a multitask manner. Then the weights are frozen, and the head is replaced with a new one of length 1 for fine-tuning. During evaluation, for each episode, the same frozen model has its last layer fine-tuned using the support set, evaluated on the query set, and reverted back to the state before fine-tuned. We tried training on both sets of CP features but neither leads to drastic improvements over the other. We decided to report the result for the model trained on the original CP to reflect the methods from Simm et al. This model is denoted as **multitask_cp**.

The loss function for pretraining is called multitask binary cross-entropy (BCE), and it is a modified binary cross entropy loss to specifically works for multitask learning with missing labels. Individual binary cross entropy losses are calculated for each task, which are then averaged into a final loss. If the label for a task is missing, then its loss is not calculated and backpropagated. In more details, the code for the multitask BCE is:

```
1. def _multitask_bce(pred, target):
2.     """Function to calculate multitask bce with missing data as -1.
3.     Mask out -1, then average bce across tasks"""
4.     eps = 1e-7
5.     mask = (target != -1).float().detach()
6.     bce = pred.clamp(min=0) - pred*target + torch.log(1.0 + torch.exp(-pred.abs()))
7.     bce[mask == 0] = 0
8.     loss = bce.sum() / (mask.sum() + eps)
9.     return(loss).
```

As can be seen from the code, the exclusion of missing values in loss calculation was implemented by defining a mask for each task with missing data (line 5), calculating the

loss for each task (line 6), then applying the mask to zero out all losses corresponding to the missing data (line 7). Then the averaged of all remaining losses were returned.

**Meta-learning models:** Inspired by human's ability to learn certain tasks very quickly with prior knowledge, meta-learning methods aim to tackle the problem of adapting to new tasks efficiently with only a few training examples. The idea is still the same: pretrain to gain transferable knowledge to generalise to new tasks. But the meta-learning methods introduce the idea of *training in the same way as testing*[58]. In particular, if we evaluate the model on $M$-samples 2-shot episodes, then we can mimic that setting during pretraining to encourage fast adaptation. That means during pretraining, we sample an episode from $D_{train}$ the same way we sample from $D_{test}$, and accumulate the loss from many episodes to update our models' weights. This process is called *episodic training*.

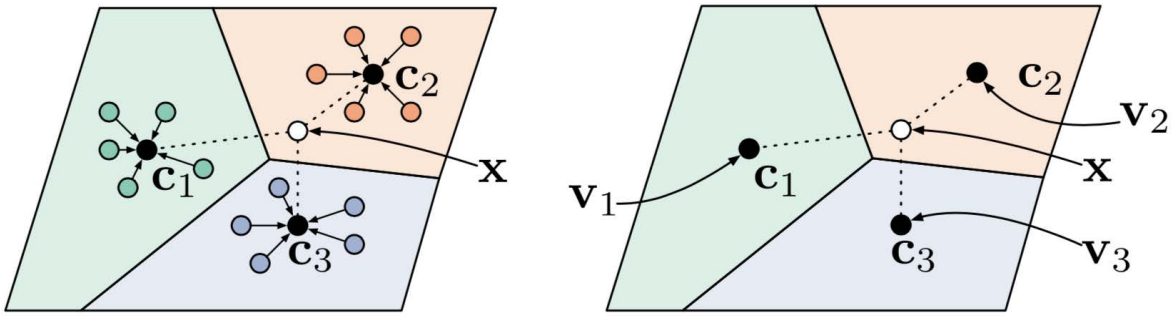
One subclass of meta-learning is *metric-based method*, which tries to learn a distance function over data samples. For example, prototypical network[38] uses a backbone model to generate an embedding. Then classification is made using a k-means clustering based on the euclidean distance from the embedding to the cluster prototypes. Since the backbone for prototypical network can be any kind of embedding generator, we try 2 versions: a ResNet50 and an FNN backbone, which generate embeddings from CP images and CP features, respectively. These models are named **protonet_img**, **protonet_cp** and **protonet_cp+**.

In more mathematical details (Figure 2.3), let the backbone model for embedding generation be $f_\theta$, and the set of feature vectors in the support set with class $c \in \mathcal{C}$ is $S_c$. We define the prototype $v_c$ as the mean vector of all vectors in $S_c$

$$\mathbf{v}_c = \frac{1}{|S_c|} \sum_{(\mathbf{x}_i, y_i) \in S_c} f_\theta(\mathbf{x}_i).$$

The distribution over classes for a given new test data point is a softmax over the inverse of distances between the test data embedding and prototypes.

$$P(y = c|\mathbf{x}) = \text{softmax}(-d_\varphi(f_\theta(\mathbf{x}), \mathbf{v}_c)) = \frac{\exp(-d_\varphi(f_\theta(\mathbf{x}), \mathbf{v}_c))}{\sum_{c' \in \mathcal{C}} \exp(-d_\varphi(f_\theta(\mathbf{x}), \mathbf{v}_{c'}))}$$




(a) Few-shot (b) Zero-shot

Figure 2.3: **How prototypical network works in the a) few-shot and b) zero-shot scenarios.** A prototype is defined for every class based on the support set. T distribution over classes of a new data point is determined by the distances between that data point and the prototypes. Figure from Snell et al.[38].

where $d_\varphi$ is any differentiable distance function. We use cosine similarity in our implementation. The loss function to optimize is then the negative log-likelihood:

$$\mathcal{L}(\theta) = -\log P_\theta(y = c|\mathbf{x})$$

*Optimisation-based* method is another subclass of meta-learning. This approach intends to make gradient-based optimisation converge within a small number of optimisation steps. MAML[39] (Model-Agnostic Meta-Learning) achieves this by obtaining good weight initialisation through pretraining, so that fine-tuning to unseen tasks can be more efficient. Thanks to MAML working with any algorithm that uses gradient descent, we provide results of a ResNet50 and an FNN after being trained by MAML (denoted as **maml_img** and **maml_cp+**).

In more mathematical details, at the start of each optimization step, sample several tasks $\tau_i$ whose associated datasets are $(\mathcal{D}^{(i)}_{\text{support}}, \mathcal{D}^{(i)}_{\text{query}})$. Let the weight at this point be $\theta$. For each of the task $\tau_i$, calculate the new weights $\theta'_i = \theta - \alpha\nabla_\theta\mathcal{L}^{(0)}_{\tau_i}(f_\theta)$, where $\alpha$ is a step size hyperparameter, using the task-associated dataset. After calculating the new weights for all sampled tasks, the actual update to $\theta$ for this optimization step is:

$$\theta \leftarrow \theta - \beta\nabla_\theta \textstyle\sum_{\tau_i \sim p(\tau)} \mathcal{L}^{(1)}_{\tau_i}(f_{\theta'_i}).$$

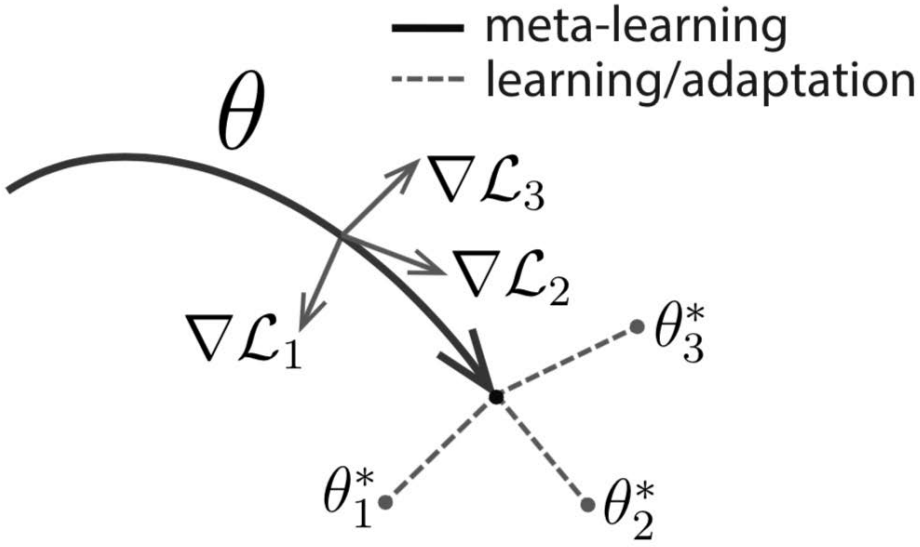



Figure 2.4: **How MAML works.** Figure from Finn et al. [39]

In essence, MAML works by instead of performing gradient descent normally, it calculates the different gradient vectors associated with different tasks, then the actual update step that it takes is in the direction middle of all of those gradient vectors(Figure 2.4). This ensures a good generalization across a variety of tasks by moving the weights closer to **every tasks**, not just one single task.

It is important to mention that unlike feature-based models, image-based models are highly computationally expensive to train. Hence to train these models and tune the hyperparameters in a reasonable time frame, only one out of six available views are used, plus random cropping and down-sizing of images are performed. With an 11GiB NVIDIA GeForce RTX 2080 Ti, the training and inference for one hyperparameter configuration takes between 1 and 2 weeks, depending on the model.

## 2.4 Results

In order to compare performances of different methods benchmarked on FSL-CP, we plot the mean AUROC across 18 test tasks of each method at different support set sizes (Fig-

Table 2.1: **p-values of the one-sided Paired Wilcoxon Sign-rank Test[b] with Alternative Hypothesis being the method in the left column outperforms the method in the upper row.** Entries left blank indicate a p-value greater or equal to 9.99e-01.

a) Support set size 8

| | protonet_cp | multitask_cp | singletask_cp | maml_cp+ | logistic_cp+ | maml_img | protonet_img |
|---|---|---|---|---|---|---|---|
| protonet_cp+ | 3.60e-01 | 7.89e-01 | 1.56e-01 | 3.79e-01 | **1.05e-03** | **3.09e-03** | **5.38e-04** |
| protonet_cp | | 8.90e-01 | 2.74e-01 | 6.12e-01 | **1.79e-03** | **5.23e-03** | **4.52e-04** |
| multitask_cp | | | 3.96e-02 | 3.04e-01 | **2.69e-04** | **1.11e-03** | **1.43e-04** |
| singletask_cp | | | | 6.03e-01 | **3.09e-03** | 8.05e-02 | **1.21e-03** |
| maml_cp+ | | | | | 8.84e-02 | 1.92e-02 | 2.74e-02 |
| logistic_cp+ | | | | | | 6.42e-01 | 9.16e-03 |
| maml_img | | | | | | | 3.47e-02 |

b) Support set size 16

| | protonet_cp | multitask_cp | singletask_cp | maml_cp+ | logistic_cp+ | maml_img | protonet_img |
|---|---|---|---|---|---|---|---|
| protonet_cp+ | 1.45e-02 | 5.42e-02 | **9.36e-04** | **2.77e-03** | **2.49e-04** | **5.34e-05** | **7.63e-06** |
| protonet_cp | | 5.41e-01 | 1.08e-02 | 2.21e-02 | **8.53e-04** | **5.35e-04** | **2.63e-04** |
| multitask_cp | | | **5.22e-04** | 1.13e-02 | **2.08e-04** | **9.54e-05** | **1.46e-04** |
| singletask_cp | | | | 6.51e-01 | **1.68e-03** | 3.49e-02 | **3.43e-04** |
| maml_cp+ | | | | | 2.01e-02 | **2.98e-03** | **1.03e-03** |
| logistic_cp+ | | | | | | 2.48e-01 | **1.50e-03** |
| maml_img | | | | | | | 1.27e-01 |

c) Support set size 32

| | protonet_cp | multitask_cp | singletask_cp | maml_cp+ | logistic_cp+ | maml_img | protonet_img |
|---|---|---|---|---|---|---|---|
| protonet_cp+ | 8.63e-02 | **7.97e-03** | **3.83e-04** | **1.42e-04** | **1.46e-04** | **3.81e-06** | **3.81e-06** |
| protonet_cp | | 6.49e-02 | **1.46e-03** | **2.67e-04** | **2.67e-05** | **1.74e-04** | **1.14e-05** |
| multitask_cp | | | **4.03e-04** | **4.56e-03** | **3.81e-06** | **3.81e-06** | **3.81e-06** |
| singletask_cp | | | | 3.61e-01 | **2.80e-03** | **1.22e-03** | **2.56e-04** |
| maml_cp+ | | | | | **3.69e-02** | **1.65e-03** | **5.23e-04** |
| logistic_cp+ | | | | | | **7.97e-03** | **1.36e-04** |
| maml_img | | | | | | | 2.55e-01 |

d) Support set size 64

| | protonet_cp | multitask_cp | singletask_cp | maml_cp+ | logistic_cp+ | maml_img | protonet_img |
|---|---|---|---|---|---|---|---|
| protonet_cp+ | 2.62e-01 | 3.43e-01 | **6.97e-03** | **4.40e-04** | **7.63e-06** | **3.81e-06** | **3.81e-06** |
| protonet_cp | | 3.88e-01 | 1.46e-02 | **6.05e-04** | **1.91e-05** | **1.46e-04** | **7.63e-06** |
| multitask_cp | | | **2.07e-04** | **2.47e-04** | **3.81e-06** | **3.81e-06** | **3.81e-06** |
| singletask_cp | | | | 9.31e-02 | **1.43e-04** | **3.81e-06** | **3.81e-06** |
| maml_cp+ | | | | | **4.10e-03** | **1.14e-05** | **1.14e-05** |
| logistic_cp+ | | | | | | **2.02e-03** | **2.11e-04** |
| maml_img | | | | | | | 4.18e-01 |

e) Support set size 96

| | protonet_cp | multitask_cp | singletask_cp | maml_cp+ | logistic_cp+ | maml_img | protonet_img |
|---|---|---|---|---|---|---|---|
| protonet_cp+ | 6.14e-01 | 2.84e-01 | 7.73e-02 | **7.25e-05** | **3.81e-06** | **3.81e-06** | **3.81e-06** |
| protonet_cp | | 2.09e-01 | 6.44e-02 | **2.29e-04** | **2.16e-04** | **3.81e-06** | **3.81e-06** |
| multitask_cp | | | 4.00e-02 | **1.45e-04** | **3.81e-06** | **3.81e-06** | **3.81e-06** |
| singletask_cp | | | | **3.36e-04** | **1.44e-04** | **3.81e-06** | **3.81e-06** |
| maml_cp+ | | | | | 1.85e-01 | **1.14e-05** | **4.44e-04** |
| logistic_cp+ | | | | | | **3.21e-04** | **3.81e-06** |
| maml_img | | | | | | | 2.33e-01 |

[b] The test compares mean AUROC (over 100 episodes) of models in 18 test tasks. Marked in bold are significant p-values at $\alpha = 0.01$.

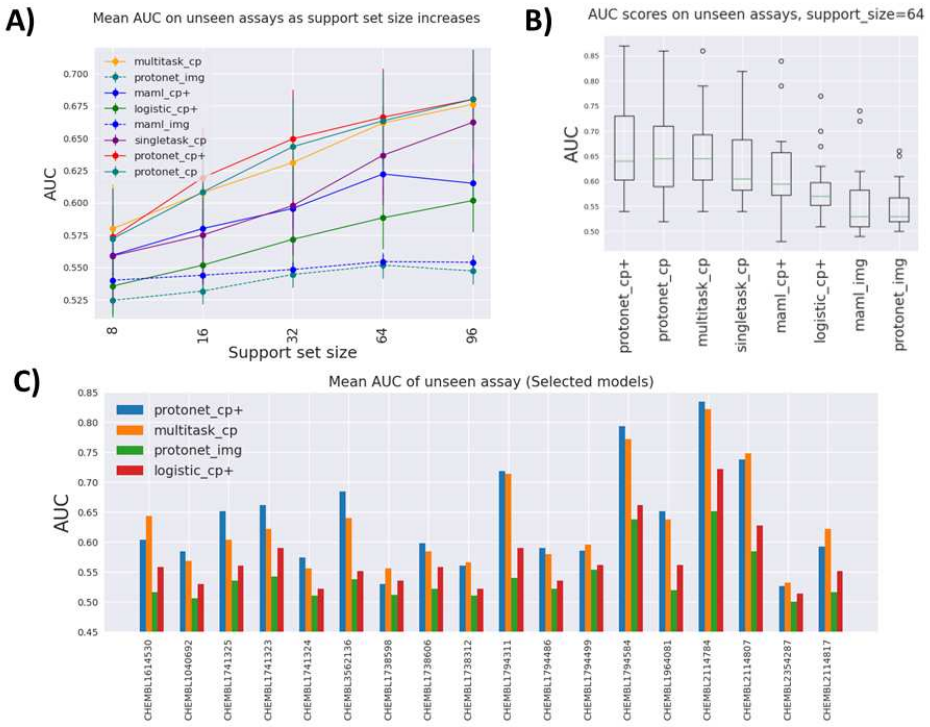



Figure 2.5: **Comparison of different models benchmarked on FSL-CP.** Figure A): Mean AUROC on test tasks as support set size increases. As there are more data available, other methods start to catch up to meta-learning models. Figure B): Distribution of AUROC across all test tasks at support set size 64. The best models tend to have larger AUROC variance. Figure C): Mean AUROC of selected models for each task across all support set sizes. For most tasks, pretraining on auxiliary tasks leads to an improvement over singletask models. However, for a few tasks this is not the case.

ure 2.5A). In addition, a paired Wilcoxon sign-rank test is performed for each pair of models as demonstrated in Table 2.1, with the alternative hypothesis that the method in the left column outperforms the method in the upper row. These figures also provide insight into how performance of each model changes as the amount of available data increases.

Figure 2.5A indicates that the best performing models overall are variants of prototypical network **protonet_cp+** and **protonet_cp**, followed closely by **multitask_cp**. Although according to the Wilcoxon signed-rank test, **protonet_cp+** only outperforms **multitask_cp** for medium-sized dataset (support set size 32). There is not sufficient evidence rejecting the null hypothesis that they perform equally well at other support set sizes with $\alpha = 0.01$. We note that **multitask_cp** even slightly outperforms **protonet_cp+** and **protonet_cp** at support set size 8.

The best singletask model **singletask_cp** is surprisingly powerful, being able to catch up with **maml_cp+** at lower support set size, and outperforms it by a wide margin at large support set size. For these singletask models with no pretraining, the availability of more data in the support set can lead to dramatic improvements in performance. It is highly likely that their performances will keep improving, and eventually might overtake other methods beyond support set size 96. On the contrary, improvements in AUROC scores of meta-learning methods slows down at higher support set size, and even drops as in the case of **maml_cp+**. However, it is worth noting that at support set size 96, some test tasks are excluded in the evaluation process due to insufficient data points. Plus, less tasks in $D_{train}$ and $D_{val}$ are included in the pretraining for meta-learning models at high support set size, due to the fact that there may not be enough datapoints to sample for episodic training. All of these factors can affect meta-learning methods' performance at higher data setting.

Image-based models such as **protonet_img** and **maml_img** substantially under-perform compared to other feature-based methods, likely because only one view out of six is used, and down-sizing of images of fairly small cells leads to drastic information loss.

We also try to leverage deep-learning-based features by concatenating the original CP

features with a ResNet50 embedding of size 1000. While in some cases it does lead to higher AUROC, as evidenced by the fact that many models use the enhanced feature, the improvements are somewhat minute. For example, when comparing **protonet_cp+** against **protonet_cp**, Table 2.1 shows insufficient evidence of improvement across tasks, and in Figure 2.5A, the additional features lead to only small improvements at support set sizes 16, 32 and 64. However, this still poses an interesting question for future research: how meaningful embedding from cell images can be produced using deep learning methods.

Better performing models have the larger spread of AUROC across test tasks, as shown in Figure 2.5B, indicating model performances are fairly dependent on tasks. This is further demonstrated in Figure 2.5C, where some tasks (e.g. CHEMBL2114784) consistently show high AUROC across model, and some (e.g. CHEMBL2354287) are not predictive at all. Additionally, for some tasks **protonet_cp+** is the best method, but in a few other cases **multitask_cp** or **singletask_cp** is the better method.

Figure 2.5C also gives insight on how much pretraining on auxiliary tasks benefits prediction. Again, this is highly task-dependent. Some tasks benefit greatly from pretraining (CHEMBL3562136, CHEMBL2114807), as seen from the improvements of the two pretraining models over the singletask models. However, pretraining can offer no improvement, or even be detrimental in tasks such as CHEMBL1738598 and CHEMBL1738312.

## 2.5 Discussion

We have presented FSL-CP, a dataset for small molecule activity few-shot prediction using cell microscopy images. This few-shot challenge mimics a screening process in early stage drug discovery, where the aim is to identify potent compounds targeting a specific protein from high-content cell images with little data. Previous efforts have been made to benchmark few-shot methods on molecules as graph-structured data[23]. But in machine learning, the primary focus of few-shot learning has been in the computer vision and natural language

processing domain. The fact that our dataset uses cell images as a molecular representation opens up opportunities to adapt state-of-the-art ideas from computer vision to enhance modeling.

This dataset allows us to establish benchmarks that compare the performances of different few-shot learning paradigms. Our result indicates feature-based prototypical network and multitask FNN pretrained on auxiliary tasks generally preforms well across all support set sizes. We also observe improvement in performance slowing down for meta-learning methods at high support set size, in contrast to singletask methods, which greatly benefits from the availability of more available data. However, more labelled compounds and their cell painting data are needed in order to accurately point out whether eventually singletask models outperforms pretrained models, and if yes, at what support set size.

Image-based models underperforms on our benchmark, and the fact that each datapoint consists of six high-definition five-channel images makes it tremendously computationally expensive to train. We had to use only one randomly cropped, downsized image to train the models in a reasonable time-frame with our infrastructure, and this leads to high information loss. Training on full-resolution cell images has been shown to offer better performance than on CP features in some settings[43]. However, in realistic drug discovery projects, larger-size images are used, and there typically are more compounds and more prediction tasks. These make pretraining on full-resolution images difficult, especially if the model needs to be regularly retrained.

A less expensive way to leverage the power of computer vision is to enhance the CP feature with an embedding out of an image using a pretrained model such as ResNet or Inception. We tried a simple approach with a ResNet50 as an embedding generator, which yielded small improvements. Since most vision models are pretrained on ImageNet, this suggests there is some transferable knowledge obtained from training on a big unrelated image database, but not enough to make a significant improvement. We expect that a more informative embedding can be achieved by pretraining the embedding generator end-

to-end on cell images with a more relevant pretraining task, such as multitask or contrastive learning[59, 60].

The benchmark also provides insight on how effective transferring knowledge from pretraining models on auxiliary tasks is, to new tasks. Mostly, new tasks benefit from such a pretraining scheme, but the degree to which different tasks improve varies. To what degree a new task benefits from pretraining is still an open question for research. As a general observation, it seems already predictive tasks tend to benefit more from pretraining. Companies aiming to pretrain their models on auxiliary tasks can use a combination of tasks from public sources as well as their own database to benefit from as much data as possible.

## 2.6 Study Limitations

One limitation of this study is the lack of self-supervised models, which are becoming more popular in recent years for pre-training. For example, SimCLR[61] is a proven contrastive learning method for learning from image data. The idea is that the model e.g. ResNet50 is pretrained by generate embeddings from a pair of images, then a contrastive loss is used to maximize agreements between these embeddings. So if the pair consists of images of the same compound (but from different views or replicates), the embeddings should be close together in the embedding space, and further otherwise. Masked autoencoder[26] has also been used as a self-supervised pretraining scheme for several downstream tasks. This method uses a U-net to reconstruct masked sections of input cell images.

Another limitation is related to a few similar prediction tasks in the test set. In Figure 2.6, we measured the Jaccard Index for the unique InChiKeys from every task pair in D_test. We observed that the majority of tasks shares very few common InChiKeys. Exceptions are tasks 737826, 737824_1 and 737825 whose targets resemble Cytochrome P450.

Taking a closer look at these three tasks 737826, 737824_1 and 737825, they have 800, 840 and 779 unique InChiKeys, respectively. 2 datapoints from 2 tasks is similar if they have

the same (InChiKey, labels) pair. Tasks (737826 and 737824_1) have 580 (InChiKey, labels) pair in common. For tasks (737826, 737825) and (737824_1, 737825) the number is 450 and 469. So around 60%-70% of the (InChiKeys, labels) pair is similar between these tasks.

In simpler words, these 3 tasks are 60%-70% 'similar' to each other. In our opinion, they are still different enough to be different tasks in the test set. However, we acknowledge that future data curation effort should take notice of similar tasks like this.

## 2.7 Reproduction of Hofmarcher et al.

This section is used to document my successful effort in reproducing the results in Hofmarcher et al. [43] at the start of my PhD. This is because even though it was my first PhD work, the codebase has been then used and adapted to other projects, including the above few-shot learning benchmark. I believe a thorough documentation of my reproduction can be helpful to other future researchers who want to build models on Cell Painting data.

### 2.7.1 Data preparation

At the time of writing this thesis, both the processed features images (Cell Painting) and the processed activity labels (ChEMBL) are publicly available to download at the authors' GitHub page https://github.com/ml-jku/hti-cnn. In addition, pre-trained weights for the models are also available to download. This helps other researchers reproduce the result immediately without weeks long model training.

However, researchers might want to curate the data by themselves, because they want either to work with activity labels from the latest ChEMBL version, or to process the images differently, or simply to learn. The github link `https://github.com/czodrowskilab/FSL_CP_DataPrep` contains every step I used for curating and processing the data, provided that the appropriate raw data has been downloaded. The data preparation pipeline needs a ChEMBL database files (e.g. chembl_31.db), which can simply be downloaded from `https:`

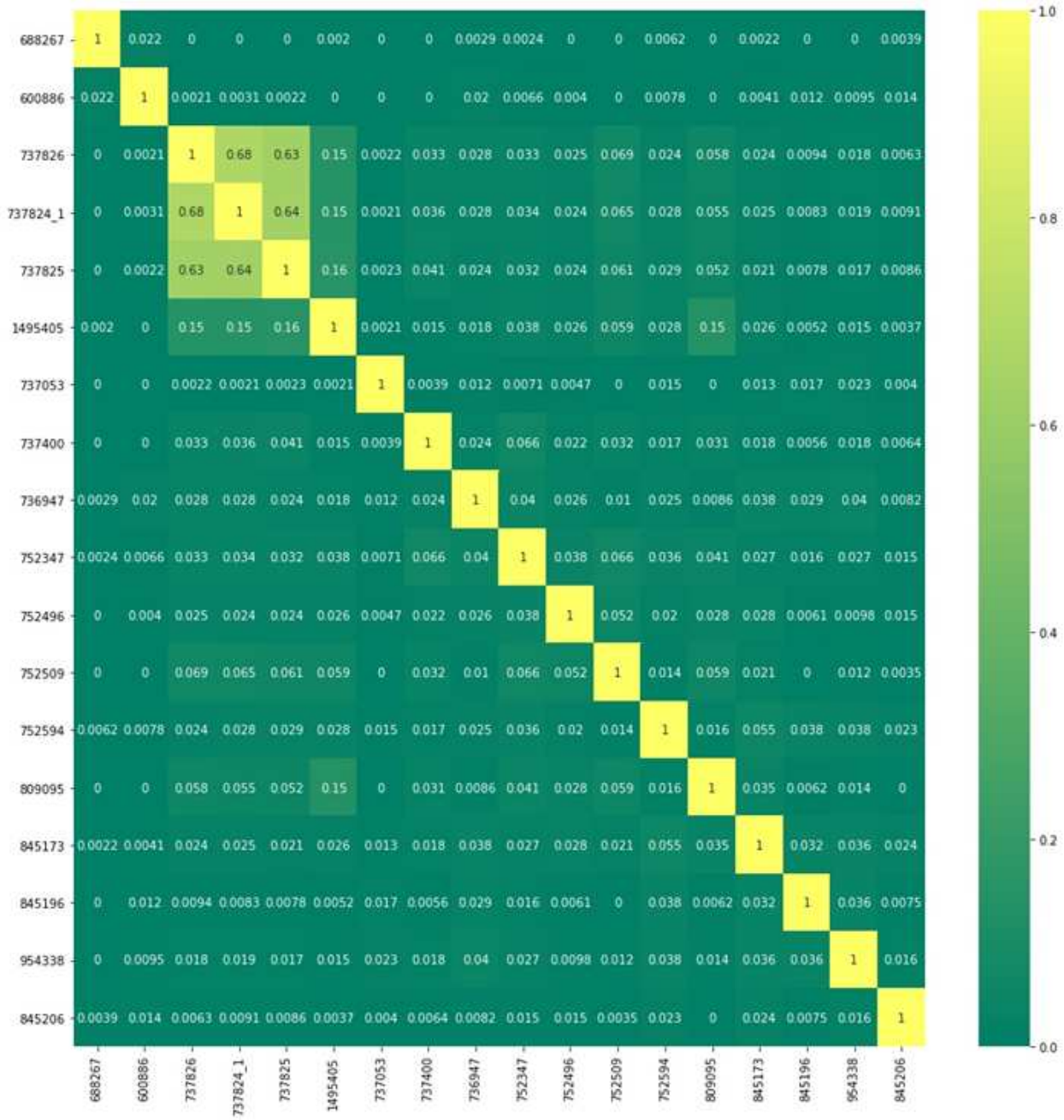



Figure 2.6: **Figure 1: Heatmap of Jaccard Index between the unique InChiKeys of 18 tasks in D_test.** The majority of tasks share very few common InChiKeys. Outliers are tasks 737826, 737824_1 and 737825 whose targets resemble Cytochrome P450. But we believe they are still different enough to be separate tasks in the test set.

//chembl.gitbook.io/chembl-interface-documentation/downloads. Hofmarcher et al used ChEMBL version 22.1 for their work. Secondly, the raw Cell Painting images are also needed. They can be downloaded from https://cellpainting-gallery.s3.amazonaws.com/index.html in the folder 'cpg0012-wawer-bioactivecompoundprofiling'. What the data processing pipeline does has already been describe in the Data Curation section.

### 2.7.2 Model Building

The models and training pipeline are also available at the authors' GitHub page. However, in this section, I will document my effort to build the modelling pipeline using PyTorch[62] and PyTorch Lightning[63]. PyTorch is a deep learning framework that allows for deep learning models development, and PyTorch Lightning is a library build on PyTorch that simplify may aspects of model training e.g. multi-GPUs training.

Firstly, the models are imported from the package timm - PyTorch Image Models [49], which is a collection of state of the art computer vision models coded in PyTorch. This package allows models to train on images with more than 3 channels (in our case, images are 5 channels), but also has ImageNet pretrained weights that users can load if they believe transfer learning from ImageNet is helpful with their tasks. To set the number of image channels to 5, when the model is loaded via *create_model* function, set the argument $in\_chans = 5$

We use PyTorch Lightning for training the model on multiple GPUs. Similar to the original paper, we use multitask BCE as the loss function, gradient clipping, and stochastic gradient descent with momentum as the optimizer. However, we use a different learning rate schedule: linear warmup cosine annealing with warmup epochs=10, max epochs=20. Plus, we resize the images to 384x384 to fit a larger batch size in the GPUs. Details of the hyperparameters can be found on the author's GitHub page. After training, predictions for the test set will be passed through a sigmoid function, then saved into a .pt file for further calculation of performance metrics.

| Model | Method | Img Size | mean AUC | std AUC | mean F1 | std F1 | AUC>0.9 | AUC>0.8 | AUC>0.7 |
|---|---|---|---|---|---|---|---|---|---|
| convnext_base_384_in22ft1k | transfer | 384x384 | 0.742 | 0.19 | 0.462 | 0.34 | 66 | 88 | 123 |
| vit_base_resnet50_384 | transfer | 384x384 | 0.720 | 0.20 | 0.429 | 0.33 | 64 | 79 | 107 |
| ViT_small_r26_s32 | end-to-end | 384x384 | 0.732 | 0.21 | 0.494 | 0.34 | 69 | 88 | 119 |
| (P)ResNet101 | end-to-end | 520x696 | 0.731 | 0.19 | 0.508 | 0.3 | 68 | 94 | 119 |

Table 2.2: **Result of the Hofmarcher et al. reproduction.** All of the models we tried (first 3 rows) performed comparably to the best model in the original paper (P)ResNet101.

### 2.7.3 Evaluation

The original paper reports mean and standard deviation of AUROC and F1 score across all tasks, as well as the number of tasks whose AUROC is larger than 0.9, 0.8 or 0.7. It is worth noting that these scores are calculated based only on the available data. For example, if a task has 90% missing labels in the test set, then the scores are calculated with the available 10%.

Using a Vision Transformer[64] "small" variant with 32x32 input patch size and a ResNet26 backbone (R26+S/32), trained end-to-end, we achieve mean AUROC across tasks 0.732 with standard deviation 0.21, and mean F1 score 0.494 with standard deviation 0.34. We achieve 69 tasks with AUROC $> 0.9$, 88 tasks with AUROC $> 0.8$, and 119 tasks with AUROC $> 0.7$. This result is comparable with the best model in the original paper. In addition, we also tried ConvNext [65] "base" variant, pretrained on ImageNet 22k and finetuned in ImageNet 1K, before being finetuned again using cell images to predict activity. This model also achieves comparable results with the best model in the original paper. Its mean AUROC across tasks is 0.742 with standard deviation 0.19, and mean F1 score 0.462 with standard deviation 0.34. We achieve 66 tasks with AUROC $> 0.9$, 88 tasks with AUROC $> 0.8$, and 123 tasks with AUROC $> 0.7$. The full results can be found in Table 2.2

# Chapter 3

# Low Concentration Cell Painting Images Enable the Identification of Highly Potent Compounds

For this chapter and the next chapter, I had the opportunity to work with the internal Cell Painting data from Janssen. One unique aspect of this dataset is that it contains Cell Painting measurements from 5 different compounds concentrations $0.16\mu M$, $0.8\mu M$, $4\mu M$, $10\mu M$, $20\mu M$. At this time, the low-concentration images ($0.16\mu M$, $0.8\mu M$, $4\mu M$) generally yield poorer bioactivity predictions compared to higher-concentration images ($10\mu M$, $20\mu M$), and are therefore less preferred for modeling. In this chapter, I will explain our approach to use low-concentration images for inference in combination with a well-performing high-concentration image model, and show how effective our approach is compared to the conventional method.

## 3.1 Motivation

Much of the research on Cell Painting images are based on two public datasets: 30K compounds from Bray et al[9] and JUMP-CP[17], both primarily contain images acquired at $10\mu M$ concentration of test compound. Here we study an internal dataset acquired at five image concentrations $0.16\mu M$, $0.8\mu M$, $4\mu M$, $10\mu M$ and $20\mu M$. For activity modelling, according to a previous internal study, we found that the higher the image concentration, the

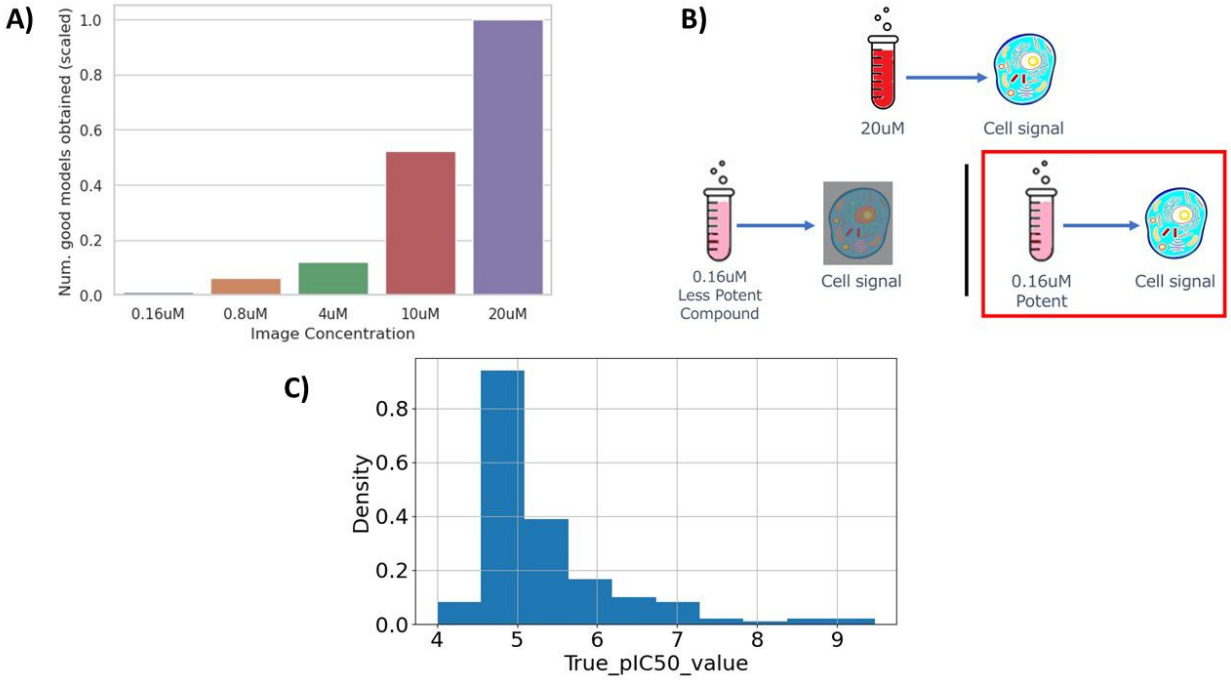



Figure 3.1: **A) Number of assays for which we could build classification models with ROC-AUC$\geq$ 0.8.** Results from a previous internal study. As image concentration increases, the more models with ROC-AUC$\geq$ 0.8 we obtain. Bar height are scaled by a factor equal to the original height of column '$20\mu M$'. This result was obtained by training the same multitask bioactivity model end-to-end with the identical training procedure for each image concentration. We also used the same evaluation procedure to calculate the ROC-AUC score for each task, subsequently quantifying the number of tasks per image concentration that reached a ROC-AUC$\geq$ 0.8. Details of the model, training procedure and evaluation procedure as can be found in Herman et al., under Experimental Procedures/Image-Informed Ligand-Based Model Building and Experimental Procedures/Model Evaluation. **B) Intuition of the behavior of low concentration images.** High concentration images show a lot more cell signals than low concentration images, which makes them more suitable for modelling. In low concentration images, only highly potent compounds induce signal from cell. **C) Distribution of true pIC50 values of 'assay 24' in the training set.** We consider compounds with pIC50$\geq$5 to be moderately potent, and pIC50$\geq$7 to be highly potent. Training a classifier for the latter tend to be more difficult than the former, due to data imbalance (few positive samples).

higher the ratio of classification models with a ROC-AUC$\geq$ 0.8 we obtained (Figure 3.1A). This result encourages using high concentration images such as $10\mu M$ and $20\mu M$ to train activity models.

Why do lower image concentrations result in poorer model performance than high image concentrations? Signal amplitude in the image features tends to increase with compound concentration (Figure 3.1B), and that a minimal signal amplitude is required for modelling. Less potent compounds, which are more abundant in chemical libraries, will require higher concentrations to induce model-compatible signal than more potent compounds, which are rarer. We propose that once a well performing model has been trained and validated at higher image concentrations, where enough compounds induce signal to enable modelling, it can be repurposed without retraining to detect similar but rarer signal at lower image concentrations, implying higher potency.

One benefit of repurposing a moderate-potency model for highly potent compound retrieval is overcoming data imbalance. Typically, when building a highly potent compound classifier, one very common problem is the shortage of positive samples (Figure 3.1C) affecting model training. As a result, we generally obtain much fewer good high-potency models than moderate-potency models. Our approach can facilitate highly potent compound retrieval in assays which only have enough data to train a good moderate-potency model.

Here, we investigate the behavior of models trained on $20\mu M$ images and inference performed with low concentration images $0.16\mu M$, $0.8\mu M$ and $4\mu M$. We show this approach can be used across a broad range of assays to repurpose a moderate-potency model for highly potent compound retrieval with high accuracy. In a drug discovery campaign, this can help prioritize more potent hits from virtual screening for experimental follow-up, and deprioritize compounds with potent off-target activities in the hit-triaging phase. We also compare our approach to the conventional method (directly building a high-potency classifier) in correctly classifying highly potent compounds. We show that this holds true for 65% of the assays in terms of AUC-ROC, and 75% of assays in terms of RIPtoP corrected AUC-PR.

## 3.2 Methods

### Data and Model Description

**Cell Painting Image Acquisition and Processing** We used the Janssen internal Cell Painting dataset, whose acquisition and processing steps can be found in Chapter 1.

**Multitask Image-based Activity Modelling** We form activity modelling as a multitask learning (MTL) problem, where a single model is trained on multiple tasks (binarized assays) simultaneously. The aim of MTL is to improve modelling capability of each task by leveraging the correlation of multiple tasks modelled jointly.

All modelling tasks are binary classification tasks, obtained from binarizing bioactivity assays which originally measures potency of a compound. Potency is measured as IC50 and correspond to the concentration at which a compound elicits 50% of a desired effect (e.g. inhibition). In line with previous literature, we use the pIC50 which is the IC50 on logarithmic scale, calculated as -log10(IC50).

In our case, we have over a hundred thousand Cell Painting profiles, and thousands of assays to train and evaluate the model. We set aside $< 5\%$ of the Cell Painting profiles for evaluation, and the rest is used for training. Compounds whose Cell Painting profiles are in the test set satisfy two criteria: 1) Have pIC50$\geq$8 in at least one of the bioassays, 2) Have a unique Murcko Scaffold[66] compared to the training set. The rest of the Cell Painting profiles are used for model training. Since the fill rate of the matrix is very low (around 5%), we use a masked Binary Cross-Entropy (BCE) loss where if we don't have activity information about a compound for a specific task, we exclude that (compound, task) pair from loss calculation.

**Training Set Folds Split** Cell Painting profiles in the training set are split into 5 folds, based on the Murcko Scaffolds of the original compound. We ensure that each fold only contains Cell Painting profiles of compounds which are structurally closest to each other.

These folds will be used for Mondrian Cross-Conformal Prediction, as described below.

**Model** We use an ensemble of 8 Multi-Layer Perceptrons (MLPs), calibrated using Mondrian Cross-Conformal Predictor (MCCP)[67]. Details of the ensemble and each MLP architecture can be found in the Supplementary Information, along with the hyperparameters for each model.

MCCP is a conformal prediction method designed to define prediction confidence in large imbalanced dataset, such as bioactivities. Our implementation of MCCP closely follows that in Sun et al.[67]. Given a calibration set, we define the **conformity measure** (CM) of a probability score output as $conformity\_measure(label, output) = |1 - label - output|$, where $label = 1$ indicates active, $label = 0$ indicates inactive, and $output$ is the probability score output from the model. Let the CM of the outputs in the calibration set for active class be $\alpha_1$, $\alpha_2$ ... $\alpha_n$, and for inactive class be $\beta_1$, $\beta_2$ ... $\beta_m$. If the CM for a new output is $\alpha_t$ (for label active) and $\beta_t$ (for label inactive),then p-values for active is $p_1^t = \frac{|i=1,..,n:\alpha_i \leq \alpha_t|}{n}$, and p-values for inactive is $p_0^t = \frac{|i=1,..,m:\beta_i \leq \beta_t|}{m}$.

In addition, a process similar to k-fold cross-validation is used. The training set is divided into k equal folds, one fold will be chosen as a calibration set for p-value calculation, and the other k-1 folds are used to train the model. The process is repeated k times, with each fold being used as the calibration set exactly once. The p-values from these repeats are then averaged to produce a single inactive p-value $p_0^t$ and active p-value $p_1^t$ for the new output.

Using significance $\varepsilon$, we obtain the conformal score for the new output as follows:

Active: $p_1^t > \varepsilon$ and $p_0^t \leq \varepsilon$

Inactive: $p_0^t > \varepsilon$ and $p_1^t \leq \varepsilon$

Uncertain: ($p_0^t > \varepsilon$ and $p_1^t > \varepsilon$) or ($p_0^t \leq \varepsilon$ and $p_1^t \leq \varepsilon$)

**Training and Inference Process** A visualization of the process can be found in the Supplementary Information. Given the test set and the training set which has been split into 5 folds, model training and inference process is as follows:

1) Train the model, which is an ensemble of 8 MLPs, on the entire training set so that we

can get the best model possible. We minimize masked BCE loss using Adam optimizer[68], and train the model for 34000 iterations. After that, compute probability scores for the test set.

2) Perform MCCP for the 5 folds of the training set. For each calibration fold choice, train the model on the other 4 folds. Calculate active and inactive p-values for each calibration fold, then average the p-values to obtain a single active p-value and inactive p-value that represents the overall level of significance of the probability scores.

3) Obtain the conformal scores for test set at significance $\varepsilon = 0.05$. The pipeline will return 1 for Active, -1 for Inactive, and 0 for Uncertain.

## Usage of Low Concentration Images for Highly Potent Compounds Retrieval

Our approach involves modifications to the above training and inference process: Using high concentration images for training and low concentration images for inference. This step repurposes a good moderate-potency model for highly potent compound retrieval. For example, if there is an assay with enough positive and negative samples to build a good model at potency threshold pIC50 $\geq 5$, and the aim is to retrieve highly potent compounds with pIC50 $\geq 7$, our approach is as follows:

**Step 1: Train a model using $20\mu M$ images to classify active compounds at potency threshold pIC50$\geq$5.** With abundant phenotypes from high concentration cell images, the model can learn to recognize signal specific to different bioassays. Besides, at this potency threshold 5, there are plenty of labelled data such that data imbalance is not an issue.

**Step 2: Inference using low concentration input images** As aforementioned, only highly potent compounds induce signals from cells at low compound concentration. Hence, if we switched to low concentration input images for inference, only highly potent compounds are identified as active.As a result, the model originally trained to classify compounds at

pIC50≥5 can be used to classify compounds at a higher potency threshold (e.g. pIC50≥ 7).

# Results

For conciseness, we name our model using this convention: [Train image concentration / Inference image concentration / train pIC50 label threshold]. For example, a model which uses $20\mu M$ images for training, $0.16\mu M$ images for inference, and learns to classify compounds with pIC50≥ 5 will be named as a $[20\mu M/0.16\mu M/5]$ model.

We report results on 57 assays. The criteria for choosing these 57 assays are:

- In the train set, each task from each assay has at least 100 labelled datapoints, and among those there are at least 25 actives and 25 inactives (at both potency threshold 5 and 7).
- In the test set, each task from each assay has at least 10 total labelled datapoints, and among those there are at least 5 actives and 5 inactives (at both potency threshold 5 and 7).
- We can obtain a good $[20\mu M/20\mu M/5]$ for this assay. (Criterion: AUC-ROC $\geq 0.7$ in a cross-validation process similar to Herman et al.[10].)

## Results From One Assay

Firstly, we give a detailed visualization of the results from one randomly chosen assay (named 'assay_24'). Our aim is to intuitively visualize the behavior of the model when using low concentration images for inference.

All stem plots in Figure 3.2 show the outputs of the models on the y-axis, and the true experimental pIC50 value on the x-axis. The black line denotes at what threshold pIC50 labels are binarized at, and the red area denotes the range of pIC50 we consider highly potent. For example, Figure 3.2A shows the result of a standard $[20\mu M/20\mu M/5]$ model. In

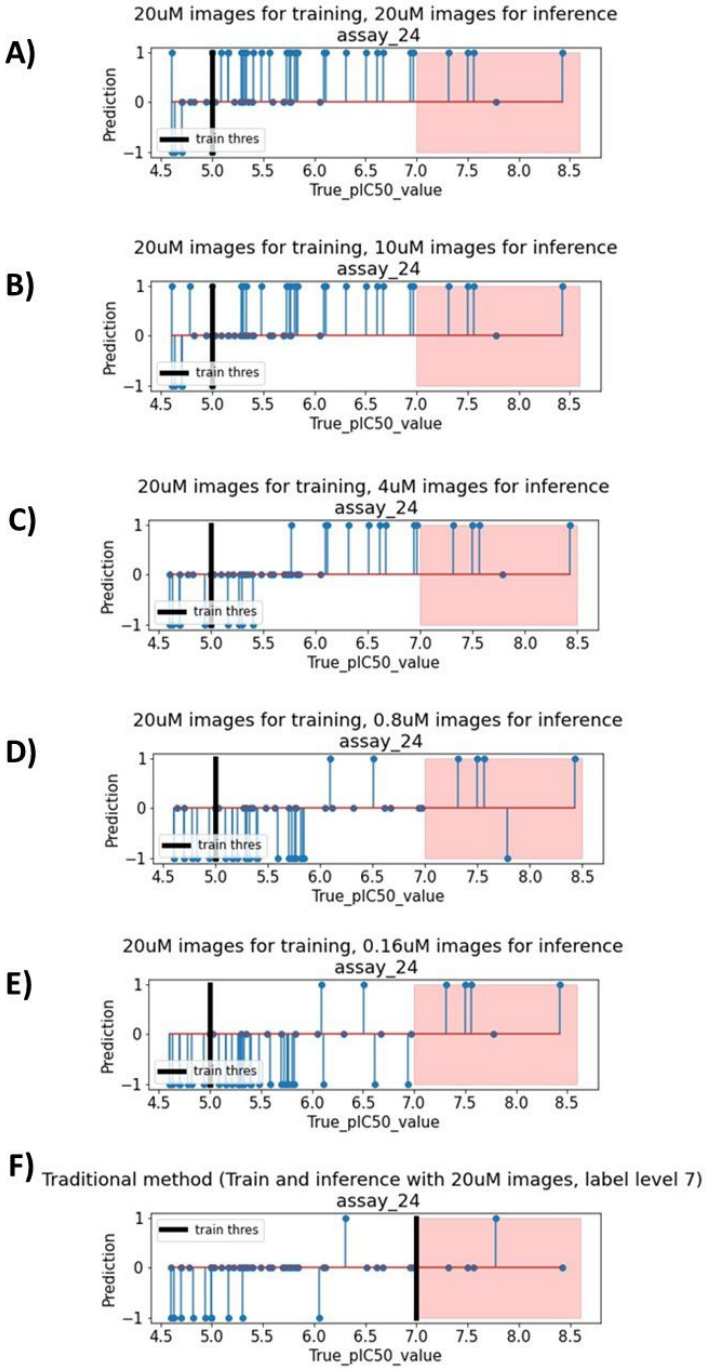



Figure 3.2: **Stem plots showing model conformal scores against the true pIC50 value.** For each plot, the y-axis denotes the conformal score for each compound. A score of 1 is positive, -1 is negative, and 0 is out of domain. The vertical black line denotes the potency threshold of the label the model was trained on. The red area denotes the range of pIC50 which we consider highly potent (in this case pIC50$\geq$7). The model behaves as a normal pIC50$\geq$5 classifier in plot A). But when using low concentration images for inference, the model specifically retrieves highly potent compounds in the red region, and skips over the moderately potent compounds. This behavior is particularly clear in plot D) and E). In fact, these two plots show, out of five highly potent compounds, our approach retrieve fours, whereas the conventional method in plot F) can only retrieve one.

this situation, the model behaves as expected: Most compounds to the right of the black line (True_pIC50_value$\geq$ 5) are classified as 1 (active), and compounds to the left of the black line are mainly classified as -1 (inactive) or 0 (uncertain) by the model. This is clearly a pIC50$\geq$ 5 model and using it for classifying pIC50$\geq$ 7 will result in many false positives. Figure 3.2C, Figure 3.2D and Figure 3.2E show what happens when we use the same $20\mu M$ pIC50$\geq$ 5 model, but perform inference with features from lower concentration images instead (still of the same compounds). It can be observed that the model now classifies less compounds as active, and tends to specifically retrieve compounds in the highly potent range (red area), skipping almost all of the moderately potent compounds. In fact, both models [$20\mu M$ / $0.16\mu M$ / 5] and [$20\mu M$ / $0.8\mu M$ / 5] classify highly potent compounds with high precision (4 correct positive classifications out of 6 positive classifications made by the model).

Figure 3.2F shows the result of a high-potency [$20\mu M/20\mu M/7$] model, dubbed as the 'conventional method' since this is what is typically done for retrieving compounds with pIC50$\geq$ 7. This conventional method employs the same training and inference process outlined in the Methods section, excluding the modification described in the "Usage of Low Concentration Images for Highly Potent Compounds Retrieval" subsection. We believe this is a suitable representation of current methods as it mirrors previous works [10, 21, 43, 22], that performed multitask bioactivity modeling using only one image concentration.

In this case, due to data imbalance affecting high-potency model (assay pIC50 distribution in Figure 3.1C), the model severely underperforms. It manages to only retrieve one highly potent compound (pIC50$\geq$ 7) out of five. In contrast, models using low concentration images for inference [$20\mu M/0.8\mu M/5$] and [$20\mu M/0.16\mu M/5$] can both retrieve four out of five highly potent compounds from this assay.

Based on the above results, we hypothesize two points regarding the use of low concentration images in retrieving highly potent compounds:

- **Hypothesis 1:** A good moderate-potency model e.g. [$20\mu M/20\mu M/5$] can be repurposed to specifically retrieve compounds with higher potency (e.g pIC50$\geq$ 7) simply by

| Assay Type | Assay Index |
|---|---|
| Other Target | 0, 3, 4 |
| Kinase | 1, 2, 6-13, 56 |
| Cell Proliferation | 5, 14-55 |

Table 3.1: **Assay type for 57 assays.**

switching to features from low concentration images for inference (without retraining the model).

- **Hypothesis 2:** In case of data imbalance adversely affecting training of a high-potency model, our approach can improve upon the high-potency model ('conventional method') in retrieving highly potent compounds.

We will further investigate these two points when showing results from other assays in the next sections.

## Results From 57 Assays

To assess **Hypothesis 1**, we calculate the precision in classifying highly potent compounds pIC50 $\geq 7$ (high-potency precision). This is because if e.g. the $[20\mu M/0.16\mu M/5]$ model is specifically retrieving highly potent compounds, then out of the compounds classified active by the model, the majority of them should be highly potent. In contrast, the $[20\mu M/20\mu M/5]$ model would have very low high-potency precision.

High-potency precision is calculated and plotted on a heatmap in Figure 3.3 for all 57 assays and displayed in the form of a fraction: #True Positives / #(True Positive + False Positive). For most assays, using low concentration images ($0.16\mu M$, $0.8\mu M$, $4\mu M$) for inference increases high-potency precision compared to the $[20\mu M/20\mu M/5]$ model. For models $[20\mu M/0.16\mu M/5]$ and $[20\mu M/0.8\mu M/5]$, high-potency precision in many assays is between 0.8 to 1. In addition, when using low concentration images for inference, fewer compounds are classified as active, but these compounds are very likely to be highly potent.

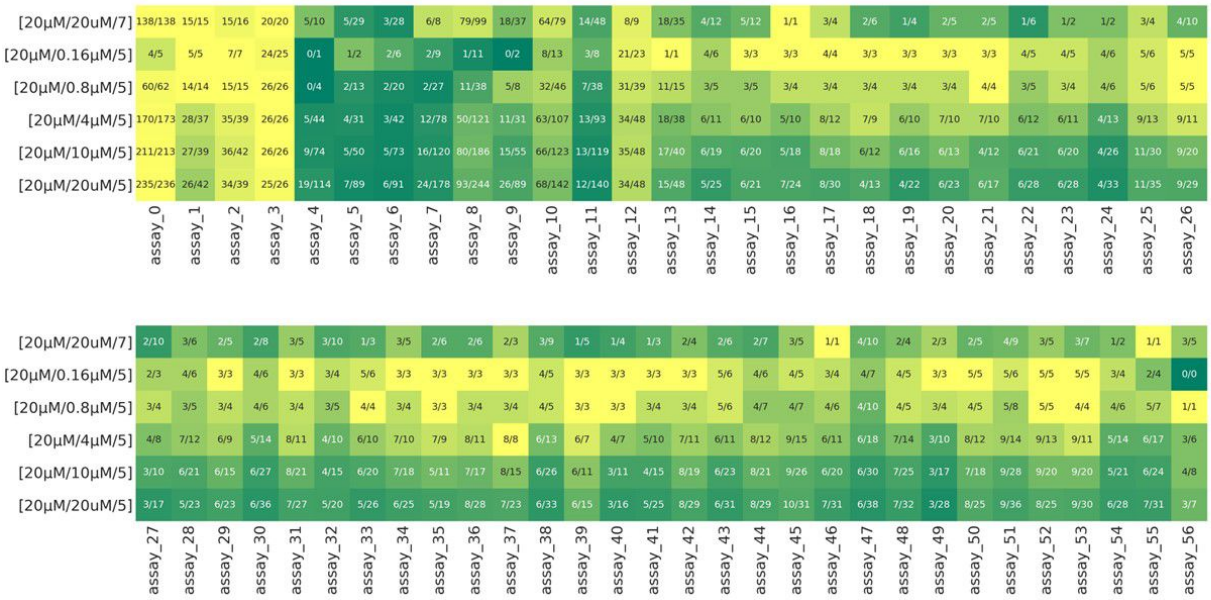



Figure 3.3: **High-potency precision heatmap.** Recorded high-potency precision of each model across 57 assays. As the inference image concentration decreases (moving from the sixth row to the second row), the model tends to be more precise at classifying highly potent compounds, resulting in a lighter color. This color gradient is consistent across all assays. The top rows are precision scores of the conventional method.


Figure 3.4: **High-potency recall heatmap.** Recorded high-potency recall of each model across 57 assays. As the inference image concentration decreases (moving from the sixth row to the second row), the model's recall decreases, resulting in a darker color. This color gradient is consistent across all assays. The top row are recall scores of the conventional method.

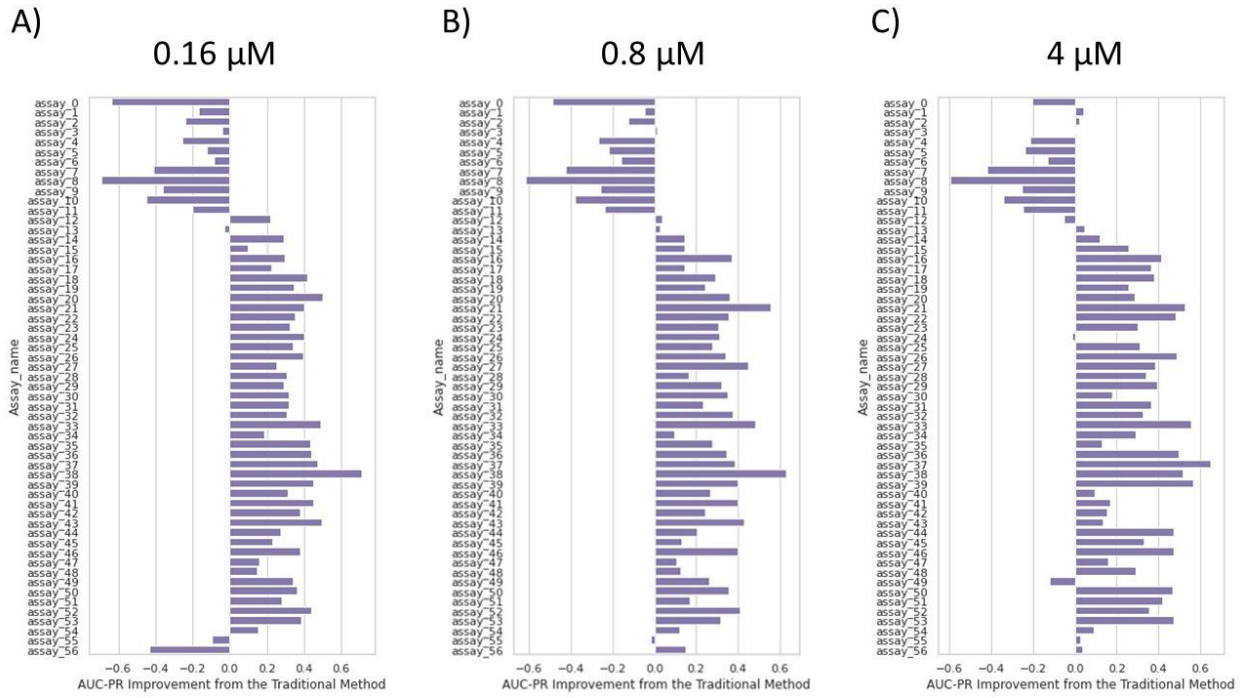



Figure 3.5: **AUC-PR improvement when using our approach compared to the conventional method.** Results across 57 assays. Our approach improves AUC-PR in 75% of assays investigated, with improvements around 0.2 to 0.5 compared to the conventional method.

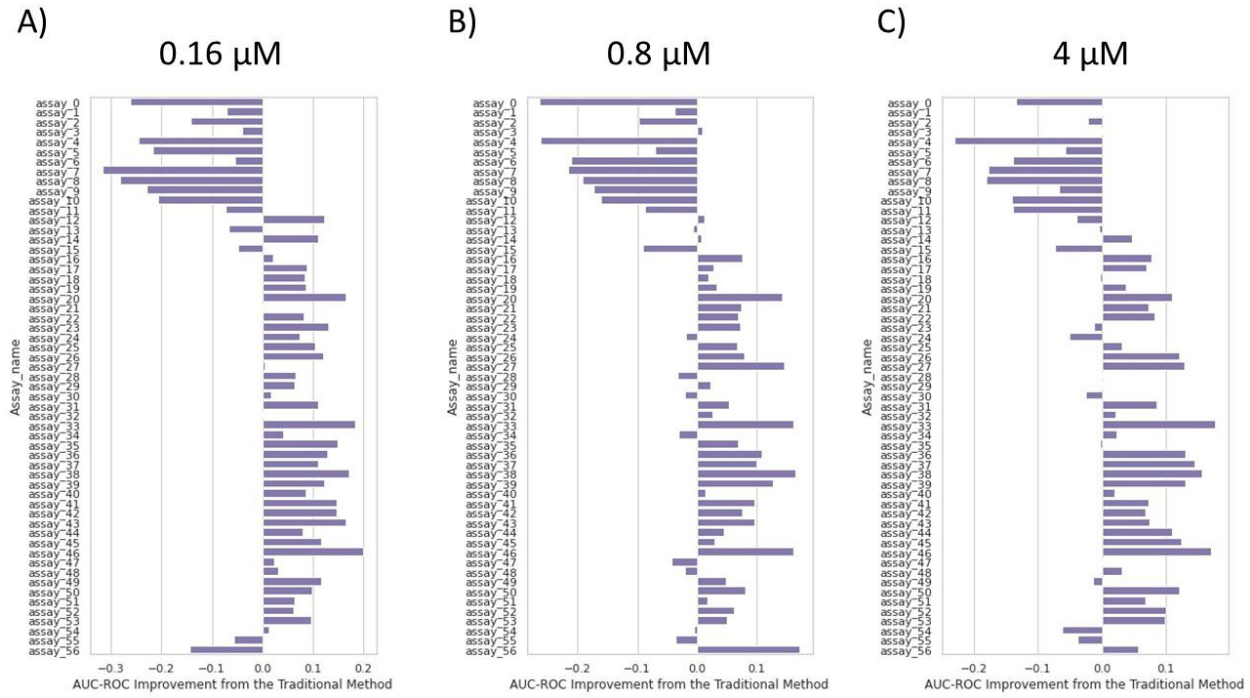



Figure 3.6: **AUC-ROC improvement when using our approach compared to the conventional method.** Results across 57 assays. Our approach improves AUC-ROC in 65% of assays investigated, with improvements around 0.1 to 0.2 compared to the conventional method.

It is also worth pointing out that high-potency precision of the models [$20\mu M/0.16\mu M/5$] and [$20\mu M/0.8\mu M/5$] is higher compared to the conventional method [$20\mu M/20\mu M/7$] (Figure 4 top row) in almost all 57 assays.

However, the improvement in high potency precision comes with decreasing recall (Figure 3.4) when using high concentration images for inference. Though notably among the low concentration images, $4\mu M$ images achieve significantly higher recall than $0.8\mu M$ and $0.16\mu M$ images while demonstrating recall comparable to, or only slightly worse than $10\mu M$ and $20\mu M$ images when used for inference with the moderate-potency models. In addition, high-potency recall of the conventional method (e 3.4 top row) is lower than those of model [$20\mu M/4\mu M/5$] for most of the 57 assays.

Overall, regarding hypothesis 1, we observe that using low concentration images for inference substantially improves high potency precision at the expense of recall compared to high concentration images. This pattern is consistent across all 57 assays investigated. In drug discovery practice, depending on the use case, higher precision may be more beneficial than higher recall. For example when using such models for an image-based virtual screen of a large compound library with a low throughput follow-up assay, or during hit triaging to deprioritize only compounds with potent off-target activities while minimizing the risk of false positive off-target flagging.

To assess **Hypothesis 2**, we propose to use Area under the ROC Curve (AUC-ROC) and Precision-Recall curve (AUC-PR). These are common metrics to compare classification models. we apply Relative Improvement of Proximity to Perfection (RIPtoP) correction to AUC-PR, as in Heyndrickx et al [69]. The idea of RIPtoP correction is that: since the random baseline of AUC-PR for each assay is the active ratio, one cannot compare AUC-PR across 2 different assays. RIPtoP correction enables cross-assay AUC-PR comparison by effectively 'rescaling' AUC-PR so that for every assay, 1 is the perfect model and 0 is the random baseline. The formula for RIPtoP correction is:

$$RIPtoP(AUC\text{-}PR) = \frac{AUC\text{-}PR - BASELINE}{1 - BASELINE}$$

Figure 3.5 shows how much AUC-PR improves over the conventional method when using for inference A) $0.16\mu M$, B) $0.8\mu M$ and C) $4\mu M$ images. Out of 57 assays, our approach improves AUC-PR by 0.2 to 0.5 in 42, 45 and 45 assays, respectively, over the conventional method, which equates to an improvement in AUC-PR in roughly 75% of assays investigated. Similarly, Figure 3.6 shows out of 57 assays, using A) $0.16\mu M$, B) $0.8\mu M$ and C) $4\mu M$ images for inference improves AUC-ROC in 40, 36 and 34 assays, respectively, over the conventional method. This equates to an improvement in AUC-ROC in approximately 65% of assays investigated, with AUC-ROC improvements generally around 0.1 to 0.2. Examining the assay types of the 57 assays in Table 3.1 reveals that the subset where our approach performs relatively poorly (assay 0 to assay 13, and assay 56) primarily consists of kinase and other target assays. In contrast, our approach significantly improves AUC-ROC and AUC-PR for cell proliferation assays.

Overall, hypothesis 2 holds for most but not all assays investigated (approximately 65% or 75% based on AUC-ROC and AUC-PR, respectively). Additionally, we find that hypothesis 2 primarily holds for cell proliferation assays.

## Case Studies

In the previous section, we report results for 57 assays chosen using a list of criteria listed at the beginning of the Result section. Since we need a systematic way to evaluate performance, the assay criteria are fairly rigid, hence there is low assay diversity. In fact, 43 out of 57 assays are Cell Proliferation assays. The others are 11 kinase assays, and 3 assays on other protein targets.

Therefore, in this section, to increase assay diversity, we manually pick a number of off- and on-target assays to test our low concentration image method on. These assays are different from the previous 57 assays in several ways: the pIC50 threshold where we consider a compound moderately or highly potent can be different, or there are plenty of

| Model Name | High Potency Precision | High Potency AUC-PR | High Potency AUC-ROC |
|---|---|---|---|
| $[20\mu M/0.16\mu M/4.6]$ | 0 (0/0) | 0 | 0.5 |
| $[20\mu M/0.8\mu M/4.6]$ | 0.429 (3/7) | 0.00512 | 0.513 |
| $[20\mu M/4\mu M/4.6]$ | 0.286 (26/91) | **0.0970** | **0.731** |
| $[20\mu M/10\mu M/4.6]$ | 0.243 (27/111) | 0.0689 | 0.693 |
| $[20\mu M/20\mu M/4.6]$ | 0.209 (27/129) | 0.0414 | 0.636 |
| $[20\mu M/20\mu M/6]$ | **0.462 (6/13)** | 0.0332 | 0.556 |

Table 3.2: **PLD Metrics Table** A high potency precision increase as inference image concentration decreases can be observed, indicating that the pIC50≥4.6 model has been repurposed for retrieving highly potent compounds with pIC50≥6. The best model in terms of high potency AUC-ROC and AUC-PR is $[20\mu M/4\mu M/4.6]$, outperforming the conventional method $[20\mu M/20\mu M/6]$.

positive samples, or the assays can even measure something that is not pIC50. Our aim is to demonstrate how we apply the method in these different settings, and in which setting the method works well.

## Phospholipidosis Assay

Drug-induced phospholipidosis (PLD) is characterized by the excess accumulation of phospholipids in tissues. Organs affected by phospholipidosis exhibit inflammatory reactions and histopathological changes[70]. Hence, PLD is considered an adverse effect and PLD assay is an essential liability assay to screen in early drug development.

For PLD assay, the model with the best AUC-ROC we have is the pIC50≥4.6 model, which is the lowest potency threshold considered for this assay. The model with the highest potency threshold is pIC50≥6, which has the worst AUC-ROC out of all our PLD models. The active ratio in the training set, when potency threshold is at 4.6 and 6, is 0.776 and 0.109, respectively. Our aim is to investigate whether we can repurpose a pIC50≥4.6 model to a model which specifically retrieves highly potent compounds pIC50≥6 (**Hypothesis 1**), and whether this method can outperform the pIC50≥6 model (**Hypothesis 2**).

It can be seen from the increase in high potency precision from 0.209 to 0.429 in Table 3.2

| Model Name | High Potency Precision | High Potency AUC-PR | High Potency AUC-ROC |
|---|---|---|---|
| $[20\mu M/0.16\mu M/4.5]$ | **1.0 (1/1)** | 0.0108 | 0.505 |
| $[20\mu M/0.8\mu M/4.5]$ | **1.0 (15/15)** | 0.161 | 0.581 |
| $[20\mu M/4\mu M/4.5]$ | **1.0 (62/62)** | 0.667 | 0.833 |
| $[20\mu M/10\mu M/4.5]$ | 0.908 (69/76) | 0.632 | 0.848 |
| $[20\mu M/20\mu M/4.5]$ | 0.721 (75/104) | 0.445 | 0.808 |
| $[20\mu M/20\mu M/5.5]$ | 0.895 (77/86) | **0.689** | **0.885** |

Table 3.3: **BSEP Metric Table** A high potency precision increase as inference image concentration decreases can be observed, indicating that the pIC50≥4.5 model has been repurposed for retrieving highly potent compounds with pIC50≥5.5. The best model in terms of high potency AUC-ROC and AUC-PR is $[20\mu M/20\mu M/5.5]$. In this case, our method does not outperform the conventional method.

that using low concentration images for inference can help specifically retrieve highly potent compounds, indicating that Hypothesis 1 holds. Although for this case, it is interesting to note that the model with the highest high potency precision is $[20\mu M/20\mu M/6]$ at 0.462. The low concentration model $[20\mu M/8\mu M/4.6]$ comes close at 0.429.

Hypothesis 2 holds, as the best model in terms of high potency AUC-PR and AUC-ROC is $[20\mu M/4\mu M/4.6]$, achieveing 0.0970 and 0.731, respectively. On the other hand, model $[20\mu M/20\mu M/6]$ AUC-ROC is close to the random baseline 0.5, likely due to few positives to train the model at pIC50 threshold 6, as the active ratio at potency threshold 6 is only 0.109. Interestingly, model $[20\mu M/20\mu M/6]$ achieves the highest high potency precision, but relatively low high potency AUC-PR and AUC-ROC at the same time. This is because this model misclassifies a lot more positives compared to other models.

## BSEP Assay

Bile salt export pump (BSEP) is the major transporter for the secretion of bile acids from hepatocytes into bile in humans. BSEP inhibition may contribute to the initiation of human drug-induced liver injury (DILI)[71]. Since DILI is a frequent cause of drug failure in development, early screening of BSEP is also vital in early drug discovery.

For BSEP assay, pIC50≥4.5 and pIC50≥5.5 classifiers are the lowest and highest potency BSEP models that we build. The active ratio in the training set, when potency threshold is at 4.5 and 5.5, is 0.825 and 0.400, respectively. We are investigating whether we can repurpose the pIC50≥4.5 model to retrieve highly potent compounds at pIC50≥5.5 (**Hypothesis 1**), and whether our method outperforms the high potency pIC50≥5.5 model (**Hypothesis 2**).

**Hypothesis 1** holds, as shown in Table 3.3. Models $[20\mu M/0.16\mu M/4.5]$, $[20\mu M/0.8\mu M/4.5]$ and $[20\mu M/4\mu M/4.5]$ specifically retrieve compounds with pIC50≥5.5, with significantly less false positives than the $[20\mu M/20\mu M/4.5]$ model.

In this case, the high potency model pIC50≥5.5 is the best model in terms of high potency AUC-PR and AUC-ROC (Table 3.3), indicating that there are still enough positive samples for training of the high potency model. Our method does not lead to an improvement in high potency AUC-PR or AUC-ROC, hence **Hypothesis 2** does not hold.

## Immunology On-Target Assay

This is an assay for an immunology protein target. pIC50≥5.3 and pIC50≥6 classifiers are the lowest and highest potency models that we build for this assay. Hence, for this case we are investigating whether we can repurpose the pIC50≥5.3 model to retrieve highly potent compounds at pIC50≥6 (**Hypothesis 1**), and whether our method outperforms the high potency pIC50≥6 model (**Hypothesis 2**). The active ratio in the training set, when potency threshold is at 5.3 and 6, is 0.437 and 0.0765.

In terms of **Hypothesis 1**, we observe a trend of increasing high potency precision as image concentration for inference decreases (Table 3.4), indicating that the model can be repurposed for highly potent compound retrieval. However, the high potency precision increase for this assay is smaller than in other cases, from 0.739 at $20\mu M$ to 0.757 at $0.8\mu M$, and the increase is not as monotonic.

Despite the low active ratio 0.0765, the high potency model $[20\mu M/20\mu M/6]$ performs well with AUC-ROC score of 0.718 and AUC-PR score of 0.258. But our method, specifically

| Model Name | High Potency Precision | High Potency AUC-PR | High Potency AUC-ROC |
|---|---|---|---|
| $[20\mu M/0.16\mu M/5.3]$ | 0.667 (10/15) | 0.183 | 0.652 |
| $[20\mu M/0.8\mu M/5.3]$ | **0.757 (28/37)** | **0.347** | 0.821 |
| $[20\mu M/4\mu M/5.3]$ | 0.709 (39/55) | 0.328 | **0.861** |
| $[20\mu M/10\mu M/5.3]$ | 0.698 (37/53) | 0.313 | 0.846 |
| $[20\mu M/20\mu M/5.3]$ | 0.739 (34/46) | 0.278 | 0.804 |
| $[20\mu M/20\mu M/6]$ | 0.727 (16/22) | 0.258 | 0.718 |

Table 3.4: **Immunology Target Metrics Table.** A high potency precision increase as inference image concentration decreases can be observed, indicating that the pIC50≥5.3 model has been repurposed for retrieving highly potent compounds with pIC50≥6. However, the increase is smaller and not as monotonic as the previous cases. The best model in terms of high potency AUC-ROC is $[20\mu M/4\mu M/5.3]$, and in terms of high potency AUC-PR is $[20\mu M/0.8\mu M/5.3]$, both outperforming the conventional method $[20\mu M/20\mu M/6]$.

models $[20\mu M/0.8\mu M/5.3]$ and $[20\mu M/4\mu M/5.3]$, improve on these scores, achieving 0.347 AUC-PR and 0.821 AUC-ROC, and 0.328 AUC-PR and 0.861 AUC-ROC, respectively. This indicates **Hypothesis 2** holds for this assay.

## Glu/Gal Assay

Another important off-target assay in early drug development is Glu/Gal. Glu/Gal is the primary assay of choice for drug-induced mitochondrial toxicity[72]. Briefly, mitochondrial dysfunction is determined by the ratio of the test substance to induce cytotoxicity in glucose and galactose culture conditions[10], hence the Glu/Gal nomenclature. The measure of mitochondrial toxicity is a ratio of two IC50 values, not one pIC50 value as in previous cases. The higher the Glu/Gal ratio is, the more indicative that the compound induces mitochondrial toxicity.

Instead of retrieving highly potent compounds, in this case we will test whether our method can specifically retrieve highly toxic compounds from this assay. We consider compounds with Glu/Gal ratio ≥5 to be highly toxic, and ratio ≥2 to be moderately toxic. The active ratio in the training set, when toxicity threshold is at 2 and 5, is 0.595 and 0.275,

| Model Name | High Toxicity Precision | High Toxicity AUC-PR | High Toxicity AUC-ROC |
|---|---|---|---|
| $[20\mu M/0.16\mu M/2]$ | 0 (0/0) | 0.0 | 0.5 |
| $[20\mu M/0.8\mu M/2]$ | **1 (2/2)** | 0.0155 | 0.508 |
| $[20\mu M/4\mu M/2]$ | 0.654 (17/26) | 0.0679 | 0.552 |
| $[20\mu M/10\mu M/2]$ | 0.553 (57/103) | 0.165 | 0.650 |
| $[20\mu M/20\mu M/2]$ | 0.483 (87/180) | 0.186 | 0.693 |
| $[20\mu M/20\mu M/5]$ | 0.531 (85/160) | **0.226** | **0.713** |

Table 3.5: **Glu/Gal Metrics Table** High potency precision tends to increase as inference image concentration decreases. This indicates the Toxicity$\geq$2 model has been repurposed for retrieving highly potent compounds with Toxicity$\geq$5. The best model in terms of high potency AUC-ROC and AUC-PR is $[20\mu M/20\mu M/5]$. In this case, our method does not outperform the conventional method.

respectively. It is also worth noting that the Glu/Gal ratios are distributed on a wide range (up to 500), but we are only interested in the thresholds in the narrow range of 2 to 5. This is because we consider every compound with ratio $\geq$5 to be equally toxic. As a result, compounds in the test set will only have Glu/Gal ratios between 0 to 20.

It can be observed in Table 3.5 that high toxicity precision increases from 0.483 of model $[20\mu M/20\mu M/2]$ to 1 of model $[20\mu M/0.8\mu M/2]$. This shows using low concentration images for inference can specifically retrieve highly toxic compounds. This shows that **Hypothesis 1** holds for this assay. We note that similar to PLD, in this assay inference using $0.16\mu M$ images returns no active compounds. It is because the compound concentration is too low that no signal related to these activities are induced in the cell.

Regarding **Hypothesis 2**, the high toxicity model $[20\mu M/20\mu M/5]$ remains the best performing model (Table 3.5) at 0.226 high toxicity AUC-PR and 0.713 high toxicity AUC-ROC. In this case, 27% of the labels are positive for classification of highly toxic compounds with Glu/Gal ratio$\geq$5, which is plenty of positive examples for a high toxicity model training. Our method does not outperform the connventional method for this assay.

## 3.3 Discussion

We have presented an approach, improving on an existing image-based small molecule activity optimization pipeline, to specifically retrieve highly potent compounds in a biological assay. We start with training a moderate-potency model with $20\mu M$ cell painting images to classify compounds with pIC50 at a potency threshold low enough so that there are still plenty of positive examples to train effectively. Then, we repurpose that well-performing model for higher potency classification, by performing inference using lower concentration images as input. In terms of application in the drug discovery pipeline, being able to classify highly potent compounds accurately can help prioritizing hits from screening for experimental follow-up based on potency. It can also help deprioritizing compounds with potent off-target activities in the hit-triaging phase. However, it should be mentioned that the improvement in retrieval of highly potent compounds with this approach comes at a cost for data generation since to benefit from our approach, additional cell painting images of different concentrations are required.

We highlighted two points that our approach can achieve. **Hypothesis 1** is that a good moderate-potency model can be repurposed to specifically retrieve compounds with higher potency, by using features from low concentration images for inference without retraining the model. We assess this point by using precision when classifying a highly potent compound, on 57 assays and 4 additional assays in the Case Studies section. We found that this behavior can be observed in almost all assays we tested on, with the majority of them being cell proliferation assays. Although using low concentration images for inference retrieves fewer active compounds, these compounds tend to be highly potent.

**Hypothesis 2** was that if data imbalance adversely affected a high potency model training, our approach could outperform the high potency model (conventional method) in classifying highly potent compounds. We assessed this point on the same selection of assays as above, using AUC-ROC and corrected AUC-PR as metrics. We found that this point holds

for around 65% to 75% of those assays. Overall, AUC-ROC scores increase by around 0.1 to 0.2, and AUC-PR scores increase by around 0.2 to 0.5, indicating an improvement over the conventional method in the majority of assays. Our approach can serve as a replacement for a conventional high potency model when activity labels for training are scarce.

# Chapter 4

# Cross Modality Learning of Cell Painting and Transcriptomics Data Improves Mechanism of Action Clustering and Bioactivity Modelling

In Janssen RNA-Seq screen is run on the same compounds as Cell Painting. Currently there are a lot of multimodal learning out there. We believe we could apply some of those techniques to perform representation learning using CP and TX. For each cell sample, the TX profile is a vector of length around 20000, each corresponds to a gene in the human transcriptome.

## 4.1 Motivation

Self-supervised representation learning is an important aspect of machine learning[60, 61, 73, 35], leveraging abundant unlabeled data to help models learn embeddings that capture underlying structures and patterns. These embeddings are beneficial for a variety of downstream tasks where labels for supervised learning is rare. In drug discovery, learning useful representations from chemical[74, 20] or biological data[37, 28] is similarly useful. This is because learned representations can not only improve performance in downstream modeling tasks with few labelled data, but also enhance our understanding of chemistry, biology, and

their interactions.

In small molecule drug discovery, high content screens such as cell painting [9] or RNA-Seq[75] are often used to quantify the changes in the biological system induced by a perturbation (e.g. after application of a treatment). The results are morphological profiles (CP data) from cell painting or gene expression profiles (TX data, short for 'Transcriptomics') from RNA-Seq for compounds, which can be further analyzed to enhance our understanding of the biological effect of different drugs. Application of CP data includes modelling of small molecules activity in different biological assays [21, 10, 13], and determining mechanism of actions of compounds[4, 5, 6]. Application of TX data includes annotating cell types[74], identifying differentially expressed genes to study their function[76, 77], and discovering potential novel biomarkers[78].

Multimodal learning is a subfield in machine learning that aims to process and integrate data from multiple modalities, originally with the goal to mimic the way human combine information from different senses (sight, sound, touch, etc.). Multimodal learning has made significant progress in recent years[79], from language-vision models[80, 81], video captioning[82], autonomous driving[83], to biomedical AI[84]. Underlying these rapid developments is the increasing adoption of self-supervised learning (SSL) methods [85], which allowed for models to be trained at an unprecedented scale. Instead of relying on expensive human annotated labels, SSL generates supervision from abundantly available unannotated data, and is then finetuned to a variety of downstream tasks of interest with little data. This technology has potential to be extremely beneficial in drug discovery, as there is a large diversity of data sources such as chemical structure, cell images, omics, quantum chemistry, etc. There has been research on multimodal learning of chemical structure-cell painting [27, 28, 29, 30], and between single-cell RNA-seq and chromatin images [37]. However, as far as we know, multimodal learning of cell painting (CP) data and bulk transcriptomics (TX) data are still unexplored.

In this chapter, we study cross modality learning[86] of CP and TX data: a multimodal

representation learning setting where we try to learn better single modality representations from CP data given unlabeled data from both CP and TX. The reason for this is that for new compounds, we would most likely only have CP data and not TX, because generating TX data is much more costly than CP. Because of that, any representation learning algorithm must be able to learn representation on both modalities, but only need CP data for embedding generation. More specifically, we benchmark two cross modality representation learning methods: contrastive learning (CL) and bimodal autoencoder (BAE), on a variety of unsupervised and supervised downstream tasks. We show that learned representation improves cluster quality for clustering of CP replicates and different modes of action (MoA), with CL embedding yielding the best results. In the supervised bioactivity multitask classification, we demonstrate that CL embedding achieves higher mean AUROC and RIPtoP-AUPRC compared to CP feature across a range of bioactivity tasks. Additionally, we provide a more detailed comparison of feature performance on bioactivity tasks grouped by protein target families. Finally, we show that in the absence of TX features for new compounds, using learned embeddings enhances performance of CP feature on tasks where TX feature excels but CP feature does not.

### 4.1.1 Data Generation

#### 4.1.1.1 Cell Painting Image Acquisition and Processing

We used the Janssen internal Cell Painting dataset, whose acquisition and processing steps can be found in Chapter 1.

#### 4.1.1.2 Transcriptomics Data Generation with Bulk RNA-Seq

U2OS cells were seeded at a density of 1600 cells/well in 384 plates on day 1. 24 hours later, on day 2, compound treat was performed with the compounds of interest + DMSO (vehicle) using the Echo Liquid Handler. Again, 24 hours later, on day 3, medium was

removed from the cells using the BlueWasher (Lightspin) followed by addition of 20$\mu$l of Cells-To-Signal lysis buffer (diluted 1/3 in PBS) per well with Multidrop. Plates were kept for 10 minutes at room temperature to get proper lysis and eventually stored at -80°$C$. After lysis, the actual HT-RNAseq protocol started. In the first step of the protocol, double-stranded complementary DNA (cDNA) was synthesized from each well with each sample being uniquely barcoded. After that, all wells from a 384 well plate were pooled together to one single tube followed by a Kapa HyperPlus Library preparation using Illumina-compatible adapters following the manufacturer's introductions. Afterwards, libraries were sequenced on an Illumina NovaSeq 6000 S4 instrument to an average depth of 1.106 reads per well.

Sequencing reads were aligned using STAR solo v2.7.6a [87] with default parameters, except for: (outFilterMultimapNmax: 1, soloCBstart: 1, soloCBlen: 10, soloUMIstart: 11, and soloUMIlen: 10), towards the human reference genome GRCh38.102 extended with a list of 92 ERCC spike-ins. The resulting UMI counts were analysed using R. After filtering for Havana and Ensembl genes, variance stabilizing transformation was applied to all samples within a plate, followed by a library size correction, taking compound and dose treatment into account, using DESeq2[88] and limma[89] R packages. Relative data versus vehicle was then calculated for each treatment within a plate such as robust adjusted Z-scores used in our work: [Sample – Median(DMSO)) / StandardDeviation(DMSO)].

### 4.1.2 Evaluation Methodology

In our dataset, each unique compound has one TX profile and several CP profiles due to replicate measurements, with the number of replicates varying across compounds. We group these into (CP, TX) pairs, resulting in just over 200k pairs for approximately 100k compounds. We then split the data into training, validation, and test sets (70/10/20) based on the Murcko scaffolds[66] of the compounds, ensuring that structurally similar compounds are in the same set. To enable a fair comparison of each feature learning algorithm, we perform feature learning on the training and validation sets and evaluate their performance

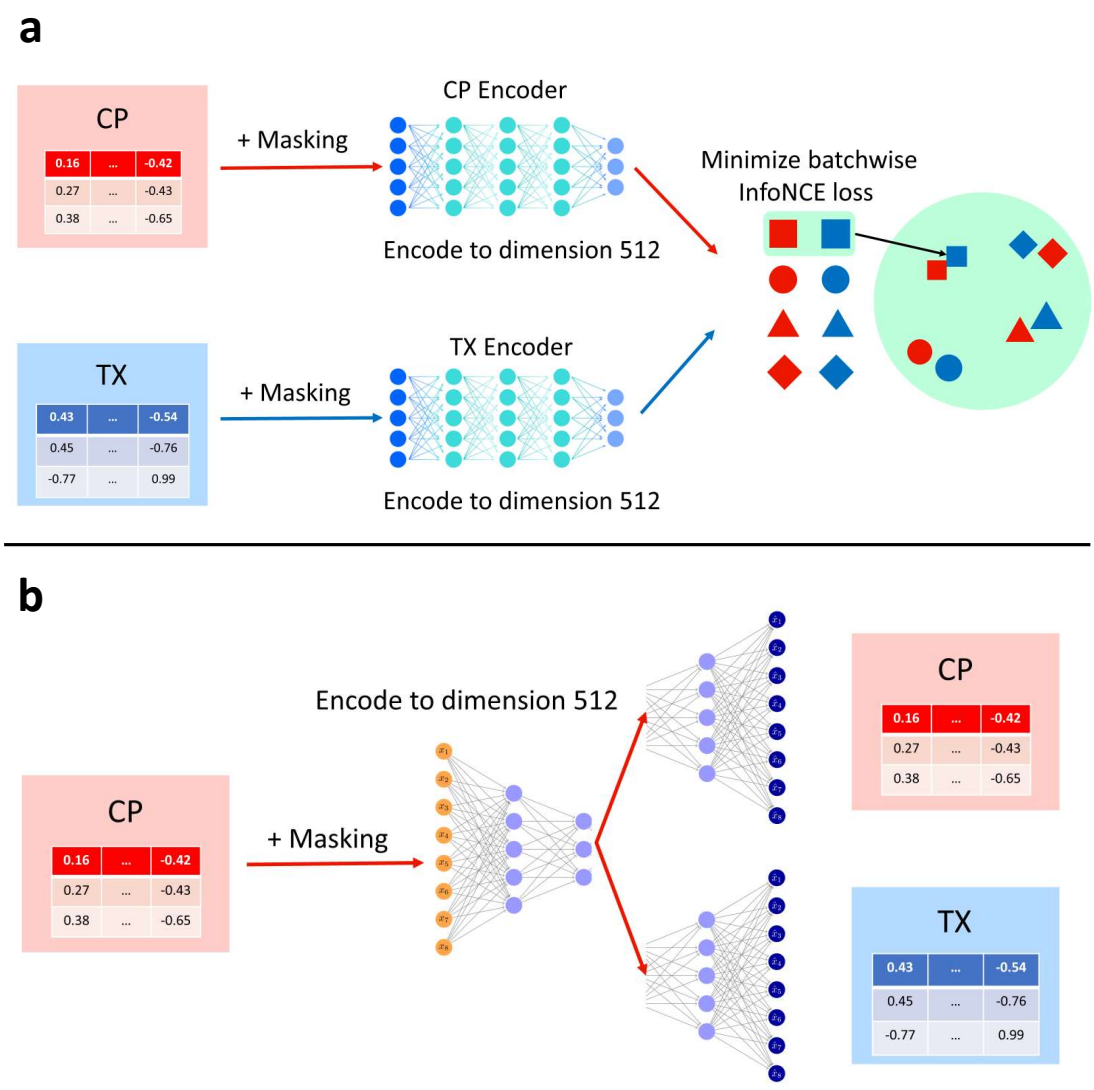



Figure 4.1: **Schema of 2 different pretraining methods.** a) Contrative learning. b) Bimodal autoenoder. For both pretraining methods data are augmented with 10% masking before being encoded. Contrastive learning pretraining aims to minimize the InfoNCE loss between pairs of the same compounds and pairs of different compounds in a batch. Bimodal autoenoder pretraining aims to minimize the average of mean square error of each reconstruction.

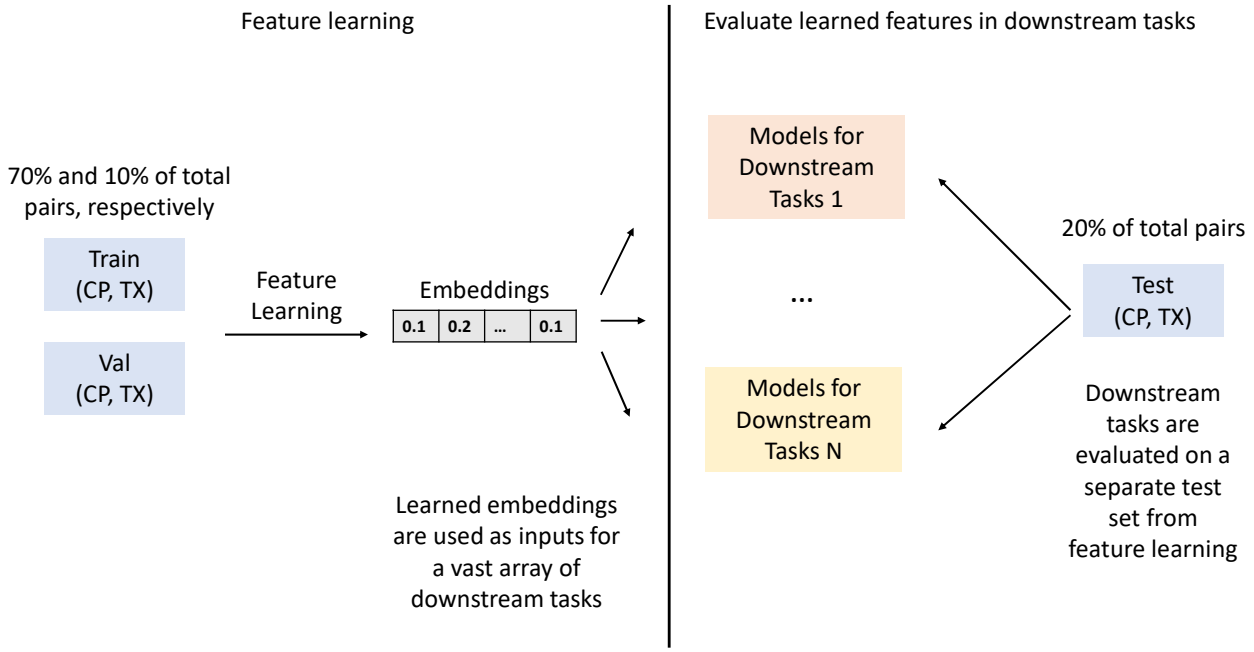



Figure 4.2: **Evaluation methodology.** We split the dataset into train, validation and test set with the ratio 70/10/20. We perform feature learning on the train and validation tasks, and evaluate the learned embedding against the original CP and TX features on the test set.

across various downstream tasks using the hold-out test set(Figure 4.2).

### 4.1.3 Contrastive Pretraining

The first feature learning algorithm we benchmark is Contrastive learning (CL): a method that involves optimizing the InfoNCE objective[73, 61] across two learned embeddings from two separate encoders. For each batch with size $N$, the InfoNCE is as follows:

$$L_{\text{InfoNCE}} = -\frac{1}{N}\sum_{i=1}^{N} \ln \frac{\exp\left(\text{sim}\left(\mathbf{x}_i, \mathbf{z}_i\right)/\tau\right)}{\sum_{j=1}^{N} \exp\left(\text{sim}\left(\mathbf{x}_i, \mathbf{z}_j\right)/\tau\right)} - \frac{1}{N}\sum_{i=1}^{N} \ln \frac{\exp\left(\text{sim}\left(\mathbf{x}_i, \mathbf{z}_i\right)/\tau\right)}{\sum_{j=1}^{N} \exp\left(\text{sim}\left(\mathbf{x}_j, \mathbf{z}_i\right)/\tau\right)},$$

where $\text{sim}\left(\mathbf{x}_i, \mathbf{z}_i\right) = \mathbf{x}_i^T \mathbf{z}_i / ||\mathbf{x}_i|| ||\mathbf{z}_i||$ is the cosine similarity between the output of CP encoder $\mathbf{x}_i$ and output of TX encoder $\mathbf{z}_i$. $\tau$ is a temperature parameter which controls how concentrated the features are in the representation space [90]. Smaller $\tau$ will make the loss concentrates more on small distances, such as similar embeddings of different compounds. Too small $\tau$ will make widely separated representations irrelevant, which heavily degenerate

the performance.

InfoNCE maximizes the similarity between the correct pairs (CP and TX profile of the same compound) and minimizes the similarity between the other random (CP, TX) pairs in the latent space. In our cross modality setting, we use both CP and TX data to pre-train the two CP and TX encoders (Figure 4.1). When generating embeddings for new compounds that do not have TX data, we would only need the CP encoder.

Our contrastive learning framework (Figure 4.1a) includes a CP encoder and a TX encoder, both of which are fully-connected neural networks (MLPs). The CP encoder has hidden layers of size [1024, 1024, 1024], while the TX encoder has hidden layers of size [4096, 4096, 4096]. The embedding size is 512. Since data augmentation is often important in contrastive learning[61, 91], we apply an augmentation to the CP and TX data before encoding: randomly masking 10% of the input features. In addition, we use a linear projection to map from each encoder's representation (dimension 512) to the embedding space (dimension 256), similar to the CLIP model from Radford et al[60]. For embedding generation, the projection head is discarded and we only use the MLP encoder. The model was trained on two NVIDIA Tesla M60 8GB GPUs, which took approximately 5 days to complete.

### 4.1.4 Bimodal Autoencoder Pretraining

Bimodal autoencoder[86] (BAE) involves one encoder for CP data, and two decoders for CP and TX data. The output of the CP encoder is the learned embedding (also known as 'latent vector' in autoencoder literature), which can be generated with the CP encoder using only CP data for new compounds. The model is trained to reconstruct both modalities when given only CP data, minimizing the average mean squared error (MSE) of the two reconstructions:

$$L_{\text{MSE-BAE}} = \frac{1}{2}\sum_{i=1}^{N}(\mathbf{u'}_i - \mathbf{u}_i)^2 + \frac{1}{2}\sum_{i=1}^{N}(\mathbf{v'}_i - \mathbf{v}_i)^2,$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are original CP and TX data, and $\mathbf{u'}_i$ and $\mathbf{v'}_i$ are CP and TX reconstructions (output of the CP decoder and TX decoder).

The CP encoder, CP decoder and TX decoder have hidden layers [1024, 512, 512], [512, 512, 1024] and [1024, 2048, 4096] (Figure 4.1b). The size of the embedding is 512. We also apply an augmentation that randomly mask 10% of the input features. This can be thought of as a masked autoencoder. The model was trained on two NVIDIA Tesla M60 8GB GPUs, which took approximately 9 days to complete. More details about hyperparameters for both pretraining methods can be found in the Supplementary Information.

### 4.1.5 Metrics

#### 4.1.5.1 Binary Classification

To assess model performance in binary classification downstream tasks, we use Area under the ROC Curve (AUROC) and Precision-Recall curve (AUPRC). We apply Relative Improvement of Proximity to Perfection (RIPtoP)[69] correction to AUPRC to 'rescale' AUPRC so that for every assay, 1 is the perfect model and 0 is the random baseline. The formula for RIPtoP correction is:

$$RIPtoP(AUPRC) = \frac{AUPRC - BASELINE}{1 - BASELINE}$$

#### 4.1.5.2 Clustering Quality

Assessing which feature type produces better cluster quality in unsupervised downstream tasks is typically done with human judges. They examine a 2-D plot obtained from a dimension reduction algorithm (e.g., t-SNE) for expected clusters. This approach, however, cannot be practically applied when there are a lot of data points and clusters which can

be unclear to the human eye. Hence, we also use a systematic metric for clustering quality called kNN accuracy (k-nearest neighbour accuracy) [28].

The intuition behind this metric is, if one feature type produced higher-quality cluster than another, fitting a kNN classifier on that feature type (with the label being the true cluster label) would yield higher accuracy. For our implementation, we use the function KNeighborsClassifier with default setting (neighbors = 5) from scikit-learn[55] to produce cluster classifications. Then accuracy is calculated between them and the true cluster labels.

## 4.2 Results

Table 4.1: **Clustering quality of each feature type, as measured by kNN accuracy in two unsupervised downstream tasks.** Contrastive learning embedding demonstrates superior clustering of CP replicates and different MoA over the original CP feature. BAE embedding only improves kNN accuracy by a small amount over CP feature.

| Feature Type | kNN Accuracy CP Replicates | kNN Accuracy MoA |
|---|---|---|
| CP | 0.416 | 0.784 |
| CL Emb | **0.805** | **0.952** |
| BAE Emb | 0.428 | 0.784 |

### 4.2.1 CP Replicates Clustering

This is an unsupervised task that investigates how well each feature type cluster CP replicates. As mentioned in the previous section, each compound has several CP replicates. As a result, when we perform unsupervised analysis, CP replicates of the same compound are assumed to lie in the same cluster, and CP replicates of different compounds are assumed to lie in different clusters.

We calculate kNN accuracy for compounds in the test set for each feature type, with the true cluster labels being test set compounds (Table 4.1). CL embedding achieves the highest

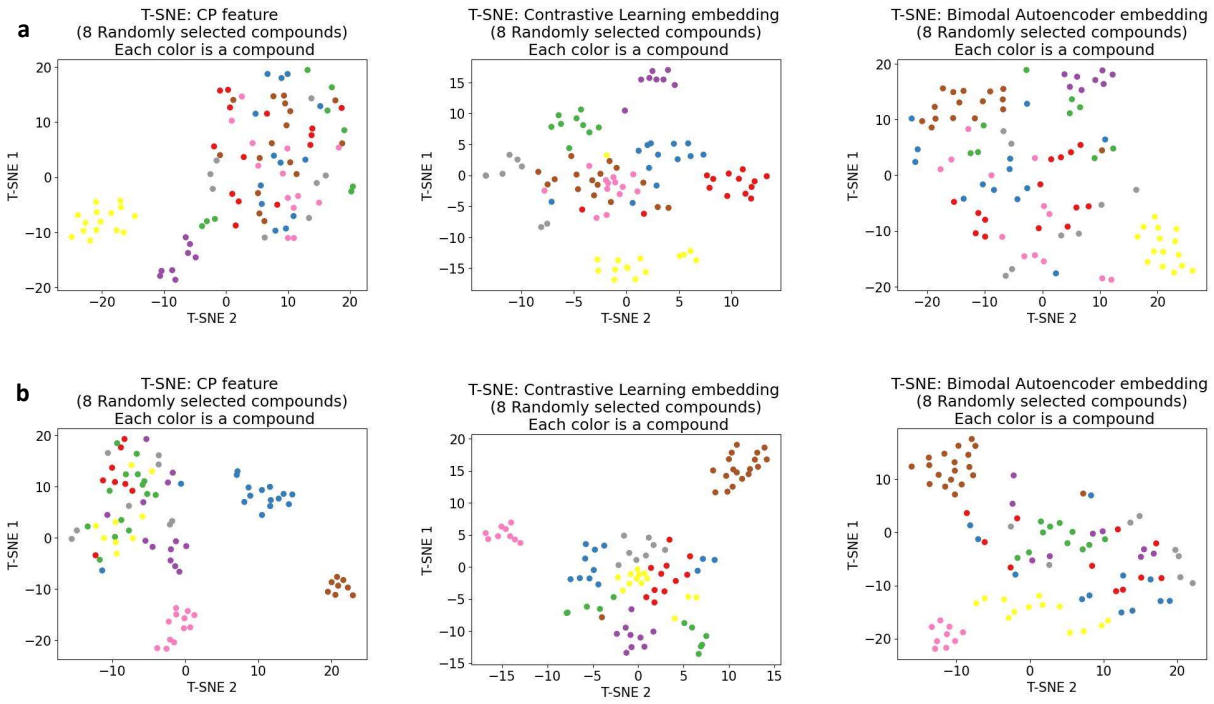



Figure 4.3: **T-SNE plots demonstrating replicate clustering ability of CP embeddings for a) 8 randomly selected compounds. b) other 8 randomly selected compounds.** It can be observed that the embeddings, especially CL, improves clustering ability of CP replicates over the original CP feature.

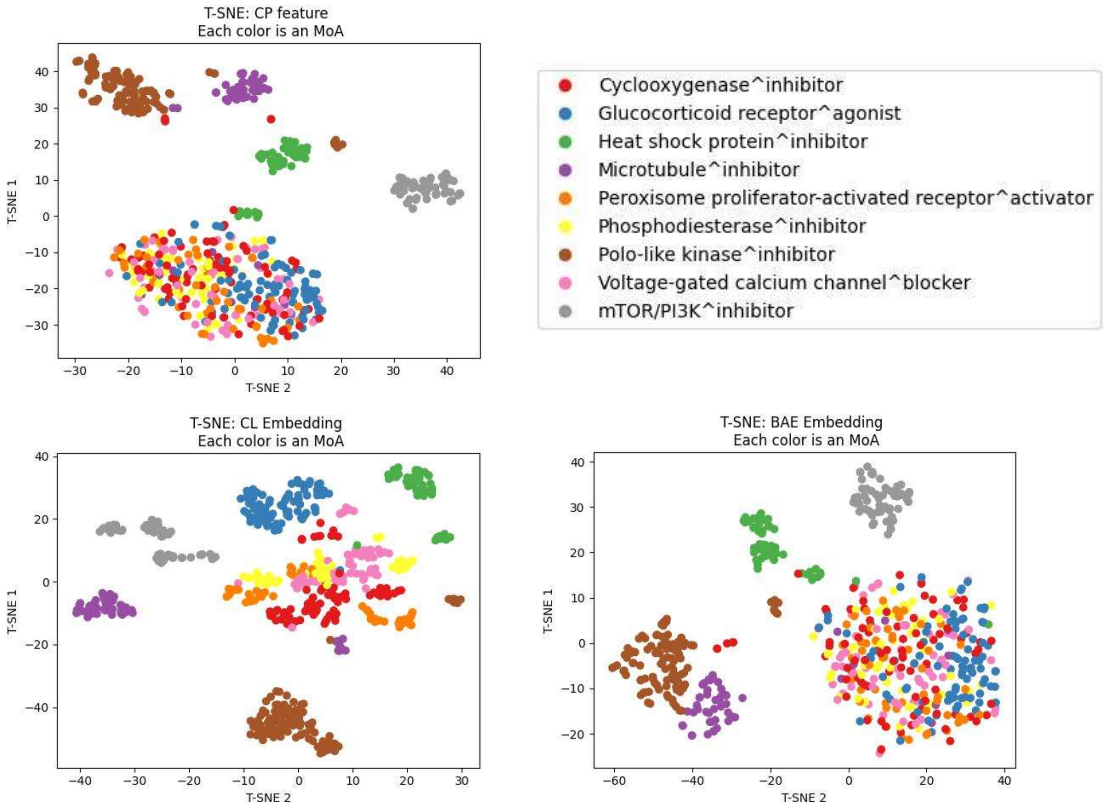



Figure 4.4: **T-SNE plots demonstrating MoA clustering ability of CP feature, CL embedding and BAE embedding.** Visually, CL embedding greatly improves cluster quality. The improvement is the most pronounced for Glucocoticoid receptor agonist and the Voltage-gated calcium channel blocker clusters.

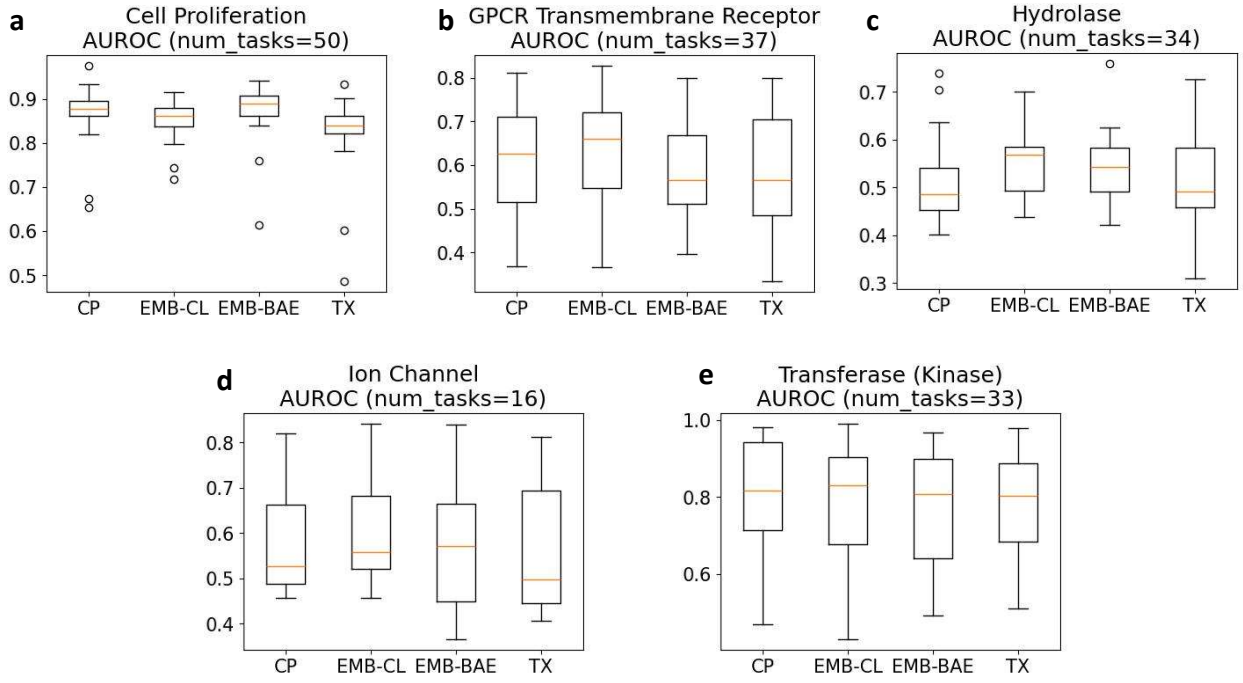



Figure 4.5: **Box plots AUROC for tasks grouped by protein target family.** a) Cell Proliferation, b) GPCR Transmembrane Receptor, c) Hydrolase, d) Ion Channel, e) Transferase (Kinase).

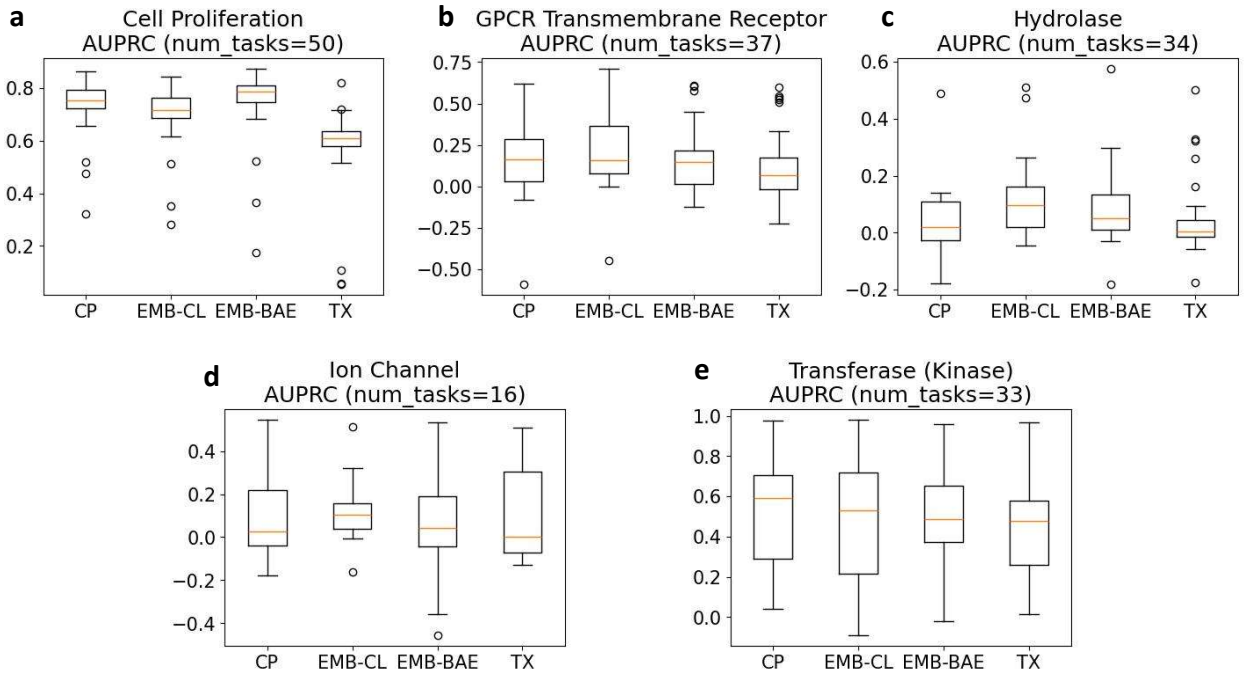



Figure 4.6: **Box plots RIPtoP-AUPRC for tasks grouped by protein target family.** a) Cell Proliferation, b) GPCR Transmembrane Receptor, c) Hydrolase, d) Ion Channel, e) Transferase (Kinase).

kNN accuracy at 0.805, drastically improves on the original CP feature, whose kNN accuracy is 0.416. BAE embedding achieves 0.428, a minor improvement to that of CP feature.

For visualization, we also produce t-SNE plots for eight randomly selected compounds from the test set (Figure 4.3a and 4.3b) using the function sklearn.manifold.TSNE from scikit-learn[55] with parameters (learning rate='auto', init='random', perplexity=10). Each color corresponds to one compound in the plots. We do not plot for the entire test set because the large number of compounds means there are thousand of colors in the plot, making clusters indiscernible. Overall, it can be observed that the t-SNE plots shows a consistent trend with the kNN accuracy table, as CL embedding displays superior clustering ability over the other feature types. For example, in Figure 4.3a, only the yellow and purple clusters are visible for CP feature and BAE embedding. In contrast, CP replicates of the same compounds are much more clearly clustered when using CL embedding.

### 4.2.2 Modes of Action (MoA) Clustering

Another unsupervised task of interest is the clustering of different Modes of Action (MoA), which gives us insight into which compounds share the same MoA[92, 27]. Similar to the first unsupervised experiment, here we calculate the kNN accuracy for each feature type, in addition to visual inspection using a t-SNE plot.

We selected 9 modes of action that compounds in the test set are best annotated with. KNN accuracy is calculated for compounds in the test set for each feature type, with the true cluster labels being the 9 MoA classes (Figure 4.1). CL embedding again achieves the highest kNN accuracy at 0.952 compared to CP features and BAE embedding, both at 0.784.

T-SNE plots are obtained using sklearn.manifold.TSNE from scikit-learn[55] with parameters (learning rate='auto', init='random', perplexity=20) (Figure 4.4). It can be seen from the CP feature and BAE embedding plots that only four MoA are clearly distinguishable using those feature types: Polo-like kinase inhibitor, Microtubule inhibitor, Heat shock protein inhibitor, and mTOR/PI3K inhibitor. The other MoA appear mixed together. This

is not the case for the CL embedding, where we can observe clear clusters formed for those other MoA as well, especially for the Glucocoticoid receptor agonist and the Voltage-gated calcium channel blocker cluster. This is consistent with what the kNN accuracies result.

### 4.2.3 Bioactivity Multitask Classification

Table 4.2: **Performances of each feature type for 703 bioactivity classification tasks.** Mean metrics ± standard deviation metrics for the mean AUROC and mean RIPtoP-AUPRC columns. #(AUROC > 0.7) denotes number of tasks that achieves AUROC > 0.7.

| Feature Type | Mean AUROC | Mean RIPtoP-AUPRC | #(AUROC > 0.7) | #(AUROC > 0.8) |
|---|---|---|---|---|
| CP | 0.680 ± 0.15 | 0.334 ± 0.25 | 290 | 169 |
| CL Emb | 0.687 ± 0.13 | 0.343 ± 0.24 | 294 | 164 |
| BAE Emb | 0.674 ± 0.14 | 0.325 ± 0.24 | 274 | 149 |
| TX | 0.659 ± 0.13 | 0.279 ± 0.22 | 252 | 126 |

Table 4.3: **Performances of each feature type for 47 bioactivity classification tasks that TX performs well (AUROC>0.7) and CP does not perform well (AUROC>0.7).** Mean metrics ± standard deviation metrics for the mean AUROC and mean RIPtoP-AUPRC columns. #(AUROC > 0.7) denotes number of tasks that achieves AUROC > 0.7.

| Feature Type | Mean AUROC | Mean RIPtoP-AUPRC | #(AUROC > 0.7) | #(AUROC > 0.8) |
|---|---|---|---|---|
| CP | 0.641 ± 0.04 | 0.359 ± 0.11 | 0 | 0 |
| CL Emb | 0.671 ± 0.06 | 0.407 ± 0.15 | 14 | 1 |
| BAE Emb | 0.656 ± 0.06 | 0.373 ± 0.12 | 13 | 0 |
| TX | 0.736 ± 0.03 | 0.468 ± 0.09 | 47 | 1 |

In this section, we compare the different feature types in the context of supervised bioactivity multitask classification[21, 10], which have useful applications in early drug discovery including virtual screening, hit triaging, and prioritizing hits for experimental follow-up. This multitask approach involves training a single model across multiple bioactivity tasks simultaneously, leveraging task correlations to enhance performance. All modelling tasks are binary classification tasks, obtained from binarizing bioactivity assays which originally measures potency (pIC50) of a compound.

We train a multitask model on the compounds in the train and validation set, and evaluate on compounds in the test set. The bioactivity label matrix from Janssen contains thousands of binarized bioactivity tasks. For model training we only select tasks that have at least 25 positive, 25 negatives, and at least 100 overall datapoints. The model architecture is a 3-layer MLP whose hyperparameters are: (hidden layer size=256, optimizer=Adam, learning rate scheduler: Cosine Annealing Warm Restarts which resets every 10 epochs learning rate=1e-4, weight decay=1e-5, batch size=128, epochs=100, dropout probability=0.3). For each feature type (CP, TX, CL Emb, BAE Emb), we train the multitask bioactivity model using the exact same setting to ensure the only variable is the feature types themselves. For evaluation, we report results on the subset of assays that have at least 25 positive and 25 negative in the test set, so as to avoid too few positives or negatives. We ended up with 703 bioactivity tasks.

Across those tasks, CL embedding has the highest mean AUROC and RIPtoP-AUPRC ($0.687 \pm 0.13$, $0.343 \pm 0.24$) among all feature types (Table 4.2), with 294 tasks having AUROC $> 0.7$ and 164 tasks having AUROC $> 0.8$. CP feature ranks second in terms of mean AUROC ($0.680 \pm 0.15$), mean RIPtoP-AUPRC ($0.334 \pm 0.25$), and number of tasks with AUROC $> 0.7$ (290). However, CP feature has the highest number of tasks with AUROC$>0.8$ (169). BAE achieves lower mean scores than both CP features and CL embedding ($0.674 \pm 0.14$ and $0.325 \pm 0.24$), with 274 tasks having AUROC $> 0.7$ and 149 tasks having AUROC $> 0.8$. TX feature has the lowest mean scores overall out of all feature types ($0.659 \pm 0.13$ and $0.279 \pm 0.22$), and lowest number of tasks that have AUROC $> 0.7$ (252) and AUROC $> 0.8$ (126). Additionally, we run a wilcoxon signed rank test for each pair of feature type (Table 6.2 , Table 6.3), with the alternative hypothesis being the row feature type outperforms the column feature type at significant value 0.05. Overall, CL embedding and CP feature both outperforms BAE embedding with statistical significance, and all three of them outperforms TX feature with statistical significance in terms of both metrics. Interestingly, despite the higher mean AUROC and RIPtoP-AUPRC,

the improvement of CL embedding over CP features is not statistically significant for the Wilcoxon test (p-value 0.0716 and 0.0841, respectively). This is because, overall, CP features outperform CL embeddings in more tasks (354 tasks compared to 349), though in many cases, the difference is minimal (Figure 6.6), leading to the higher mean scores of CP features.

#### 4.2.3.1 Performance on Assays Grouped by Protein Target Family

Next, we investigate the performance of different feature types on tasks grouped by five protein target families: G-protein-coupled Transmembrane Receptors, Hydrolase, Ion Channel, Transferase (Kinase), and Cell Proliferation (though not a protein target, it is still a significant assay category). These groups comprise 37, 34, 16, 33, and 50 assays, respectively. We focus on the protein families that are the most well-annotated among the 703 tasks. Of the remaining 553 tasks, 480 are not yet annotated with a protein target family, and 53 are annotated but belong to protein targets with much fewer tasks.

Overall, for groups of tasks that CP feature relatively underperforms, such as GPCR Transmembrane Receptor and Hydrolase tasks, CL embedding tends to outperform CP features in terms of AUROC (Figure 4.5b, c) and RIPtoP-AUPRC (Figure 4.6b, c) with statistical significance according to a Wilcoxon signed-rank test at significance level 0.05 (Table 6.6, Table 6.7, Table 6.8, Table 6.9,). Ion Channel is also one of such groups, with CL embedding outperforming CP feature in terms of AUROC with statistical significance (Figure 4.5d, Table 6.10), but not in terms of RIPtoP-AUPRC where neither feature type outperform another with statistical significance (Figure 4.6d, Table 6.11). On the other hand, for groups of tasks that CP feature already performs well, for example Transferase (Kinase), CP feature tends to outperform both CL and BAE embeddings in terms of AUROC (Figure 4.5e), and RIPtoP-AUPRC (Figure 4.6e), with statistical significance (Table 6.12, Table 6.13). Another group that CP feature already perform well on is Cell Proliferation, for which CP feature outperforms CL embedding with statistical significance (Table 6.4, Table 6.5) in terms of both metrics (Figure 4.5a, Figure 4.6a). However, it is worth noting

that BAE embedding outperforms both CP features and CP embedding on Cell Proliferation tasks in terms of both metrics with statistical significance.

#### 4.2.3.2 Performance on Subset of Assays that TX Performs Well but CP Does Not

A key motivation for exploring the cross-modality learning setting is the higher cost of generating TX data via RNA-Seq compared to cell painting data. Consequently, new compounds are more likely to have only cell painting data, and can no longer take advantage of well-performing transcriptomics models. This analysis focuses on tasks where TX feature excels but CP feature does not, aiming to assess whether the learned embeddings can enhance CP feature performance in these tasks. Specifically, from the 703 tasks previously mentioned, we apply an additional filter to select tasks where TX features achieve an AUROC $> 0.7$ and CP features have an AUROC $< 0.7$, yielding 47 tasks.

When using TX features, the mean AUROC and RIPtoP-AUPRC are 0.736 and 0.468, respectively (Table 4.3). On the other hand, the mean scores for CP features are lower, at $0.641 \pm 0.04$ and $0.359 \pm 0.11$. Both CL and BAE embeddings improve upon the performance of CP features. CL embeddings achieves the higher mean scores between the two ($0.671 \pm 0.06$ and $0.407 \pm 0.15$), with statistically significant improvements in both metrics at the 0.05 level, according to the Wilcoxon signed-rank test (Table 6.14 and Table 6.15). BAE embedding shows slightly lower mean scores ($0.656 \pm 0.06$ and $0.373 \pm 0.12$), with only the AUROC improvement being statistically significant, even though the mean RIPtoP-AUPRC is still higher than that of CP features. Furthermore, the learned embeddings improve 14 tasks for CL embedding and 13 tasks for BAE embedding, out of the 47 tasks, to an AUROC $> 0.7$.

### 4.2.4 Modelling of Hallmark Gene Sets

In this section, we tried to compare learned features with the original Cell Painting features in the downstream task of modelling the Hallmark gene sets [93].

One way transcriptomics data is used to study the effect of treatments on a biological system is through gene set variation analysis (GSVA) [94]. The GSVA family of algorithm aims to enables pathway-centric analyses by aggregating measurements of different genes into **gene sets**. In other words, biological understanding of the treatment's effect on the cell state can then be inferred by linking changes in these gene sets to specific biological processes. The Molecular Signatures Database (MSigDB) is one of the most widely used and comprehensive databases of gene sets for performing GSVA. However, the utility of the database is reduced by increased redundancy across, and heterogeneity within, gene sets. The Hallmark gene sets tried to overcome this problem through carefully refining gene sets using automated approaches and expert curation. The result is 50 Hallmark gene sets that each conveys a specific biological state or process and displays coherent expression, reducing both variation and redundancy compared to the MSigDB gene sets.

#### Generating and Categorizing Hallmark Scores

To generate Hallmark scores labels, we ran the GSVA algorithm from the GSVA package using R on our RNA-Seq data, which transformed the gene-by-compound RNA-Seq data matrix into a corresponding gene-set-by-compound Hallmark score matrix, with 50 columns corresponding to 50 Hallmark gene sets. In addition, since the Hallmark data were in float numbers, we binarized the data to turn the modelling problem into classification instead of regression. The binarization steps are:

1. For each gene set, calculate the standard deviation of DMSO controls across plates ($DMSOstd$)

2. Calculate plate-wise measurement differences: [$Compounds–DMSOcontrol$]

3. Label an entry 'up' if the difference is larger than $3 * DMSOstd$, 'down' if the difference is less than $-3 * DMSOstd$, and 'silent' if the difference is in between $3 * DMSOstd$ and $-3 * DMSOstd$,

4. Create 2 classification tasks for each gene set: i) 'up' or 'not up' (silent + down), ii) 'down' or 'not down' ('silent' + 'up').

5. Repeat the process for all 50 Hallmark signatures.

In the end, the 50 Hallmark gene sets were categorized into 100 binary classification tasks, each classify whether genes in a certain gene set have increased expression or reduced expression.

**Hallmark Modelling Results**

Table 4.4: **Hallmark scores classification average over 100 classification tasks.**

| Feature Type | Mean AUROC | Mean RIPtoP-AUPRC | #(Highest AUROC) | #(Highest AUPRC) |
|---|---|---|---|---|
| CP | 0.6693 | 0.1780 | 49 | 55 |
| CL Emb | 0.6694 | 0.1779 | 51 | 45 |
| BAE Emb | 0.6333 | 0.1396 | 0 | 0 |

Table 4.5: **Top ten (sorted by average AUROC) best performing gene sets.**

| Gene Set | AUROC-CP | AUROC-CL | AUROC-BAE |
|---|---|---|---|
| HALLMARK-E2F-TARGETS-DOWN | 0.880 | **0.886** | 0.845 |
| HALLMARK-MYC-TARGETS-V2-DOWN | **0.839** | 0.834 | 0.809 |
| HALLMARK-G2M-CHECKPOINT-DOWN | 0.829 | **0.832** | 0.792 |
| HALLMARK-CHOLESTEROL-HOMEOSTASIS-UP | **0.833** | 0.828 | 0.782 |
| HALLMARK-MYC-TARGETS-V1-DOWN | **0.818** | 0.811 | 0.786 |
| HALLMARK-MYC-TARGETS-V2-UP | **0.794** | 0.793 | 0.747 |
| HALLMARK-E2F-TARGETS-UP | 0.760 | **0.795** | 0.733 |
| HALLMARK-MYC-TARGETS-V1-UP | 0.780 | **0.785** | 0.719 |
| HALLMARK-TNFA-SIGNALING-VIA-NFKB-UP | 0.747 | **0.753** | 0.713 |
| HALLMARK-MTORC1-SIGNALING-UP | 0.739 | **0.747** | 0.698 |

Similar to the bioactivity modeling task, we trained an MLP multitask model on the compounds in the train and validation set, and evaluate on compounds in the test set. For

Table 4.6: **Bottom ten (sorted by average AUROC) worst performing gene sets.**

| Gene Set | AUROC-CP | AUROC-CL | AUROC-BAE |
|---|---|---|---|
| HALLMARK-IL6-JAK-STAT3-SIGNALING-DOWN | **0.606** | 0.596 | 0.583 |
| HALLMARK-KRAS-SIGNALING-DN-DOWN | **0.602** | 0.595 | 0.578 |
| HALLMARK-KRAS-SIGNALING-DN-UP | **0.599** | 0.599 | 0.574 |
| HALLMARK-NOTCH-SIGNALING-DOWN | **0.596** | 0.590 | 0.571 |
| HALLMARK-WNT-BETA-CATENIN-SIGNALING-UP | **0.601** | 0.590 | 0.556 |
| HALLMARK-APICAL-SURFACE-DOWN | **0.594** | 0.584 | 0.564 |
| HALLMARK-SPERMATOGENESIS-UP | **0.595** | 0.587 | 0.557 |
| HALLMARK-APICAL-SURFACE-UP | **0.583** | 0.579 | 0.550 |
| HALLMARK-NOTCH-SIGNALING-UP | **0.580** | 0.575 | 0.550 |
| HALLMARK-WNT-BETA-CATENIN-SIGNALING-DOWN | **0.578** | 0.575 | 0.544 |

each feature type (CP, TX, CL Emb, BAE Emb), we train the multitask bioactivity model using the exact same setting to ensure the only variable is the feature types themselves. The MLP has hidden layers [512, 512, 512], Adam optimizer, learning rate $1e-3$, weight decay $1e-6$, train on 100 epochs with batch size 256, and the epoch with the lowest validation lost is used for inference.

Firstly, when we use TX features, the AUROC score for every Hallmark tasks is 0.95 or above, with the top 10 most predictive tasks being over 0.99. The fact that TX data are highly correlated with the Hallmark tasks was expected, since the latter was created from the former using GSVA, and the model was simply learning the GSVA algorithm in a data-driven way.

Secondly, we observed that the learned embeddings did not lead to improvements over the CP features in terms of mean AUROC and RIPtoP-AUPRC across 100 classification tasks (Table 4.4). In fact, for BAE embedding the scores got significantly worse. CL embedding appeared to perform on par with the original CP features on average, improving performance of around half the tasks but worsening performance of the other half. Details of the top 10 and bottom 10 most predictive task (sorted by the average of AUROC of CP, CL and BAE) can be found in Table 4.5, Table 4.6. These results suggested our current implementation of CP and BAE had yet to produce representations that achieved perfect translation from CP to TX.

## 4.3 Discussion

We benchmarked two cross-modality representation learning algorithms, contrastive learning (CL) and bimodal autoencoder (BAE), for cell painting (CP) and transcriptomics (TX) data. The aim was to learn a new embedding from two modalities that can be generated with just one modality (CP). This can address the real-life problem where new compounds will only have CP data, as generating TX data is much more costly. We show both visually and through calculating kNN accuracy, that learned representations improve cluster quality for clustering of CP replicates and different modes of action (MoA), demonstrating the learning of new biological knowledge. In particular, CL embedding yielded the best results, while BAE only achieved minor improvements over the CP features.

In the supervised bioactivity multitask classification, we demonstrated that CL embedding achieves higher mean AUROC and RIPtoP-AUPRC compared to CP feature across a range of bioactivity tasks, though the improvements are not statistically significant with a Wilcoxon signed rank test at significant value 0.05. Mean scores of BAE embedding and TX features ranks third and fourth, respectively. Additionally, we observed that CL embedding outperformed CP feature in GPCR Transmembrane Receptor, Hydrolase and Ion Channel protein families. CP feature outperformed both learned embeddings in Transferase tasks, but BAE embedding outperforms CP features in Cell Proliferation tasks. Overall, for groups of tasks that CP feature relatively underperforms, CL embedding outperformed CP features. Finally, we show that in the absence of TX features for new compounds, using learned embeddings enhanced performance of CP feature on tasks where TX feature exceled but CP feature did not.

Limitations of this study includes: 1) Simple approach to encoding CP and TX data. We simply encoded the data with a simple MLP, without any denoising or quantization. We expect that for weak supervision pretraining tasks such as CL or BAE, good denoising and quantization can potentially help overcome experimental noise and help the algorithms learn

useful representations. In the next chapter, we will discuss potential algorithmic and architectural upgrades. 2) More work could be done on assay annotations/categorisation to find out which subgroup of assays works best, and which subgroup has worse performance. This can offer a more granular look into the applications that the learned embeddings work best, as well as which kind of biological information CL and BAE learned during the representation learning process.

# Chapter 5

# Conslusion and Future Directions

## 5.1 Summary of the Research

Overall, I had described my work to improve the Image-based Activity Modelling pipeline for early drug discovery using machine learning techniques. Firstly, in chapter 2, I aimed to improve the pipeline through comparing several machine learning models and choose the best one. In particular, I introduced FSL-CP, a dataset designed for few-shot prediction of small molecule bioactivity using cell microscopy images. This few-shot framework simulated an early-stage drug discovery scenario where the objective is to identify potent compounds targeting specific proteins from high-content cell images with minimal data. While few-shot learning has been explored for molecular activity using graph-based representations of molecules, most advancements in few-shot learning have been concentrated in computer vision and natural language processing. By employing cell images as a molecular representation, our dataset creates an opportunity to apply state-of-the-art techniques from computer vision to improve predictive modeling.

This dataset established benchmarks to evaluate the performance of various few-shot learning paradigms. Our findings showed that feature-based prototypical networks and multitask feedforward neural networks (FNNs) pretrained on auxiliary tasks generally perform well across a range of support set sizes. It could be observed, however, that meta-learning methods saw diminishing returns with larger support sets, whereas single-task methods benefitted considerably from additional data. To more definitively assess whether single-task

models can eventually surpass pretrained models—and at which support set size, additional labeled compounds and cell painting data were required.

Image-based models currently underperformed on our benchmark, primarily due to the computational demands associated with processing each sample, which includes six high-definition, five-channel images. To manage training within practical limits, we used only a single randomly cropped, resized image per compound, resulting in significant information loss. Training on full-resolution cell images had shown potential for improved performance over CP features in some cases. However, in real-world drug discovery contexts, larger images, a higher number of compounds, and more prediction tasks are common, making routine pretraining on full-resolution images impractical, especially when frequent retraining is necessary.

An alternative approach to leveraging computer vision was to enhance CP features by embedding images through a pretrained model like ResNet or Inception. We tested this using ResNet50 as an embedding generator, yielding minor performance gains. Given that most pretrained vision models rely on ImageNet, this indicated some transferability of knowledge from unrelated image databases, though not sufficient for major improvements. We expected that more substantial gains could be achieved by pretraining embedding generators end-to-end on cell images with a relevant task, such as multi-task or contrastive learning.

Our benchmark also sheded light on the effectiveness of knowledge transfer from models pretrained on auxiliary tasks to novel tasks. While many tasks benefit from such pretraining, the degree of improvement varies. Determining how much a new task will benefit from pretraining remains an open research question, though preliminary observations suggest that predictive tasks often gain more from pretraining. Groups looking to pretrain models on auxiliary tasks could benefit by combining publicly available tasks with proprietary datasets to maximize the amount of data available.

In chapter 3, we presented an approach for optimizing image-based small molecule activity pipelines to effectively retrieve highly potent compounds in biological assays. Initially, we

trained a moderate-potency model using $20\mu M$ cell painting images to classify compounds at a pIC50 potency threshold that retains a sufficient number of positive samples for effective training. We then adapted this well-performing model for higher potency classification by performing inference with images at lower concentrations. Within the drug discovery pipeline, accurately identifying highly potent compounds could streamline hit prioritization for experimental follow-up based on potency and facilitate the deprioritization of compounds with off-target activity during hit triaging. However, this approach requires additional cell painting images at varying concentrations, which increases data generation costs.

Our approach demonstrated two key hypotheses. Hypothesis 1 is that a strong moderate-potency model can be repurposed to selectively retrieve higher potency compounds by using features from low-concentration images for inference without requiring model retraining. We evaluate this hypothesis by assessing precision in identifying highly potent compounds across 57 assays and 4 additional case studies, finding that nearly all assays, especially cell proliferation assays, support this approach. Although fewer active compounds are retrieved with low-concentration images, these tend to be the most potent compounds.

Hypothesis 2 suggests that if data imbalance hinders the training of a high-potency model, our method could surpass conventional high-potency models in identifying highly potent compounds. We tested this on the same selection of assays, using AUC-ROC and corrected AUC-PR metrics. Results show that in 65–75% of assays, our approach performs better, with AUC-ROC scores improving by approximately 0.1 to 0.2 and AUC-PR scores by 0.2 to 0.5. This improvement indicates that our method can effectively replace traditional high-potency models when activity labels are limited.

Finally, in chapter 3, we benchmarked two cross-modality representation learning algorithms, contrastive learning (CL) and bimodal autoencoder (BAE), on cell painting (CP) and transcriptomics (TX) data. One of the goals was to learn a shared embedding from both modalities, that can be generated solely from one (CP). This is vital in settings where CP is available but TX is not, due to the relatively high cost of running an RNA-Seq screen.

Our results, confirmed both visually and through kNN accuracy, demonstrated that these learned representations enhance clustering quality for CP replicates and various modes of action (MoA), thus revealing new biological insights. Specifically, the CL embedding achieved superior results, while BAE provides only slight improvements over CP features.

In the supervised bioactivity multitask classification, the CL embedding outperformed CP features in mean AUROC and RIPtoP-AUPRC across several bioactivity tasks, though the Wilcoxon signed-rank test at significance level 0.05 did not confirm statistical significance. BAE embedding and TX features ranked third and fourth in mean scores, respectively. Notably, CL embedding surpassed CP features in GPCR Transmembrane Receptor, Hydrolase, and Ion Channel protein families, while CP features remained superior in Transferase tasks, and BAE embedding exceled over CP features in Cell Proliferation tasks. Overall, in tasks where CP features underperform, CL embedding often showed superior results. Lastly, we demonstrated that in the absence of TX data for new compounds, learned embeddings improve CP feature performance in tasks where TX features typically excel.

In the Hallmark gene set modeling tasks, The learned embedding does not lead to improvements over the CP features in terms of average AUROC and RIPtoP-AUPRC across 100 classification tasks, though for some gene sets contrastive learning embedding performs better than CP feature and vice versa. This indicates out implementation of CP and BAE had yet to produce representations that achieved perfect translation from CP to TX.

## 5.2 Future Directions

Firstly, at the time of writing, few-shot and zero-shot prediction for molecular properties, including applications in drug discovery and traditional machine learning, have increasingly adopted self-supervised transformer-based models. These models are generally pretrained on large unlabeled datasets in a self-supervised way to learn useful representations that can enhance performance on low-data tasks. Multitask learning has also seen significant adop-

tion due to its straightforward implementation and effectiveness in transferring knowledge across low-data tasks [95, 23, 10, 22]. In contrast, meta-learning approaches, which focus on enhancing the model's capability to adapt to tasks with little data, such as metric-based (e.g., ProtoNet) and optimization-based methods (e.g., MAML), have not yet gained similar popularity, though recent studies still indicate their competitive performance for low-data tasks [96, 95]. Directions to improve zero-shot and few-shot capability of models include, but not limited to: 1) learning better representations, as discussed later; 2) algorithmic advancements in meta-learning to further enhance adaptation in low-data settings; and 3) combination of these techniques. Useful representations can be learned through self-supervision. Then learned representation can be used as input to a meta-learning algorithm that can further facilitate quick adaptation to tasks with few datapoints, or as input to multitask learning which facilitates knowledge transfers between low data tasks.

Secondly, regarding multiple concentrations in Cell Painting, I believe that future research could significantly benefit from generating Cell Painting data across different concentrations. Our work in Chapter 3 exemplifies this, but additional motivations exist. For instance, previous studies have shown concentration-dependent effects that caused cellular damage in Cell Painting screens [24] or triggered distinct pathway activations in proteomics screens [97]. Generating Cell Painting data at varying compound concentrations could provide deeper insights and help overcome such concentration-dependent effects. Moreover, prior research has recorded proteomics data in a dose-response fashion, measuring relative protein intensity to produce a dose-response curve for each drug-protein combination (x-axis: drug concentration; y-axis: relative protein intensity) [98]. Such dataset demonstrates that dose–response measurements reveal information unavailable from single-dose analyses, such as the finding that histone deacetylase (HDAC) inhibitors strongly down-regulate the T cell receptor complex, impairing human T cell activation. Since Cell Painting can be performed at a higher throughput than proteomics screens, it is feasible to generate data for many compounds at different doses, potentially enhancing modeling and leading to new biological insights.

Thirdly, I would like to discuss more about unimodal representation learning of Cell Painting data for future research. Ultimately, I think that the goal of unimodal representation learning is the deconvolution of the rich but noisy Cell Painting data, and subsequently the correlation of the deconvoluted signals with more interpretable MoA or biological pathways. For examples, algorithms such as DINO, masked autodencoder and SimCLR were used to learn useful representation from the raw cell images themselves [26, 25]. Representations can also be learned from the Cell Profiler extracted features themselves, such as in the work of Pahl et al[24], which showed using a subset of the Cell Profiler features yields much higher similarities between the Cell Painting profiles of compounds that have the same MoA. One potentially effective approach that we have not seen done for CP data is quantization - mapping loage continuous CP values into a smaller discrete set. In traditional machine learning quantization is mainly used to lower the bit rate of parameters in a model so that it can be efficiently trained using less computational resources. However in our case, effective quantization can additionally overcome the noisiness of CP data, and potentially making signal deconvolution easier. This can also help in some multimodal applications where CP data is used as labels for supervision, or condition for some molecule generation tasks.

Finally, multimodal representation learning is a promising and exciting direction, driven by the success of large models that process and generate data across image, text, and audio modalities. This approach is especially compelling for drug discovery, given the wide range of available data modalities, including chemistry, omics, quantum mechanics (QM), electronic health records, clinical trials, etc. Each modality provides distinct information, from the chemical structure of a drug to its interactions within biological systems. A multimodal foundation model capable of integrating these diverse modalities could unlock substantial chemical and biological insights and new applications. However, achieving this remains challenging due to the limited availability and high cost of generating such comprehensive datasets. Our work in Chapter 4 represents an early step toward achieving this goal, though we recognize its limitations and propose several improvements: 1) *Architectural upgrades.*

Instead of relying on extracted features, future work could directly train on Cell Painting images using computer vision encoders/decoders, such as ResNet [44] or Vision Transformer [64]. Additionally, transformer architectures have been effective in encoding transcriptomics data, as demonstrated in natural language processing, where they excel at learning expressive data representations [34, 32, 33]. 2) *Effective quantization or tokenization of data.* Transformer-based methods above also address noisy transcriptomics measurements by introducing quantization or tokenization strategies. We believe this is essential, as it could mitigate the high level of noise typically encountered in biological data. 3) *Further characterization of learned embeddings in additional applications.* By examining the performance of learned embeddings across different applications, we could identify tasks that benefit from these embeddings and those where performance may lag. This analysis would also offer insights into the specific types of biological knowledge captured during pretraining.

# Chapter 6

# Supplementary Information

## 6.1 Chapter 2: Few-shot Model Hyperparameters

Here we include more details about the hyperparameters of benchmark models in Chapter 2. If the reader wants to reproduce the result of the work, we would encourage running the codes from our GitHub repository.

**protonet_cp+:**

num_episodes _train = 50000

num_episodes_val = 100

loss function: $nn.CrossEntropyLoss()$

optimizer: $optim.Adam(model.parameters(), lr = 0.0001)$

learning rate scheduler: $StepLR(optimizer, step_size = 20000, gamma = 0.1)$

Backbone model: 3-hidden-layer Fully-connected Neural Network

Size of output layer of backbone model = 256

Distance = 'Euclidean'

**protonet_cp:**

num_episodes _train = 50000

num_episodes_val = 100

loss function: $nn.CrossEntropyLoss()$

optimizer: $optim.Adam(model.parameters(), lr = 0.0001)$

learning rate scheduler: $StepLR(optimizer, gamma = 0.1)$

Backbone model: 3-hidden-layer Fully-connected Neural Network

Size of output layer of backbone model = 512

Distance = 'Euclidean'

**protonet_img:**

num_episodes _train = 30000

num_episodes_val = 100

loss function: $nn.CrossEntropyLoss()$

optimizer: $optim.Adam(model.parameters(), weight\_decay = 1e-4)$

learning rate scheduler: $StepLR(optimizer, step_size = 10000, gamma = 0.1)$

Image transformation: $RandomCrop(300), Resize(200)$

Backbone model: ResNet50

Size of output layer of backbone model = 1600

Distance = 'CosineSimilarity'

**maml_cp+:**

num_episodes _train = 32000

num_episodes_val = 100

adaptation_steps=3

loss function: $nn.BCEWithLogitsLoss()$

optimizer: $optim.Adam(lr = 0.001)$

Image transformation: $RandomCrop(100), Resize(85)$

Model: ResNet50

MAML Fast adaptation learning rate = 0.01

**maml_img:**

num_episodes _train = 32000

num_episodes_val = 100

adaptation_steps=3

loss function: $nn.BCEWithLogitsLoss()$

optimizer: $optim.Adam(lr = 0.001)$

Model: 3-hidden-layer Fully-connected Neural Network

MAML Fast adaptation learning rate = 0.01

**singletask_cp:**

max_epochs = 50

loss function: $nn.BCEWithLogitsLoss()$

optimizer: $optim.Adam(lr = 0.0001)$

learning rate scheduler: $StepLR(optimizer, step_size = 100, gamma = 0.1)$

Model: 3-hidden-layer Fully-connected Neural Network

**multitask_cp:**

pretrain max_epochs = 50

pretrain loss function: *multitask_bce* (Binary Cross Entropy)

pretrain optimizer: $optim.SGD(lr = 1e-2, momentum = 0.9, weight\_decay = 1e-4)$

pretrain learning rate scheduler: $StepLR(optimizer, step\_size = 20, gamma = 0.1)$

Model: 3-hidden-layer Fully-connected Neural Network

inference max_epochs = 50

inference loss function: $nn.BCEWithLogitsLoss()$

inference optimizer: $optim.Adam(lr = 0.0001)$

inference learning rate scheduler: $StepLR(optimizer, step\_size = 100, gamma = 0.1)$

**logistic_cp+:**

We did a RandomizedSearchCV on these hyperparameters:

'C' : [0.01, 0.1, 1.0, 10.0, 100.0]

## 6.2 Chapter 2: Cohen's Kappa Confusion Matrix Formula Derivation

We start from the first formula

$$\kappa = \frac{p_o - p_e}{1 - p_e}.$$

$p_o$ by definition is the relative observed agreement among raters. Hence it can be written in terms of confusion matrix terms as:

$$\frac{TP + TN}{TP + TN + FP + FN}.$$

On the other hand, $p_e$ is the hypothetical probability of chance agreement. Let $\widehat{p_{k12}}$ be the probability that both rater 1 and 2 will classify the same class as $k$; $\widehat{p_{k1}}$ and $\widehat{p_{k2}}$ be the probability that rater 1 or rater 2, respectively, will classify the same class as $k$. Assume independence between 2 raters, we have $\widehat{p_{k12}} = \widehat{p_{k1}}\widehat{p_{k2}}$ Note that in binary classification $k$ is either positive or negative. In other words, $k \in \{P, N\}$

By definition of $p_e$,

$$p_e = \sum_k \widehat{p_{k12}} p_e = \sum_k \widehat{p_{k1}}\widehat{p_{k2}} p_e = \sum_k \frac{n_{k1}}{M}\frac{n_{k2}}{M} p_e = \frac{1}{M^2}[(n_{P1}n_{P2}) + (n_{N1}n_{N2})],$$

where $n_{k1}$ is the number of time rater 1 predict $k$, and $M$ is the total items to classify.

$\frac{n_{k1}}{M}$ is an estimation of $\widehat{p_{k1}}$. The same for rater 2 and when $k = N$. The terms in the equation can now be calculated using terms from the confusion matrix, hence,

$$p_e = \frac{1}{M^2}[(TP + FP)(TP + FN) + (TN + FN)(TN + FP)],$$

and,

$$p_o = \frac{TP + TN}{M}$$

Now, we calculate the numerator of the original Cohen's Kappa formula:

$$\begin{aligned}
p_o - p_e &= \frac{1}{M^2}[(TP + TN + FP + FN)(TP + TN) \\
&\quad - (TP + FP)(TP + FN) - (TN + FN)(TN + FP)] \\
&= \frac{1}{M^2}[(TP + FP)(TP + TN) + (TN + FN)(TP + TN) \\
&\quad - (TP + FP)(TP + FN) - (TN + FN)(TN + FP)] \\
&= \frac{1}{M^2}[(TP + FP)(TP + TN) + (TN + FN)(TP + TN) \\
&\quad - (TP + FP)(TP + FN) - (TN + FN)(TN + FP)] \\
&= \frac{1}{M^2}[(TP + FP)(TP + TN - TP - FN) \\
&\quad - (TP + TN - TN - FP)(TP + FN)] \\
&= \frac{2}{M^2}(TP \times TN - FP \times FN).
\end{aligned}$$

Then, the denominator:

$$
\begin{aligned}
1 - p_e &= \frac{1}{M^2}[(TP + TN + FP + FN)^2 \\
&\quad - (TP + FP)(TP + FN) - (TN + FN)(TN + FP)] \\
&= \frac{1}{M^2}[(TP + FP)^2 + (FN + TN)^2 + 2(TP + FP)(TN + FN) \\
&\quad - (TP + FP)(TP + FN) - (TN + FN)(TN + FP)] \\
&= \frac{1}{M^2}[(TP + FP)^2 + (FN + TN)^2 + (TP + FP)(TN - TP) \\
&\quad + (TN + FN)(TP - TN)] \\
&= \frac{1}{M^2}[(TP + FP)(FP + TN) + (TP + FP)(TN - TP - FP - TN) \\
&\quad + (TN + FN)(TP + FN) - (TN + FN)(TP - TN - TP - FN)] \\
&\quad + (TP + FP)^2 + (TN + FN)^2] \\
&= \frac{1}{M^2}[(TP + FP)(TP + TN - TP - FN) - \\
&\quad (TP + TN - TN - FP)(TP + FN)] \\
&= \frac{1}{M^2}[(TP + FP)(FP + TN) + (TN + FN)(TP + FN)].
\end{aligned}
$$

Combining the numerator and the denominator gives the final formula:

$$
\kappa = \frac{2 \times (TP \times TN - FN \times FP)}{(TP + FP) \times (FP + TN) + (TP + FN) \times (FN + TN)}.
$$

## 6.3 Chapter 3: Model Architecture and Additional Details about Training Procedure

### Model Hyperparameters

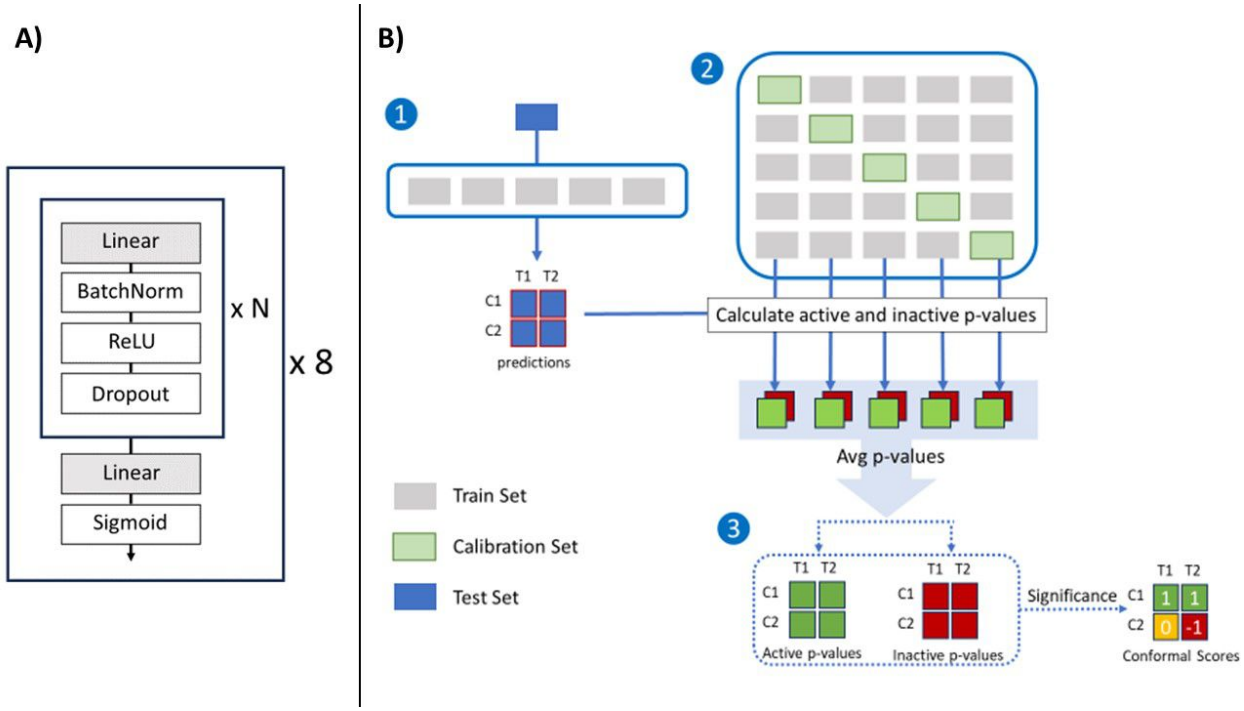



Figure 6.1: **A) Architecture of the model.** The model is an ensemble of 8 Multi-Layer Perceptrons, with N 'blocks' consisting of a Linear layer, BatchNorm, ReLU activation and Dropout. At the end of the model is the final linear layer (corresponding to the end tasks) followed by sigmoid to produce the probability scores for the different end tasks. **B) Training and Inference Process.** Firstly, the model is trained on the entire train set, and inference is done on the test set (for example, compounds C1, C2 in tasks T1, T2). Then, MCCP is performed for the 5 folds of the training set, acquiring an active p-value and an inactive p-value for each output probability score. Finally, the conformal scores for the outputs are calculated using the p-values, which returns 1 for Active, -1 for Inactive, and 0 for Uncertain.

| Model | Hidden Layers Sizes | Dropout | Learning Rate | Weight Decay |
|---|---|---|---|---|
| 0 | (512, 512, 512, 512) | 0.5 | 0.0005 | 0.0 |
| 1 | (1536, 1536, 1536) | 0.3 | 0.0001 | 0.0 |
| 2 | (256, 256) | 0.5 | 0.0001 | 1e-05 |
| 3 | (1024, 1024) | 0.1 | 0.0005 | 0.0001 |
| 4 | (1024, 1024, 1024) | 0.5 | 0.0001 | 0.0001 |
| 5 | (2048, 2048, 2048) | 0.3 | 5e-05 | 1e-05 |
| 6 | (512, 512, 512) | 0.0 | 5e-05 | 1e-05 |
| 7 | (1024, 1024) | 0.1 | 1e-05 | 1e-05 |

Table 6.1: **Hyperparameters for each of the MLP in the ensemble.**

## 6.4 Chapter 3: Stem Plots Visualization of Assays in Case Study

We provide stem plot visualisation for the assays in the Case Study section of chapter 3. Generally, the findings in that section agree with the stem plot visualisations.

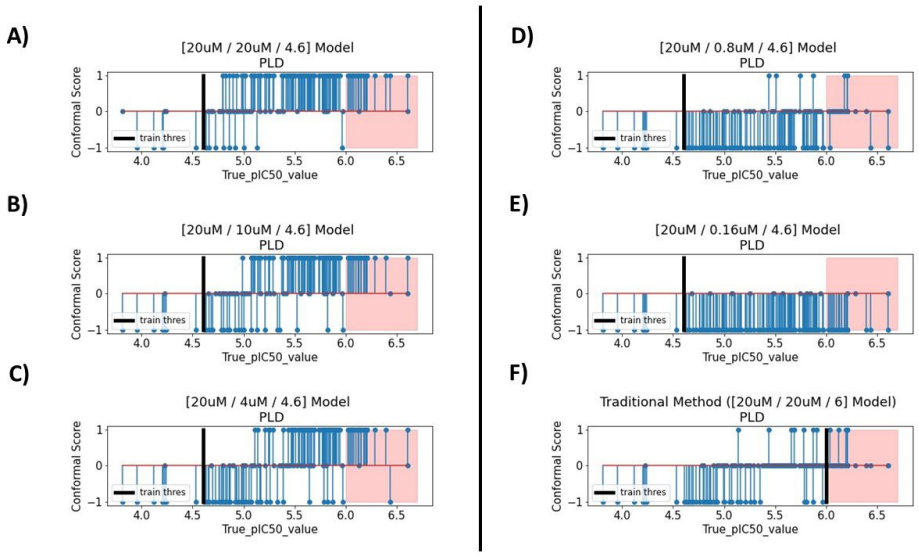



Figure 6.2: **PLD Assay**

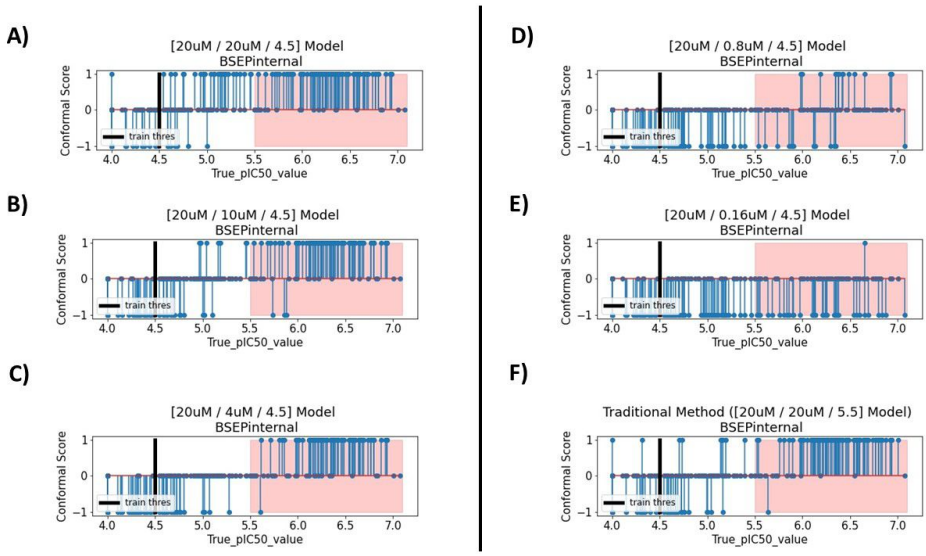



Figure 6.3: **BSEP Assay**

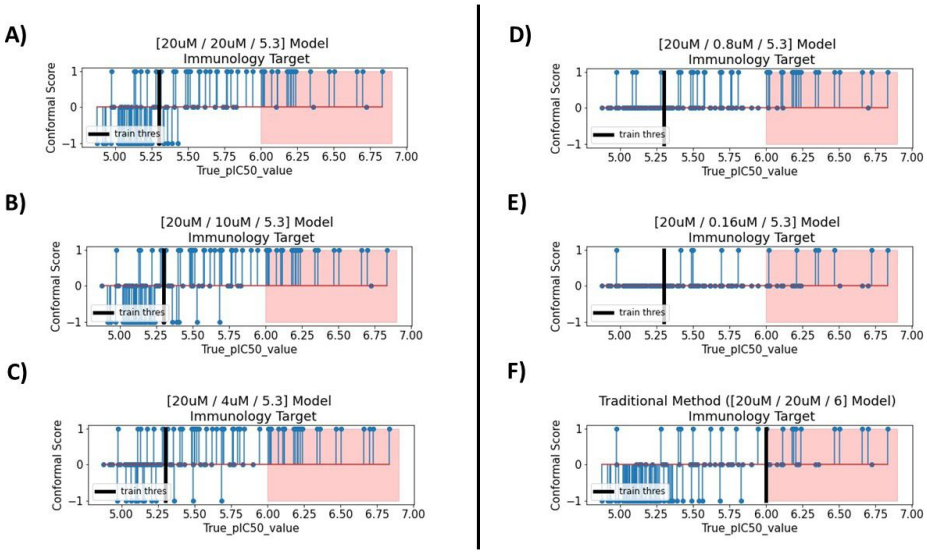



Figure 6.4: **Immunology Target Assay**

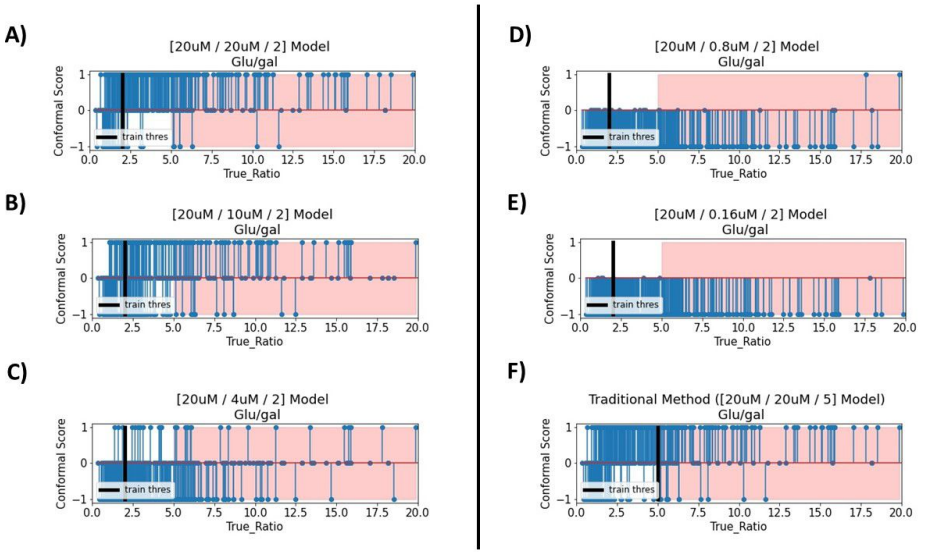



Figure 6.5: **Glu/gal Assay**

## 6.5 Chapter 4: Wilcoxon Signed Rank Tests P-values Tables

Table 6.2: **Wilcoxon signed rank tests P-values of the AUROC for 703 bioactivity tasks in Table 4.2.** The alternative hypothesis is that the AUROC of the row feature being higher than the column feature. P-values smaller than the significant value 0.05 are in bold.

| Feature Type | CP | CL Emb | BAE Emb | TX |
|---|---|---|---|---|
| CP | - | 9.28e-01 | **2.73e-04** | **1.02e-10** |
| CL Emb | 7.16e-02 | - | **3.81e-08** | **1.35e-19** |
| BAE Emb | 1 | 1 | - | **5.51e-07** |
| TX | 1 | 1 | 1 | - |

Table 6.3: **Wilcoxon signed rank tests P-values of the RIPtoP-AUPRC for 703 bioactivity tasks in Table 4.2.** The alternative hypothesis is that the RIPtoP-AUPRC of the row feature being higher than the column feature. P-values smaller than the significant value 0.05 are in bold.

| Feature Type | CP | CL Emb | BAE Emb | TX |
|---|---|---|---|---|
| CP | - | 9.16e-01 | **2.83e-03** | **2.59e-22** |
| CL Emb | 8.41e-02 | - | **2.07e-07** | **4.79e-39** |
| BAE Emb | 9.97e-01 | 1 | - | **2.09e-20** |
| TX | 1 | 1 | 1 | - |

Table 6.4: **Wilcoxon signed rank tests P-values of the AUROC for Cell Proliferation tasks in Figure 4.5.** The alternative hypothesis is that the AUROC of the row feature being higher than the column feature. P-values smaller than the significant value 0.05 are in bold.

| Feature Type | CP | CL Emb | BAE Emb | TX |
|---|---|---|---|---|
| CP | - | **1.26e-05** | 9.98e-01 | **1.18e-09** |
| CL Emb | 1 | - | 1 | **5.74e-06** |
| BAE Emb | **1.54e-03** | **1.30e-07** | - | **3.58e-09** |
| TX | 1 | 1 | 1 | - |

Table 6.5: **Wilcoxon signed rank tests P-values of the RIPtoP-AUPRC for Cell Proliferation tasks in Figure 4.6.** The alternative hypothesis is that the RIPtoP-AUPRC of the row feature being higher than the column feature. P-values smaller than the significant value 0.05 are in bold.

| Feature Type | CP | CL Emb | BAE Emb | TX |
|---|---|---|---|---|
| CP | - | **2.00e-05** | 1 | **3.78e-10** |
| CL Emb | 1 | - | 1 | **5.44e-10** |
| BAE Emb | **1.45e-04** | **7.47e-07** | - | **4.27e-10** |
| TX | 1 | 1 | 1 | - |

Table 6.6: **Wilcoxon signed rank tests P-values of the AUROC for GPCR Transmembrane Receptor tasks in Figure 4.5.** The alternative hypothesis is that the AUROC of the row feature being higher than the column feature. P-values smaller than the significant value 0.05 are in bold.

| Feature Type | CP | CL Emb | BAE Emb | TX |
|---|---|---|---|---|
| CP | - | 9.63e-01 | **1.12e-02** | **1.53e-02** |
| CL Emb | **3.67e-02** | - | **2.47e-03** | **1.34e-03** |
| BAE Emb | 9.89e-01 | 9.98e-01 | - | **7.14e-02** |
| TX | 9.85e-01 | 9.99e-01 | 9.29e-01 | - |

Table 6.7: **Wilcoxon signed rank tests P-values of the RIPtoP-AUPRC for GPCR Transmembrane Receptor tasks in Figure 4.6.** The alternative hypothesis is that the RIPtoP-AUPRC of the row feature being higher than the column feature. P-values smaller than the significant value 0.05 are in bold.

| Feature Type | CP | CL Emb | BAE Emb | TX |
|---|---|---|---|---|
| CP | - | 9.71e-01 | 2.00e-01 | 6.21e-02 |
| CL Emb | **2.88e-02** | - | **2.88e-02** | **4.41e-03** |
| BAE Emb | 8.00e-01 | 9.71e-01 | - | 3.22e-02 |
| TX | 1 | 1 | 1 | - |

Table 6.8: **Wilcoxon signed rank tests P-values of the AUROC for Hydrolase tasks in Figure 4.5.** The alternative hypothesis is that the AUROC of the row feature being higher than the column feature. P-values smaller than the significant value 0.05 are in bold.

| Feature Type | CP | CL Emb | BAE Emb | TX |
|---|---|---|---|---|
| CP | - | 9.99e-01 | 9.86e-01 | **7.50e-01** |
| CL Emb | **7.83e-04** | - | 2.64e-01 | **8.71e-02** |
| BAE Emb | **1.40e-02** | 7.36e-01 | - | 1.29e-01 |
| TX | 2.50e-01 | 9.13e-01 | 8.71e-01 | - |

Table 6.9: **Wilcoxon signed rank tests P-values of the RIPtoP-AUPRC for Hydrolase tasks in Figure 4.6.** The alternative hypothesis is that the RIPtoP-AUPRC of the row feature being higher than the column feature. P-values smaller than the significant value 0.05 are in bold.

| Feature Type | CP | CL Emb | BAE Emb | TX |
|---|---|---|---|---|
| CP | - | 1 | 9.88e-01 | 4.24e-01 |
| CL Emb | **5.37e-05** | - | 7.93e-02 | **1.08e-02** |
| BAE Emb | **1.15e-02** | 9.21e-01 | - | 6.87e-02 |
| TX | 5.76e-01 | 9.89e-01 | 9.31e-01 | - |

Table 6.10: **Wilcoxon signed rank tests P-values of the AUROC for Ion Channel tasks in Figure 4.5.** The alternative hypothesis is that the AUROC of the row feature being higher than the column feature. P-values smaller than the significant value 0.05 are in bold.

| Feature Type | CP | CL Emb | BAE Emb | TX |
|---|---|---|---|---|
| CP | - | 1 | 3.48e-01 | 3.85e-01 |
| CL Emb | **9.77e-04** | - | 1.88e-01 | 6.54e-02 |
| BAE Emb | 6.88e-01 | 8.39e-01 | - | 6.52e-01 |
| TX | 6.52e-01 | 9.47e-01 | 3.85e-01 | - |

Table 6.11: **Wilcoxon signed rank tests P-values of the RIPtoP-AUPRC for Ion Channel tasks in Figure 4.6.** The alternative hypothesis is that the RIPtoP-AUPRC of the row feature being higher than the column feature. P-values smaller than the significant value 0.05 are in bold.

| Feature Type | CP | CL Emb | BAE Emb | TX |
|---|---|---|---|---|
| CP | - | 8.75e-01 | 3.26e-01 | 4.10e-01 |
| CL Emb | 1.50e-01 | - | 1.80e-01 | 1.02e-01 |
| BAE Emb | 7.15e-01 | 8.50e-01 | - | 6.33e-01 |
| TX | 6.33e-01 | 9.18e-01 | 4.10e-01 | - |

Table 6.12: **Wilcoxon signed rank tests P-values of the AUROC for Transferase (Kinase) tasks in Figure 4.5.** The alternative hypothesis is that the AUROC of the row feature being higher than the column feature. P-values smaller than the significant value 0.05 are in bold.

| Feature Type | CP | CL Emb | BAE Emb | TX |
|---|---|---|---|---|
| CP | - | **1.55e-03** | **8.95e-03** | **3.63e-02** |
| CL Emb | 9.98e-01 | - | 2.81e-01 | 5.25e-01 |
| BAE Emb | 9.91e-01 | 7.19e-01 | - | 5.18e-01 |
| TX | 9.64e-01 | 4.75e-01 | 4.82e-01 | - |

Table 6.13: **Wilcoxon signed rank tests P-values of the RIPtoP-AUPRC for Transferase (Kinase) tasks in Figure 4.6.** The alternative hypothesis is that the RIPtoP-AUPRC of the row feature being higher than the column feature. P-values smaller than the significant value 0.05 are in bold.

| Feature Type | CP | CL Emb | BAE Emb | TX |
|---|---|---|---|---|
| CP | - | **2.45e-03** | **1.79e-02** | **3.58e-03** |
| CL Emb | 9.98e-01 | - | 1.25e-01 | 1.36e-01 |
| BAE Emb | 9.82e-01 | 8.75e-01 | - | 2.75e-01 |
| TX | 9.96e-01 | 8.64e-01 | 7.25e-01 | - |

Table 6.14: **Wilcoxon signed rank tests P-values of the AUROC for 47 bioactivity tasks in Table 4.3.** The alternative hypothesis is that the AUROC of the row feature being higher than the column feature. P-values smaller than the significant value 0.05 are in bold.

| Feature Type | CP | CL Emb | BAE Emb |
|---|---|---|---|
| CP | - | 1 | 9.62e-01 |
| CL Emb | **4.29e-04** | - | 7.97e-02 |
| BAE Emb | **3.77e-02** | 9.20e-01 | - |

Table 6.15: **Wilcoxon signed rank tests P-values of the RIPtoP-AUPRC for 47 bioactivity tasks in Table 4.3.** The alternative hypothesis is that the RIPtoP-AUPRC of the row feature being higher than the column feature. P-values smaller than the significant value 0.05 are in bold.

| Feature Type | CP | CL Emb | BAE Emb |
|---|---|---|---|
| CP | - | 9.99e-01 | 7.58e-01 |
| CL Emb | **1.37e-03** | - | **8.63e-03** |
| BAE Emb | 2.42e-01 | 9.91e-01 | - |

## 6.6 Chapter 4: Histogram of Difference between CP and CL Features

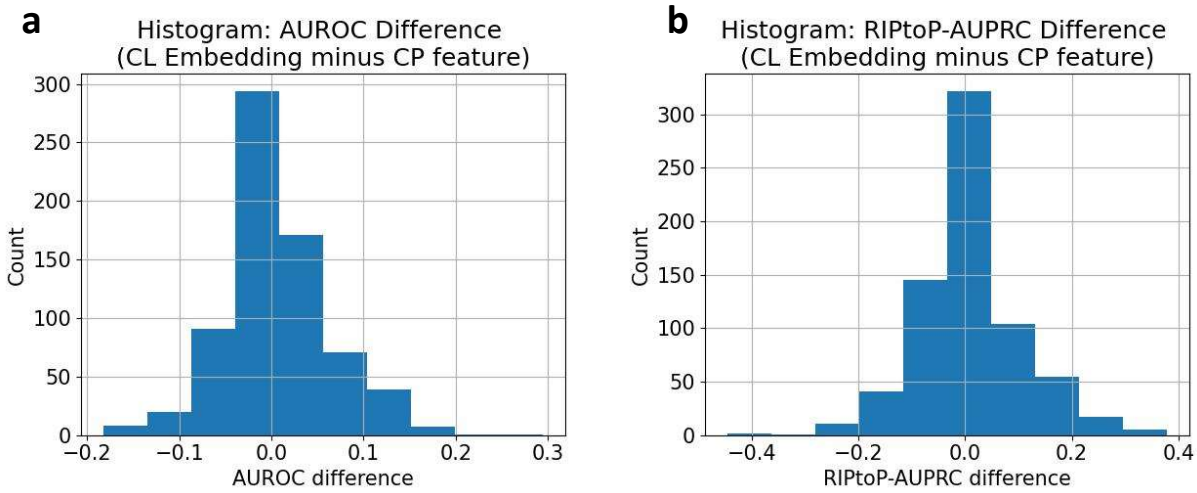
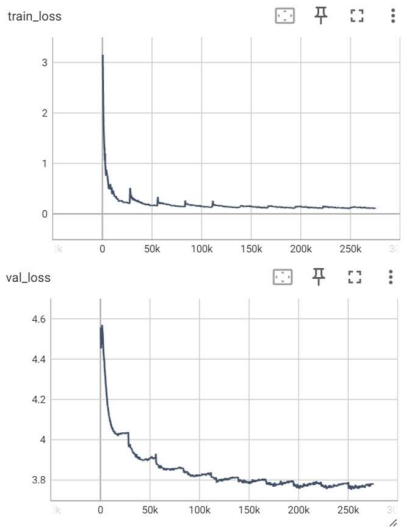



Figure 6.6: **Histogram of a) AUROC b) RIPtoP-AUPRC difference (CL embedding minus CP feature).**

## 6.7 Chapter 4: Loss curves of the Pretraining Process

**a**

| | |
|---|---|
| Embedding shape | 512 |
| Batch size | 1024 |
| Optimizer | AdamW |
| Learning rate | 1e-3 |
| Weight decay | 1e-6 |
| Learning rate scheduler | CosineAnnealingWarmRestarts |
| Num epochs between each restart | 101 |
| Temperature (contrastive loss) | 0.1 |
| CP encoder hidden layers | [1024, 1024, 1024] |
| TX encoder hidden layers | [4096, 4096, 4096] |
| Epochs | 2000 |
| Gradient clipping (norm) value | 5 |
| Dropout | 0.1 |




**b**

| | |
|---|---|
| Embedding shape/ Latent dimension | 512 |
| Batch size | 512 |
| Optimizer | Adam |
| Learning rate | 1e-3 |
| Weight decay | 1e-6 |
| Learning rate scheduler | CosineAnnealingWarmRestarts |
| Num epochs between each restart | 101 |
| CP encoder hidden layers | [1024, 512, 512] |
| CP decoder hidden layers | [512, 512, 1024] |
| TX encoder hidden layers | [1024, 2048, 4096] |
| Epochs | 2000 |
| Gradient clipping (norm) value | 5 |
| Dropout | 0.1 |

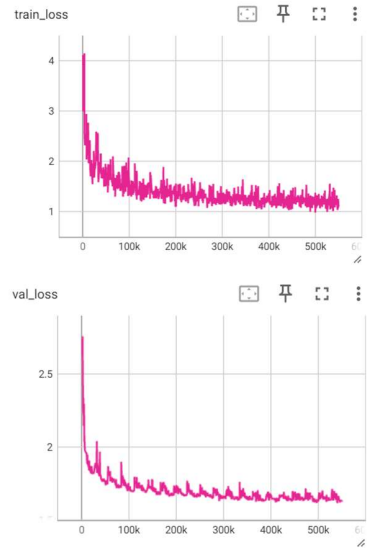



Figure 6.7: **Hyperparameters and train/validation loss curves for a) contrastive learning, b) bimodal autoencoder.**

# Bibliography


[1] John G Moffat, Joachim Rudolph, and David Bailey. "Phenotypic screening in cancer drug discovery - past, present and future". en. In: *Nat. Rev. Drug Discov.* 13.8 (Aug. 2014), pp. 588–602.

[2] Cory M Johannessen, Paul A Clemons, and Bridget K Wagner. "Integrating phenotypic small-molecule profiling and human genetics: the next phase in drug discovery". en. In: *Trends Genet.* 31.1 (Jan. 2015), pp. 16–23.

[3] Juan C Caicedo, Shantanu Singh, and Anne E Carpenter. "Applications in image-based profiling of perturbations". In: *Current Opinion in Biotechnology* 39 (2016), pp. 134–142. ISSN: 0958-1669. DOI: https://doi.org/10.1016/j.copbio.2016.04.003. URL: https://www.sciencedirect.com/science/article/pii/S0958166916301112.

[4] Felix Reisen et al. "Linking phenotypes and modes of action through high-content screen fingerprints". en. In: *Assay Drug Dev. Technol.* 13.7 (Sept. 2015), pp. 415–427.

[5] Daniel W Young et al. "Integrating high-content screening and ligand-target prediction to identify mechanism of action". en. In: *Nat. Chem. Biol.* 4.1 (Jan. 2008), pp. 59–68.

[6] Vebjorn Ljosa et al. "Comparison of methods for image-based profiling of cellular morphological responses to small-molecule treatment". en. In: *J. Biomol. Screen.* 18.10 (Dec. 2013), pp. 1321–1329.

[7] Claudio Collinet et al. "Systems survey of endocytosis by multiparametric image analysis". In: *Nature* 464.7286 (Mar. 2010), pp. 243–249. ISSN: 1476-4687. DOI: 10.1038/nature08779. URL: https://doi.org/10.1038/nature08779.

[8] Florian Fuchs et al. "Clustering phenotype populations by genome-wide RNAi and multiparametric imaging". en. In: *Mol. Syst. Biol.* 6.1 (June 2010), p. 370.

[9] Mark-Anthony Bray et al. "Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes". In: *Nature Protocols* 11.9 (2016), 1757–1774. DOI: 10.1038/nprot.2016.105.

[10] Dorota Herman et al. "Leveraging Cell Painting Images to Expand the Applicability Domain and Actively Improve Deep Learning Quantitative Structure–Activity Relationship Models". In: *Chemical Research in Toxicology* 36.7 (July 2023), pp. 1028–1036. ISSN: 0893-228X. DOI: 10.1021/acs.chemrestox.2c00404. URL: https://doi.org/10.1021/acs.chemrestox.2c00404.

[11] Marina Garcia de Lomana, Paula Andrea Marin Zapata, and Floriane Montanari. "Predicting the Mitochondrial Toxicity of Small Molecules: Insights from Mechanistic Assays and Cell Painting Data". In: *Chemical Research in Toxicology* 36.7 (July 2023), pp. 1107–1120. ISSN: 0893-228X. DOI: 10.1021/acs.chemrestox.3c00086. URL: https://doi.org/10.1021/acs.chemrestox.3c00086.

[12] Anika Liu et al. "Using chemical and biological data to predict drug toxicity". In: *SLAS Discovery* 28.3 (2023), pp. 53–64. ISSN: 2472-5552. DOI: https://doi.org/10.1016/j.slasd.2022.12.003. URL: https://www.sciencedirect.com/science/article/pii/S2472555222137147.

[13] Srijit Seal et al. "Integrating cell morphology with gene expression and chemical structure to aid mitochondrial toxicity detection". In: *Communications Biology* 5.1 (Aug. 2022), p. 858. ISSN: 2399-3642. DOI: 10.1038/s42003-022-03763-5. URL: https://doi.org/10.1038/s42003-022-03763-5.

[14] Mohammad Akbarzadeh et al. "Morphological profiling by means of the Cell Painting assay enables identification of tubulin-targeting compounds". In: *Cell Chemical Biology* 29.6 (2022), 1053–1064.e3. ISSN: 2451-9456. DOI: 10.1016/j.chembiol.2021.12.009. URL: http://dx.doi.org/10.1016/j.chembiol.2021.12.009.

[15] Srijit Seal et al. "Merging bioactivity predictions from cell morphology and chemical fingerprint models using similarity to training data". In: *Journal of Cheminformatics* 15.1 (June 2023), p. 56. ISSN: 1758-2946. DOI: 10.1186/s13321-023-00723-x. URL: https://doi.org/10.1186/s13321-023-00723-x.

[16] Mark-Anthony Bray et al. "A dataset of images and morphological profiles of 30 000 small-molecule treatments using the Cell Painting assay". In: *GigaScience* 6.12 (Jan. 2017). giw014. ISSN: 2047-217X. DOI: 10.1093/gigascience/giw014. eprint: https://academic.oup.com/gigascience/article-pdf/6/12/giw014/25513584/giw014_reviewer-2-report-(original-submission).pdf. URL: https://doi.org/10.1093/gigascience/giw014.

[17] Srinivas Niranj Chandrasekaran et al. "JUMP Cell Painting dataset: morphological impact of 136,000 chemical and genetic perturbations". In: *bioRxiv* (2023). DOI: 10.1101/2023.03.23.534023. eprint: https://www.biorxiv.org/content/early/2023/03/27/2023.03.23.534023.full.pdf. URL: https://www.biorxiv.org/content/early/2023/03/27/2023.03.23.534023.

[18] David R. Stirling et al. "CellProfiler 4: improvements in speed, utility and usability". In: *BMC Bioinformatics* 22.1 (Sept. 2021). ISSN: 1471-2105. DOI: 10.1186/s12859-021-04344-9. URL: http://dx.doi.org/10.1186/s12859-021-04344-9.

[19] Darko Butina, Matthew D Segall, and Katrina Frankcombe. "Predicting ADME properties in silico: methods and models". en. In: *Drug Discov. Today* 7.11 (June 2002), S83–8.

[20] Ross Irwin et al. "Chemformer: a pre-trained transformer for computational chemistry". In: *Machine Learning: Science and Technology* 3.1 (Jan. 2022), p. 015022. ISSN: 2632-2153. DOI: 10.1088/2632-2153/ac3ffb. URL: http://dx.doi.org/10.1088/2632-2153/ac3ffb.

[21] Jaak Simm et al. "Repurposing High-Throughput Image Assays Enables Biological Activity Prediction for Drug Discovery". In: *Cell Chemical Biology* 25.5 (2018), 611–618.e3. ISSN: 2451-9456. DOI: https://doi.org/10.1016/j.chembiol.2018.01.015. URL: https://www.sciencedirect.com/science/article/pii/S2451945618300370.

[22] Johan Fredin Haslum et al. "Cell Painting-based bioactivity prediction boosts high-throughput screening hit-rates and compound diversity". In: *Nature Communications* 15.1 (Apr. 2024). ISSN: 2041-1723. DOI: 10.1038/s41467-024-47171-1. URL: http://dx.doi.org/10.1038/s41467-024-47171-1.

[23] Megan Stanley et al. "FS-Mol: A Few-Shot Learning Dataset of Molecules". In: *NeurIPS 2021 Track Datasets and Benchmarks* (2021). DOI: https://doi.org/10.1016/j.chembiol.2018.01.015. URL: https://www.sciencedirect.com/science/article/pii/S2451945618300370.

[24] Axel Pahl et al. "Morphological subprofile analysis for bioactivity annotation of small molecules". In: *Cell Chemical Biology* 30.7 (2023), 839–853.e7. ISSN: 2451-9456. DOI: https://doi.org/10.1016/j.chembiol.2023.06.003. URL: https://www.sciencedirect.com/science/article/pii/S2451945623001599.

[25] Vladislav Kim et al. "Self-supervision advances morphological profiling by unlocking powerful image representations". In: *bioRxiv* (2024). DOI: 10.1101/2023.04.28.538691. eprint: https://www.biorxiv.org/content/early/2024/01/03/2023.04.28.538691.full.pdf. URL: https://www.biorxiv.org/content/early/2024/01/03/2023.04.28.538691.

[26] Oren Kraus et al. *Masked Autoencoders for Microscopy are Scalable Learners of Cellular Biology*. 2024. arXiv: 2404.10242 [cs.CV]. URL: https://arxiv.org/abs/2404.10242.

[27] Ana Sanchez-Fernandez et al. "CLOOME: contrastive learning unlocks bioimaging databases for queries with chemical structures". In: *Nature Communications* 14.1 (Nov. 2023). ISSN: 2041-1723. DOI: 10.1038/s41467-023-42328-w. URL: http://dx.doi.org/10.1038/s41467-023-42328-w.

[28] Chenyu Wang et al. *Removing Biases from Molecular Representations via Information Maximization*. 2023. arXiv: 2312.00718 [cs.LG]. URL: https://arxiv.org/abs/2312.00718.

[29] Cuong Q. Nguyen, Dante Pertusi, and Kim M. Branson. *Molecule-Morphology Contrastive Pretraining for Transferable Molecular Representation*. 2023. arXiv: 2305.09790 [q-bio.BM]. URL: https://arxiv.org/abs/2305.09790.

[30] Shuangjia Zheng et al. "Cross-Modal Graph Contrastive Learning with Cellular Images". In: *Advanced Science* (June 2024). ISSN: 2198-3844. DOI: 10.1002/advs.202404845. URL: http://dx.doi.org/10.1002/advs.202404845.

[31] Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. "scGen predicts single-cell perturbation responses". In: *Nature Methods* 16.8 (July 2019), 715–721. ISSN: 1548-7105. DOI: 10.1038/s41592-019-0494-8. URL: http://dx.doi.org/10.1038/s41592-019-0494-8.

[32] Minsheng Hao et al. "Large-scale foundation model on single-cell transcriptomics". In: *Nature Methods* (June 2024). ISSN: 1548-7105. DOI: 10.1038/s41592-024-02305-7. URL: http://dx.doi.org/10.1038/s41592-024-02305-7.

[33] Haotian Cui et al. "scGPT: toward building a foundation model for single-cell multi-omics using generative AI". In: *Nature Methods* (Feb. 2024). ISSN: 1548-7105. DOI: 10.1038/s41592-024-02201-0. URL: http://dx.doi.org/10.1038/s41592-024-02201-0.

[34] Christina V. Theodoris et al. "Transfer learning enables predictions in network biology". In: *Nature* 618.7965 (May 2023), 616–624. ISSN: 1476-4687. DOI: 10.1038/s41586-023-06139-9. URL: http://dx.doi.org/10.1038/s41586-023-06139-9.

[35] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[36] Tom B. Brown et al. "Language Models are Few-Shot Learners". In: *arXiv:2005.14165 [cs.CL]* (2020). arXiv: 2005.14165 [cs.CL]. URL: https://arxiv.org/abs/2005.14165.

[37] Karren Dai Yang et al. "Multi-domain translation between single-cell imaging and sequencing data using autoencoders". In: *Nature Communications* 12.1 (Jan. 2021). ISSN: 2041-1723. DOI: 10.1038/s41467-020-20249-2. URL: http://dx.doi.org/10.1038/s41467-020-20249-2.

[38] Jake Snell, Kevin Swersky, and Richard S. Zemel. "Prototypical Networks for Few-shot Learning". In: *arXiv:1703.05175 [cs.LG]* (2017). arXiv: 1703.05175 [cs.LG].

[39] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *arXiv:1703.03400 [cs.LG]* (2017). arXiv: 1703.03400 [cs.LG].

[40] Ruiying Geng et al. "Induction Networks for Few-Shot Text Classification". In: *arXiv:1902.10482 [cs.CL]* (2019). arXiv: 1902.10482 [cs.CL].

[41] Oriol Vinyals et al. "Matching Networks for One Shot Learning". In: *arXiv:1606.04080 [cs.LG]* (2017). arXiv: 1606.04080 [cs.LG].

[42] David Mendez et al. "ChEMBL: towards direct deposition of bioassay data". In: *Nucleic Acids Research* 47.D1 (Nov. 2018), pp. D930–D940. ISSN: 0305-1048. DOI: 10.1093/nar/gky1075. eprint: https://academic.oup.com/nar/article-pdf/47/D1/D930/27437436/gky1075.pdf. URL: https://doi.org/10.1093/nar/gky1075.

[43] Markus Hofmarcher et al. "Accurate Prediction of Biological Assays with High-Throughput Microscopy Images and Convolutional Networks". In: *Journal of Chemical Information and Modeling* 59.3 (2019), pp. 1163–1171. DOI: 10.1021/acs.jcim.8b00670. eprint: https://doi.org/10.1021/acs.jcim.8b00670. URL: https://doi.org/10.1021/acs.jcim.8b00670.

[44] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *arXiv:1512.03385 [cs.CV]* (2015). arXiv: 1512.03385 [cs.CV]. URL: https://arxiv.org/abs/1512.03385.

[45] J. Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: (2009).

[46] Lauren Schiff et al. "Integrating deep learning and unbiased automated high-content screening to identify complex disease signatures in human fibroblasts". In: *Nature Communications* 13.1 (Mar. 2022), p. 1590. ISSN: 2041-1723. DOI: 10.1038/s41467-022-28423-4. URL: https://doi.org/10.1038/s41467-022-28423-4.

[47] Christian Szegedy et al. "Going Deeper with Convolutions". In: *arXiv:1409.4842 [cs.CV]* (2014). arXiv: 1409.4842 [cs.CV].

[48] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *arXiv:1512.00567 [cs.CV]* (2015). arXiv: 1512.00567 [cs.CV].

[49] Ross Wightman. *Pytorch Image Models (TIMM)*. URL: https://timm.fast.ai/.

[50] Anne Carpenter et al. "CellProfiler: Image analysis software for identifying and quantifying cell phenotypes". In: *Genome biology* 7 (Feb. 2006), R100. DOI: 10.1186/gb-2006-7-10-r100.

[51] Jacob Cohen. "A Coefficient of Agreement for Nominal Scales". In: *Educational and Psychological Measurement* 20.1 (1960), pp. 37–46. DOI: 10.1177/001316446002000104. eprint: https://doi.org/10.1177/001316446002000104. URL: https://doi.org/10.1177/001316446002000104.

[52] David Rogers and Mathew Hahn. "Extended-Connectivity Fingerprints". In: *Journal of Chemical Information and Modeling* 50.5 (2010). PMID: 20426451, pp. 742–754. DOI: 10.1021/ci100050t. eprint: https://doi.org/10.1021/ci100050t. URL: https://doi.org/10.1021/ci100050t.

[53] Andrea Mauri et al. "DRAGON software: An easy approach to molecular descriptor calculations". In: *MATCH Communications in Mathematical and in Computer Chemistry* 56 (Jan. 2006), pp. 237–248.

[54] Othman Soufan et al. "DPubChem: a web tool for QSAR modeling and high-throughput virtual screening". In: *Scientific Reports* 8.1 (June 2018), p. 9110. ISSN: 2045-2322. DOI: 10.1038/s41598-018-27495-x. URL: https://doi.org/10.1038/s41598-018-27495-x.

[55] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[56] Andreas Mayr et al. "DeepTox: Toxicity Prediction using Deep Learning". In: *Frontiers in Environmental Science* 3 (2016). ISSN: 2296-665X. DOI: 10.3389/fenvs.2015.00080. URL: https://www.frontiersin.org/articles/10.3389/fenvs.2015.00080.

[57] Weihua Hu et al. "Strategies for Pre-training Graph Neural Networks". In: *arXiv:1905.12265 [cs.LG]* (2020). arXiv: 1905.12265 [cs.LG].

[58] Lilian Weng. "Meta-Learning: Learning to Learn Fast". In: *lilianweng.github.io* (2018). Accessed: 2023-04-26. URL: https://lilianweng.github.io/posts/2018-11-30-meta-learning/.

[59] Krishna Chaitanya et al. "Contrastive learning of global and local features for medical image segmentation with limited annotations". In: *arXiv:2006.10511 [cs.CV]* (2020). arXiv: 2006.10511 [cs.CV].

[60] Alec Radford et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *arXiv:2103.00020 [cs.CV]* (2021). arXiv: 2103.00020 [cs.CV]. URL: https://arxiv.org/abs/2103.00020.

[61] Ting Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: (2020). arXiv: 2002.05709 [cs.LG]. URL: https://arxiv.org/abs/2002.05709.

[62] Jason Ansel et al. "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation". In: *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, 2024. DOI: 10.1145/3620665.3640366. URL: https://pytorch.org/assets/pytorch2-2.pdf.

[63] William Falcon and The PyTorch Lightning team. *PyTorch Lightning*. Version 1.4. 2019. DOI: 10.5281/zenodo.3828935. URL: https://github.com/Lightning-AI/lightning.

[64] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929.

[65] Zhuang Liu et al. *A ConvNet for the 2020s*. 2022. arXiv: 2201.03545 [cs.CV]. URL: https://arxiv.org/abs/2201.03545.

[66] Guy W. Bemis and Mark A. Murcko. "The Properties of Known Drugs. 1. Molecular Frameworks". In: *Journal of Medicinal Chemistry* 39.15 (Jan. 1996), 2887–2893. ISSN: 1520-4804. DOI: 10.1021/jm9602928. URL: http://dx.doi.org/10.1021/jm9602928.

[67] Jiangming Sun et al. "Applying Mondrian Cross-Conformal Prediction To Estimate Prediction Confidence on Large Imbalanced Bioactivity Data Sets". In: *Journal of Chemical Information and Modeling* 57.7 (July 2017), pp. 1591–1598. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.7b00159. URL: https://doi.org/10.1021/acs.jcim.7b00159.

[68] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[69] Wouter Heyndrickx et al. "MELLODDY: Cross-pharma Federated Learning at Unprecedented Scale Unlocks Benefits in QSAR without Compromising Proprietary Information". In: *Journal of Chemical Information and Modeling* (Aug. 2023). ISSN: 1549-9596. DOI: 10.1021/acs.jcim.3c00799. URL: https://doi.org/10.1021/acs.jcim.3c00799.

[70] Nora Anderson and Jürgen Borlak. "Drug-induced phospholipidosis". en. In: *FEBS Lett.* 580.23 (Oct. 2006), pp. 5533–5540.

[71] Raquel Rodríguez-Pérez and Grégori Gerebtzoff. "Identification of bile salt export pump inhibitors using machine learning: Predictive safety from an industry perspective". In: *Artificial Intelligence in the Life Sciences* 1 (2021), p. 100027. ISSN: 2667-3185. DOI: https://doi.org/10.1016/j.ailsci.2021.100027. URL: https://www.sciencedirect.com/science/article/pii/S2667318521000271.

[72] Laleh Kamalian et al. "Acute Metabolic Switch Assay Using Glucose/Galactose Medium in HepaRG Cells to Detect Mitochondrial Toxicity". en. In: *Curr Protoc Toxicol* 80.1 (May 2019), e76.

[73] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. *Representation Learning with Contrastive Predictive Coding*. 2019. arXiv: 1807.03748 [cs.LG]. URL: https://arxiv.org/abs/1807.03748.

[74] Chloe X Wang, Lin Zhang, and Bo Wang. "One Cell At a Time (OCAT): a unified framework to integrate and analyze single-cell RNA-seq data". en. In: *Genome Biol.* 23.1 (Apr. 2022), p. 102.

[75] Zhong Wang, Mark Gerstein, and Michael Snyder. "RNA-Seq: a revolutionary tool for transcriptomics". In: *Nature Reviews Genetics* 10.1 (Jan. 2009), 57–63. ISSN: 1471-0064. DOI: 10.1038/nrg2484. URL: http://dx.doi.org/10.1038/nrg2484.

[76] Lorena Zubovic et al. "The altered transcriptome of pediatric myelodysplastic syndrome revealed by RNA sequencing". en. In: *J. Hematol. Oncol.* 13.1 (Oct. 2020), p. 135.

[77] Alicia Oshlack, Mark D Robinson, and Matthew D Young. "From RNA-seq reads to differential expression results". en. In: *Genome Biol.* 11.12 (Dec. 2010), p. 220.

[78] Meinusha Govindarajan et al. "High-throughput approaches for precision medicine in high-grade serous ovarian cancer". en. In: *J. Hematol. Oncol.* 13.1 (Oct. 2020), p. 134.

[79] Yongshuo Zong, Oisin Mac Aodha, and Timothy Hospedales. *Self-Supervised Multimodal Learning: A Survey*. 2024. arXiv: 2304.01008 [cs.LG]. URL: https://arxiv.org/abs/2304.01008.

[80] Shagun Uppal et al. *Multimodal Research in Vision and Language: A Review of Current and Emerging Trends*. 2020. arXiv: 2010.09522 [cs.CV]. URL: https://arxiv.org/abs/2010.09522.

[81] Lucas Beyer et al. *PaliGemma: A versatile 3B VLM for transfer*. 2024. arXiv: 2407.07726 [cs.CV]. URL: https://arxiv.org/abs/2407.07726.

[82] Shaoxiang Chen, Ting Yao, and Yu-Gang Jiang. "Deep Learning for Video Captioning: A Review". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 6283–6290. DOI: 10.24963/ijcai.2019/877. URL: https://doi.org/10.24963/ijcai.2019/877.

[83] Keli Huang et al. *Multi-modal Sensor Fusion for Auto Driving Perception: A Survey*. 2022. arXiv: 2202.02703 [cs.CV]. URL: https://arxiv.org/abs/2202.02703.

[84] Julián N Acosta et al. "Multimodal biomedical AI". en. In: *Nat. Med.* 28.9 (Sept. 2022), pp. 1773–1784.

[85] Linus Ericsson et al. "Self-Supervised Representation Learning: Introduction, advances, and challenges". In: *IEEE Signal Processing Magazine* 39.3 (May 2022), 42–62. ISSN: 1558-0792. DOI: 10.1109/msp.2021.3134634. URL: http://dx.doi.org/10.1109/MSP.2021.3134634.

[86] Jiquan Ngiam et al. *Multimodal deep learning*. 2011.

[87] Benjamin Kaminow, Dinar Yunusov, and Alexander Dobin. "STARsolo: accurate, fast and versatile mapping/quantification of single-cell and single-nucleus RNA-seq data". In: *bioRxiv* (2021). DOI: 10.1101/2021.05.05.442755. eprint: https://www.biorxiv.org/content/early/2021/05/05/2021.05.05.442755.full.pdf. URL: https://www.biorxiv.org/content/early/2021/05/05/2021.05.05.442755.

[88] Simon Anders Michael Love. *DESeq2*. 2017. DOI: 10.18129/B9.BIOC.DESEQ2. URL: https://bioconductor.org/packages/DESeq2.

[89] Gordon Smyth [Cre, Aut], Yifang Hu [Ctb], Matthew Ritchie [Ctb], Jeremy Silver [Ctb], James Wettenhall [Ctb], Davis McCarthy[Ctb], Di Wu [Ctb], Wei Shi [Ctb], Belinda Phipson [Ctb], AaronLun [Ctb], Natalie Thorne [Ctb], Alicia Oshlack [Ctb], Carolynde Graaf [Ctb], Yunshun Chen [Ctb], Mette Langaas [Ctb], EgilFerkingstad [Ctb], Marcus Davy [Ctb], Francois Pepin [Ctb], Dongseok Choi [Ctb]. *limma*. 2017. DOI: 10.18129/B9.BIOC.LIMMA. URL: https://bioconductor.org/packages/limma.

[90] Feng Wang and Huaping Liu. *Understanding the Behaviour of Contrastive Loss*. 2021. arXiv: 2012.09740 [cs.LG]. URL: https://arxiv.org/abs/2012.09740.

[91] Yonglong Tian, Dilip Krishnan, and Phillip Isola. "Contrastive Multiview Coding". In: (2020). arXiv: 1906.05849 [cs.CV]. URL: https://arxiv.org/abs/1906.05849.

[92] Rens Janssens et al. "Fully unsupervised deep mode of action learning for phenotyping high-content cellular images". In: *Bioinformatics* 37.23 (July 2021), pp. 4548–4555. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btab497. eprint: https://academic.oup.com/bioinformatics/article-pdf/37/23/4548/50579588/btab497_supplementary_data.pdf. URL: https://doi.org/10.1093/bioinformatics/btab497.

[93] Arthur Liberzon et al. "The Molecular Signatures Database Hallmark Gene Set Collection". In: *Cell Systems* 1.6 (Dec. 2015), 417–425. ISSN: 2405-4712. DOI: 10.1016/j.cels.2015.12.004. URL: http://dx.doi.org/10.1016/j.cels.2015.12.004.

[94] Sonja Hänzelmann, Robert Castelo, and Justin Guinney. "GSVA: gene set variation analysis for microarray and RNA-Seq data". In: *BMC Bioinformatics* 14.1 (Jan. 2013).

ISSN: 1471-2105. DOI: 10.1186/1471-2105-14-7. URL: http://dx.doi.org/10.1186/1471-2105-14-7.

[95] Son V. Ha, Lucas Leuschner, and Paul Czodrowski. "FSL-CP: a benchmark for small molecule activity few-shot prediction using cell microscopy images". In: *Digital Discovery* 3.4 (2024), 719–727. ISSN: 2635-098X. DOI: 10.1039/d3dd00205e. URL: http://dx.doi.org/10.1039/D3DD00205E.

[96] Daniel Vella and Jean-Paul Ebejer. "Few-Shot Learning for Low-Data Drug Discovery". In: *Journal of Chemical Information and Modeling* 63.1 (2023). PMID: 36410391, pp. 27–42. DOI: 10.1021/acs.jcim.2c00779. eprint: https://doi.org/10.1021/acs.jcim.2c00779. URL: https://doi.org/10.1021/acs.jcim.2c00779.

[97] Dylan C. Mitchell et al. "A proteome-wide atlas of drug mechanism of action". In: *Nature Biotechnology* 41.6 (Jan. 2023), 845–857. ISSN: 1546-1696. DOI: 10.1038/s41587-022-01539-0. URL: http://dx.doi.org/10.1038/s41587-022-01539-0.

[98] Stephan Eckert et al. "Decrypting the molecular basis of cellular drug phenotypes by dose-resolved expression proteomics". In: *Nature Biotechnology* (May 2024). ISSN: 1546-1696. DOI: 10.1038/s41587-024-02218-y. URL: http://dx.doi.org/10.1038/s41587-024-02218-y.