Utrecht University

Master Thesis

Challenging the i.i.d. assumption in molecular modelling

Author: Cas Wognum (6934501)

1st supervisor:	Dr. ir. R. W. (Ronald) Poppe	
daily supervisor:	P. (Prudencio) Tossou	(Valence Discovery)
2nd reader:	Dr. M. (Thijs) van Ommen	

A thesis submitted in fulfillment of the requirements for the double Master of Science degree in Artificial Intelligence and Game and Media Technology

August 21, 2022

"Biology is likely far too complex and messy to ever be encapsulated as a simple set of neat mathematical equations. But just as mathematics turned out to be the right description language for physics, biology may turn out to be the perfect type of regime for the application of AI"

Demis Hassabis, CEO of DeepMind and Isomorphic Labs

Abstract

Molecular scoring, in which a machine learning model is used as an in-silico proxy for an otherwise expensive and slow in-silico or in-vivo experiment, is a promising direction to improve the efficiency of the drug discovery process. While it is commonly assumed that OOD generalization is needed in molecular scoring, a principled and complete problem specification of the OOD problem as encountered in ongoing drug discovery programs is still missing.

We therefore propose the Molecular Out-Of-Distribution (MOOD) framework, which consists of two parts. With the MOOD *specification*, we propose a new set of evaluation standards that more closely matches the situations encountered in ongoing drug discovery programs. Specifically, we propose the usage of a continuous, representation-dependent and distance-based OOD metric to characterize, compare and replicate realistic distribution shifts. In the MOOD *investigation*, we use the newly proposed evaluation standards to benchmark various tools on how they affect OOD generalization.

We find that current evaluation standards do not match the situations encountered in practice and that while some effective methods have been developed over the years to improve generalization, more efforts are needed to close the gap between advances in academia and industry pain points. To that end, we hope that MOOD can help to inform future research directions and to more efficiently direct resources in ongoing drug discovery programs.

Acknowledgements

I feel incredibly lucky to have been surrounded by bright, passionate and warm-hearted people throughout the pursuit of this thesis. Thank you to all for your trust, your guidance and your friendship.

To Prudencio Tossou, for his incredible patience with me, for the open-hearted discussions and for the many laughs. Énan tchè nou wé!

To Ronald Poppe, for giving me the space to explore my own interests and for his personal and attentionate supervision. Dankjewel!

To the whole team at Valence Discovery, for the warm welcome in Montréal, for their inspiring expertise and for their contagious enthusiasm. Merci!

To Ariane, my friends and my family, for their unconditional love and support.

This work would not have been possible without you. I am happy that our paths crossed and look forward to what's next.

This work was supported by Mitacs through the Mitacs Accelerate International-To Canada Program.

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

1

Introduction

Drug-discovery is the process of identifying and testing potential new medicines (or *drugs*). It is a multi-disciplinary process that builds upon state-of-the-art knowledge from fields such as biology, chemistry, and computer science. In many cases, a drug is a *small molecule* that either *inhibits* or *activates* a *protein* to achieve a medicinal effect. Even when molecules with the desired activity can be found, this does not automatically imply that they are viable drugs. For example, a candidate drug could be too toxic to humans or too difficult to organically synthesize. Due to the presence of several rate-limiting steps and its overall high failure rate, the average cost of bringing a single new drug to market is in the billions and the whole process generally requires over a decade (1, 2, 3).

Computational (or *in-silico*) methods hold the promise to improve the efficiency of the drug discovery process. These methods could enable efficient exploration of the chemical space and reduce the need for slow, expensive and arguably unethical experiments. To model the highly complex biochemical systems of interest in drug discovery, *machine learning* (ML) has proven an effective and efficient tool (4, 5, 6). More recently, the application of *deep learning* (DL) to biochemistry has attracted lots of attention (7, 8, 9, 10, 11, 12). DL comprises a particular class of ML methods which can extract sophisticated representations directly from the data, opening up possibilities in cases where a system is either not yet well enough understood or too complex to manually engineer a set of informative features. Over the years, a diverse toolbox of ML techniques has been developed to support human researchers in the experiments they conduct throughout the drug discovery process. In this study, we focus on *molecular scoring*, in which an ML model is trained to predict the outcome of a biochemical experiment.

1. INTRODUCTION

While ML has disrupted several scientific disciplines, such as natural language processing and computer vision, it has proven more difficult to achieve successes outside of academia. A common assumption in ML is that the train and test data are independent and identically distributed. This is known as the *i.i.d.* assumption. In the i.i.d. setting, ML models are prone to learn statistically significant but spurious correlations that do not generalize well to differently distributed datasets. In practice, however, the i.i.d. assumption is difficult to satisfy. This causes ML models to disappoint outside of the controlled environments of academia (13). In drug discovery, a model could, for example, be expected to fail on molecules that are structurally dissimilar from the training data. The decrease in performance of an ML model on differently distributed data is formalized by the *domain shift* problem. Finding solutions to it is an active field of research (14, 15), which is broadly referred to as out-of-distribution (OOD) generalization.

This raises the question to what extent the i.i.d. assumption holds for the molecular datasets used in drug discovery. After all, human chemists do not randomly sample the chemical space, but build upon empirical and theoretical knowledge to bias their exploration. A chemist will, for example, iteratively make small structural changes to refine a molecule. We could reasonably expect this human bias to transfer over to the datasets used in ML. Additionally, due to the sheer size of the space of drug-like molecules (estimates range from 10^{20} (16) to 10^{63} (17)), we could expect to regularly encounter out-of-distribution molecules on which our models do not perform as well. Yet the i.i.d. assumption is, either implicitly or explicitly, assumed to hold in most molecular scoring tasks in ongoing drug discovery programs. In this thesis we therefore set out to challenge the i.i.d assumption in molecular scoring by providing a complete specification of the OOD problem as encountered in drug discovery programs and by investigating the effect and importance of various tools to improve generalization through the lens of this new specification.

The rest of this thesis is structured as follows: Chapter 2 provides a succinct summary of relevant background knowledge by discussing the drug discovery process and the role of machine learning within it. Chapter 3 provides an overview of prior work that is more directly related to the research conducted in this thesis. Chapter 4 describes the Molecular Out-Of-Distribution (or *MOOD*) framework, which is the main contribution of this thesis and constitutes various experiments and the results thereof. Finally, Chapter 5 summarizes the outcomes of this research and suggests future research directions.

2

Background

The goal of this chapter is to provide the reader with the relevant background knowledge to understand the rest of this thesis. In Section 2.1, we will give an overview of the different steps of the drug discovery process. In Section 2.2, we will detail the role of ML within that process, specifically focusing on the applications of molecular scoring. Finally, as the low quantity and quality of data in molecular scoring pose an additional challenge compared to other ML domains, Section 2.3 will discuss molecular data and some of the ways this challenge is being tackled.

2.1 The Drug Discovery Process

The *drug discovery and development* process encompasses all the steps by which a new drug is brought to market (see Figure 2.1). The steps are completed sequentially and the number of candidate drugs that is considered in each step decreases exponentially. Due to the sequential nature of this process, there are several rate-limiting steps (i.e. bottlenecks) that have to be optimized simultaneously to improve the overall efficiency. In this work, the focus is on drug discovery. Drug discovery can generally be divided into four phases: target discovery, hit generation, hit-to-lead and lead optimization. In this section, we will discuss how each of these steps is traditionally structured.

In *target discovery*, the goal is to identify and validate a drug target (19). A target is a biochemical entity in the human cell that has a causal association with a disease. There exist different types of targets (e.g. a macromolecule such as a protein or a nucleic acid) and there are different mechanisms of action to affect a target and achieve a medicinal effect (e.g. inhibition or activation). In target discovery it is essential that we have a fundamental

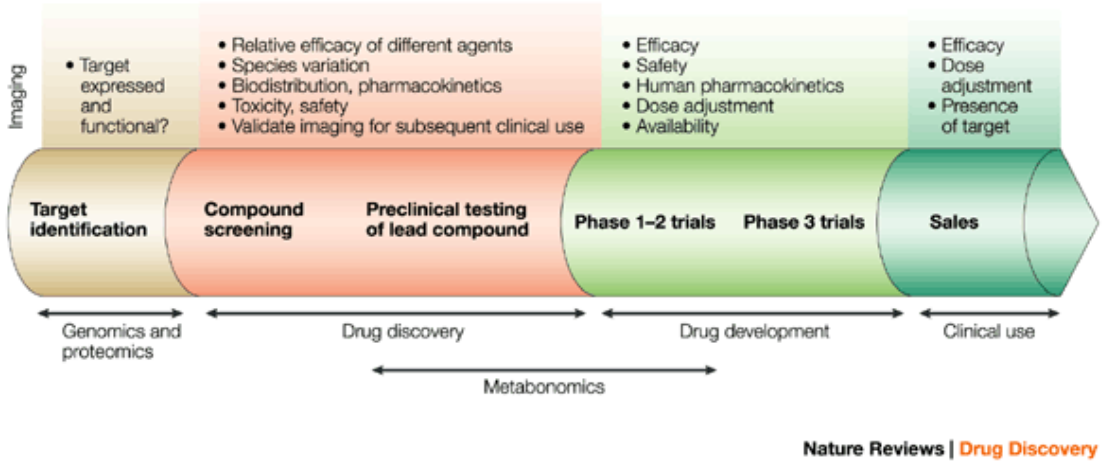

Figure 2.1: The drug discovery and development process - Target discovery and drug discovery are considered separate phases here. In this work, they are considered to be part of the same phase. Taken from Rudin et al. (18).

understanding of both the function and structure of the human cell on a molecular level. This is why target discovery in pharamaceutical companies is closely entangled with basic research in academia and not always considered part of the drug discovery process.

In *hit generation*, the goal is to identify a set of molecules from a chemical library with promising activity against the target. This is accomplished by a high-throughput screening (HTS) (20). HTS is a costly, large scale experiment that measures the ability of a molecule to modulate a target of interest. This can be done practically in a specialized lab or in-silico (e.g. through simulations). The hit molecules are used as a starting point for further experimentation in the hit-to-lead phase.

In *hit-to-lead*, the goal is to iteratively filter and refine the hit molecules to come to a set of *lead molecules* which all surpass a predetermined, *in-vitro* activity threshold against the drug target. This is done through a cyclic approach called the *design-make-test-analyze* (DMTA) cycle (22) (see Figure 2.2). This cycle is repeated until a satisfactory number of lead molecules is found or until time or money runs out.

In *lead-optimization*, the goal is to safely transfer the in-vitro efficacy of the lead molecules to *in-vivo* efficacy by optimizing for multiple pragmatic properties (23). These properties are generally summarized as absorption, distribution, metabolism, excretion and toxicity

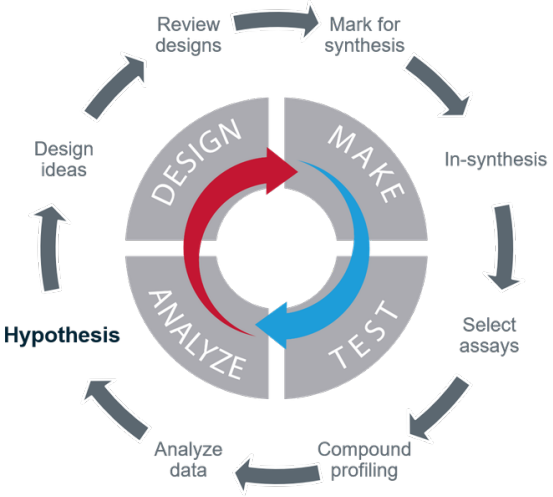

Figure 2.2: The DMTA cycle - Through this cyclic process a set of molecules is iteratively refined. Taken from (21).

(ADMET) and describe how a drug enters, spreads through and leaves the body and how harmful the drug is to humans. Other properties do exist and the specific properties of interest depend on the program. For example, a drug which is taken orally has different requirements than a drug which is given by an intravenous injection. When a drug is found with the desired binding activity and the right set of ADMET properties, the drug discovery process ends and the drug is prepared for clinical trials.

Now that we have provided a basic overview of the drug discovery process, we will next discuss the role of ML within this process.

2.2 Applications of ML in Drug Discovery

The usage of data-driven methods in drug discovery is not a new idea. These methods are commonly referred to as Quantitative Structure-Activity Relationship (QSAR) models and aim to - as the name suggests - establish a quantitative relationship between a molecule's structure and its activity. ML provides one way to implement a QSAR model (24).

Over the years a diverse toolkit of such machine learning methods (6) has been developed to speed up and automate different steps of the drug discovery process. In most of these methods, the aim is to accurately reproduce and automate the experiments that human chemists conduct throughout the drug discovery process, but some other machine learning

applications go beyond our theoretical understanding and open up entirely new research directions (25, 26). As machine learning can more efficiently leverage more data from more complex environments than humans can, its application to drug discovery tackles multiple bottlenecks simultaneously. With an increase in compute and data, machine learning holds the promise to push the boundaries of what is ought to be possible.

Providing an exhaustive summary of all the different ML methods in drug discovery is beyond the scope of this thesis. In this work we focus on *molecular scoring*. In molecular scoring, a model predicts a molecule's activity (e.g. the binding activity, solubility or the toxicity) and thus serves as an in-silico proxy to an in-vitro or in-vivo experiment. These models serve three main purposes in drug discovery:

1. **Virtual screening**: In virtual screening (4, 27, 28, 29), a molecular scoring model is used as a scalable method to automatically and efficiently *screen* (i.e. rank) the molecules in large, chemical libraries. These libraries consist of readily purchasable molecules that are offered by chemical marketplaces such as Molport. Since all molecules are thus readily available, it is relatively easy to experimentally validate the model's predictions. As such, virtual screening offers a cheaper and faster alternative to HTS and to the "test" step in the DMTA cycle.

2. **De-novo generation (optimization)**: In *de-novo generation* (30, 31, 32), the goal is to design generative models from which candidate drugs can be sampled. As generative methods do not select candidates from a predefined, chemical library, they can cover a larger portion of the chemical space they learn to represent than in virtual screening. De-novo generation is often split in *generation* (i.e. generating a *valid* molecule) and *optimization* (i.e. generating a *good* molecule). Since a good molecule optimizes multiple constraints simultaneously (e.g. binding, ADMET, etc.), this constitutes the biggest challenge. In de-novo generation, molecular scoring models function as cheap proxies for the actual experiments that we are trying to optimize (33). Generative models are applied in the "design" step of the DMTA cycle and are thus applied iteratively to propose a batch of molecules. After experimentally verifying a batch, the model is retrained.

3. **QSAR modeling**: By carefully assessing how a ML model comes to its prediction, it can serve as a QSAR model. In QSAR modeling, it is essential that a relation is established between the structure and activity of a molecule that can then be

intuitively communicated to a biologist or chemist. This is outside the scope of this thesis and will not be discussed any further.

It is worth noting that in molecular scoring, a model's predictions inform which molecules will be experimentally validated next. Since this experimental validation is expensive, it is important that we can efficiently balance between *exploration* and *exploitation* throughout the different DMTA cycles and this requires not just good generalization in terms of performance, but also well-calibrated uncertainty estimates.

Now that we have discussed the applications of ML in drug discovery, we will next discuss some of the data challenges that are specific to this domain.

2.3 Molecular Data

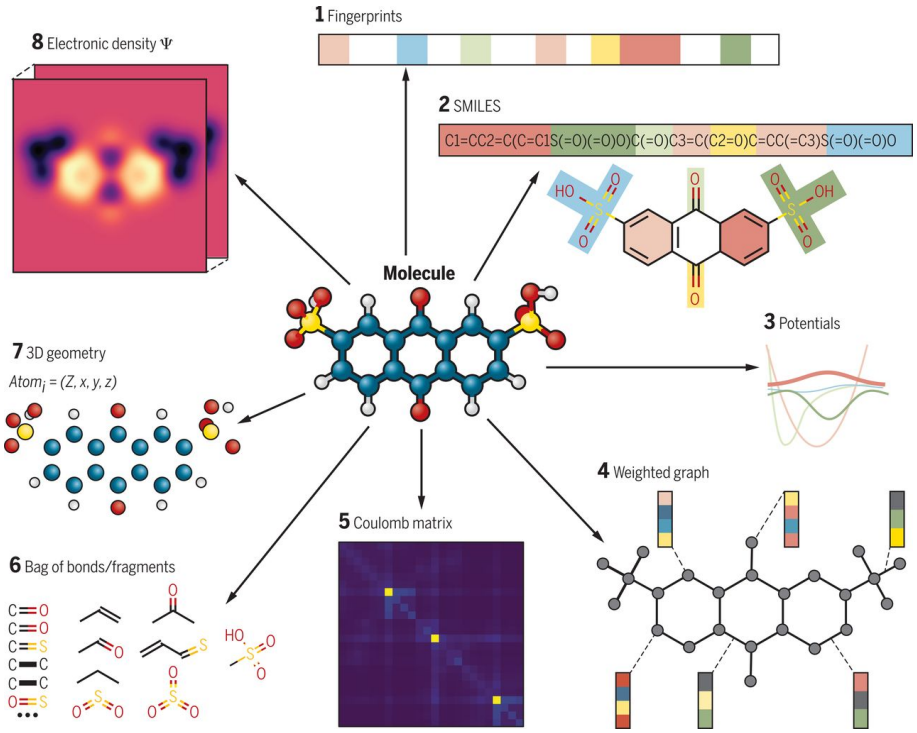

Figure 2.3: Molecular representations - Different representations for molecular data in machine learning. Taken from Sanchez-Lengeling et al. (32).

A machine learning model is only as good as the data it is trained on. In molecular scoring, data is gathered by running biological assays (or *bioassays*). Bioassays are experiments that are conducted in a laboratory. They are expensive and subject to many sources

of variation that are hard to control for. Due to the high costs, the amount of data for any particular task is small. For example, most datasets in Therapeutics Data Commons (TDC) (34), a collection of publicly available machine learning datasets for drug discovery, contain in the order of 10^2 to 10^5 molecules. The majority of these molecules tests negative against the biological activity of interest, resulting in an imbalanced dataset. For example, when predicting a molecule's binding activity against a target, most molecules won't show any interesting activity due to the strong structural constraints required for binding. Yet in using the model, it is exactly the molecules that do bind to the target that are of interest as these form the set of drug candidates. Compared to other machine learning domains such as computer vision and natural language processing, the lower quantity and quality of data in drug discovery poses an additional challenge.

There's several techniques to address this challenge. Most relevant to this thesis, is the usage of different ways to represent molecules and the usage of different pre-training techniques to make model training more efficient.

2.3.1 Molecular representations

A molecule can be represented in a variety of ways (see Figure 2.3) that differ in the information they carry, their suitability for generalization, and their compatibility with machine learning techniques. *Molecular fingerprints*, such as the Extended-Connectivity Fingerprint (ECFP) (35) and the Molecular ACCess System (MACCS) (36) fingerprint, compute a structural descriptor of a molecule. These structural descriptors are informed by domain knowledge. Although more advanced representations have been developed since they were first introduced, fingerprints remain a popular option in drug discovery due to their ease of use and competitive performance in terms of both efficacy and efficiency (37). With the rise of deep learning, neural networks are increasingly used to directly learn such representations from the data (38). A natural way of describing a molecule in this setting is through a graph in which the nodes represent atoms and the edges represent atomic bonds. Using this representation, *graph neural networks* (39) (GNNs) have been successfully applied in drug discovery. Another popular option to represent a molecular structure is the simplified molecular-input line-entry system (or *SMILES*) string (40). SMILES defines a grammar to encode a molecule's structure in text. With this textual representation, we can leverage the more mature line of research in natural language processing to model the data. For example, modeling chemical reactions can now be framed as a translation problem between two SMILES strings (41). Picking the right molecular representation is

an important factor in optimizing the performance and generalization of machine learning in drug discovery.

2.3.2 Pre-training techniques

Another technique to improve performance in a low data setting is to pre-train a model on different but related tasks and then fine-tune. To ensure that the pre-training is actually helpful to the downstream tasks, we can leverage domain knowledge. Specifically, we know that molecules are governed by quantum mechanics (42) and that their structure is important to their function (43, 44). Therefore, during pre-training, we can for example use a molecule's quantum properties (45, 46) or its most-likely three-dimensional configurations (called *conformers*) (47). Since this data is not task-specific, we can leverage more data sources. Additionally, there also exist accurate simulators that, given enough compute, can be used to construct a sufficiently large dataset. Alternatively, we can leverage pre-training techniques from other domains. Specifically, the usage of a text-based molecular representation allows the naive application of pre-training techniques from NLP (11, 48). Despite the structural differences between molecular and liguistic data, these methods have achieved some success. The goal of pre-training is to learn a representation that is expected to efficiently transfer to the downstream tasks relevant in drug discovery, making the fine-tuning more data efficient.

3

Related Work

While this work presents a first complete specification of the OOD problem as it is encountered in live drug discovery programs, the importance of generalization in molecular scoring has long been established. We divide the related work into three categories:

- **Model analysis**: Work in this category uses the data a model was trained on or the model itself to identify a part of the chemical space on which the model can be expected to perform well. In molecular scoring, the concept of the *applicability domain* falls into this category and will be discussed in Section 3.1.
- **Model selection**: Work in this category uses careful evaluation to select one model out of many that is expected to generalize the best. In molecular scoring, the usage of the different *data splits* falls into this category and will be discussed in Section 3.2.
- **Model design**: Work in this category relaxes the i.i.d. assumption and aims to design algorithms and representations that generalize better by reducing the reliance on spurious correlations. In molecular scoring, the exploration of *domain generalization* and *domain adaptation* algorithms falls into this category and will be discussed in Section 3.3.

Finally, as it is important that we can efficiently balance between exploration and exploitation and well-calibrated uncertainty estimates are thus essential, related work on *uncertainty estimation* will be discussed in Section 3.4.

3.1 Model Analysis: Applicability Domain

The concept of the Applicability Domain (AD) was first introduced in the early 2000s as one of multiple guidelines to follow when replacing in-vitro and in-vivo experiments by

in-silico alternatives (49, 50), specifically QSAR models. While the AD concept stems from cheminformatics, it bears many similarities to the idea of OOD detection in ML (51). Intuitively, the AD identifies a part of the chemical space on which the model is expected to perform well. Practically, the AD of a model is defined with respect to the data it was trained on.

Central to various AD definitions is a notion of similarity between the training data and any other molecule (52). Similarity can be defined with respect to various molecular characteristics, such as the molecular structure, its physico-chemical properties or a task-specific activity. Approaches differ in which characteristics they use and the criteria they employ to compare them (53, 54, 55). Criteria can for example be distance-based, range-based or probability-based. At test time, the established criteria can be used to determine whether a molecule falls within the model's applicability domain and thus whether the model's prediction can be trusted.

Prior work reduces a measure of similarity down to a binary decision, i.e. whether a molecule falls in or outside of the AD (or, similarly, whether a molecule is in- or out-of-distribution). Different from prior work, we argue that this distinction is somewhat arbitrary and rather aim to study how a model's performance evolves as the similarity to the training data changes.

3.2 Model Selection: Data Split

In ML, it is common practice to select one out of many models for further usage. A typical example of this is hyper-parameter tuning. To do so effectively, this requires a *selection criterion* that is informed by how we expect to use the model in practice. In drug discovery, the popular usage of different data-splits other than the default, random split can be thought of as a form of OOD generalization through model selection (56). The underlying assumption is that these splitting methods better replicate the distribution shifts encountered in live drug discovery programs. Scoring and selecting models based on these splits is expected to not only give a more accurate indication of prospective performance, but is also expected to lead to models that generalize better.

Several data splits have been proposed, some of which we will discuss next. Different from prior work, we argue that there is no single best split, but rather describe a protocol to

choose a split given a dataset *and* molecular representation. We do so by characterizing the distribution shifts encountered in live drug discovery programs and by evaluating different splits based on their ability to replicate this shift.

3.2.1 Temporal split

To increase representativeness, an obvious option would be to use a temporal split (57, 58). In this split, each data point is associated with a timestamp that encodes the moment at which it was collected during a real drug discovery program. The test set is then comprised of the molecules that were collected the latest. Due to directly leveraging the temporal information in a representative drug discovery program, this provides a good estimate of the distribution shifts encountered in practice and a good indication of the model's prospective performance (59). In practice, however, this temporal data is generally not (publicly) available and approximate splitting methods are needed.

3.2.2 Scaffold-based split

A *scaffold* is a concept that intuitively refers to the structural core of a molecule. There is no universal definition that formalizes this intuition (see Figure 3.1), but in ML research the Bemis-Murcko (BM) scaffold (60) is the prevalent option. In ML for drug discovery, the *scaffold-based split* has replaced the random split as the standard splitting method in many popular benchmarks (34, 61, 62). With this split, the data is stratified by scaffold after which the different strata are randomly split in a train and test set. This provides a better approximation of the temporal split and is thus more representative of prospective performance (58).

3.2.3 Extrapolation-oriented split

In parallel to ML practitioners, computational chemists have also proposed various splitting methods for the evaluation of QSAR models (52). One such method was the *extrapolation-oriented split* (64). This work proposes an algorithm to select molecules that are on the perimeter of the dataset. It does so by repeatedly finding the two molecules that are furthest away from one another (according to the Euclidean distance of their representation) and adding these to the test set. When compared to other splits, using the extrapolation-oriented split for model selection is found to lead to models that extrapolate (or *generalize*) better (64).

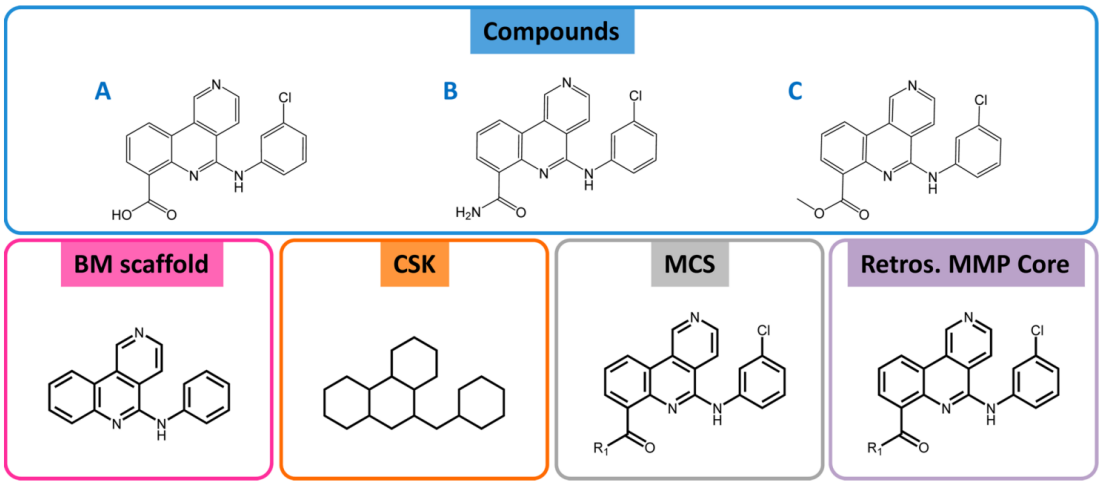

Figure 3.1: Molecular scaffolds - Visualization of different definitions of molecular scaffolds, taken from Hu et al. (63).

3.3 Model Design: Domain Generalization and Adaptation

The i.i.d. assumption is central to machine learning, but unrealistic for many practical applications. In the i.i.d. setting, models are prone to rely on statistically significant but spurious correlations (or *shortcuts*) between the input features and the labels (13). This makes them susceptible to biases in the dataset and hinders OOD generalization, especially in the low-data setting, resulting in disappointing performance in practice (61). Several learning paradigms exist that relax the i.i.d. assumption when testing a model's generalization performance (e.g. meta-learning, transfer learning, and domain adaptation). These paradigms differ to the extent in which they put constraints on the similarity and availability of the train and test data (14, 65).

Although many of these paradigms can be applied to molecular scoring, we focus on *domain generalization* (66, 67) (DG) and *domain adaptation* (15) (DA). In DG and DA, a model is trained on several related, but distinct data distributions called *domains*. In practice we rely on human bias to define the domains (e.g. the scaffold of a molecule) because it's difficult to do so formally. After training, the generalization performance of the model is evaluated, without any further fine-tuning, on an set of differently distributed test domains. The main difference between DA and DG, is that in DG the test domains are unknown during training. Specifically, we focus on unsupervised DA, in which the test domains are known, but can only be used as an unsupervised signal during training.

In both the DG and DA paradigm, a model that relies on domain-specific biases will be penalized.

Generally, the goal in DG and DA is to learn domain-invariant representations of the input which are expected to generalize better. This can be related to causal inference (68) by describing the data labeling process as a structural causal model (SCM) (69). Pearl's seminal work on causality (70) tells us that solely observational data cannot convey causal relations, but in domain generalization each domain can be considered a constrained intervention of the original SCM. As the causal relations are invariant across domains, we can consequently hope to learn them from the interventional distribution by learning a domain invariant representation. In molecular scoring, this would hopefully result in learning the structural or quantum-mechanical properties and dynamics that *cause* a molecule's activity.

There exists prior work that benchmarks DA and DG methods for molecular scoring (34, 61, 62). These benchmarks maintain various definitions of a molecule's domain (e.g. the number of atoms, the scaffold or the year a molecule was patented) and benchmark how effective different methods are for generalizing across domains. In line with conclusions in different disciplines (56), there is a significant drop in performance with a domain-based split compared to a random split. Also in line with earlier conclusions, most DA and DG methods tend to do on par or worse than the default empirical risk minimization (ERM) (71). Different from prior work, we benchmark DA and DG methods on a test set that replicates the covariate shifts encountered in ongoing drug discovery programs.

3.4 Uncertainty Estimation

Uncertainty estimation aims to quantify a model's confidence by replacing point-wise predictions by distributional ones. Well-calibrated uncertainty is essential in interactive settings where a model's predictions inform the next actions, such as in active learning and Bayesian optimization. In these types of settings, it is important to know what a model does not know to effectively balance between *exploration* and *exploitation*. In drug discovery, for example, the first few rounds are generally used to improve the model's performance on new chemistry where the uncertainty is high (i.e. explore) before starting to optimize the activity of interest (i.e. exploit) (72). The worst kind of error in such settings is a confidently wrong model.

3. RELATED WORK

A common distinction in uncertainty estimation is between *aleatoric* and *epistemic* uncertainty (73). Aleatoric uncertainty is caused by inherently random aspects of the data generation process (e.g. the noise in bioassay readouts). This uncertainty can not be reduced by improving the model. Epistemic uncertainty on the other hand is caused by a lack of knowledge by the model. With few exceptions (74), most methods use the variance of the predictions by multiple, similar models as a proxy to directly estimate the epistemic uncertainty.

A naive approach to obtain multiple predictions of similar models is to actually train multiple models from scratch (75). While this is an effective approach in practice, the required compute might not be available when models or datasets become sufficiently large, specifically in deep learning. To circumvent this issue, various methods have been proposed that reduce the computational needs (e.g. methods based on dropout (76) or by using an ensemble of model checkpoints (77)). Outside of deep learning, there is the Random Forest (78) and Gaussian Process (79), which take a more principled approach to uncertainty estimation.

Due to the importance of well-calibrated uncertainty estimates in molecular scoring, we argue that any work investigating the OOD performance of different ML methods should include the uncertainty estimates in its evaluation. Specifically because prior work suggests that the accuracy of a model's uncertainty estimates decrease under distribution shifts (80, 81), we evaluate a model's OOD generalization along two axes: the accuracy of its predictions (or *performance*) and the accuracy of its uncertainty estimates (or *calibration*).

4

Molecular Out-Of-Distribution (MOOD)

This work proposes the Molecular Out-Of-Distribution (MOOD) framework. MOOD consists of two parts:

1. A complete **problem specification** of OOD generalization for molecular scoring. The goal of this specification is to align different stakeholders on the details of the problem that we are trying to solve and to scope what a solution would look like. Specifically, we hope that this can close the gap between advances in academia and industry pain points. Section 4.1 will describe what distribution shifts we encounter in ongoing drug discovery programs, how these shifts affect generalization and how we can best best replicate such shifts for model evaluation.

2. An **investigation** of the effect and importance of different tools to improve OOD generalization in molecular scoring. The goal of this investigation is to get an overview of how various techniques affect generalization through the lens of the newly proposed problem specification. Specifically, we hope that this can inform future study directions and that this can more efficiently direct resources in ongoing drug discovery programs. Section 4.2 will detail what tools we consider in this study and how they compare to one another.

Finally, to promote reproducibility, Section 4.3 will detail the **experimental setup**.

4.1 The MOOD specification

The goal of the MOOD specification is to make explicit what problem we are trying to solve when we are talking about OOD generalization in molecular scoring. Specifically, the goal is to come to a set of evaluation standards that closely align with the situations encountered in ongoing drug discovery programs and that can be followed by the community to guide future advances in molecular scoring. The first evaluation standard has already been established in Section 3.4 and dictates that molecular scoring techniques should be evaluated based on both their performance and their calibration. This section introduces the second (and final) standard: A novel protocol to obtain a test set that is similarly difficulty to the OOD molecules encountered in downstream applications of molecular scoring. Before we get there though, we first need to define what it means to be OOD in molecular scoring.

4.1.1 Where to generalize to?

Drug discovery is a vast domain and even within just molecular scoring, there exist many forms of OOD generalization that are interesting to pursue. One could for example aim to generalize from small molecules to macro-molecules or across different, but related bioassays. In this work, the focus is on generalizing across the molecular representation space as this is a common use case for molecular scoring in ongoing drug discovery programs.

We argue that in this setting the type of distribution shift we encounter is the *covariate shift*. Let $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$ with $(\mathbf{x}_i, \mathbf{y}_i) \sim \Pr_S(X) \times \Pr_S(Y)$ be a training set of molecule-activity pairs. We denote $\Pr_S(X)$ as the train distribution over the input space of molecules and $\Pr_S(Y)$ as the train distribution over the output space of activity values. Assuming that we have a learned predictor of the activity given the molecules: $f : X \to Y$, we want the predictor to generalize to $T = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{m}$ with $(\mathbf{x}_i, \mathbf{y}_i) \sim \Pr_T(X) \times \Pr_T(Y)$ and $\Pr_T(X) \neq \Pr_S(X)$. We further assume that $\Pr_S(Y) = \Pr_T(Y)$ and $\Pr_S(Y|X) = \Pr_T(Y|X)$. To verify whether this is a realistic scenario in practice, we consider the two popular applications of molecular scoring and consider how these affect $\Pr(X)$, $\Pr(Y)$ and $\Pr(Y|X)$:

- In both virtual screening and optimization, we assume there to be a shift in the input space distribution $\Pr(X)$. Due to the sheer size of the molecular space and the human bias in exploring it so far, we can expect to regularly encounter molecules

that do not resemble the molecules in the train set. This will be discussed in more detail in Section 4.1.4.

- In both virtual screening and optimization, we assume that the underlying conditional distribution $\Pr(Y|X)$ (i.e. the data generation process) remains the same since we train the model to serve as a proxy of that data generation process.
- That there is no shift in target space distribution $\Pr(Y)$ is less obvious.
  - In virtual screening, it is a common assumption that there is no such shift. Since we do not select which molecules to apply our model to but rather rely on quantity to discover new hits, we can think of this as randomly sampling the molecular space and can assume the target distribution to remain the same.
  - In de-novo generation, one might expect a shift since the model is used to bias a generative process to produce molecules that have an optimized activity. However, since the optimization process is iterative and we retrain a model after experimentally testing a batch of generated molecules, this shift is minimized. Additionally, the shift is in the prediction space of the model and not in the actual target space. To account for such a shift in the prediction space, we can use task-dependent metrics that focus on the performance of the model in the direction (i.e. minimize or maximize) or class that matters most during optimization.

Now that we have established the kind of generalization that is of interest in this study, we will next define what it means for a molecule to be OOD in this setting.

4.1.2 A continuous, distance-based OOD definition

Whether a data point is in- or out-of-distribution is often framed as a binary decision (e.g. OOD detection in ML and the AD of QSAR models). We argue that such a sharp distinction is arbitrary and propose to use a continuous, distance-based definition instead. Intuitively, the further a query molecule is from the train set, the higher the expected error of the model (i.e. the "more OOD" that molecule is). If needed, one can still easily go from the continuous to the binary setting by thresholding the distance. By assuming a continuous definition, however, it becomes easier to characterize the types of covariate shifts encountered in ongoing drug discovery programs (see Section 4.1.4).

This raises the question what distance metric to use. While this is an interesting research question on its own, we adopt the k-NN distance metric between a query molecule and the train set as proposed in prior work (81). While it is likely better to pick a dataset-dependent k, we fix $k = 5$ since early experiments showed that the derived conclusions were stable under multiple values of k. The pairwise distance between two molecules is computed between their representations and the associated distance metric is representation dependent: For binary representations we use the Tanimoto distance and for continuous representations we use the Euclidean distance. An important outcome of this is that **the distance between two molecules depends on the representation used during training**. It is worth noting that this departs from prior work, especially on the AD, in which an OOD metric is defined independently of the molecular representation. While using a representation-independent distance metric makes sense from a human perspective, an ML model does not have the same context and only knows about the information contained in the molecular representation. From the model's perspective, the distance between two molecules should therefore be solely defined in terms of their representation.

To validate the suitability of the assumed, representation-dependent k-NN distance as an OOD metric, we will next examine how performance and calibration are affected as the distance to the train set grows.

4.1.3 Validation of the assumed OOD definition

A good OOD metric captures our intuition that a baseline model (i.e. a model that relies on the i.i.d. assumption) is more likely to make errors the further a query molecule is from the train set. As a way to validate the assumed, distance-based OOD metric, we thus run a hyper-parameter search for three baseline algorithms on various datasets (see Figure 4.1) and using various representations (see Table 4.1) and evaluate how performance is affected as the distance to the train set grows. As baselines we use the Random Forest (RF) (78), Gaussian Process (GP) (79) and an ensemble (75) of Multi-Layer Perceptrons (MLP) (82). Besides the performance, we also evaluate how calibration is affected as the distance to the train set grows. To support the usage of set-based performance and calibration metrics (see Table 4.2), we bin the model's predictions based on their distance to the train set and compute the performance and calibration per bin. For the full experimental details, see Section 4.3.1. For the resulting figures, see Figure 4.2, Figure 4.3 and Figure 4.4.

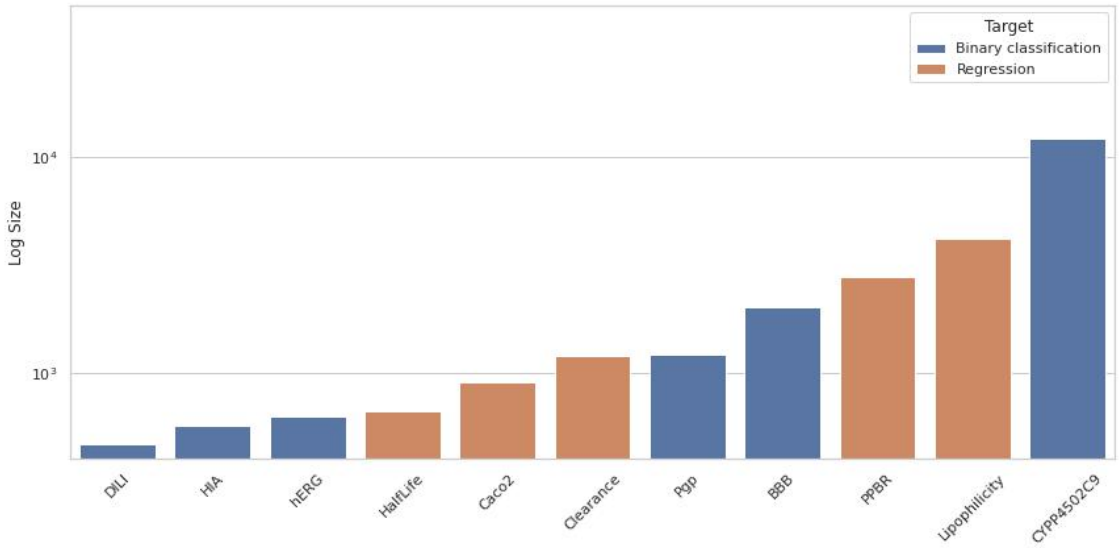

Figure 4.1: Dataset overview - An overview of the 11 ADMET datasets used in this study. All datasets were selected from TDC to be diverse in terms of their size, task and characteristics. There is 4 absorption, 2 distribution, 1 metabolism, 2 excretion and 2 toxicity datasets. While not having target-specific datasets, the chosen datasets do include 2 datasets that measure protein inhibition. For a detailed description of each dataset, we refer to the TDC documentation (34).

Name		Learned	Description
MACCS	(36)	No	Binary vector describing the absence or presence of expert-informed structural patterns.
ECFP6	(35)	No	Binary vector describing the absence or presence of structural patterns with radius 3.
Desc2D	(83)	No	Continuous vector of 2D physico-chemical properties from RDKit, such as the molecular weight, the number of valence electrons and the LogP value.
WHIM	(84)	No	Continuous vector describing the 3D shape.
Graphormer	(45)	Yes	Embedding from a Transformer pre-trained to predict the HOMO-LUMO gap from the 2D graph.
ChemBERTa	(11)	Yes	Embedding from a Transformer pre-trained with the BERT objective (85) on SMILES strings

Table 4.1: Molecular representation overview - An overview of the different molecular representations used in this study. All representations were chosen based on their popularity in the scientific literature and their public availability. There is 4 engineered representations and 2 representations obtained from pre-trained Transformer NNs (86).

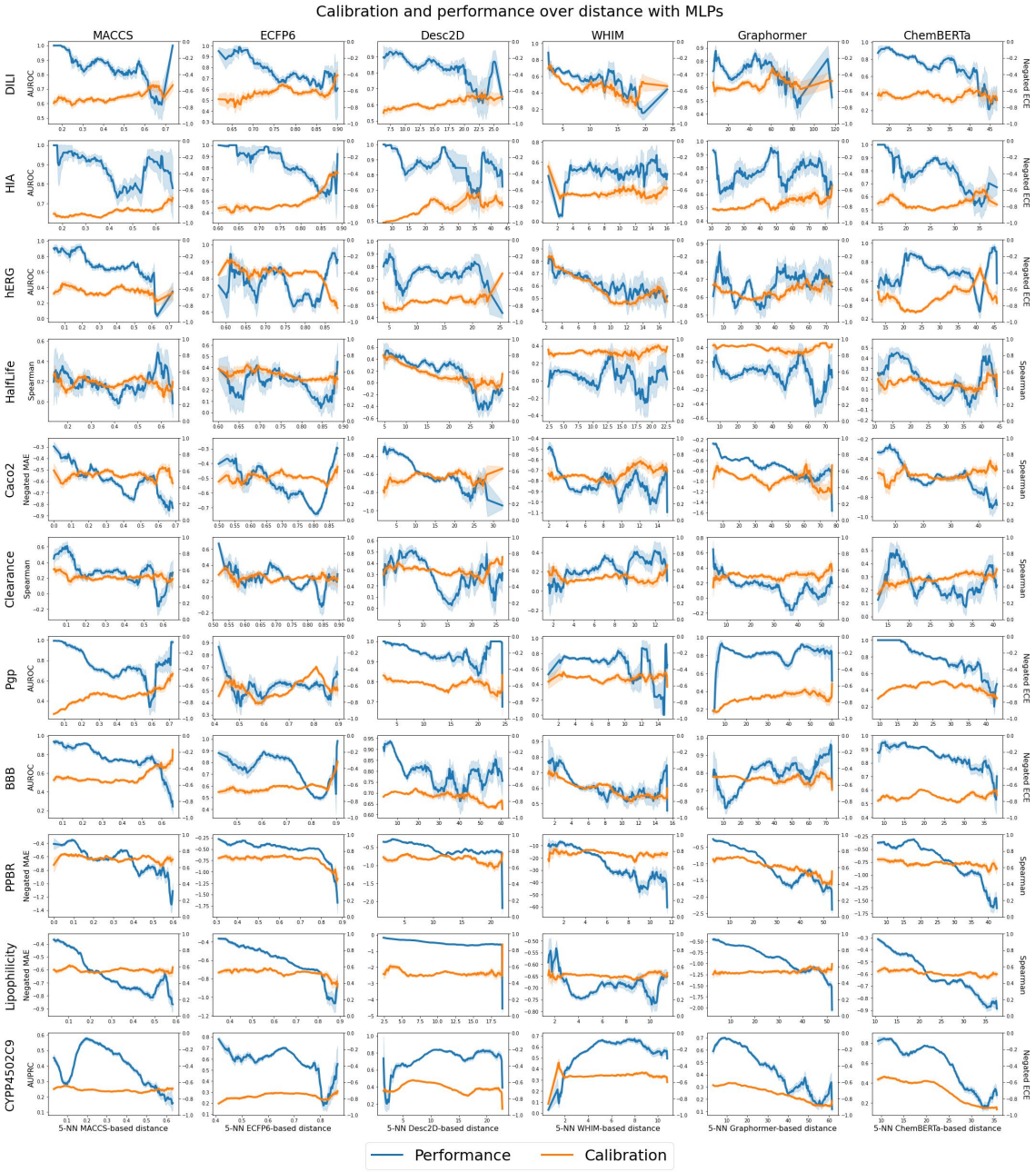

Figure 4.2: Performance and calibration over distance for MLP ensembles - In this grid, each column represents a molecular representation and each row represents a dataset. The datasets are ordered by their size from small (at the top) to large (at the bottom). All y-axes are visualized in such a way that higher is better. For the MLP ensemble, we find $\rho = -0.516$ for the performance and $\rho = -0.011$ for the calibration. This figure is best viewed digitally.

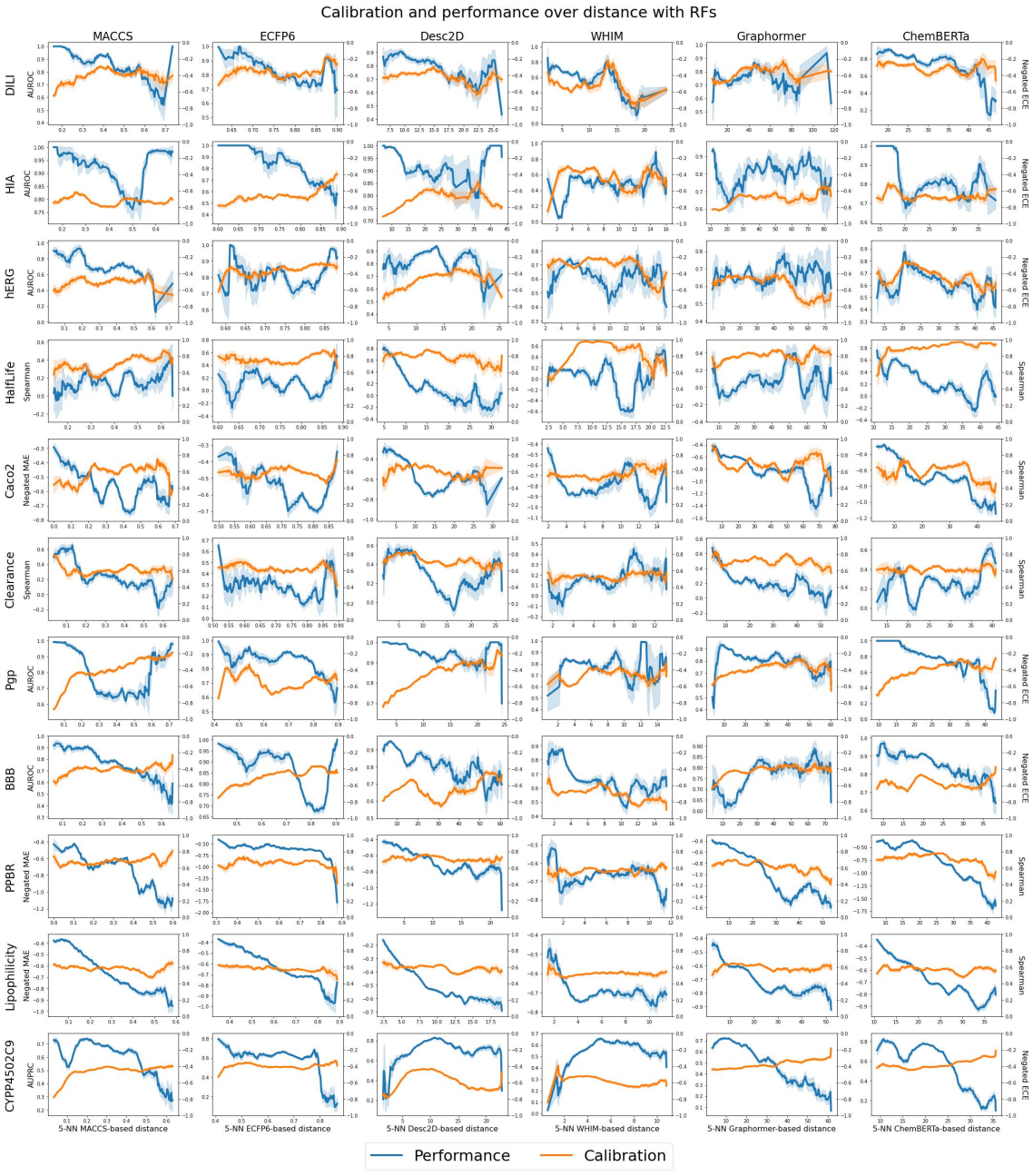

Figure 4.3: Performance and calibration over distance for RFs - In this grid, each column represents a molecular representation and each row represents a dataset. The datasets are ordered by their size from small (at the top) to large (at the bottom). All y-axes are visualized in such a way that higher is better. For the RF, we find $\rho = -0.467$ for the performance and $\rho = 0.069$ for the calibration. This figure is best viewed digitally.

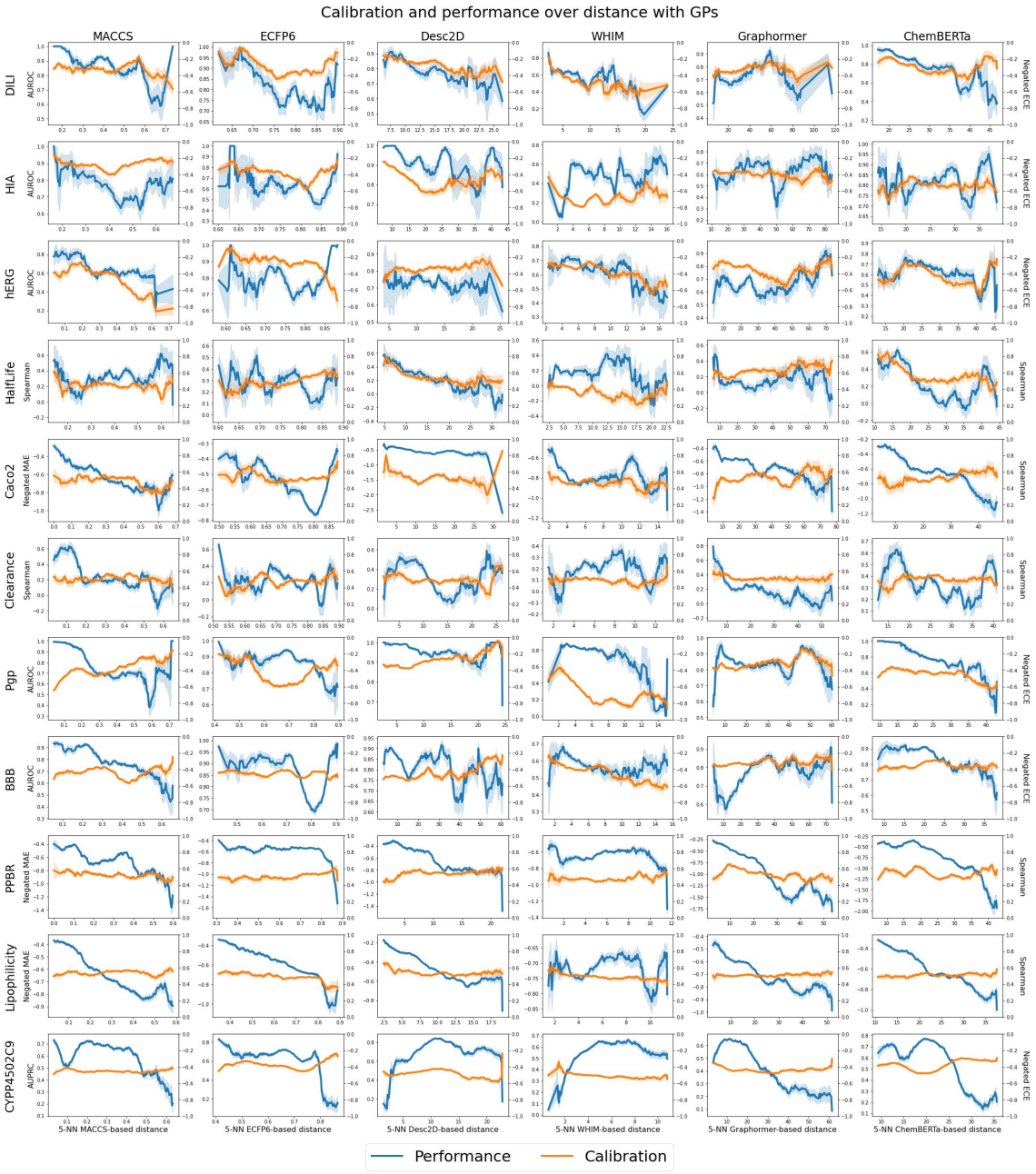

Figure 4.4: Performance and calibration over distance for GPs - In this grid, each column represents a molecular representation and each row represents a dataset. The datasets are ordered by their size from small (at the top) to large (at the bottom). All y-axes are visualized in such a way that higher is better. For the GP, we find $\rho = -0.475$ for the performance and $\rho = -0.096$ for the calibration. This figure is best viewed digitally.

Dataset	Performance metric	Calibration metric
DILI	AUROC	ECE
HIA	AUROC	ECE
hERG	AUROC	ECE
HalfLife	Spearman	Spearman
Caco2	MAE	Spearman
Clearance	Spearman	Spearman
Pgp	AUROC	ECE
BBB	AUROC	ECE
PPBR	MAE	Spearman
Lipophilicity	MAE	Spearmna
CYPP4502C9	AUPRC	ECE

Table 4.2: Overview of calibration and performance metrics per dataset - Metrics used to measure performance and calibration. For the performance metrics, we follow the metrics prescribed by the associated TDC benchmark (34). For the calibration metrics, we use the Expected Calibration Error (ECE) (80, 87) for all classification datasets and we use the Spearman correlation between the uncertainty and MAE for all regression datasets.

Visually inspecting these figures, we observe a tendency for performance to drop over distance, whereas calibration seemingly remains more stable. Quantifying this trend using the mean Spearman correlation coefficient across (model, dataset, representation) triplets, we find $\rho = -0.486$ between the distance and performance and $\rho = -0.013$ between the distance and calibration[1]. This provides quantitative evidence that performance indeed drops over distance, while calibration is mostly unaffected, which matches our expectations of a good OOD metric.

By computing the correlation measures separately for each of the datasets (see Table 4.3) and each of the representations (see Table 4.4), we observe that this trend is consistent across the board. For the performance, it now also becomes clear that the strength of the trend seems to largely depend on the performance metric. One interesting difference between the metrics is that MAE, with the strongest correlation, is computed per-sample while AUROC, AUPRC and Spearman are computed for a set of samples. This suggests that at least some of the differences could be explained as a side-effect of the binning

[1]The Spearman correlation is computed using SciPy (88). The documentation of the p-value notes "The p-values are not entirely reliable but are probably reasonable for datasets larger than 500 or so." Since we use a smaller number of bins and thus do not surpass that threshold, the p-values are not reported.

Dataset	Performance ρ	Calibration ρ
DILI	-0.722	-0.029
HIA	-0.216	0.241
hERG	-0.251	-0.103
HalfLife	-0.264	-0.056
Caco2	-0.671	0.043
Clearance	-0.294	-0.046
Pgp	-0.473	0.309
BBB	-0.482	0.081
PPBR	-0.766	-0.254
Lipophilicity	-0.810	-0.236
CYPP4502C9	-0.400	-0.090

Table 4.3: Performance and calibration over distance per dataset - The Spearman correlation between the distance and binned performance or calibration, aggregated by dataset. The datasets are sorted by size from smallest (top) to largest (bottom).

Representation	Performance ρ	Calibration ρ
MACCS	-0.658	0.125
ECFP6	-0.505	-0.003
Desc2D	-0.530	-0.082
WHIM	-0.206	-0.233
Graphormer	-0.323	0.116
ChemBERTa	-0.696	0.001

Table 4.4: Performance and calibration over distance per representation - The Spearman correlation between the distance and binned performance or calibration, aggregated by representation.

Representation	Correlation ↓	Slope ↑	Intercept ↑
ChemBERTa	-0.970	-0.534	-0.155
Desc2D	-0.915	-0.477	-0.186
ECFP6	-0.955	-0.579	-0.058
Graphormer	-0.906	-0.553	-0.462
MACCS	-0.934	-0.512	-0.364
WHIM	-0.139	-0.062	-0.658

Table 4.5: Slope and intercept on Lipophilicity - The resulting slopes and intercepts of fitting a linear function to the performance and distance of different representations on the Lipophilicity dataset. The slopes are adjusted for the differences in range of the respective distance functions. To be consistent with earlier figures, we use the negated MAE to measure performance in Lipophilicity.

procedure we employ to compute set-based metrics (e.g. because the number of samples differs from bin to bin). For the the calibration, we have unfortunately not been able to find an explanation for the differences in correlation between different datasets or different representations.

Finally, it is worth noting that by using the Spearman correlation, we only look at the correlation between the rank of the distance and the rank of the performance or calibration. This is done on purpose, as we just want to validate our intuition that performance and calibration drop over distance, but do not want to assume a specific function. This does, however, hide some useful information. It could for example be that two representations with a similar correlation differ in the speed by which the performance decreases. For example, if we assume a linear function and just look at the Lipophilicity dataset, we can compute the intercept and slope for different representations (see Table 4.5). By doing this, we can for example see that while ECFP6 and MACCS have a similar correlation and slope, the intercept of ECFP6 is a lot higher, which tells us that ECFP6 performs better for this dataset. Similarly, while WHIM does poorly at low distances (i.e. the intercept is low), the drop in performance is minimal when the distance grows (i.e. the slope is close to 0). At higher distances, WHIM could thus be the better option. This shows that a *validated* OOD metric can serve as an insightful tool for model selection.

While for the assumed k-NN distance the general tendency is as expected, an interesting research question could be to find a distance metric that better correlates with the difficulty

of generalizing. Additionally, there are some interesting patterns that could be worth investigating further. For example, in many of these plots, performance goes up again after having dropped. Another interesting observation is that at high distances models can do worse than random (e.g. the AUROC score drops below 0.5 or the Spearman correlation drops below 0), suggesting that the model suffers from having learned spurious correlations. For MOOD, however, we consider there to be enough evidence that the chosen distance metric is a reasonable choice and leave the exploration of other distance metrics and related research questions to future work.

Now that we have found a distance metric that can serve as a continuous OOD metric, we can use it to characterize the covariate shift from any training dataset to any set of (unlabeled) molecules in any representation space. This will lead us to present our first, major result in the next section: A protocol for replicating the covariate shifts in molecular scoring as encountered in ongoing drug discovery programs.

4.1.4 A protocol for replicating realistic shifts

In line with popular ML practice, MOOD distinguishes between two different data splits that both serve a different purpose. The *train-test split* splits the initial dataset in a train set and test set (or *holdout* set). The test set is not at all used during model training and its purpose is to finally provide an accurate indication of prospective performance in downstream applications. The resulting training set is once more split using the *train-val split* (this will be discussed further in Section 4.2.1). This results in the final training set, which is used to optimize the model's parameters, and a validation set, which is used for model selection and early stopping.

To get an accurate estimate of the downstream performance of a model, we thus want the hold-out test set to be similarly difficult to the molecules encountered in downstream tasks. Using our continuous, distance-based OOD definition, we can now characterize, compare and thus replicate covariate shifts. It is important to note that while we will be discussing a particular example, **the protocol we describe is more generally applicable to replicate the expected covariate shift for molecular scoring in any drug discovery program**.

Step 1: Compile a set of molecules representative of the downstream task

The first step is to compile a list of (unsupervised) molecules that you want to use your model on. Consider both popular use cases of molecular scoring models:

- **Virtual screening:** In virtual screening, a molecular scoring model is used to screen a chemical library of readily available compounds. Since such a library is known beforehand, we can simply collect all molecules we will be applying our model to. In this study, we randomly sample 50.000 molecules from Molport.

- **De-novo generation:** In de-novo generation, we apply a molecular scoring model to the molecules sampled from a generative process. This generative process can be subject to multiple constraints and changes during training due to the optimization procedure, making it difficult to know beforehand what molecules it will generate exactly. However, by training the generative model with a subset of the constraints that do not involve the molecular scoring model or with general drug-like constraints, we can get a reasonable estimate. In this study, we sample 50.000 molecules from REINVENT (33), trained to match the ChEMBL distribution (89).

Step 2: Compute the distance from each of the representatives to the train set

For each of the molecules collected in Step 1 and each of the molecular representations we are interested in trying, we compute the distance to the train dataset.

Step 3: Characterize various data splits

For each of the train-test splits that we are considering, use it to split the dataset in a train and test set. For each representation, compute the distance from the test samples to the train set. If the split is stochastic, this can be repeated for multiple seeds to get a more reliable estimate.

Step 4: Rank the different splits based on their representativeness

Using the two distance distributions computed in Step 2 and Step 3, we can rank each of the considered splitting methods by computing the Wasserstein distance between these two distributions. The splitting method that has the lowest mean Wasserstein distance to the distributions of downstream representatives should be used as the train-test split.

To demonstrate the above protocol, we use it to evaluate the representativeness of four different splitting methods. Besides the widely adopted random split and Bemis-Murcko scaffold split (58), we also investigate the usage of the extrapolation oriented split (64) (which we hereafter refer to as the "perimeter" split) and propose the maximum dissimilarity split. For both the perimeter split and the maximum dissimilarity split, we group the data points according to the k-means clustering of their representations ($k = 25$), where the representative of each group is the cluster center. To ensure we can use k-means with each representation, we first compute a continuous representation using an Empirical Kernel Map with 512 randomly sampled points. For the perimeter split, we add the two groups that are furthest away from one another to the test set (according to the Euclidean distance of their representatives) until the desired test size is reached. For the maximum dissimilarity split, we find the two groups that are furthest away from one another and add one to the train set and the other to the test set. We then repeatedly add the group that is closest to the original test group to the test set until the desired test size is reached. Compared to the random and scaffold split, the maximum dissimilarity and perimeter split increase the distance between train and test.

We visualize the ranking results of the protocol in Figure 4.5. Additionally, we summarize the prescribed splits in Figure 4.6. Interestingly, the proportion of prescribed splits differs significantly from representation to representation and what is prescribed often does not match common practice. For example, while the random split is believed to be an unsuited choice, we show that for the Desc2D and WHIM representations this is actually the most representative splitting method. While this may seem unintuitive at first (e.g. due to the sheer size of the molecular space and the biased exploration of it so far), it makes more sense when considering that the range of these representations (i.e. the variety of 3D shapes and bio-physical properties of small molecules) is rather limited compared to a molecule's structural features. For many of the other representations, however, the currently popular splits are actually not difficult enough and the perimeter or maximum dissimilarity split is a better choice. **This suggests that evaluation standards in academia do not align with the situations encountered in ongoing drug discovery programs**. Consequently, this could result in a gap between advances in academia and industry pain points.

This raises the question of how big this gap is. We can answer this question by comparing the performance and calibration of models evaluated with a scaffold split and models

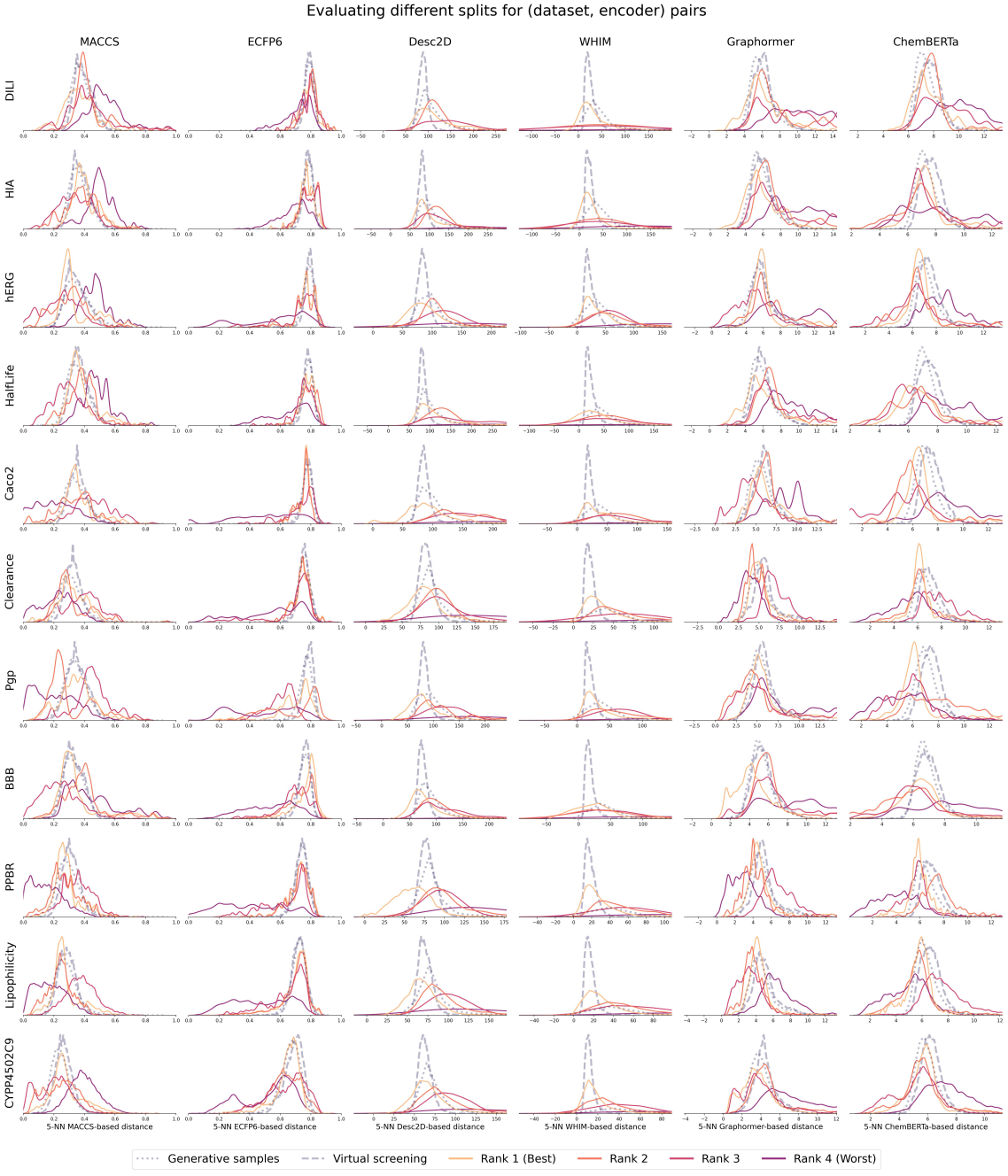

Figure 4.5: Visualization of the split prescription protocol - To get a realistic estimate of downstream performance, we can characterize, compare and replicate the covariate shifts encountered in ongoing drug discovery programs. In this grid, each column represents a molecular representation and each row represents a dataset. The datasets are ordered by their size from small (at the top) to large (at the bottom). This figure is best viewed digitally.

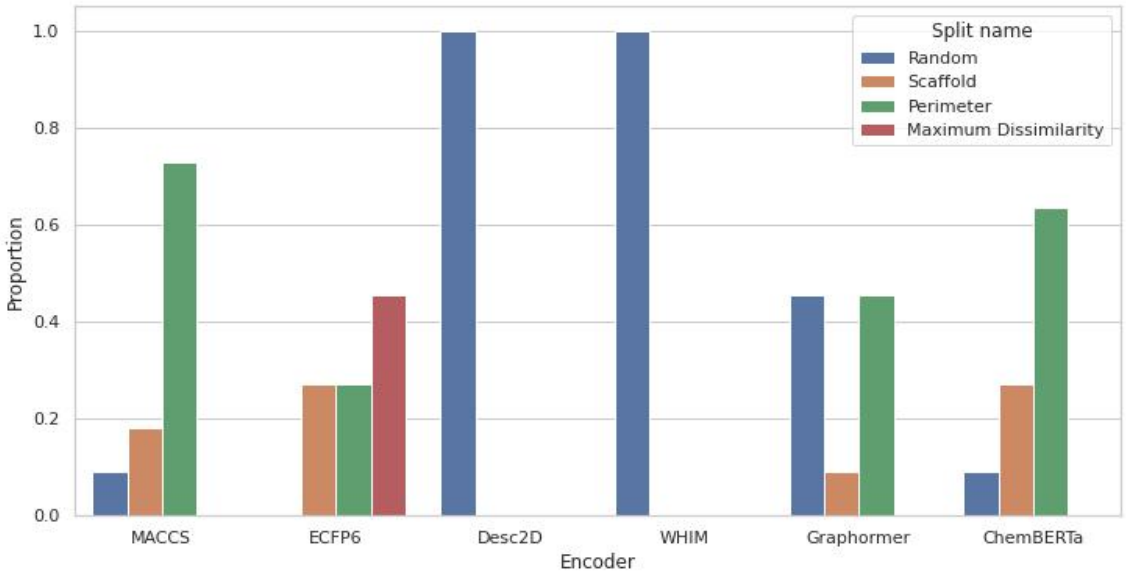

Figure 4.6: Proprotion of prescribed splits following from the proposed protocol - For each of the encoders, this plot shows the proportion of splits that are most representative of downstream molecules according to the protocol proposed in Section 4.1.4.

evaluated with the prescribed split. However, as we do not yet have all the data needed, we will revisit this question in Section 4.2.3.

This concludes the MOOD specification. In summary, the MOOD specification consists of two evaluation standards that ensure the model is evaluated in a way that closely matches how the model will be used in downstream applications. First of all, models in molecular scoring should always be evaluated in terms of both their performance and their calibration. Secondly, the train-test split should be chosen by following the described protocol to replicate a realistic covariate shift. We hope that the MOOD specification can guide future advances and reduce the gap between advances in academia and industry painpoints.

4.2 The MOOD investigation

Now that we have a complete specification of the OOD problem as it is encountered in ongoing drug discovery programs, we use the resulting evaluation standards to investigate the *effect* and *importance* of different tools to improve generalization through this new lens. To this end, we first need to establish which tools can be used to improve generalization.

4.2.1 Tools to improve generalization

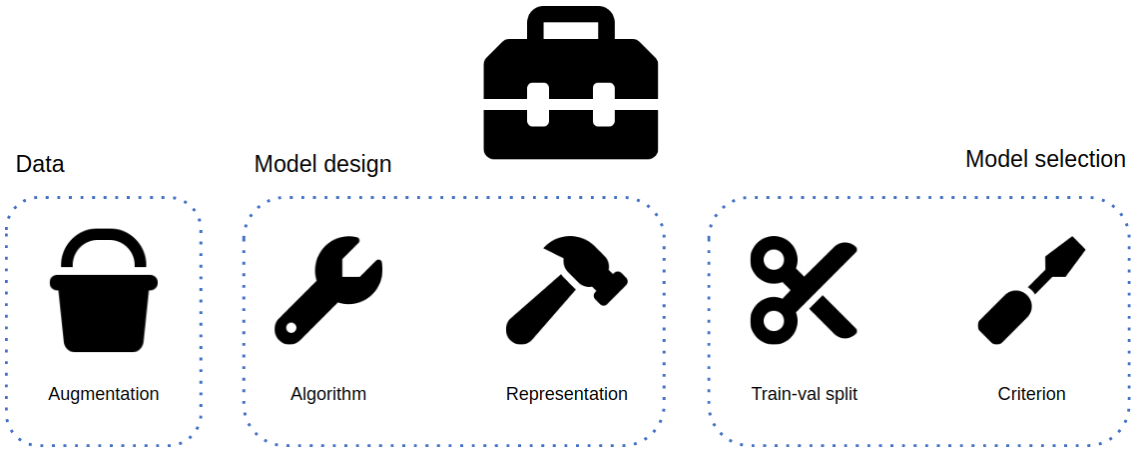

Figure 4.7: A visual overview of tools to improve generalization - Across domains, we found that these 5 tools are frequently used to improve the generalization of ML models. While data augmentation is an effective and popular tool in other domains, data augmentation techniques in molecular scoring are not yet well established.

Assuming that we start with a fixed dataset, we consider there to be a total of five tools ML practitioners can employ to improve generalization. We split these tools in three categories (see also Figure 4.7):

1. **Data**: In other domains (e.g. computer vision) a popular and effective technique to improve generalization is to augment a dataset by a set of transformations that respect the symmetries of the problem (e.g. by rotating an image, you do not alter the class of the object in that image). In molecular scoring, to the best of our knowledge, there is no such set of commonly accepted transformations. While we think this to be a promising path moving forward, we therefore do not consider data augmentation in this study.

2. **Model design**:

   - *Algorithm*: Some algorithms have inductive biases that are better suited for generalization, as discussed in Section 3.3.
   - *Representation*: Different representations carry different information and are differently suited for generalization, as discussed in Section 2.3.

Criterion	Description
Default	The mean validation performance.
Domain Weighted	The mean weighted validation performance, where the weight of each sample is one over the domain frequency of the domain it is part of.
Distance Weighted	The mean weighted validation performance, where the weight of each sample is its distance to the train set.
~~Calibration~~	~~The mean validation calibration~~
~~Calibration x Metric~~	~~The mean validation calibration $\in [0, 1]$ times the mean validation performance.~~

Table 4.6: An overview of the different model selection criteria - A criterion dictates how to compare different models. We ran experiments for all five criteria listed in this table, but due to a bug in the code the results for *Calibration* and *Calibration x Metric* were unusable and have been filtered out.

3. **Model selection**:

   - *Train-val split*, how to split of a subset of the data on which the model will be evaluated for model selection, as discussed in Section 3.2 and Section 4.1.4. While for the train-test split representativeness was the most important, for the train-val split this is less established. One could e.g. also expect a random train-val split to work best as a model then sees more diverse chemistry during training.
   - *Criterion*: To select one model out of many, we need a criterion that dictates which model is best. To the best of our knowledge, there is no prior work on this and common practice is to simply select based on the metric you're optimizing for, but we suspect that this can be an effective tool to improve generalization, especially when optimizing for both calibration and performance simultaneously. For the options considered in this study, see Table 4.6.

Now that we established which tools ML practicioners can use to improve generalization, we next investigate how these different tools compare to one another.

4.2.2 The effect and importance of different tools

The goal of the MOOD investigation is to inform future research directions and to more efficiently direct resources in ongoing drug discovery programs. To this end, we are interested in both the *effect* and the *importance* of each tool. The effect indicates how

Category	Tool	Options	
Model design	Algorithm	**Baseline:**	RF, GP, MLP
		DG:	MTL, VREx, IB-ERM
		DA:	CORAL, DANN, Mixup
Model design	Representation	**Structural:**	MACCS, ECFP6, WHIM
		Biochemical:	RDKit 2D
		Pre-trained:	Graphormer, ChemBERTa
Model selection	Train-val split	**Baseline:**	Random, BMS scaffold
		Generalization:	Perimeter, Max Dissimilarity
Model selection	Criterion	**Performance:**	Default, Domain-Weighted, Distance-Weighted
		~~**Calibration:**~~	~~Calibration~~
		~~**Mixed:**~~	~~Calibration x Metric~~
-	-	**Initial seed:**	$\in [0, 1024)$

Table 4.7: An overview of the different options evaluated in the MOOD RCT - For each RCT trial, we randomly sample an option for each of the trials. Due to a bug in the code, the results for two of the criteria were unusable and have been filtered out.

different options of a tool differ in terms of their performance and calibration (e.g. how does a domain adaptation algorithm compare to a baseline algorithm). The importance then describes the variance of the effect to indicate how impactful that tool is on the final performance and calibration relative to the other tools (e.g. how impactful is a change in algorithm compared to a change in representation).

To have some universality to our conclusions, we want to test at least a few number of different options for each of the considered tools. However, even with just a few options for each, testing all possible combinations quickly becomes infeasible due to the curse of dimensionality. To not introduce any spurious correlations, we conduct a Randomized Control Trial (RCT) for each of the datasets (for the full experimental details, see Section 4.3.2). In the RCT, we randomly sample an option for each of the tools (see Table 4.7), run a hyper-parameter search and record the final test performance and calibration. Besides three baseline algorithms, we also include three DA (CORAL (90), DANN (91) and Mixup (92)) and three DG algorithms (MTL (93), VREx (94) and IB-ERM (95)). In line with the MOOD specification, we split the initial dataset in a train and test set using the splitting method that is prescribed by the protocol. In total, we ran a 100 trials per dataset and trained over 110.000 models throughout this RCT. The final distribution of

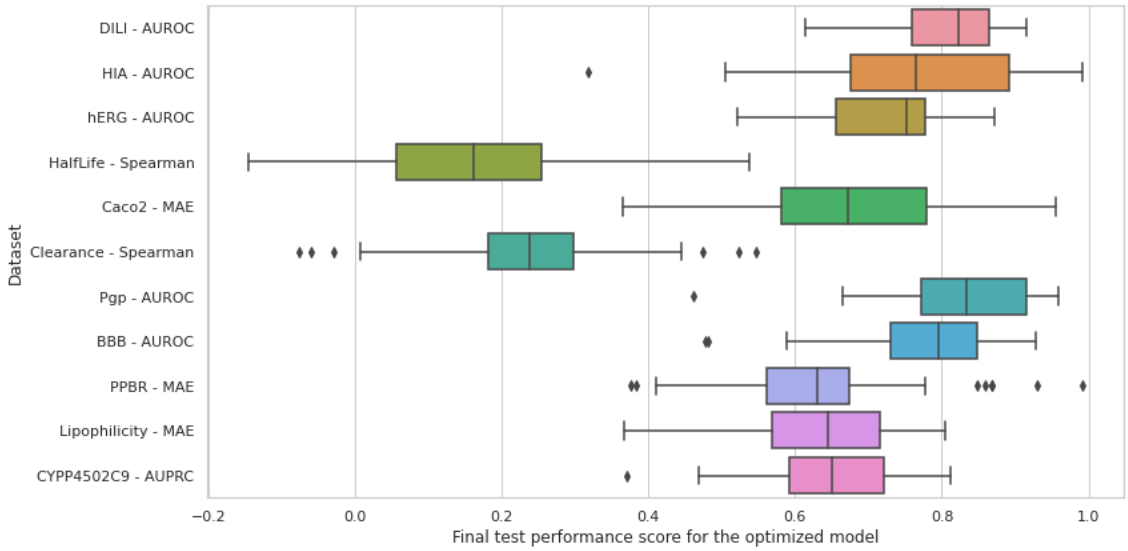

Figure 4.8: Distribution of test performance scores per dataset - The distribution of final test peformance scores (i.e. MAE, Spearman, AUROC or AUPRC), achieved throughout the different RCT trials. Aggregated per dataset. The datasets are ordered from small (at the top) to large (at the bottom)

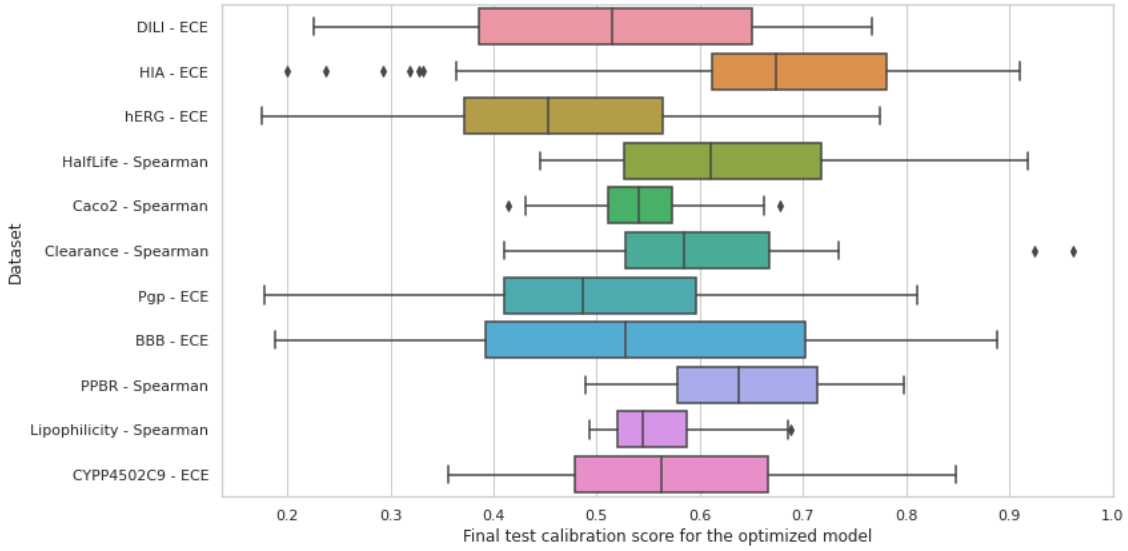

Figure 4.9: Distribution of test calibration scores per dataset - The distribution of final test calibration scores (i.e. ECE or Spearman), achieved throughout the different RCT trials. Aggregated per dataset. The datasets are ordered from small (at the top) to large (at the bottom)

performance and calibration scores on the test set is visualized in Figure 4.8 and Figure 4.9. While not frequent, an interesting observation is that the best model from the hyper-parameter search can still do worse than random (e.g. the Spearman correlation is lower than 0 or the AUROC score is lower than 0.5), suggesting that the model suffers from having learned spurious correlations that do well for the validation set, but not for the test set. Per dataset, per tool and between each ordered pair of that tool's options, we compute the mean, signed difference in performance and in calibration between the two options. The resulting distributions of differences are visualized in Figure 4.10 and Figure 4.11. As expected, these distributions are symmetric around zero.

Finally, to visualize the effect and importance of each of the tools, we introduce the *improvement ratio*. The improvement ratio is computed separately per dataset and is defined as the signed difference divided by the maximum difference for that dataset. Using the improvement ratio instead of just the (absolute, relative or plain) difference is simply a means to compare options and tools across datasets, despite the different metrics used to measure performance and calibration[1]. The effect is then computed as the mean importance ratio of an option and the importance of a tool is computed as the variance of its options' effects. To make the importance easier to interpret, we divide it by the variance of an equal number of options of randomly grouped seeds. An importance of 1 thus implies that the seed is of equal importance as the tool of interest. The effects for the test performance are visualized in Figure 4.12 and the effects for the test calibration are visualized in Figure 4.13. Finally, the importance of the different tools is given in Table 4.8.

Performance

When it comes to the performance, an interesting observation is that the only tool of significant importance is the molecular representation. All other tools have an importance score lower than 1.0 and are thus less influential than the seed. That the model selection tools are less important than the model design tools is expected. After all, in model selection we can only select the best performing model, but not improve the performance of the models we are choosing from. However, when visualizing the difference between the best test score found throughout the hyper-parameter search and the test score of the

[1]While using the relative difference or change might seem more logical, some of the metrics have negative values. Since there is no commonly accepted way to compute the relative difference in this scenario, we opted to use the improvement ratio instead.

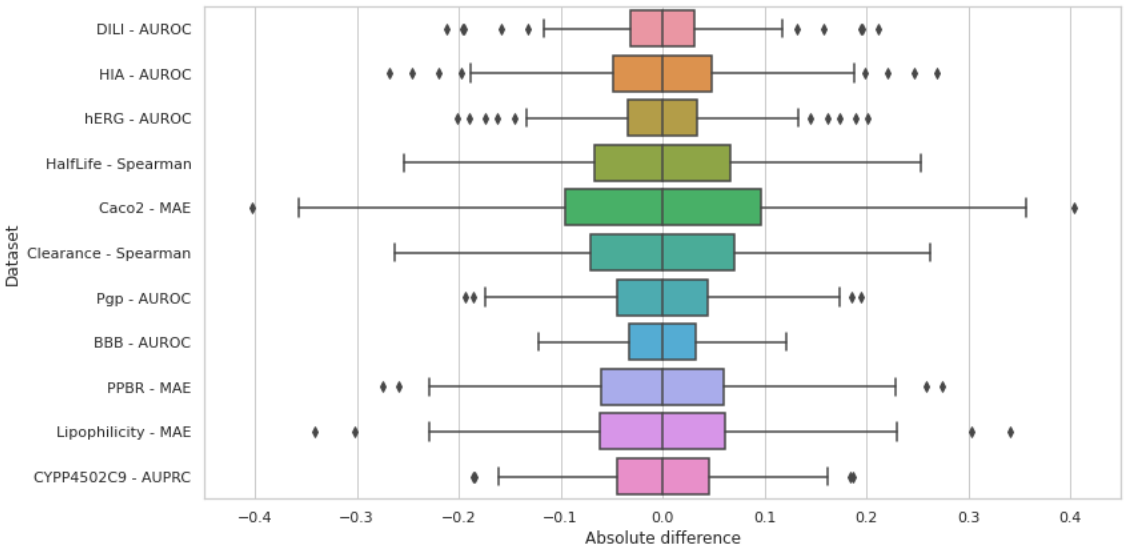

Figure 4.10: Distribution of differences in test performance - The distribution of mean, signed differences in performance between a tool's different options, aggregated per dataset. The datasets are ordered from small (at the top) to large (at the bottom)

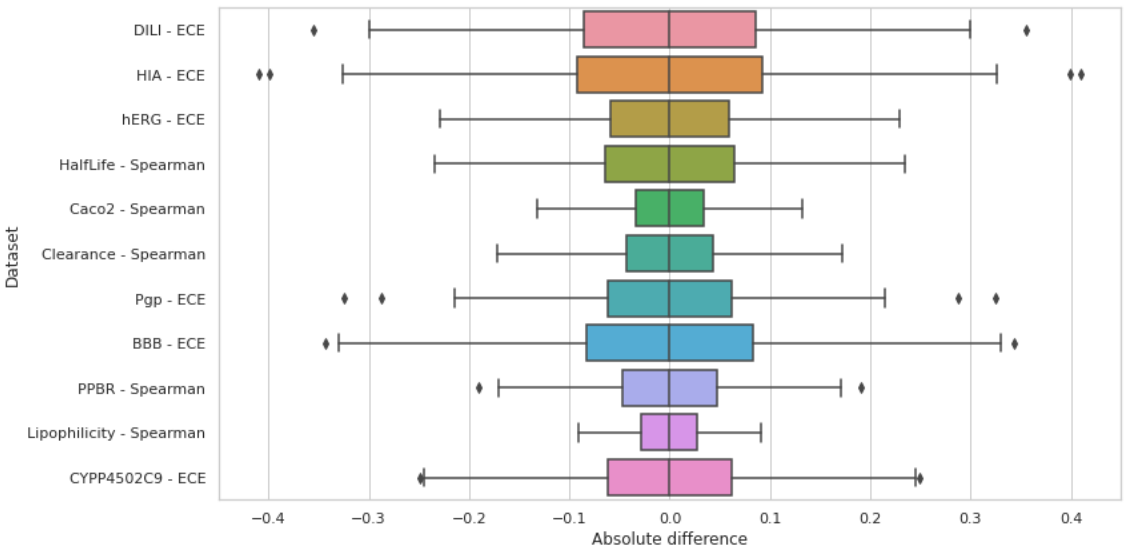

Figure 4.11: Distribution of differences in test calibration - The distribution of mean, signed differences in calibration between a tool's different options, aggregated per dataset. The datasets are ordered from small (at the top) to large (at the bottom)

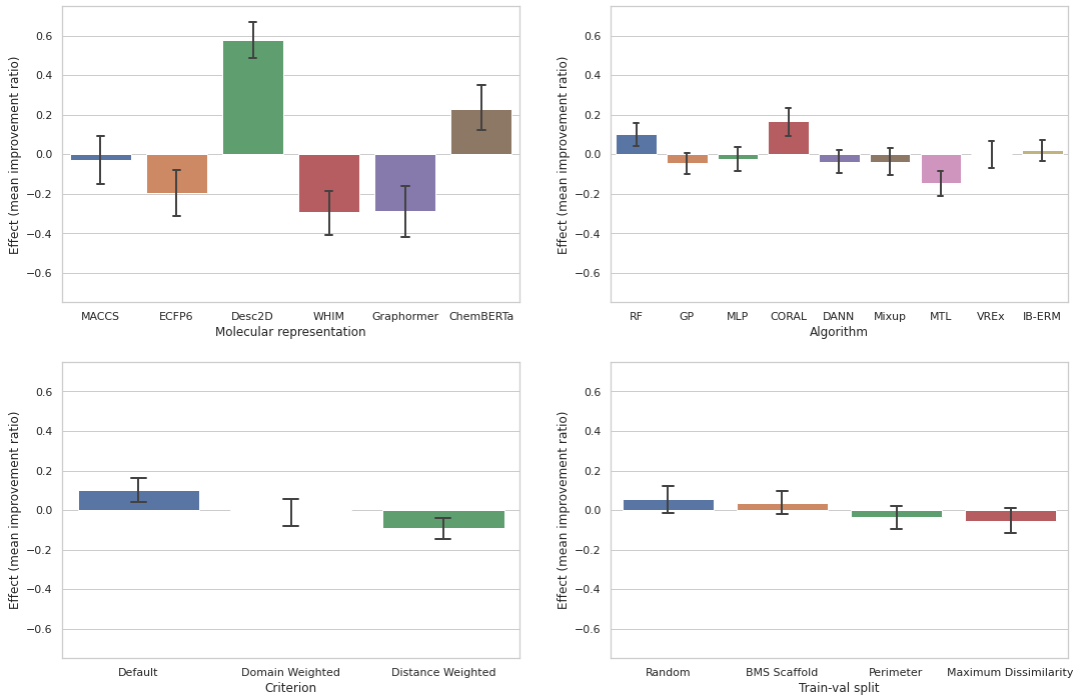

Figure 4.12: Comparing different tools on their test performance - The effect and importance of different tools to improve a model's test performance.

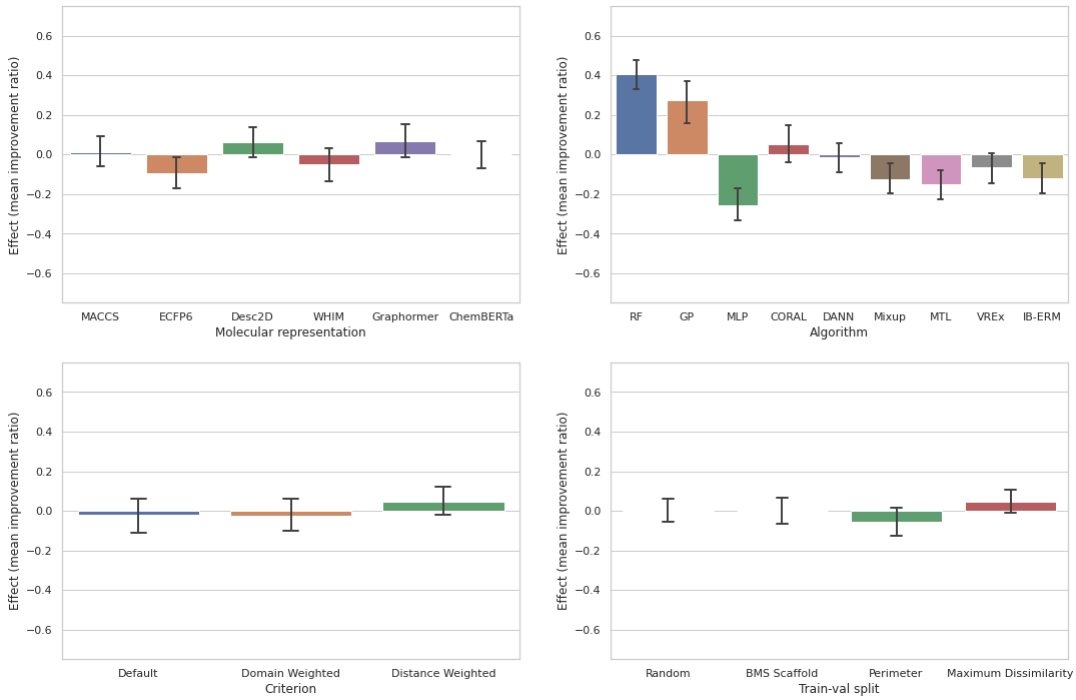

Figure 4.13: Comparing different tools on their test calibration - The effect and importance of different tools to improve a model's test calibration.

Tool	Performance importance	Calibration performance
Algorithm	0.983	2.000
Molecular representation	4.691	1.856
Train-val split	0.765	1.099
Criterion	0.806	1.154

Table 4.8: Importance of different tools to improve generalization - The importance captures the difference in performance between a tool's different options. The larger this difference, the more influential and thus important the tool. For interpretability, these scores are divided by the score of the seed. A score higher than 1 thus indicates that the tool is more influential than changing the seed.

model that was selected based on the criterion (see Figure 4.14), we observe that there often still is quite a large gap. **This suggests that model selection tools could be explored further as an effective tool to improve generalization.**

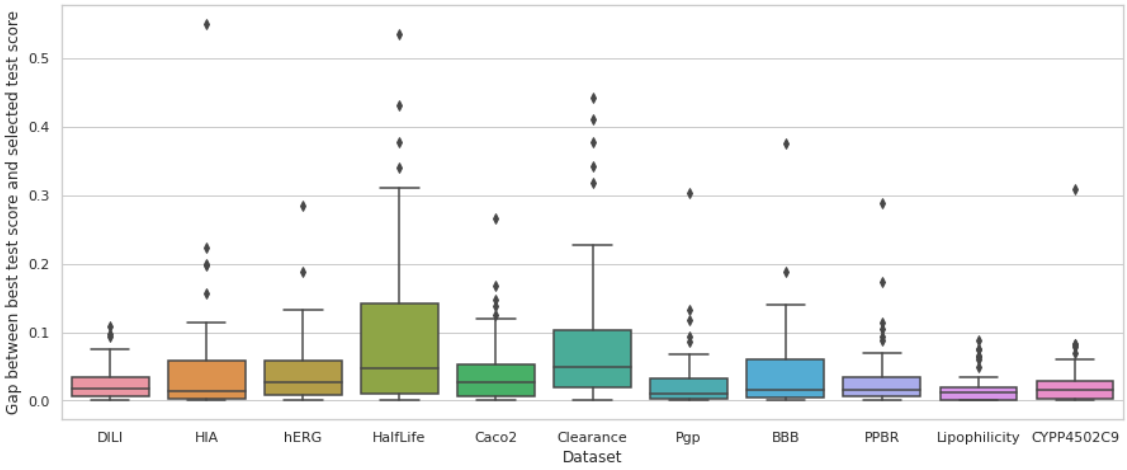

Figure 4.14: Difference between the test score of the best and selected model - Absolute differences between the maximum test score and the test score of the model that was selected during the hyper-parameter search. The datasets are ordered by size, from small on the left to large on the right.

Perhaps more surprising, is that the algorithm is also of negligible importance. While the best algorithm is from DA (CORAL), the baseline algorithms remain competitive to DA and DG algorithms. This is in line with conclusions from similar benchmarks (34, 56, 62). RF, a vanilla ML algorithm, outperforms most other methods without using unlabeled data from the test set and while being more efficient to train and easy to interpret.

Finally, considering the molecular representation, it is interesting to observe that the Desc2D representation does so well. A possible explanation could be that this is due to the prescribed test split. The Desc2D split is consistently prescribed the random split (see Figure 4.6) and since this split results in a test set that is close to the train set, we can expect performance to be higher (see Section 4.1.3). However, the WHIM representation is also consistently prescribed the random split, but its performance is actually worse than for all other representations. Similarly, it is surprising to see ECFP6 to do poorly as it is a popular choice in molecular scoring. This can be explained by the fact that the related distance metric quickly saturates (i.e. the molecules from downstream applications are far from the training set) and that its prescribed splits are thus more difficult, making ECFP6 ill-suited for generalization. **This implies that molecular representations should be evaluated based on both the information they carry and their suitability for generalization.**

Calibration

What stands out when inspecting the results for the test calibration, is that all tools have an importance higher than 1.0, which is in stark contrast to the results on the test performance. Visually inspecting the spread of the effects in the figures, one would actually expect the importance to be similar. One possible explanation could be that the seed is a surprisingly effective tool at improving importance, whereas it has less impact on the calibration.

Furthermore, it is interesting to observe that the GP and RF algorithm clearly outperform all deep-learning based methods when it comes to their calibration. Together with the earlier conclusion on the test performance, this shows that **RF remains a strong baseline in molecular scoring**, which newly proposed methods should be compared against.

4.2.3 Gap between current standards and the MOOD framework

In Section 4.1.4, we suggested that since the prescribed splits often do not match the commonly used scaffold split, there could be a gap between advances in academia and industry pain points. For this to be the case, the evaluation standards proposed by the MOOD specification should lead to different results than the currently standard evaluation

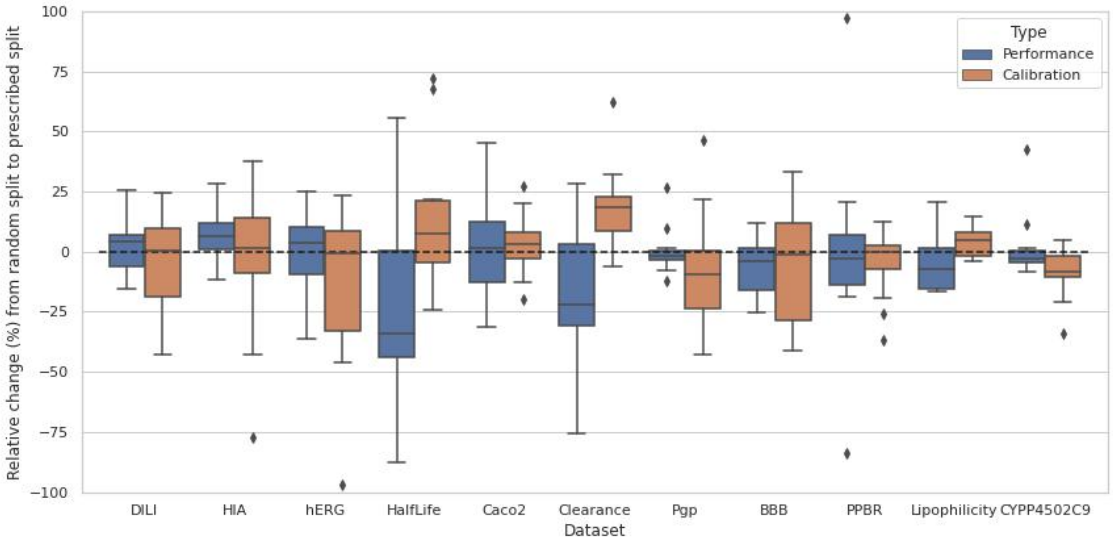

Figure 4.15: Comparing the the scaffold split and prescribed split for different datasets - An indication of the relative difference in performance and calibration on the test set between the scaffold split and the prescribed split. The datasets are ordered by their size from small on the left to large on the right.

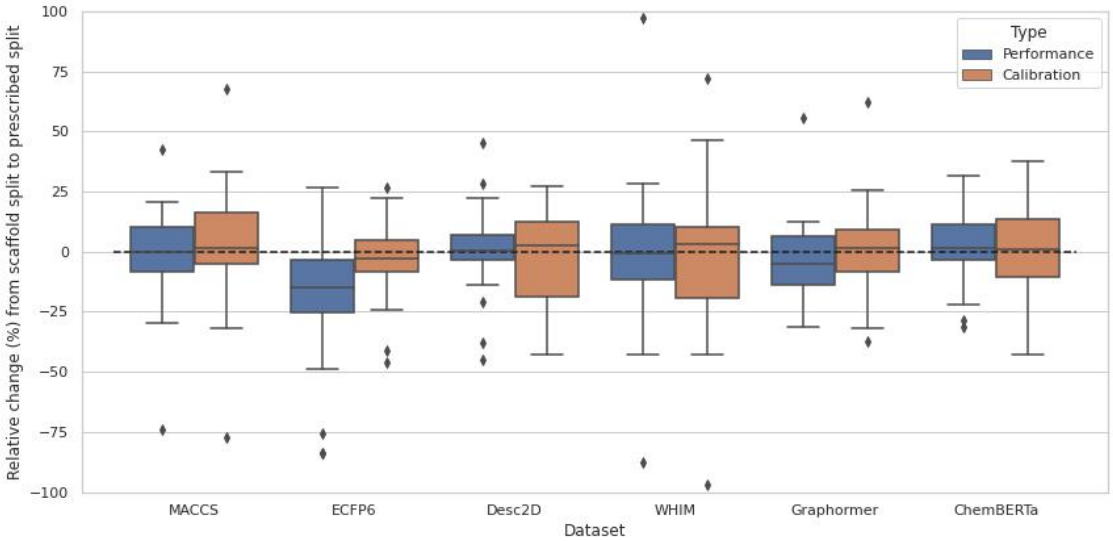

Figure 4.16: Comparing the scaffold split and prescribed split for different representations - An indication of the relative difference in performance and calibration on the test set between the scaffold split and the prescribed split.

standards. To get an estimate[1] of this difference, we combine the results collected for the baseline algorithms in Section 4.1.3 (as an indication of the performance and calibration on a scaffold split) and the results for for the baseline algorithm in Section 4.2.2 (as an indication of performance and calibration on the prescribed split). We visualize the differences in Figure 4.15 per dataset and in Figure 4.16 per representation.

This shows that on average, **we still over-estimate performance by using the scaffold split**. The drop in performance is larger for some of the representations than for others. Desc2D and WHIM, which are prescribed the random split by the MOOD specification are relatively unaffected while there is a large drop in performance for ECFP6.

This concludes the MOOD investigation. Crucially, we find that the representation is extra important when trying to improve OOD generalization in molecular scoring as it not only determines the information accessible to the model, but also dictates how similar (and thus how difficult) the molecules in the downstream applications are to the train set from the model's perspective. Other than that, we find that besides RF, which remains a surprisingly strong baseline, there is no clear winners.

4.3 Experimental setup

In this study, we run a total of two experiments. The *baseline experiment* (see Section 4.1.3) is used to validate the OOD metric by finding how a model's performance and calibration evolve as the distance to the train set grows. The *RCT experiment* (see Section 4.2.2) is used to find the effect and importance of different tools. This section will detail the experimental setup of these two experiments.

4.3.1 Baseline experiment

For each baseline, representation and dataset (for the specific options, see Section 4.1.3)) and seed $\in \{0, 1, 2, 3, 4\}$, we perform a hyper-parameter search through Bayesian Optimization using Optuna (96) and store the best model found. First, the dataset is split

[1]Due to excessive computational costs, we decided not to run a set of dedicated experiments for this. This means that the experiments are not perfectly comparable. Specifically, the train set used in the baseline experiment is considerably smaller and less diverse and in the RCT experiment we also use different model selection criteria and train-val splits. While we - based on earlier, smaller experiments - do not expect vastly different conclusions with a more comparable setup, these results should be taken as an approximation.

three times with different splitting methods to ensure that the molecules in the validation set and two hold-out test sets cover a large range of the related distance metric. In line with common practice, we use a scaffold split for the validation set and use the random and maximum dissimilarity split (see Section 4.1.4) for the test sets. Except for the two binary representations (MACCS and ECFP6), all representations are standardized using z-normalization. The hyper-parameter search consists of 50 trials in total. In each trial, we train the model using the train set and use early stopping on the validation set. The selected model is the one with the best performance on the validation set, according to the metric associated with the dataset (see Table 4.2). Using the best model found in the hyper-parameter search, we generate predictions for all molecules in the validation set and two test sets. We bin these predictions based on their distance and use bootstrapping ($n = 1000$) to compute the mean and variance of the performance and calibration of all predictions in the bin. To increase smoothness, we use overlapping bins and remove all bins with less than 25 samples (leaving about 180 overlapping bins in total). To have a consistent visualization, we negate the value of all metrics that should be minimized (i.e. MAE and ECE) so that a higher value is always better.

4.3.2 RCT experiment

Per dataset, we randomly sample 100 combinations from all possible combinations as summarized in Table 4.7. For each combination, we then run a hyper-parameter search through Bayesian Optimization using Optuna (96). First, we split the dataset in a train and test using the prescribed splitting method following the MOOD specification protocol and then split the train set once more in a validation set and final train set using the splitting method that was randomly sampled at the start. For each hyper-parameter search, we run a total of 50 trials, each of which consists of 5 independent train-val splits. For each split, we train an ensemble of 5 models (except for RF and GP, for which we train just a single model) and evaluate it - in terms of performance and calibration - on both the validation and holdout test set. The 5 validation scores of each trial are aggregated in a single criterion score, which dictates how good the model is compared to the other models trained in this hyper-parameter search. The test scores are aggregated in a single score by simply taking the mean. Except for the two binary representations (MACCS and ECFP6), all representations are standardized using z-normalization. For the DA algorithms, we use the unlabeled data from the test set as the target domain. For the DG algorithms, we use k-means clustering ($k = 8$) to define the domains.

5

Conclusion

5.1 Summary

The goal of this work was to challenge the i.i.d. assumption in molecular scoring. Specifically, this meant finding a set of evaluation standards that closely aligns with the situations encountered in ongoing drug discovery programs. Adopting such a set of evaluation standards will ensure that advances in academia better align with the pain points of industry, turning ML in an even more powerful tool to improve the efficiency of the drug discovery process.

To that end, we proposed the MOOD specification. By assuming a continuous, distance-based and, crucially, representation dependent OOD metric, we could compute the distance of any set of unlabeled molecules to any train set in any representation space. This allowed us to characterize the covariate shifts encountered in ongoing drug discovery programs and led us to our first, major contribution: A protocol that ranks different splitting methods based on their ability to replicate a realistic covariate shift. The usage of that protocol to prescribe a train-test split is joined by the standard of evaluating both the performance and calibration of a model to complete the MOOD specification.

The MOOD investigation naturally followed from the MOOD specification and aimed to answer the question of how different tools compare when we adhere to this new set of evaluation standards. To have some universality to our conclusions, while remaining computationally feasible and without introducing spurious correlations, we opted to conduct a Randomized Control Trial. We found that RF outperforms almost all other methods, in terms of both performance and calibration. Additionally, we found that choosing the

molecular representation is extra important as it not only affects the information accessible to the model, but also dictates how similar two molecules are from the model's perspective.

Through the MOOD framework, we hope to have proposed a set of principled guidelines that can help close the gap between advances in academia and industry pain-points.

5.2 Future work

During the pursuit of this thesis, there were many questions we would have liked to explore further, but that we unfortunately did not have the resources for. The in our opinion three most promising continuations are summarized below.

5.2.1 Expand the RCT

The most straight-forward continuation of this work is to collect more results for the RCT. While we aimed to choose a diverse set of options to compare, there was only so much we could include within the constraints of this thesis. For the datasets for example, it would be interesting to include larger datasets and to include some datasets outside of ADMET. For the algorithms, it would be interesting to include more advanced deep learning architectures, such as Graph Neural Networks (97). For the criterion and train-val split, we have seen (in Figure 4.14) that there is still plenty of space left to improve. This raises the question of what makes a good criterion and validation set, which up until now, to the best of our knowledge, is unanswered. For the molecular representation, we have seen that it is the most impactful tool we have to improve OOD generalization. With the successes of large-scale models in other domains (e.g. Natural Language Processing) and increasing efforts to replicate these success within drug discovery (9, 48, 98), it would be interesting to understand the implications of these models for OOD generalization in molecular scoring.

5.2.2 Investigating different OOD metrics

In this work, we assumed the k-NN distance as our OOD metric. While we observed that by using this distance the general tendency is as expected, an interesting line of work would be to explore different distance metrics. Additionally, it would be interested to expand the applicability of the OOD metric to more advanced representations, as for example encountered in multi-view learning or by concatenating two other representations.

Given a validated OOD metric, it would additionally be interesting to evaluate its usage within model selection. As we shortly discussed in Section 4.1.3, a good OOD metric only tells us that there is a trend. By evaluating what function best fits that trend for different algorithms or representations, this could help us select a model that generalizes better.

5.2.3 Synergy

In the MOOD investigation, we have looked at the impact of different tools in isolation. It seems reasonable, however, to expect certain pairs of options to do better than just the sum of their parts. For example, a particular molecular representation might be well suited to be used with a particular algorithm. An investigation of the synergistic (and antagonistic) effects of different tools in OOD generalization for molecular scoring would be an interesting research direction.

References

[1] Olivier J Wouters, Martin McKee, and Jeroen Luyten. **Estimated research and development investment needed to bring a new medicine to market, 2009-2018**. *Jama*, **323**(9):844–853, 2020. 1

[2] Joseph A DiMasi, Henry G Grabowski, and Ronald W Hansen. **Innovation in the pharmaceutical industry: new estimates of R&D costs**. *Journal of health economics*, **47**:20–33, 2016. 1

[3] Steven M Paul, Daniel S Mytelka, Christopher T Dunwiddie, Charles C Persinger, Bernard H Munos, Stacy R Lindborg, and Aaron L Schacht. **How to improve R&D productivity: the pharmaceutical industry's grand challenge**. *Nature reviews Drug discovery*, **9**(3):203–214, 2010. 1

[4] Antonio Lavecchia. **Machine-learning approaches in drug discovery: methods and applications**. *Drug discovery today*, **20**(3):318–331, 2015. 1, 6

[5] Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. **Applications of machine learning in drug discovery and development**. *Nature Reviews Drug Discovery*, **18**(6):463–477, 2019. 1

[6] Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. **The rise of deep learning in drug discovery**. *Drug discovery today*, **23**(6):1241–1250, 2018. 1, 5

[7] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. **Highly accurate protein structure prediction with AlphaFold**. *Nature*, **596**(7873):583–589, 2021. 1

[8] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. **Improved protein structure prediction using potentials from deep learning**. *Nature*, **577**(7792):706–710, 2020. 1

[9] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. **Do Transformers Really Perform Badly for Graph Representation?** In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 1, 46

[10] Yu Shi, Shuxin Zheng, Guolin Ke, Yifei Shen, Jiacheng You, Jiyan He, Shengjie Luo, Chang Liu, Di He, and Tie-Yan Liu. **Benchmarking Graphormer on Large-Scale Molecular Modeling Datasets**. *arXiv preprint arXiv:2203.04810*, 2022. 1

[11] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. **ChemBERTa: Large-scale self-supervised pretraining for molecular property prediction**. *arXiv preprint arXiv:2010.09885*, 2020. 1, 9, 21

[12] Philippe Schwaller, Benjamin Hoover, Jean-Louis Reymond, Hendrik Strobelt, and Teodoro Laino. **Unsupervised attention-guided atom-mapping**. 2020. 1

[13] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. **Shortcut learning in deep neural networks**. *Nature Machine Intelligence*, **2**(11):665–673, 2020. 2, 14

[14] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. **Domain Generalization: A Survey**. *CoRR*, **abs/2103.02503**, 2021. 2, 14

[15] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. **A brief review of domain adaptation**. *Advances in data science and information engineering*, pages 877–894, 2021. 2, 14

[16] Peter Ertl. **Cheminformatics analysis of organic substituents: identification of the most common substituents, calculation of substituent properties, and automatic identification of drug-like bioisosteric groups**. *Journal of chemical information and computer sciences*, **43**(2):374–380, 2003. 2

[17] Regine S Bohacek, Colin McMartin, and Wayne C Guida. **The art and practice of structure-based drug design: a molecular modeling perspective**. *Medicinal research reviews*, **16**(1):3–50, 1996. 2

[18] Markus Rudin and Ralph Weissleder. **Molecular imaging in drug discovery and development**. *Nature reviews Drug discovery*, **2**(2):123–131, 2003. 4

[19] Mark A Lindsay. **Target discovery**. *Nature Reviews Drug Discovery*, **2**(10):831–838, 2003. 3

[20] Ricardo Macarron, Martyn N Banks, Dejan Bojanic, David J Burns, Dragan A Cirovic, Tina Garyantes, Darren VS Green, Robert P Hertzberg, William P Janzen, Jeff W Paslay, et al. **Impact of high-throughput screening in biomedical research**. *Nature reviews Drug discovery*, **10**(3):188–195, 2011. 4

[21] **Hypothesis management in the DMTA cycle**, Apr 2021. 5

[22] Steven S Wesolowski and Dean G Brown. **The strategies and politics of successful design, make, test, and analyze (dmta) cycles in lead generation**. *Lead Generation*, pages 487–512, 2016. 4

[23] Albert P Li. **Screening for human ADME/Tox drug properties in drug discovery**. *Drug discovery today*, **6**(7):357–366, 2001. 4

[24] Ross D King, Jonathan D Hirst, and Michael JE Sternberg. **New approaches to QSAR: neural networks and machine learning**. *Perspectives in Drug Discovery and Design*, **1**(2):279–290, 1993. 5

[25] Patrick Cramer. **AlphaFold2 and the future of structural biology**. *Nature Structural & Molecular Biology*, **28**(9):704–705, 2021. 6

[26] Alexander B. Tong, Jason D. Burch, Daniel McKay, Carlos Bustamante, Michael A. Crackower, and Hao Wu. **Could AlphaFold revolutionize chemical therapeutics?** *Nature Structural & Molecular Biology*, **28**(10):771–772, Oct 2021. 6

[27] Kirill Veselkov, Guadalupe Gonzalez, Shahad Aljifri, Dieter Galea, Reza Mirnezami, Jozef Youssef, Michael Bronstein, and Ivan Laponogov. **HyperFoods: Machine intelligent mapping of cancer-beating molecules in foods**. *Scientific reports*, **9**(1):1–12, 2019. 6

[28] Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, D Boscaini, MM Bronstein, and BE Correia. **Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning**. *Nature Methods*, **17**(2):184–192, 2020. 6

[29] Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackermann, et al. **A deep learning approach to antibiotic discovery**. *Cell*, **180**(4):688–702, 2020. 6

[30] Tiago Sousa, João Correia, Vítor Pereira, and Miguel Rocha. **Generative Deep Learning for Targeted Compound Design**. *Journal of Chemical Information and Modeling*, **61**(11):5343–5361, Nov 2021. 6

[31] Julien Horwood and Emmanuel Noutahi. **Molecular design in synthetically accessible chemical space via deep reinforcement learning**. *ACS omega*, **5**(51):32984–32994, 2020. 6

[32] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. **Inverse molecular design using machine learning: Generative models for matter engineering**. *Science*, **361**(6400):360–365, 2018. 6, 7

[33] Thomas Blaschke, Josep Arús-Pous, Hongming Chen, Christian Margreitter, Christian Tyrchan, Ola Engkvist, Kostas Papadopoulos, and Atanas Patronov. **REINVENT 2.0: an AI tool for de novo drug design**. *Journal of Chemical Information and Modeling*, **60**(12):5918–5922, 2020. 6, 29

[34] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. **Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development**. *Proceedings of Neural Information Processing Systems, NeurIPS Datasets and Benchmarks*, 2021. 8, 13, 15, 21, 25, 40

[35] David Rogers and Mathew Hahn. **Extended-connectivity fingerprints**. *Journal of chemical information and modeling*, **50**(5):742–754, 2010. 8, 21

[36] Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. **Reoptimization of MDL keys for use in drug discovery**. *Journal of chemical information and computer sciences*, **42**(6):1273–1280, 2002. 8, 21

[37] Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas. **Molecular fingerprint similarity search in virtual screening**. *Methods*, **71**:58–63, 2015. 8

[38] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian P Kelley, Andrew Palmer, Volker Settels, et al. **Are learned molecular representations ready for prime time?** 2019. 8

[39] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. **A comprehensive survey on graph neural networks**. *IEEE transactions on neural networks and learning systems*, **32**(1):4–24, 2020. 8

[40] David Weininger. **SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules**. *Journal of chemical information and computer sciences*, **28**(1):31–36, 1988. 8

[41] Igor V Tetko, Pavel Karpov, Ruud Van Deursen, and Guillaume Godin. **State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis**. *Nature communications*, **11**(1):1–11, 2020. 8

[42] Peter W Atkins and Ronald S Friedman. *Molecular quantum mechanics*. Oxford university press, 2011. 9

[43] Chanin Nantasenamat, Chartchalerm Isarankura-Na-Ayudhya, Thanakorn Naenna, and Virapong Prachayasittikul. **A practical overview of quantitative structure-activity relationship**. 2009. 9

[44] A Crum-Brown and TR Fraser. **The connection of chemical constitution and physiological action**. *Trans R Soc Edinb*, **25**(1968-1969):257, 1865. 9

[45] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. **Do transformers really perform badly for graph representation?** *Advances in Neural Information Processing Systems*, **34**:28877–28888, 2021. 9, 21

[46] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. **Self-Supervised Graph Transformer on Large-Scale Molecular Data**, 2020. 9

[47] Hannes Stärk, Dominique Beaini, Gabriele Corso, Prudencio Tossou, Christian Dallago, Stephan Günnemann, and Pietro Liò. **3D Infomax improves GNNs for Molecular Property Prediction**. *arXiv preprint arXiv:2110.04126*, 2021. 9

[48] Nathan Frey, Ryan Soklaski, Simon Axelrod, Siddharth Samsi, Rafael Gomez-Bombarelli, Connor Coley, and Vijay Gadepally. **Neural Scaling of Deep Chemical Models**. 2022. 9, 46

[49] Joanna S Jaworska, M Comber, C Auer, and CJ Van Leeuwen. **Summary of a workshop on regulatory acceptance of (Q) SARs for human health and environmental endpoints.** *Environmental health perspectives*, **111**(10):1358–1360, 2003. 12

[50] OECD. *Guidance Document on the Validation of (Quantitative) Structure-Activity Relationship [(Q)SAR] Models*. 2014. 12

[51] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. **Generalized out-of-distribution detection: A survey**. *arXiv preprint arXiv:2110.11334*, 2021. 12

[52] Kunal Roy, Supratik Kar, and Rudra Narayan Das. *Understanding the basics of QSAR for applications in pharmaceutical sciences and risk assessment*, chapter Validation of QSAR Models, pages 231–289. Academic press, 2015. 12, 13

[53] Domenico Gadaleta, Giuseppe Felice Mangiatordi, Marco Catto, Angelo Carotti, and Orazio Nicolotti. **Applicability domain for QSAR models: where theory meets reality**. *International Journal of Quantitative Structure-Property Relationships (IJQSPR)*, **1**(1):45–63, 2016. 12

[54] Joanna Jaworska, Nina Nikolova-Jeliazkova, and Tom Aldenberg. **QSAR applicability domain estimation by projection of the training set in descriptor space: a review**. *Alternatives to laboratory animals*, **33**(5):445–459, 2005. 12

[55] Tatiana I Netzeva, Andrew P Worth, Tom Aldenberg, Romualdo Benigni, Mark TD Cronin, Paola Gramatica, Joanna S Jaworska, Scott Kahn, Gilles Klopman, Carol A Marchant, et al. **Current status of**

methods for defining the applicability domain of (quantitative) structure-activity relationships: The report and recommendations of ECVAM workshop 52. *Alternatives to Laboratory Animals*, **33**(2):155–173, 2005. 12

[56] Ishaan Gulrajani and David Lopez-Paz. **In Search of Lost Domain Generalization**. *CoRR*, **abs/2007.01434**, 2020. 12, 15, 40

[57] Robert P Sheridan. **Time-split cross-validation as a method for estimating the goodness of prospective prediction.** *Journal of chemical information and modeling*, **53**(4):783–790, 2013. 13

[58] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. **Analyzing learned molecular representations for property prediction**. *Journal of chemical information and modeling*, **59**(8):3370–3388, 2019. 13, 30

[59] Katsuhisa Morita, Tadahaya Mizuno, and Hiroyuki Kusuhara. **Investigation of a Data Split Strategy Involving the Time Axis in Adverse Event Prediction Using Machine Learning**. *Journal of Chemical Information and Modeling*, **0**(0):null, 0. PMID: 35971760. 13

[60] Guy W Bemis and Mark A Murcko. **The properties of known drugs. 1. Molecular frameworks**. *Journal of medicinal chemistry*, **39**(15):2887–2893, 1996. 13

[61] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. **WILDS: A Benchmark of in-the-Wild Distribution Shifts**. *CoRR*, **abs/2012.07421**, 2020. 13, 14, 15

[62] Yuanfeng Ji, Lu Zhang, Jiaxiang Wu, Bingzhe Wu, Long-Kai Huang, Tingyang Xu, Yu Rong, Lanqing Li, Jie Ren, Ding Xue, et al. **DrugOOD: Out-of-Distribution (OOD) Dataset Curator and Benchmark for AI-aided Drug Discovery–A Focus on Affinity Prediction Problems with Noise Annotations**. *arXiv preprint arXiv:2201.09637*, 2022. 13, 15, 40

[63] Ye Hu, Dagmar Stumpfe, and Jürgen Bajorath. **Computational exploration of molecular scaffolds in medicinal chemistry: miniperspective**. *Journal of medicinal chemistry*, **59**(9):4062–4076, 2016. 14

[64] Csaba Szántai-Kis, István Kövesdi, György Kéri, and László Örfi. **Validation subset selections for extrapolation oriented QSPAR models**. *Molecular diversity*, **7**(1):37–43, 2003. 13, 30

[65] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Wenjun Zeng, and Tao Qin. **Generalizing to Unseen Domains: A Survey on Domain Generalization**. *arXiv preprint arXiv:2103.03097*, 2021. 14

[66] Gilles Blanchard, Gyemin Lee, and Clayton Scott. **Generalizing from several related classification tasks to a new unlabeled sample**. *Advances in neural information processing systems*, **24**:2178–2186, 2011. 14

[67] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. **Domain generalization via invariant feature representation**. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013. 14

[68] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. **Invariant risk minimization**. *arXiv preprint arXiv:1907.02893*, 2019. 15

[69] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. 15

[70] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018. 15

[71] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999. 15

[72] Paul Bertin, Jarrid Rector-Brooks, Deepak Sharma, Thomas Gaudelet, Andrew Anighoro, Torsten Gross, Francisco Martinez-Pena, Eileen L Tang, Cristian Regep, Jeremy Hayter, et al. **Recover: sequential model optimization platform for combination drug repurposing identifies novel synergistic compounds in vitro**. *arXiv preprint arXiv:2202.04202*, 2022. 15

[73] Eyke Hüllermeier and Willem Waegeman. **Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods**. *Machine Learning*, **110**(3):457–506, 2021. 16

[74] Moksh Jain, Salem Lahlou, Hadi Nekoei, Victor Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. **DEUP: Direct epistemic uncertainty prediction**. *arXiv preprint arXiv:2102.08501*, 2021. 16

[75] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. **Simple and scalable predictive uncertainty estimation using deep ensembles**. *Advances in neural information processing systems*, **30**, 2017. 16, 20

[76] Yarin Gal and Zoubin Ghahramani. **Dropout as a bayesian approximation: Representing model uncertainty in deep learning**. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. 16

[77] Hugh Chen, Scott Lundberg, and Su-In Lee. **Checkpoint ensembles: Ensemble methods from a single training process**. *arXiv preprint arXiv:1710.03282*, 2017. 16

[78] Leo Breiman. **Random forests**. *Machine learning*, **45**(1):5–32, 2001. 16, 20

[79] Carl Edward Rasmussen. **Gaussian processes in machine learning**. In *Summer school on machine learning*, pages 63–71. Springer, 2003. 16, 20

[80] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. **Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift**. *Advances in neural information processing systems*, **32**, 2019. 16, 25

[81] Kehang Han, Balaji Lakshminarayanan, and Jeremiah Liu. **Reliable graph neural networks for drug discovery under distributional shift**. *arXiv preprint arXiv:2111.12951*, 2021. 16, 20

[82] Simon Haykin and N Network. **A comprehensive foundation**. *Neural networks*, **2**(2004):41, 2004. 20

[83] Greg Landrum. **RDKit: Open-Source Cheminformatics Software**. 2016. 21

[84] Paola Gramatica. **WHIM descriptors of shape**. *QSAR & Combinatorial Science*, **25**(4):327–332, 2006. 21

[85] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. **Bert: Pre-training of deep bidirectional transformers for language understanding**. *arXiv preprint arXiv:1810.04805*, 2018. 21

[86] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. **Attention is all you need**. *Advances in neural information processing systems*, **30**, 2017. 21

[87] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. **Obtaining well calibrated probabilities using bayesian binning**. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. 25

[88] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. **SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python**. *Nature Methods*, **17**:261–272, 2020. 25

[89] David Mendez, Anna Gaulton, A Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, María Paula Magariños, Juan F Mosquera, Prudence Mutowo, Michał Nowotka, et al. **ChEMBL: towards direct deposition of bioassay data**. *Nucleic acids research*, **47**(D1):D930–D940, 2019. 29

[90] Baochen Sun and Kate Saenko. **Deep coral: Correlation alignment for deep domain adaptation**. In *European conference on computer vision*, pages 443–450. Springer, 2016. 35

[91] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. **Domain-Adversarial Training of Neural Networks**. 2015. 35

[92] Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. **Improve Unsupervised Domain Adaptation with Mixup Training**, 2020. 35

[93] Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. **Domain generalization by marginal transfer learning**. *arXiv preprint arXiv:1711.07910*, 2017. 35

[94] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. **Out-of-Distribution Generalization via Risk Extrapolation (REx)**, 2020. 35

[95] Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish. **Invariance Principle Meets Information Bottleneck for Out-of-Distribution Generalization**, 2021. 35

[96] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. **Optuna: A Next-generation Hyperparameter Optimization Framework**. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019. 43, 44

[97] Max Welling and Thomas N Kipf. **Semi-supervised classification with graph convolutional networks**. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016. 46

[98] Xinyuan Lin, Chi Xu, Zhaoping Xiong, Xinfeng Zhang, Ningxi Ni, Bolin Ni, Jianlong Chang, Ruiqing Pan, Zidong Wang, Fan Yu, et al. **PanGu Drug Model: Learn a Molecule Like a Human**. *bioRxiv*, 2022. 46