Drug Discovery under Covariate Shift with Domain-Informed Prior Distributions over Functions

Leo Klarner [1] Tim G. J. Rudner [1] Michael Reutlinger [2] Torsten Schindler [2]
Garrett M. Morris [1] Charlotte M. Deane [1] Yee Whye Teh [1]

Abstract

Accelerating the discovery of novel and more effective therapeutics is an important pharmaceutical problem in which deep learning is playing an increasingly significant role. However, real-world drug discovery tasks are often characterized by a scarcity of labeled data and significant covariate shift—a setting that poses a challenge to standard deep learning methods. In this paper, we present Q-SAVI, a probabilistic model able to address these challenges by encoding explicit prior knowledge of the data-generating process into a prior distribution over functions, presenting researchers with a transparent and probabilistically principled way to encode data-driven modeling preferences. Building on a novel, gold-standard bioactivity dataset that facilitates a meaningful comparison of models in an extrapolative regime, we explore different approaches to induce data shift and construct a challenging evaluation setup. We then demonstrate that using Q-SAVI to integrate contextualized prior knowledge of drug-like chemical space into the modeling process affords substantial gains in predictive accuracy and calibration, outperforming a broad range of state-of-the-art self-supervised pre-training and domain adaptation techniques.

1. Introduction

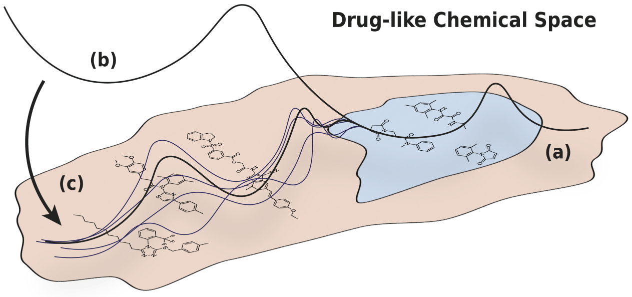


Figure 1. In early-stage drug discovery, bioactivity labels are usually only available for a small and biased subset of compounds (a). However, predictions are often most useful for novel molecules that are dissimilar to the ones already explored—an evaluative setting in which deep learning algorithms perform unreliably (b). We demonstrate that using a regularizing distribution over functions to encode prior knowledge of drug-like chemical space into the modeling process improves the predictive performance and calibration of neural networks in this extrapolative regime (c).

Discovering novel drug candidates that are able to safely and effectively treat neglected diseases or combat multidrug-resistant pathogens is a challenging biomedical research problem of considerable scientific and societal importance. Leveraging modern deep learning algorithms to accurately predict clinically relevant molecular properties and reduce the need for time- and resource-intensive experiments has the potential to significantly accelerate the development of promising and innovative chemical leads in drug discovery.

A key feature of practical early-stage drug discovery research is the application of predictive models to novel compounds that are *structurally or functionally dissimilar* to molecules that have already been explored (see Figure 1). In such an *extrapolative regime*, the practical utility of machine learning systems hinges on their ability to (a) robustly generalize to unexplored areas of chemical space and (b) reliably indicate when they fail to do so by generating well-calibrated predictive uncertainty estimates. However, standard deep learning algorithms often perform poorly under covariate shift, generating both incorrect and highly miscalibrated predictions (Ovadia et al., 2019; Koh et al., 2021). This is particularly problematic in the context of early-stage drug discovery, where experimental labels are expensive to acquire and therefore only available for a small and often highly biased subset of compounds.

[1] Department of Statistics, University of Oxford, United Kingdom [2] Pharma Research & Early Development, Roche, Switzerland. Correspondence to: Leo Klarner <leo.klarner@stats.ox.ac.uk>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

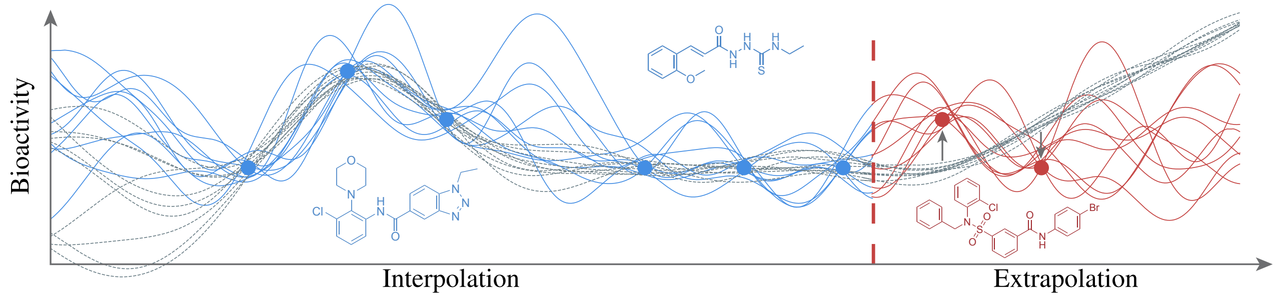


Figure 2. When trained on a small and highly biased subset of chemical space, standard neural networks (**gray**) rarely generalize well in the extrapolative regime. Our approach enables the construction of a problem-informed regularizing prior distribution over functions to place soft constraints on a neural network's hypothesis space, enabling better generalization and uncertainty quantification under covariate shift. In-distribution training points are shown in (**blue**) and out-of-distribution test points are shown in (**red**).

To improve the predictive performance of deep learning algorithms in such resource-constrained, low-data settings, we may wish to use relevant prior knowledge about the problem domain to specify inductive biases that make some predictive functions more likely than others. Common approaches to imbuing neural networks with useful inductive biases include (a) pre-training them on larger, potentially unlabeled datasets (Finn et al., 2017; Tan et al., 2018; Bommasani et al., 2021) and (b) adjusting their architectures to mirror appropriate invariances of their input domain (Bronstein et al., 2017; Satorras et al., 2021). However, these approaches are only an indirect—and often insufficiently precise—way of translating explicit modeling preferences into constraints over a neural network's hypothesis space.

In this paper, we present an alternative approach. To encode domain-informed prior knowledge of the data-generating process into neural network training, we specify a prior distribution over the space of **Q**uantitative **S**tructure-**A**ctivity mappings evaluated at a carefully selected set of context points, and perform **V**ariational **I**nference in the resulting probabilistic model (see Figure 2). We will refer to this method as **Q-SAVI**.

To demonstrate the practical utility of this approach, we construct a robust evaluation setup based on a carefully pre-processed bioactivity dataset. We then apply several different techniques to induce strong covariate and label shifts, resulting in challenging and practically meaningful train-test splits. Finally, we use Q-SAVI to specify explicit and problem-informed prior knowledge of drug-like chemical space and show that this substantially improves the predictive accuracy and calibration of deep learning algorithms in an out-of-distribution setting, outperforming a range of strong self-supervised pre-training, domain adaptation, and ensembling techniques.

Code and datasets are provided at:
`https://github.com/leojklarner/Q-SAVI`.

2. Predicting Properties to Discover Drugs

The overarching objective of small molecule drug discovery is to identify compounds that modulate a biological target of interest and elicit a therapeutically beneficial response. Unfortunately, the process of discovering a promising candidate to take into clinical trials is difficult and often unsuccessful, as the search space of viable drug-like molecules $\mathcal{X} = \{m_1, m_2, ...\}$ is vast, with estimates of $|\mathcal{X}|$ ranging from 10^{20} to 10^{60} (Bohacek et al., 1996; Ertl, 2003; Polishchuk et al., 2013). This is compounded by the inherent experimental limitations of medicinal chemistry, meaning that labels can only be acquired for a vanishingly small subset of compounds $\mathcal{X}' \subset \mathcal{X}$, with $|\mathcal{X}'| \ll |\mathcal{X}|$. Naturally, this has generated substantial interest in training supervised machine learning algorithms on available data $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) | \mathbf{x}_i \in \mathcal{X}', \mathbf{y}_i \in \mathcal{Y}\}_{i=1}^{N}$ to predict the properties of compounds in $\mathcal{X} \setminus \mathcal{X}'$.

As the purpose of such models is to accelerate the discovery of novel and more effective therapeutics, predictions are usually desired most on compounds that are meaningfully dissimilar to molecules in $\mathcal{X}'$. Were $\mathcal{X}'$ sampled uniformly from $\mathcal{X}$, that is, $\mathcal{X}' \sim \mathcal{U}(\mathcal{X})$, these predictions would be made in an interpolative regime, in which standard regularization techniques such as weight decay, dropout (Srivastava et al., 2014), and batch normalization (Ioffe and Szegedy, 2015) constitute effective approaches to minimizing the expected loss on new samples from $\mathcal{U}(\mathcal{X})$.

In practice, however, the composition of $\mathcal{X}'$ is largely determined by empirical considerations such as compound availability and the preferences and intuitions of medicinal chemists, resulting in a highly biased subsample $\mathcal{X}' \sim \tilde{p}_{\mathcal{X}}$. This means that, in order to reliably predict the properties of novel and scientifically interesting compounds, it is essential for machine learning algorithms to perform well in an extrapolative regime. As this requirement is distinct from in-distribution generalization, standard approaches to regularization are unlikely to be effective.

Instead, we propose an alternative regularization scheme—Q-SAVI—that builds on the fact that we are able to approximately sample from $\mathcal{U}(\mathcal{X})$ through large chemical databases such as ZINC (Irwin et al., 2020) or GDB (Polishchuk et al., 2013) to specify arbitrary modeling preferences on $\mathcal{X} \setminus \mathcal{X}'$. Specifically, we construct a probabilistic model of neural network *functions* and define a tractable prior distribution over parametric function mappings evaluated at points in $\mathcal{U}(\mathcal{X})$. We then extend this probabilistic model to include a label-space prior over $\mathcal{X}$, which encodes contextualized information on $\tilde{p}_{\mathcal{X}}$ and $\mathcal{Y}$, and demonstrate empirically that variational inference in this probabilistic model results in neural networks that make accurate predictions in regions of chemical space that they can reliably extrapolate to while generating well-calibrated predictive uncertainty estimates that indicate when correct predictions are unlikely.

3. Related Work

Starting with foundational attempts to link the electronic properties of different substituents to the reactivity (Hammett, 1937) and bioactivity (Hansch et al., 1962) of benzoic acid derivatives, the problem of predicting the properties of a molecule from its structure has long received considerable attention (Cherkasov et al., 2014). While simpler algorithms such as support vector machines (Cortes and Vapnik, 1995) and random forests (Breiman, 2001) remain a popular choice for such quantitative structure-activity relationship (QSAR) models, recent years have seen substantial interest in applying modern deep learning algorithms to this task (Ma et al., 2015; Gawehn et al., 2016; Zhang et al., 2017), including important attempts to improve their performance in low-data and out-of-distribution regimes.

Self-supervised pre-training techniques. To this end, Hu et al. (2019) and Rong et al. (2020) have introduced a range of self-supervised objectives to pre-train graph neural networks and graph transformers on a set of unlabeled molecular structures to generate initializations that can be efficiently fine-tuned on downstream tasks. However, the out-of-distribution generalization of their approaches was only assessed on scaffold splits—a setting that may underestimate of covariate and label shift encountered in many practical applications (Wallach and Heifets, 2018).

Domain adaptation techniques. Building on the fact that biases in the data collection process are often known at training time, domain adaptation and generalization techniques (Ganin et al., 2016; Sun and Saenko, 2016; Sagawa et al., 2019; Arjovsky et al., 2019) aim to improve the performance of deep learning algorithms in out-of-distribution settings by leveraging pre-specified domain indicators. However, these methods—originally developed for image data—have been found to provide limited benefits in the context of molecular property prediction (Ji et al., 2022).

Bayesian inference-based techniques. Bayesian Neural Networks (BNNs; Neal (1996)) provide a principled probabilistic framework for posterior inference over neural networks parameters and have long been explored in the context of drug discovery (Burden and Winkler, 1999; Burden et al., 2000). Even though they conceptually guarantee robustness in low-data regimes, their empirical performance often falls short of ensembling techniques or even standard stochastic gradient descent (Ovadia et al., 2019; Foong et al., 2019; Farquhar et al., 2020), including in the context of molecular property prediction (Ryu et al., 2019; Zhang et al., 2019).

While these approaches may improve the robustness of deep learning algorithms in some settings, they are limited in the extent to which they can encode problem-specific modeling preferences that, for example, encourage high predictive uncertainty away from the training data or specify prior knowledge of synthetic accessibility and patentability. For instance, the standard parameter-space formulation of BNNs precludes the specification of semantically meaningful prior information due to the highly non-linear and complex relationship between a neural network's parameters and the functions they encode.

Building on recent work that aims to address the shortcomings of BNNs (e.g., in specifying meaningful prior distributions and providing reliable uncertainty quantification) via function-space variational inference (Sun et al., 2019; Rudner et al., 2021; 2022b), we reframe QSAR modeling as inferring a posterior distribution over functions. We do so by specifying a prior distribution over function mappings along with a prior distribution over function evaluation points and performing variational inference in this probabilistic model, which allows us to explicitly encode prior beliefs about the distribution over functions as well as about the structure of the input space into neural network training.

4. Quantitative Structure-Activity VI

Consider the supervised learning setup outlined in Section 2, with the objective of training a machine learning model on the experimental labels of N independent and identically distributed samples drawn from a biased subset of chemical space, resulting in the data realizations $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N = (\mathbf{x}_\mathcal{D}, \mathbf{y}_\mathcal{D})$ of inputs $\mathbf{x}_i \in \mathcal{X}' \subset \mathcal{X}$ and labels $\mathbf{y}_i \in \mathcal{Y}$, where $\mathcal{Y} \subseteq \mathbb{R}^K$ for regression and $\mathcal{Y} \subseteq \{0, 1\}^K$ for classification tasks with K labels.

Let $p_{\mathbf{Y}|f(\mathbf{X};\mathbf{\Theta})}$ be an observation model of the labels $\mathbf{Y}$ given a latent stochastic function $f(\mathbf{X}; \mathbf{\Theta}) : \mathcal{X} \times \mathbb{R}^P \to \mathcal{Y}$ induced by a set of stochastic parameters $\mathbf{\Theta} \in \mathbb{R}^P$ and evaluated at a set of input points $\mathbf{X} \in \mathcal{X}$. Additionally, let $p_{f(\mathbf{X};\mathbf{\Theta})}$ be a prior distribution over such latent stochastic functions. $p_{\mathbf{Y}|f(\mathbf{X};\mathbf{\Theta})}(\mathbf{y}_\mathcal{D} \mid f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta}))$ is then the likelihood of observing labels $\mathbf{y}_\mathcal{D}$ under $f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta})$ — a realization of the stochastic function evaluated at inputs $\mathbf{x}_\mathcal{D}$.

Instead of formulating the posterior inference problem as finding the posterior distribution over stochastic parameters $\boldsymbol{\Theta}$, we follow Rudner et al. (2021) and reframe variational inference in stochastic neural networks as finding a posterior distribution over the *latent stochastic functions* $f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\Theta})$ at the training points $\mathbf{x}_{\mathcal{D}}$. In particular, while the parameter-space Bayesian inference problem is given by

$$p_{\boldsymbol{\Theta} \mid \mathcal{D}}(\boldsymbol{\Theta} \mid \mathcal{D}) = \frac{p_{\mathbf{Y} \mid \boldsymbol{\Theta}, \mathbf{X}}(\mathbf{y}_{\mathcal{D}} \mid \boldsymbol{\theta}, \mathbf{x}_{\mathcal{D}}) \, p_{\boldsymbol{\Theta}}(\boldsymbol{\theta})}{p_{\mathbf{Y} \mid \mathbf{X}}(\mathbf{y}_{\mathcal{D}} \mid \mathbf{x}_{\mathcal{D}})}, \tag{1}$$

the inference problem over $f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta})$ is expressed by

$$\begin{aligned} & p_{f(\mathbf{X}; \boldsymbol{\Theta}) \mid \mathcal{D}}(f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta}) \mid \mathcal{D}) \\ & = \frac{p_{\mathbf{Y} \mid f(\mathbf{X}; \boldsymbol{\Theta})}(\mathbf{y}_{\mathcal{D}} \mid f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta})) \, p_{f(\mathbf{X}; \boldsymbol{\Theta})}(f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta}))}{p_{\mathbf{Y} \mid \mathbf{X}}(\mathbf{y}_{\mathcal{D}} \mid \mathbf{x}_{\mathcal{D}})}, \end{aligned} \tag{2}$$

which includes an explicit dependence on the function-space prior evaluated at the training points $p_{f(\mathbf{X}; \boldsymbol{\Theta})}(f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta}))$, which allows us to specify arbitrary preferences for suitable parametric function mappings f.

To show how the inference problem in Equation (1) and Equation (2) are related, note that for a prior distribution over parameters $p_{\boldsymbol{\Theta}}$, the prior distribution $p_{f(\mathbf{X}; \boldsymbol{\theta})}$ over $f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta})$ induced by $p_{\boldsymbol{\Theta}}$ is given by

$$\begin{aligned} & p_{f(\mathbf{X}; \boldsymbol{\Theta})}(f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta})) \\ & \quad = \int_{\mathbb{R}^P} p_{\boldsymbol{\Theta}}(\boldsymbol{\theta}') \, \delta(f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta}) - f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta}')) \, \mathrm{d}\boldsymbol{\theta}', \end{aligned} \tag{3}$$

and, similarly, the posterior distribution $p_{f(\mathbf{X}; \boldsymbol{\Theta}) \mid \mathcal{D}}$ over $f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta})$ induced by the posterior distribution over parameters $p_{\boldsymbol{\Theta} \mid \mathcal{D}}$ is given by

$$\begin{aligned} & p_{f(\mathbf{X}; \boldsymbol{\Theta}) \mid \mathcal{D}}(f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta}) \mid \mathcal{D}) \\ & = \int_{\mathbb{R}^P} p_{\boldsymbol{\Theta} \mid \mathcal{D}}(\boldsymbol{\theta}' \mid \mathcal{D}) \, \delta(f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta}) - f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta}')) \, \mathrm{d}\boldsymbol{\theta}', \end{aligned} \tag{4}$$

where $\delta(\cdot)$ is the Dirac delta function (Wolpert, 1993; Rudner et al., 2022a). In the remainder of this section, subscripts will be dropped from probability density functions when the dependence is clear from context.

We will now extend this function-space formulation of Bayesian inference to define a probabilistic model that is able to integrate prior knowledge of the full input space $\mathcal{X}$ beyond a biased subset of training points $\mathbf{x}_{\mathcal{D}} \subseteq \mathcal{X}'$. Specifically, we extend the probabilistic model above to the random variables $f(\{\mathbf{X}, \mathbf{X}_{\mathcal{C}}\}; \boldsymbol{\Theta})$ and $\mathbf{X}_{\mathcal{C}}$, where $\mathbf{X}_{\mathcal{C}} \subseteq \mathcal{X} \setminus \mathcal{X}'$ is a set of *context points*, yielding the posterior distribution

$$\begin{aligned} & p(f(\{\mathbf{x}_{\mathcal{D}}, \mathbf{x}_{\mathcal{C}}\}; \boldsymbol{\theta}), \mathbf{x}_{\mathcal{C}} \mid \mathcal{D}) \\ & = \frac{p(\mathbf{y}_{\mathcal{D}} \mid f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta})) \, p(f(\{\mathbf{x}_{\mathcal{D}}, \mathbf{x}_{\mathcal{C}}\}; \boldsymbol{\theta}) \mid \mathbf{x}_{\mathcal{D}}, \mathbf{x}_{\mathcal{C}}) \, p(\mathbf{x}_{\mathcal{C}})}{p(\mathbf{y}_{\mathcal{D}} \mid \mathbf{x}_{\mathcal{D}})} \end{aligned} \tag{5}$$

where, for a stochastic function evaluation $f(\{\mathbf{x}_{\mathcal{D}}, \mathbf{x}_{\mathcal{C}}\}; \boldsymbol{\theta})$ defined by a valid stochastic process over $f(\cdot\,; \boldsymbol{\Theta})$, the likelihood $p(\mathbf{y}_{\mathcal{D}} \mid f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta}))$ and marginal likelihood $p_{\mathbf{Y} \mid \mathbf{X}}(\mathbf{y}_{\mathcal{D}} \mid \mathbf{x}_{\mathcal{D}})$ are independent of $f(\mathbf{X}_{\mathcal{C}}; \boldsymbol{\Theta})$ and $\mathbf{X}_{\mathcal{C}}$ by marginal consistency.

For non-linear function mappings $f : \mathcal{X} \times \mathbb{R}^P \to \mathcal{Y}$ parameterized by high-dimensional $\boldsymbol{\Theta} \in \mathbb{R}^P$, the inference problem specified in Equation (5) is analytically intractable. Instead, we may frame it variationally as

$$\min_{q_{\boldsymbol{\Theta}} \in \mathcal{Q}_{q_{\boldsymbol{\Theta}}}} \mathbb{D}_{\mathrm{KL}}(q_{f(\{\mathbf{X}, \mathbf{X}_{\mathcal{C}}\}; \boldsymbol{\Theta}), \mathbf{X}_{\mathcal{C}}} \,\|\, p_{f(\{\mathbf{X}, \mathbf{X}_{\mathcal{C}}\}; \boldsymbol{\Theta}), \mathbf{X}_{\mathcal{C}} \mid \mathcal{D}}) \tag{6}$$

for some variational distribution over parameters $q_{\boldsymbol{\Theta}}$ in a variational family $\mathcal{Q}_{q_{\boldsymbol{\Theta}}}$ (Wainwright and Jordan, 2008). Letting the variational distribution factorize as

$$q_{f(\{\mathbf{X}, \mathbf{X}_{\mathcal{C}}\}; \boldsymbol{\Theta}), \mathbf{X}_{\mathcal{C}}} \doteq q_{f(\{\mathbf{X}, \mathbf{X}_{\mathcal{C}}\}; \boldsymbol{\Theta})} q_{\mathbf{X}_{\mathcal{C}}} = q_{f(\{\mathbf{X}, \mathbf{X}_{\mathcal{C}}\}; \boldsymbol{\Theta})} p_{\mathbf{X}_{\mathcal{C}}}, \tag{7}$$

and assuming that $q_{\mathbf{X}_{\mathcal{C}}} = p_{\mathbf{X}_{\mathcal{C}}}$, we can reformulate the inference problem above in a simplified form as

$$\min_{q_{\boldsymbol{\Theta}} \in \mathcal{Q}_{q_{\boldsymbol{\Theta}}}} \mathbb{E}_{p_{\mathbf{X}_{\mathcal{C}}}} \left[\mathbb{D}_{\mathrm{KL}}(q_{f(\mathbf{X}; \boldsymbol{\Theta}) \mid \mathbf{X}_{\mathcal{C}}} \,\|\, p_{f(\mathbf{X}; \boldsymbol{\Theta}) \mid \mathbf{X}_{\mathcal{C}}, \mathcal{D}}) \right], \tag{8}$$

which can in turn be equivalently expressed as

$$\begin{aligned} \max_{q_{\boldsymbol{\Theta}} \in \mathcal{Q}_{q_{\boldsymbol{\Theta}}}} \Big\{ & \mathbb{E}_{q_{\boldsymbol{\Theta}} p_{\mathbf{X}_{\mathcal{C}}}} [\log p(\mathbf{y}_{\mathcal{D}} \mid f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta}))] \\ & - \mathbb{D}_{\mathrm{KL}}(q_{f(\{\mathbf{X}, \mathbf{X}_{\mathcal{C}}\}; \boldsymbol{\Theta})} \,\|\, p_{f(\{\mathbf{X}, \mathbf{X}_{\mathcal{C}}\}; \boldsymbol{\Theta})}) \Big\}. \end{aligned} \tag{9}$$

If $p_{\boldsymbol{\Theta}}$ is chosen to be an isotropic Gaussian distribution and $\mathcal{Q}_{q_{\boldsymbol{\Theta}}}$ is the family of mean-field Gaussian distributions, the prior and variational distributions in Equation (9) can be approximated using the local linearization scheme introduced in Rudner et al. (2022a). These approximations result in a factorized variational objective, making stochastic variational inference and stochastic gradient-based optimization techniques applicable (Hinton and van Camp, 1993; Graves, 2011; Hoffman et al., 2013; Blundell et al., 2015).

By enabling the specification of the context point distribution $p_{\mathbf{X}_{\mathcal{C}}}$ and the prior distribution over functions $p_{f(\{\mathbf{X}, \mathbf{X}_{\mathcal{C}}\}; \boldsymbol{\Theta})}$, this framework enables us to explicitly encode arbitrary modeling preferences as distributions that place high probability mass on relevant regions of the input domain and specify prior knowledge of preferred parametric function mappings on unlabelled data points.

After optimizing the variational objective with respect to the parameters of $q_{\boldsymbol{\Theta}}$, we obtain samples from approximate posterior predictive distribution through

$$q(\mathbf{y}_* \mid \mathbf{x}_*) \approx \frac{1}{M_*} \sum\nolimits_{j=1}^{M_*} p(\mathbf{y}_* \mid f(\mathbf{x}_*; \boldsymbol{\theta}^{(j)})), \tag{10}$$

with $\boldsymbol{\theta}^{(j)} \sim q_{\boldsymbol{\Theta}}$ and M_* being the number of Monte Carlo samples used to estimate the predictive distribution.

5. Empirical Evaluation

To demonstrate the practical utility of Q-SAVI, we establish a robust evaluation setup: In Section 5.1, we argue that many commonly-used bioactivity datasets may not be able to meaningfully assess the extrapolative power of supervised machine learning algorithms and present a carefully cleaned and pre-processed alternative dataset and in Section 5.2, we define an appropriate set of statistics to quantify covariate and label shifts in chemical space and use them to investigate the extent to which different splitting techniques induce data shift. In Section 5.3, we then use this experimental setup to demonstrate that employing Q-SAVI to incorporate domain-informed prior knowledge into the modeling process leads to significant gains in predictive accuracy and calibration, outperforming a range of strong self-supervised pre-training, domain adaptation, and ensembling techniques. Finally, in Section 5.4, we show that these strong empirical results extend to real-world production settings by evaluating our method on the time-split data presented in Ma et al. (2015).

5.1. Curating an Appropriate Dataset

A fundamental obstacle to training and evaluating QSAR models in the public domain is the scarcity of sufficiently large datasets with high-quality labels (Schneider et al., 2020). Even though collections of publicly available bioactivity data exist (Wu et al., 2018; Huang et al., 2021), they are often sourced directly from repositories of high-throughput screening (HTS) data such as PUBCHEM (Kim et al., 2019), CHEMBL (Mendez et al., 2019) or TOXCAST (Richard et al., 2016) without significant filtering or pre-processing. While this approach maximizes the number of available data points, it may reduce the discriminative power of model performance comparisons. For instance, a well-known problem of confirmatory dose-response screens—which make up the bulk of measurements in the above repositories—is that they usually contain a large number of reproducible false positive readouts (in many cases up to 95% of hits (Thorne et al., 2010)) caused by molecular substructures that interfere with an assay's readout system (Baell and Holloway, 2010; Dahlin et al., 2015). Using such data without further processing runs the risk of simply testing for the ability of algorithms to memorize these substructures instead of assessing meaningful extrapolative performance (Klarner et al., 2022).

To curate a dataset of sufficient quality to enable an informative comparison of predictive models, we used the measurement meta-data of bioactivity and toxicity screens to prioritize certain data points for further inspection. After surveying the publications associated with the most promising datasets, we selected a high-quality screening campaign for inhibitors of the development of liver-stage malaria parasites for further processing (Antonova-Koch et al., 2018). Specifically, we retrieved and reprocessed the raw measurement data to remove likely false positives and other experimental artifacts, yielding a binary classification dataset with 7,301 inactive and 849 active molecules, each measured in biological duplicate and confirmed as a true positive or negative through a set of quality-assuring counter-screens (see Appendix A for full details).

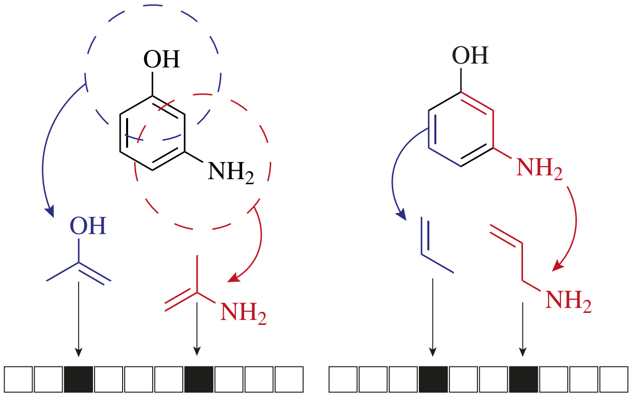


Figure 3. Schematic representation of extended-connectivity fingerprints (**left**) and RDKIT fingerprints (**right**). Both methods operate on the topological graph of a molecule and enumerate all labeled subgraphs up to a certain diameter, differing only in the space of substructures they consider. While RDKITFPS enumerate subgraphs of any shape, ECFPS are restricted to radial substructures. Once extracted, this set of subgraphs is hashed into a fixed-length bit vector.

5.2. Inducing and Quantifying Data Shift

Featurization. Commonly used techniques to numerically represent the structural properties of a molecule include strings, graphs, and topological fingerprints. For the following experiments, each molecule was featurized as both an extended-connectivity fingerprint (ECFP; Rogers and Hahn (2010)) and an RDKIT fingerprint (RDKITFPS), using the respective implementations in the open-source cheminformatics package RDKIT (Landrum et al., 2022). An illustration of this process is presented in Figure 3.

Statistics for covariate and label shift. To evaluate the extent to which different train-test splits induce covariate and label shift, we identified a set of suitable two-sample test statistics and used it to quantify the dissimilarity of the marginal covariate and label distributions of the respective training and test sets $\mathcal{D}_{\text{tr}} = (\mathbf{X}_{\text{tr}}, \mathbf{y}_{\text{tr}})$ and $\mathcal{D}_{\text{te}} = (\mathbf{X}_{\text{te}}, \mathbf{y}_{\text{te}})$.

Since $\mathbf{y}_{\text{tr}}$ and $\mathbf{y}_{\text{te}}$ consist of binary indicators of antimalarial activity, well-established categorical statistics such as Fisher's exact test (Upton, 1992) are applicable. In the following, its negative logarithmic p-value is used as a scalar indicator of label shift.

Defining a corresponding statistic to quantify covariate shift between two sets of molecules is more challenging, as they constitute disjoint sets of discrete objects. For this purpose, we used the maximum mean discrepancy (MMD) metric (Gretton et al., 2012) to quantify the difference between two samples of molecules as the distance between the embeddings of their expectations in a reproducing kernel Hilbert space (RKHS) defined by some mapping $\phi : \mathcal{X} \to \mathcal{H}$ and an associated kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}$. An empirical estimator of this statistic is obtained by

$$\begin{aligned}
&\mathrm{MMD}^2(\mathbf{X}_{\mathrm{tr}}, \mathbf{X}_{\mathrm{te}}) \\
&= \|\mathbb{E}_{\mathbf{x}\in\mathbf{X}_{\mathrm{tr}}}[\phi(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\in\mathbf{X}_{\mathrm{te}}}[\phi(\mathbf{x})]\|^2_{\mathcal{H}} \\
&= \mathbb{E}_{\mathbf{x}_i,\mathbf{x}_j\in\mathbf{X}_{\mathrm{tr}}}[k(\mathbf{x}_i,\mathbf{x}_j)] + \mathbb{E}_{\mathbf{x}_i,\mathbf{x}_j\in\mathbf{X}_{\mathrm{te}}}[k(\mathbf{x}_i,\mathbf{x}_j)] \\
&\quad - 2\mathbb{E}_{\mathbf{x}_i\in\mathbf{X}_{\mathrm{tr}},\mathbf{x}_j\in\mathbf{X}_{\mathrm{te}}}[k(\mathbf{x}_i,\mathbf{x}_j)]\,,
\end{aligned}$$

using the Jaccard/Tanimoto similarity coefficient

$$k_{\mathrm{jac}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \cap \mathbf{x}_j}{\mathbf{x}_i \cup \mathbf{x}_j} = \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{\|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - \langle \mathbf{x}_i, \mathbf{x}_j \rangle}$$

as an appropriate similarity metric, both due to its established use in the cheminformatics community (Bajusz et al., 2015) and the favorable properties of the RKHS that it induces. The MMD statistic is only valid if the mean embedding $\mathbb{E}_{\mathbf{x}\in\mathbf{X}}[\phi(\mathbf{x})]$ is injective, which is the case for strictly positive definite kernels operating in discrete domains (Borgwardt et al., 2006), such as k_{jac} (Bouchard et al., 2013).

Random and scaffold splits. Equipped with the appropriate statistical tools to quantify distributional similarities, we investigated the extent to which different train-test splits are able to emulate practically relevant covariate and label shifts, beginning with the two most popular approaches of splitting data either randomly or by scaffold. While *random splits* are commonly used in many domains, they are known to produce unrealistically optimistic performance estimates in the context of molecular property prediction. This is a consequence of the biased composition of many experimental datasets, which often contain structurally similar compounds from so-called chemical series. As these often exhibit very similar properties, distributing them evenly across data splits leads to a de-facto overlap between training and test sets that incentivizes overfitting and memorization (Wallach and Heifets, 2018). *Scaffold splits* attempt to mitigate this shortcoming by mapping each molecule to an overarching compound class—usually its Bemis-Murcko scaffold (Bemis and Murcko, 1996; 1999)—and splitting the data so that all molecules of a given scaffold are assigned to the same partition. However, this approach often results in a similar pathology, as even molecules with nominally different scaffolds can exhibit a high degree of structural and functional similarity, as illustrated in Figure 5.

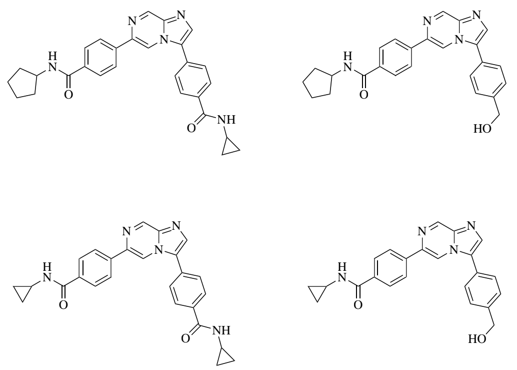


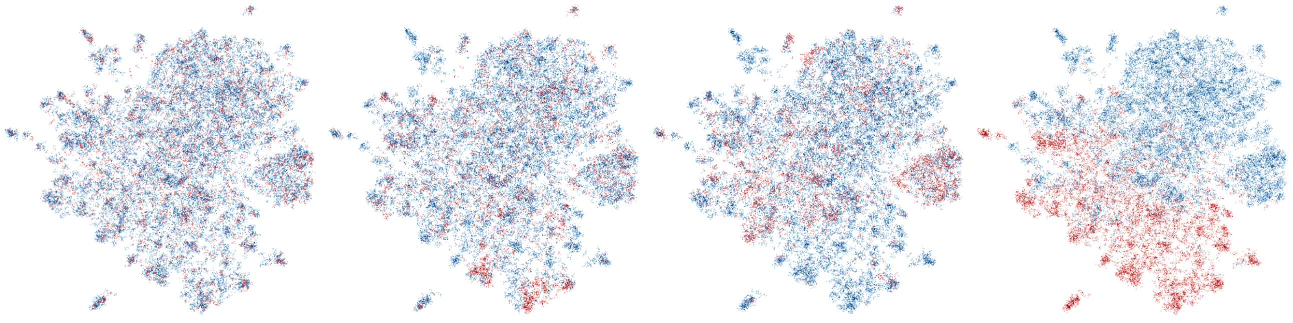
Figure 5. Even molecules with nominally different Bemis-Murcko scaffolds can exhibit a high degree of structural and functional similarity. Depicted are four structurally similar molecules from our antimalarial dataset that are assigned to four different scaffolds.

Molecular weight and spectral splits. To facilitate the comparison of models in an extrapolative regime, we explored two alternative approaches. A straightforward *molecular weight split* was used to induce data shift by assigning molecules into training and test sets based on a molecular weight cut-off, relying on the correlation of molecular size and binding strength to also induce strong label shift (Hopkins et al., 2014). More rigorously, we developed a clustering-based *spectral split* to generate data splits that are guaranteed to exhibit maximal covariate shift under the MMD statistic. By interpreting the Jaccard kernel Gram matrix $\mathbf{W}^{\mathrm{jac}} \in [0,1]^{|\mathbf{X}_{\mathcal{D}}|\times|\mathbf{X}_{\mathcal{D}}|}$ of a given set of molecules $\mathbf{X}_{\mathcal{D}}$ as the weighted adjacency matrix of a fully-connected similarity graph S, well-established spectral clustering algorithms (Von Luxburg, 2007) can be employed to identify an optimal partitioning of S that maximizes the similarity within and minimizes the similarity between partitions.

We present a comparison of the resulting covariate and label shift statistics in Table 1, which shows that molecular weight and spectral clustering-based splits generate a significantly more extrapolative evaluation setup than random and scaffold splits. This is substantiated by the qualitative visualization presented in Figure 4.

Table 1. A summary of the covariate and label shifts induced by the different train-test splits presented in Section 5.2, using rdkit and extended-connectivity (EC) fingerprints. Covariate shift is quantified as the Jaccard kernel-based MMD statistic, while label shift is quantified as the negative log p-value of Fisher's exact test.

Split	Covariate Shift *(rdkit, EC)*	Label Shift *(rdkit, EC)*
Random	0.00, 0.00	0.00
Scaffold	0.08, 0.07	4.23
Weight	0.14, 0.10	**61.96**
Spectral	**0.34**, **0.25**	17.49, 50.05



Random split. Scaffold split. Molecular weight split. Spectral split.

Figure 4. A visual comparison of the covariate shift induced by different train-test splits using rdkit fingerprints, colored in **blue** and **red** respectively. While random and scaffold splits lead to relatively similar training and test sets, molecular weight and spectral splits induce significantly stronger covariate shifts. The plots were generated using UMAP dimensionality reduction (McInnes et al., 2018).

5.3. Model Construction, Baselines & Results

Model construction. Using the increasingly data-shifted splits constructed in Sections 5.1 and 5.2, we assessed Q-SAVI with respect to its ability to improve the predictive accuracy and calibration of deep learning algorithms under covariate and label shifts. By leveraging the option to specify both an arbitrary context point distribution $p_{\mathbf{X}_c}$ and a prior distribution over parametric function mappings $p_{f(\{\mathbf{X},\mathbf{X}_c\};\boldsymbol{\Theta})}$, we used Q-SAVI to encode relevant information about both the input domain and the label space of the problem setup into the model. Specifically, we precomputed the featurizations of a uniform subsample of the ZINC database of commercially available compounds (Irwin et al., 2020) and used them to construct a uniform context point distribution $p_{\mathbf{X}_c} = \mathcal{U}(\bar{\mathcal{X}})$ over a set of 2×10^6 synthetically accessible drug-like molecules $\bar{\mathcal{X}}$. Additionally, we used the prior distribution $p_{f(\{\mathbf{X},\mathbf{X}_c\};\boldsymbol{\Theta})}$ over parametric mappings to encode an informative function-space prior that encourages high predictive uncertainty in unexplored regions of chemical space, counteracting the likelihood term in Equation (9) to generate better predictive uncertainty estimates.

Baselines. We compared the performance of the resulting probabilistic model to a range of standard baselines and state-of-the-art pre-training and domain adaptation techniques. The simplest of these models is regularized **logistic regression**, which is expected to underperform in an extrapolative regime due to the linearity of its logit function. While **random forest classifiers** (Breiman, 2001) represent a more flexible baseline with strong in-distribution generalization guarantees, they generally exhibit coarser decision boundaries at the fringes of the training distribution that are unlikely to perform well on covariate-shifted inputs. Standard deep learning methods such as multi-layer perceptrons **(MLPs)** have an even higher representational capacity, yet also generally underperform under data shift, yielding both incorrect and highly overconfident predictions (Ovadia et al., 2019; Koh et al., 2021). **Deep ensembles** are an effective technique to improve the predictive performance of MLPs by averaging the predictive distributions of a set of independently trained neural networks (Lakshminarayanan et al., 2017). To investigate the extent to which existing self-supervised pre-training techniques and more expressive model architectures impact the performance of deep learning algorithms in this setting, we fine-tuned graph isomorphism networks (**GINs**; Xu et al. (2018)) provided by Hu et al. (2019) both from scratch and from initializations that were pre-trained on compounds from the ZINC database using *context prediction* and *attribute masking* objectives. Additionally, we fine-tuned the graph transformer (**GROVER**) proposed by Rong et al. (2020) from a pre-trained initialization that was optimized on molecules from the ZINC and ChEMBL databases using self-supervised contextual property and graph-level motif prediction techniques. Finally, we adapted a range of domain adaptation and generalization techniques, including invariant risk minimization (**IRM**; Arjovsky et al. (2019)), group-distributionally robust training (**GroupDRO**; Sagawa et al. (2019)), domain-adversarial networks (**DANN**; Ganin et al. (2016)), and deep correlation alignment (**DeepCoral**; Sun and Saenko (2016)) from Ji et al. (2022) who provided them with data split-specific domain indicators.

Training and evaluation. To facilitate a fair comparison, we carried out an extensive hyperparameter search for every model, data split, and featurization. After an initial division of the data into training and test sets, the same data-splitting technique was applied again to derive a representative validation set. The hyperparameter setting with the lowest negative log-likelihood on that validation set was then used to train ten independent models using different random seeds. Full implementation details and hyperparameter ranges are provided in Appendix B.

Table 2. An overview of the test set performance of each model for each data split and featurization technique, quantified by the AUC-ROC (↑) and the Brier score (↓). All entries indicate the mean and standard errors computed over 10 independent training runs with different random seeds. The best models within a margin of statistical significance are highlighted in bold.

	Model & Featurization	Spectral Split		Weight Split		Scaffold Split		Random Split	
		ECFP	rdkitFP	ECFP	rdkitFP	ECFP	rdkitFP	ECFP	rdkitFP
AUC-ROC (↑)	Logistic Regression	.583±.000	.551±.000	.626±.000	.632±.000	.684±.000	.698±.000	.704±.000	.687±.000
	Random Forest	.576±.009	.552±.006	.592±.006	.567±.004	.605±.004	.642±.003	.696±.002	.690±.002
	MLP	.574±.006	.571±.003	.614±.004	.577±.005	.625±.010	.631±.014	**.720±.002**	.692±.004
	Deep Ensemble	.589±.006	.571±.002	.644±.001	.594±.002	.679±.001	**.697±.003**	**.720±.001**	**.710±.003**
	GIN	.549±.009	.551±.007	.582±.007		.664±.005		.685±.004	
	GIN (attr masking)	.588±.004	.559±.010	.625±.004		**.700±.002**		.705±.002	
	GIN (context pred)	.541±.005	.566±.009	.621±.003		.674±.003		**.713±.003**	
	Grover	.574±.002	.544±.006	.623±.003		.689±.003		.701±.001	
	Q-SAVI	**.606±.003**	**.603±.006**	**.650±.002**	**.643±.003**	.657±.004	**.701±.002**	.708±.001	.681±.002
Brier score (↓)	Logistic Regression	.131±.000	.111±.000	.051±.000	.049±.000	.101±.000	.100±.000	.087±.000	.088±.000
	Random Forest	.133±.000	**.110±.000**	.055±.000	.058±.000	.104±.000	.102±.000	**.085±.000**	**.086±.000**
	MLP	.133±.001	.111±.000	.050±.000	.055±.002	.103±.000	.108±.003	.087±.000	.088±.000
	Deep Ensemble	.133±.001	**.110±.000**	.048±.000	.052±.001	.101±.000	.100±.000	.086±.000	.086±.000
	GIN	.132±.001	.112±.001	.050±.000		.103±.000		.090±.001	
	GIN (attr masking)	**.130±.000**	.114±.002	.049±.000		**.100±.000**		.087±.000	
	GIN (context pred)	.134±.000	.113±.001	.050±.000		.101±.000		.087±.000	
	Grover	.134±.001	.111±.001	.049±.000		.101±.000		.088±.000	
	Q-SAVI	**.130±.000**	.112±.003	**.047±.000**	**.048±.000**	.102±.000	**.099±.000**	.088±.000	.090±.000

Following model training and hyperparameter selection, the predictive accuracy and calibration of the estimated test-set label probabilities were characterized by the area under the ROC curve (AUC-ROC) and the Brier Score, as these enable the direct comparison of models across test sets with different label distributions (see Table 2). Additionally, each algorithm's performance was characterized by the area under the precision-recall curve (AUC-PRC) and the adaptive calibration error (ACE; Nixon et al. (2019)), which closely mirror the AUC-ROC and Brier Score (see Table 5).

Results. The predictive accuracy and calibration metrics presented in Tables 2 and 5 (see Appendix B.1) demonstrate that Q-SAVI achieves significant performance gains in an out-of-distribution setting. On the spectral and molecular weight splits—the evaluation settings with the strongest covariate and label shift—Q-SAVI outperformed all other algorithms by a substantial and statistically significant margin in terms of predictive accuracy. Similarly, its predictive uncertainty estimates were significantly better calibrated than all other algorithms on the molecular weight split and most other algorithms on the ECFP-based spectral split.

On the substantially less data-shifted scaffold and random splits, relatively simple machine learning algorithms (e.g., random forests and deep ensembles) as well as more sophisticated self-supervised pre-training-based approaches consistently achieved the best predictive performance.

In line with the empirical observations of Ji et al. (2022), IRM, GroupDRO, DANN, and DeepCoral—domain adaptation and generalization techniques originally developed for images—were found to perform worse than most other techniques across most splits and featurizations (see Table 5).

5.4. Merck Molecular Activity Challenge

As a complementary assessment of the practical utility of Q-SAVI, we evaluated the method on the Merck Molecular Activity Challenge (Ma et al., 2015). Consisting of 15 datasets from real-world production settings, it provides time-split training and test sets that represent the data shift encountered throughout a molecular optimization campaign (Sheridan, 2013). As the compound structures are only provided in the form of anonymized atom-pair descriptors in count and bit vector form, using a uniform subsample of a large chemical database as a context point distribution is not possible.

Table 3. Covariate and label shift of time-split data from the Merck Molecular Activity Challenge. Covariate shift is quantified as the (multi-)set Jaccard kernel-based MMD statistic, while label shift is quantified as the two-sample Kolmogorov–Smirnov test statistic.

Dataset	Label Shift	Covariate Shift	
		Count Vector	Bit Vector
HIVPROT	0.579	0.132	0.162
DPP4	0.375	0.112	0.125
NK1	0.419	0.071	0.062

Table 4. A summary of the test set performance of each model for each of the datasets from the Merck Molecular Activity Challenge, quantified by the mean squared error (↓). All entries indicate the mean and standard error computed over 10 independent training runs with different random seeds. The best models within a margin of statistical significance are highlighted in bold.

Model	HIVPROT		DPP4		NK1	
	count vector	bit vector	count vector	bit vector	count vector	bit vector
L_1-Regression	1.137±.000	0.714±.000	1.611±.000	1.130±.000	0.482±.000	0.442±.000
L_2-Regression	0.999±.000	0.723±.000	1.495±.000	1.143±.000	0.498±.000	0.436±.000
Random Forest	0.815±.009	0.834±.010	1.473±.008	1.461±.012	0.458±.002	0.438±.002
MLP	0.768±.014	2.118±.015	1.393±.024	1.094±.029	0.443±.007	0.399±.006
Q-SAVI	**0.682±.019**	**0.664±.028**	**1.332±.017**	**1.028±.027**	**0.436±.007**	**0.387±.012**

Instead, our evaluation focused on the three most covariate- and label-shifted datasets (see Table 3), repurposing the remaining data as an anonymized context point distribution. All methods were evaluated following the protocol outlined in Section 5, with full details presented in Appendix C. The performance metrics for our method and the baseline algorithms investigated in Ma et al. (2015) are presented in Table 4, demonstrating that Q-SAVI performs favorably across every setting and outperforms all other models on the strongly data-shifted HIVPROT, DPP4, and NK1 datasets by a substantial and statistically significant margin.

6. Summary and Conclusions

The objective of early-stage drug discovery is to identify lead compounds that exhibit sufficient evidence of modulating a given disease phenotype—as well as suitable safety profiles—to qualify them for further investigation in in-vivo studies. Computational techniques that reliably predict the properties of novel molecules in unexplored regions of chemical space have the potential to substantially accelerate this time- and resource-intensive process. Motivated by the practical importance of developing such methods, we derived Q-SAVI, a probabilistic model that allows encoding explicit, problem-informed prior knowledge about the prediction domain into neural network training.

To construct a robust experimental setup and facilitate a practically meaningful evaluation of the proposed method, we carefully pre-processed a high-quality bioactivity dataset and explored different domain-specific statistics to quantify distribution shifts in this setting. Using these statistics to highlight the limited extent to which commonly used random and scaffold splits are able to induce meaningful covariate and label shifts, we built on two alternative molecular weight- and spectral clustering-based approaches to construct challenging train-test splits. Leveraging this extrapolative evaluation setup, we demonstrated that using Q-SAVI to provide neural networks with relevant and contextualized information on drug-like chemical space significantly improves both the predictive accuracy and calibration of neural network models, outperforming a range of state-of-the-art self-supervised pre-training, ensembling, and domain adaptation techniques.

The main limitation of the proposed method compared to standard training regimes is its increased computational cost, due to the amortized cost of having to pre-process a suitable context point distribution and the direct cost of having to perform each forward pass over both a mini-batch and a sample of context points. However, by keeping the size of each context set sample to be roughly comparable to the size of each mini-batch, we found this increase in computational cost to be manageable—especially in comparison to the computational cost of pre-training and fine-tuning related self-supervised methods or deep ensembles.

Promising avenues for future work include an investigation into how using Q-SAVI to specify problem-informed modeling preferences may improve the performance of deep learning algorithms for drug discovery applications that heavily rely on out-of-distribution generalization. For instance, the approach could be used to construct an acquisition function for an active learning loop to propose structural modifications that optimize the therapeutic properties of an existing lead compound (Nicolaou et al., 2007; Gómez-Bombarelli et al., 2018), as Q-SAVI generates robust predictions and additionally enables researchers to explicitly specify desirable exit vectors. It may also accelerate the discovery of novel compound classes that exhibit similar pharmacological properties to already explored molecules (Böhm et al., 2004; Hu et al., 2017), enabling the optimization of certain pharmacokinetic properties or the circumvention of patent restrictions. More broadly, we hope that this work encourages further research into the utility of probabilistic inference and domain-informed prior distributions over functions for drug discovery and beyond.

Acknowledgments

We thank anonymous reviewers for useful feedback. LK is funded by a Clarendon Scholarship. TGJR is funded by a Qualcomm Innovation Fellowship. We gratefully acknowledge the Oxford Advanced Research Computing service for providing computing resources and infrastructure.

References

Antonova-Koch, Y., Meister, S., Abraham, M., Luth, M. R., Ottilie, S., Lukens, A. K., Sakata-Kato, T., Vanaerschot, M., Owen, E., Jado, J. C., et al. (2018). Open-source discovery of chemical leads for next-generation chemoprotective antimalarials. *Science*, 362(6419):eaat9446.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2019). Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Baell, J. B. and Holloway, G. A. (2010). New substructure filters for removal of pan assay interference compounds (pains) from screening libraries and for their exclusion in bioassays. *Journal of Medicinal Chemistry*, 53(7):2719–2740.

Bajusz, D., Racz, A., and Heberger, K. (2015). Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7(1):1–13.

Bemis, G. W. and Murcko, M. A. (1996). The properties of known drugs. 1. molecular frameworks. *Journal of Medicinal Chemistry*, 39(15):2887–2893.

Bemis, G. W. and Murcko, M. A. (1999). Properties of known drugs. 2. side chains. *Journal of Medicinal Chemistry*, 42(25):5095–5099.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural networks. volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, Lille, France. PMLR.

Bohacek, R. S., McMartin, C., and Guida, W. C. (1996). The art and practice of structure-based drug design: a molecular modeling perspective. *Medicinal Research Reviews*, 16(1):3–50.

Böhm, H.-J., Flohr, A., and Stahl, M. (2004). Scaffold hopping. *Drug Discovery Today: Technologies*, 1(3):217–224.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H.-P., Schölkopf, B., and Smola, A. J. (2006). Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57.

Bouchard, M., Jousselme, A.-L., and Doré, P.-E. (2013). A proof for the positive definiteness of the jaccard index matrix. *International Journal of Approximate Reasoning*, 54(5):615–626.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. (2017). Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42.

Burden, F. R., Ford, M. G., Whitley, D. C., and Winkler, D. A. (2000). Use of automatic relevance determination in qsar studies using bayesian neural networks. *Journal of Chemical Information and Computer Sciences*, 40(6):1423–1430.

Burden, F. R. and Winkler, D. A. (1999). Robust qsar models using bayesian regularized neural networks. *Journal of Medicinal Chemistry*, 42(16):3183–3187.

Cherkasov, A., Muratov, E. N., Fourches, D., Varnek, A., Baskin, I. I., Cronin, M., Dearden, J., Gramatica, P., Martin, Y. C., Todeschini, R., et al. (2014). Qsar modeling: where have you been? where are you going to? *Journal of Medicinal Chemistry*, 57(12):4977–5010.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273–297.

Dahlin, J. L., Nissink, J. W. M., Strasser, J. M., Francis, S., Higgins, L., Zhou, H., Zhang, Z., and Walters, M. A. (2015). Pains in the assay: chemical mechanisms of assay interference and promiscuous enzymatic inhibition observed during a sulfhydryl-scavenging hts. *Journal of Medicinal Chemistry*, 58(5):2091–2113.

Ertl, P. (2003). 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.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Farquhar, S., Osborne, M. A., and Gal, Y. (2020). Radial Bayesian neural networks: Beyond discrete support in large-scale Bayesian deep learning. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1352–1362. PMLR.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR.

Foong, A. Y. K., Li, Y., Hernández-Lobato, J. M., and Turner, R. E. (2019). 'in-between' uncertainty in Bayesian neural networks.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Gawehn, E., Hiss, J. A., and Schneider, G. (2016). Deep learning in drug discovery. *Molecular Informatics*, 35(1):3–14.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. (2018). Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276.

Graves, A. (2011). Practical variational inference for neural networks. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, page 2348–2356, Red Hook, NY, USA. Curran Associates Inc.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

Hammett, L. P. (1937). The effect of structure upon the reactions of organic compounds. benzene derivatives. *Journal of the American Chemical Society*, 59(1):96–103.

Hansch, C., Maloney, P. P., Fujita, T., and Muir, R. M. (1962). Correlation of biological activity of phenoxyacetic acids with hammett substituent constants and partition coefficients. *Nature*, 194(4824):178–180.

Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., et al. (2020). Array programming with numpy. *Nature*, 585(7825):357–362.

Hinton, G. E. and van Camp, D. (1993). Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, COLT '93, page 5–13, New York, NY, USA. Association for Computing Machinery.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347.

Hopkins, A. L., Keserü, G. M., Leeson, P. D., Rees, D. C., and Reynolds, C. H. (2014). The role of ligand efficiency metrics in drug discovery. *Nature Reviews Drug Discovery*, 13(2):105–121.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., and Leskovec, J. (2019). Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*.

Hu, Y., Stumpfe, D., and Bajorath, J. (2017). Recent advances in scaffold hopping: miniperspective. *Journal of Medicinal Chemistry*, 60(4):1238–1246.

Huang, K., Fu, T., Gao, W., Zhao, Y., Roohani, Y., Leskovec, J., Coley, C. W., Xiao, C., Sun, J., and Zitnik, M. (2021). Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *arXiv preprint arXiv:2102.09548*.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR.

Irwin, J. J., Tang, K. G., Young, J., Dandarchuluun, C., Wong, B. R., Khurelbaatar, M., Moroz, Y. S., Mayfield, J., and Sayle, R. A. (2020). Zinc20—a free ultralarge-scale chemical database for ligand discovery. *Journal of Chemical Information and Modeling*, 60(12):6065–6073.

Ji, Y., Zhang, L., Wu, J., Wu, B., Huang, L.-K., Xu, T., Rong, Y., Li, L., Ren, J., Xue, D., et al. (2022). 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*.

Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B., et al. (2019). Pubchem 2019 update: improved access to chemical data. *Nucleic Acids Research*, 47(D1):D1102–D1109.

Klarner, L., Reutlinger, M., Schindler, T., Deane, C., and Morris, G. (2022). Bias in the benchmark: Systematic experimental errors in bioactivity databases confound multi-task and meta-learning algorithms. In *ICML 2022 2nd AI for Science Workshop*. PMLR.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. (2021). Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan,

S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6402–6413.

Landrum, G. et al. (2022). Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E., and Svetnik, V. (2015). Deep neural nets as a method for quantitative structure–activity relationships. *Journal of Chemical Information and Modeling*, 55(2):263–274.

McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Mendez, D., Gaulton, A., Bento, A. P., Chambers, J., De Veij, M., Félix, E., Magariños, M. P., Mosquera, J. F., Mutowo, P., Nowotka, M., et al. (2019). Chembl: towards direct deposition of bioassay data. *Nucleic Acids Research*, 47(D1):D930–D940.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*.

Neal, R. M. (1996). Bayesian Learning for Neural Networks.

Nicolaou, C. A., Brown, N., and Pattichis, C. S. (2007). Molecular optimization using computational multi-objective methods. *Current Opinion in Drug Discovery and Development*, 10(3):316.

Nixon, J., Dusenberry, M. W., Zhang, L., Jerfel, G., and Tran, D. (2019). Measuring calibration in deep learning. In *CVPR Workshops*, volume 2.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. (2019). Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems 32*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Polishchuk, P. G., Madzhidov, T. I., and Varnek, A. (2013). Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of Computer-aided Molecular Design*, 27(8):675–679.

Richard, A. M., Judson, R. S., Houck, K. A., Grulke, C. M., Volarath, P., Thillainadarajah, I., Yang, C., Rathman, J., Martin, M. T., Wambaugh, J. F., et al. (2016). Toxcast chemical landscape: paving the road to 21st century toxicology. *Chemical Research in Toxicology*, 29(8):1225–1251.

Rogers, D. and Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754.

Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., and Huang, J. (2020). Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33:12559–12571.

Rudner, T. G., Chen, Z., Teh, Y. W., and Gal, Y. (2021). Tractable function-space variational inference in bayesian neural networks. In *Advances in Neural Information Processing Systems*.

Rudner, T. G. J., Chen, Z., Teh, Y. W., and Gal, Y. (2022a). Tractabe Function-Space Variational Inference in Bayesian Neural Networks.

Rudner, T. G. J., Smith, F. B., Feng, Q., Teh, Y. W., and Gal, Y. (2022b). Continual Learning via Sequential Function-Space Variational Inference. In *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR.

Ryu, S., Kwon, Y., and Kim, W. Y. (2019). A bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification. *Chemical Science*, 10(36):8438–8446.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. (2019). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*.

Satorras, V. G., Hoogeboom, E., and Welling, M. (2021). E (n) equivariant graph neural networks. In *International Conference on Machine Learning*, pages 9323–9332. PMLR.

Schneider, P., Walters, W. P., Plowright, A. T., Sieroka, N., Listgarten, J., Goodnow, R. A., Fisher, J., Jansen, J. M., Duca, J. S., Rush, T. S., et al. (2020). Rethinking drug design in the artificial intelligence era. *Nature Reviews Drug Discovery*, 19(5):353–364.

Sheridan, R. P. (2013). Time-split cross-validation as a method for estimating the goodness of prospective prediction. *Journal of Chemical Information and Modeling*, 53(4):783–790.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Sun, B. and Saenko, K. (2016). Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 443–450. Springer.

Sun, S., Zhang, G., Shi, J., and Grosse, R. B. (2019). Functional variational Bayesian neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*, pages 270–279. Springer.

Thorne, N., Auld, D. S., and Inglese, J. (2010). Apparent activity in high-throughput screening: origins of compound-dependent assay interference. *Current Opinion in Chemical Biology*, 14(3):315–324.

Upton, G. J. (1992). Fisher's exact test. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 155(3):395–402.

Van Rossum, G. and Drake Jr, F. L. (1995). *Python tutorial*, volume 620. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Wainwright, M. J. and Jordan, M. I. (2008). *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc., Hanover, MA, USA.

Wallach, I. and Heifets, A. (2018). Most ligand-based classification benchmarks reward memorization rather than generalization. *Journal of Chemical Information and Modeling*, 58(5):916–932.

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

Wes McKinney (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.

Wolpert, D. H. (1993). Bayesian backpropagation over i-o functions rather than weights. In Cowan, J., Tesauro, G., and Alspector, J., editors, *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann.

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

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2018). How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

Zhang, L., Tan, J., Han, D., and Zhu, H. (2017). From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug Discovery Today*, 22(11):1680–1685.

Zhang, Y. et al. (2019). Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning. *Chemical Science*, 10(35):8154–8163.

Appendix

A. Data Curation and Pre-Processing

To generate an appropriate dataset of reliably labeled bioactivity measurements, we retrieved and reprocessed high-throughput screening data generated by Antonova-Koch et al. (2018) as part of a campaign to discover novel chemoprotective antimalarial drug candidates.

The authors established a cell-based phenotypic screening pipeline to identify compounds that inhibit the development of luciferase-expressing liver-stage *Plasmodium falciparum* parasites. After assaying a commercially-available chemical library of 538 273 of drug-like small molecules in a single-point primary screen, they selected the 9963 most promising compounds for a series of confirmatory dose-response screens. Specifically, an 8-point dilution series was used to assess, in duplicate, the potency and efficacy of each compound in the original assay (*Pbluc*). Additionally, the tendency of the assayed compounds to produce false positives and other experimental artifacts was investigated by performing a series of counter-screens that measure hepatic cytotoxicity (*HepG2tox*) and interference with the luciferase-based luminescent readout (*Ffluc*). The fact that all bioactivity measurements are (1) generated using biological duplicates and (2) associated with quantitative measures that reflect their likelihood to produce confounding experimental artifacts substantially improves the reliability of the resulting labels.

To facilitate the integration of bioactivity and counter-screen measurements and make the data more amenable to predictive modeling, the IC_{50} values that quantify the concentration at which a molecule produces half of its maximum inhibitory effect were converted to binary labels. Specifically, all compounds with an $\mathrm{IC}_{50} \leq 1.5\mu\mathrm{M}$ were denoted as active while all compounds with an $\mathrm{IC}_{50} \geq 3\mu\mathrm{M}$ were denoted as inactive, discarding 652 compounds with $1.5\mu\mathrm{M} \leq \mathrm{IC}_{50} \leq 3\mu\mathrm{M}$ and assigning qualified IC_{50} values to the appropriate class (see Figure 6 for a diagram of the IC_{50} distribution and the applied thresholds).

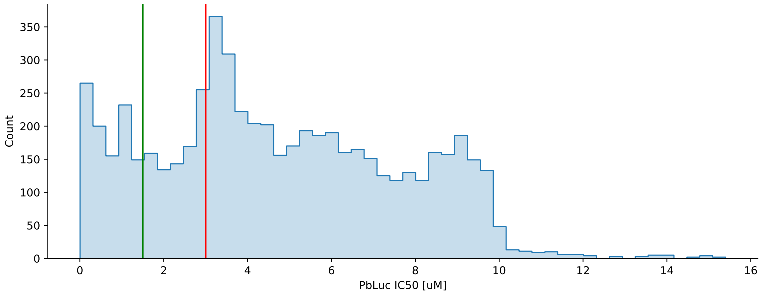


Figure 6. A histogram of the distribution of non-qualified *Pbluc* IC_{50} values. The green and red vertical lines indicate the thresholds (set to $\mathrm{IC}_{50} = 1.5\mu\mathrm{M}$ and $\mathrm{IC}_{50} = 3\mu\mathrm{M}$) that determine whether a compound is assigned to the active or inactive class.

In order to integrate information from the *HepG2tox* and *Ffluc* counter-screens and filter out problematic compounds that are likely false positives or risk confounding the evaluation in other ways, the thresholds outlined in Antonova-Koch et al. (2018) were applied. In particular, problematic compounds were discarded due to causing hepatotoxicity or assay interference if their respective IC_{50} values met at least one of the criteria outlined in Equation (A.1) and Equation (A.2)

$$\textit{HepG2tox}\ \mathrm{IC}_{50} < 2 \cdot \textit{Pbluc}\ \mathrm{IC}_{50} \wedge \textit{HepG2tox}\ \mathrm{IC}_{50} < c_{\max} \tag{A.1}$$

$$\textit{Ffluc}\ \mathrm{IC}_{50} < 2 \cdot \textit{Pbluc}\ \mathrm{IC}_{50} \wedge \textit{Ffluc}\ \mathrm{IC}_{50} < c_{\max}, \tag{A.2}$$

where $c_{\max}$ denotes the maximum concentration a compound was assayed at. These filtering criteria categorized 764 compounds as inhibiting hepatocyte viability and 446 compounds as interfering with the luminescence readout, including an overlap of 49. Removing these compounds from the dataset results in a total of 8150 compounds, of which 7301 (90%) are labeled as inactive and 849 (10%) are labeled as active.

B. Additional Experimental Details

This section provides additional implementational details of our experimental setup. Appendix B.1 presents the test set performance of each model under different data splitting and featurization techniques using the AUC-PRC and the ACE scores. Additional experimental details are provided in Appendix B.2, describing the implementation and hyperparameter screening ranges for each model in our empirical evaluation, namely logistic regression (Appendix B.2.1), random forest classifiers (Appendix B.2.2), multi-layer perceptrons (Appendix B.2.3), deep ensembles (Appendix B.2.4), pre-trained graph neural networks (Appendix B.2.5), GROVER (Appendix B.2.6), various domain adaptation and generalization techniques (Appendix B.2.7), and Q-SAVI (Appendix B.2.8). Detailed ablation plots that explore the influence of different hyperparameters on the performance of Q-SAVI are presented in Appendix B.3.

All experiments and analyses were performed in Python (Van Rossum and Drake Jr, 1995), using a range of general-purpose packages to aid model development and analysis (Harris et al., 2020; Waskom, 2021; Wes McKinney, 2010; Virtanen et al., 2020). All code that is necessary to reproduce the results presented in this work is available in the following repository: https://github.com/leojklarner/Q-SAVI.

B.1. Test Set Performances in AUC-PRC and ACE

Following model training and hyperparameter selection, the predictive accuracy and calibration of the estimated test-set label probabilities were assessed using the area under the ROC curve (AUC-ROC) and the Brier score. These metrics allow for a direct comparison of models across test sets with different label distributions (see Table 2). In addition, the performance of each algorithm was evaluated using the area under the precision-recall curve (AUC-PRC) and the adaptive calibration error (ACE; Nixon et al. (2019)). AUC-PRC and ACE are particularly well-suited for imbalanced datasets, and provide a characterization of model performance that closely aligns with AUC-ROC and BRIER SCORE (see Table 5). Note that their performance is not comparable across data splits, as it depends on the label distribution of a given test set—while the AUC-ROC of a no-skill classifier is 0.5, its AUC-PRC is given by the positive label probability $p(y = 1)$ of the test set.

Table 5. An overview of the test set performance of each model for each data splitting and featurization technique, quantified by the AUC-PRC (↑) and the ACE (↓) scores. All entries indicate the mean and standard error computed over 10 independent training runs with different random seeds. The best models within a margin of statistical significance are highlighted in bold.

	Model & Featurization	Spectral Split		Weight Split		Scaffold Split		Random Split	
		ECFP	rdkitFP	ECFP	rdkitFP	ECFP	rdkitFP	ECFP	rdkitFP
AUC-PRC (↑)	Logistic Regression	.211±.000	.140±.000	.106±.000	**.112±.000**	.211±.000	.211±.000	.248±.000	.225±.000
	Random Forest	.207±.005	.141±.002	.090±.002	.089±.003	.165±.002	.200±.002	**.292±.002**	**.294±.002**
	MLP	.208±.003	.144±.001	.090±.003	.089±.004	.180±.005	.184±.007	.270±.004	.233±.006
	Deep Ensemble	.217±.003	.144±.001	.114±.001	.102±.003	.209±.002	.209±.002	.288±.002	.266±.004
	GIN	.183±.006	.149±.005	.082±.004		.202±.005		.218±.005	
	GIN (attr masking)	.192±.003	.152±.005	.108±.003		**.245±.004**		.251±.003	
	GIN (context pred)	.188±.004	.152±.004	.098±.002		.206±.005		.254±.005	
	Grover	.199±.001	.139±.002	.106±.001		.204±.004		.227±.003	
	IRM	.154±.004	.145±.004	.086±.004		.178±.005		.176±.004	
	GroupDRO	.166±.003	.159±.002	.102±.003		.202±.003		.172±.005	
	DANN	.156±.003	.155±.005	.095±.005		.184±.004		.202±.003	
	DeepCoral	.154±.004	.151±.004	.091±.003		.194±.003		.212±.003	
	Q-SAVI	**.221±.003**	**.165±.004**	**.121±.002**	**.111±.003**	.197±.003	.216±.003	.239±.002	.208±.004
ACE (↓)	Logistic Regression	.061±.000	.055±.000	.041±.000	.034±.000	.026±.000	**.025±.000**	.018±.000	.024±.000
	Random Forest	.078±.001	**.033±.001**	.074±.001	.087±.001	.029±.001	**.025±.001**	**.016±.001**	.035±.001
	MLP	.079±.003	.052±.003	.035±.003	.055±.007	.029±.002	.044±.011	.029±.001	.026±.002
	Deep Ensemble	.078±.004	.050±.001	.025±.001	.053±.005	**.022±.001**	**.025±.001**	.023±.001	.019±.001
	GIN	.064±.004	.047±.007	.036±.003		.033±.003		.026±.003	
	GIN (attr masking)	**.053±.002**	.057±.009	.038±.002		.030±.001		.020±.001	
	GIN (context pred)	.078±.002	.051±.005	.034±.003		.028±.002		**.015±.001**	
	Grover	.074±.004	**.035±.002**	.036±.002		.038±.002		.020±.001	
	IRM	.071±.003	.067±.002	.044±.002		.035±.001		.024±.002	
	GroupDRO	.060±.003	**.035±.003**	.039±.002		.036±.002		.026±.001	
	DANN	.057±.002	.046±.003	.035±.003		.028±.001		.030±.002	
	DeepCoral	.097±.006	**.035±.002**	.041±.004		.036±.002		.026±.002	
	Q-SAVI	**.052±.001**	.043±.013	**.015±.001**	**.016±.001**	.036±0.002	**.025±.002**	.021±.001	.024±.002

B.2. Model Implementations and Hyperparameter Ranges

To ensure a fair and meaningful comparison of the evaluated machine learning models, the hyperparameters of each algorithm were independently optimized for every data split and featurization technique. The following sections provide comprehensive details about the implementation and hyperparameter ranges used for each model in our empirical evaluation.

- Logistic Regression (Section B.2.1)
- Random Forest Classifiers (Section B.2.2)
- Multi-layer Perceptrons (Section B.2.3)
- Deep Ensembles (Section B.2.4)
- Pre-trained Graph Neural Networks (Section B.2.5)
- GROVER (Section B.2.6)
- Domain Adaptation and Generalization Techniques (Section B.2.7)
- Our Probabilistic Regularization Scheme (Section B.2.8)

B.2.1. Logistic Regression

The **logistic regression models** were trained with the scikit-learn library (Pedregosa et al., 2011) using the LIBLINEAR solver (Fan et al., 2008) with a maximum of 1000 iterations and a stopping tolerance of 1×10^{-4}. They were independently fit for all hyperparameter combinations specified inTable 6, using the combination with the best unweighted validation set log-likelihood to choose the best hyperparameter setting to evaluate on the held-out test set.

Table 6. Hyperparameters for Logistic Regression

Model	Hyperparameter	Search Space
Linear Regression	regularization type	$\ell 1$, $\ell 2$
	regularization strength	1.0×10^{-4}, $2.6 \times 10^{-4} \ldots$, 3.8×10^{3}, 1.0×10^{4}
	class weight	none, balanced

B.2.2. Random Forest Classifiers

The **random forest models** were trained with the scikit-learn library (Pedregosa et al., 2011) using 100 decision trees and the GINI splitting criterion. They were independently fit for all hyperparameter combinations specified in Table 7, using the combination with the best unweighted validation set log-likelihood to choose the best hyperparameter setting to evaluate on the held-out test set.

Table 7. Hyperparameters for Random Forest Classifiers

Model	Hyperparameter	Search Space
Random Forest	maximum depth	5, 15, 26, 36, 47, 57, 68, 78, 89, 100
	min. samples per split	5, 15, 50, 100
	min. samples per leaf	1, 5, 10, 30, 100
	class weight	none, balanced

B.2.3. Multi-Layer Perceptrons

The **multi-layer perceptrons** were implemented with the PyTorch library (Paszke et al., 2019), using rectified linear units (Nair and Hinton, 2010) as activation functions. Their weights were initialized using a Normal distribution $\mathcal{N}(0, 1)$ truncated at $\pm 2\sigma$, with biases initialized at zero. These parameters were optimized on the training set using the AdamW stochastic gradient descent optimizer (Loshchilov and Hutter, 2017) with a batch size of 128 and the cross-entropy loss for a maximum of 500 epochs, using early stopping to terminate training if the unweighted log-likelihood on the validation set did not decrease for more than 10 epochs, reverting to the checkpoint with best validation set log-likelihood for evaluating their performance for hyperparameter optimization and the subsequent on the held-out test set. Batch normalization and dropout were applied after the ReLU non-lineary. The full hyperparameter search space is presented in Table 8.

Table 8. Hyperparameters for Multi-layer Perceptrons

Model	Hyperparameter	Search Space
Multi-layer Perceptron	learning rate	1×10^{-4}, 1×10^{-3}
	weight decay	1×10^{-3}, 1×10^{-2}, 1×10^{-1}
	number of layers	2, 4, 6
	embedding dimension	32, 64
	batch normalization (BN)	yes, no
	BN running statistics	yes, no
	dropout	0.0, 0.2, 0.5
	class weight	none, balanced

B.2.4. Deep Ensembles

The **deep ensembles** were trained using an identical setup to the multi-layer perceptrons, with the distinction that $M = 5$ independent networks were trained with different random seeds and evaluated with respect to their average log-likelihood on the validation set. Similarly, at inference time the class probabilities were averaged across ensembles. The full hyperparameter search space is presented in Table 9 and is identical to Table 8.

Table 9. Hyperparameters for Deep Ensembles

Model	Hyperparameter	Search Space
Deep Ensemble	learning rate	1×10^{-4}, 1×10^{-3}
	weight decay	1×10^{-3}, 1×10^{-2}, 1×10^{-1}
	number of layers	2, 4, 6
	embedding dimension	32, 64
	batch normalization (BN)	yes, no
	BN running statistics	yes, no
	dropout	0.0, 0.2, 0.5
	class weight	none, balanced

B.2.5. Pre-trained Graph Neural Networks

The graph featurization pipeline, architectures, and pre-trained initializations of the graph isomorphism networks presented in Hu et al. (2019) were retrieved from the paper's official GitHub repository and fine-tuned on the training set using the AdamW optimizer (Loshchilov and Hutter, 2017) with a batch size of 128 and the cross-entropy loss for a maximum of 500 epochs, using early stopping to terminate training if the unweighted log-likelihood on the validation set did not decrease for more than 10 epochs and reverting to the checkpoint with best validation set log-likelihood for evaluating their performance for hyperparameter optimization and the subsequent on the held-out test set. The full hyperparameter search space is presented in Table 10. The pre-trained initializations were provided for networks with 5 layers of 300 hidden units, set up using batch normalization with running statistics.

Table 10. Hyperparameters for Pre-trained GINs

Model	**Hyperparameter**	**Search Space**
Pre-trained GINs	learning rate	1×10^{-4}, 3×10^{-3}, 1×10^{-3}, 3×10^{-3}, 1×10^{-2}
	weight decay	1×10^{-3}, 1×10^{-2}, 1×10^{-1}
	dropout	0.0, 0.2, 0.5
	class weight	none, balanced

B.2.6. GROVER

All code, models, and initializations required to fine-tune the pre-trained graph transformers presented in Rong et al. (2020) was retrieved from the paper's official GitHub repository and fine-tuned on the training set with a batch size of 128 for a maximum of 500 epochs, using early stopping to terminate training if the unweighted log-likelihood on the validation set did not decrease for more than 10 epochs and reverting to the checkpoint with best validation set log-likelihood for evaluating their performance for hyperparameter optimization and the held-out test set. The hyperparameters specifying the number of layers and their embedding dimension indicate the size of the MLP fit on top of the pre-trained molecular representations produced by the GROVER base model and were chosen to be identical to the other MLP-based deep learning algorithms. The full hyperparameter search space is presented in Table 11.

Table 11. Hyperparameters for GROVER

Model	**Hyperparameter**	**Search Space**
GROVER	learning rate	1×10^{-4}, 1×10^{-3}
	weight decay	1×10^{-3}, 1×10^{-2}, 1×10^{-1}
	dropout	0.0, 0.2, 0.5
	number of layers	2, 4, 6
	embedding dimension	32, 64

B.2.7. Domain Adaptation Techniques

All code and featurization utilities required to run the evaluated domain adaptation and generalization techniques, namely invariant risk minimization (**IRM**), group-distributionally robust training (**GroupDRO**), domain-adversarial networks (**DANN**) and deep correlation alignment (**DeepCoral**), were adapted from Ji et al. (2022) and provided with data split-specific domain indicators. For this, the training set was additionally split into three domains, either using spectral clustering, molecular weight thresholds, a grouped scaffold split, or random partitions. All models used the default architecture choice in Ji et al. (2022)—a graph isomorphism network with 4 layers and 128 hidden units—and trained according to the respective optimization procedures. The full hyperparameter range is presented in Table 12.

Table 12. Hyperparameters for Domain Adaptation Techniques

Model	Hyperparameter	Search Space
IRM/GroupDRO/DANN/DeepCoral	learning rate	1×10^{-4}, 1×10^{-3}
	weight decay	1×10^{-3}, 1×10^{-2}, 1×10^{-1}
	dropout	0.0, 0.2, 0.5

B.2.8. Q-SAVI

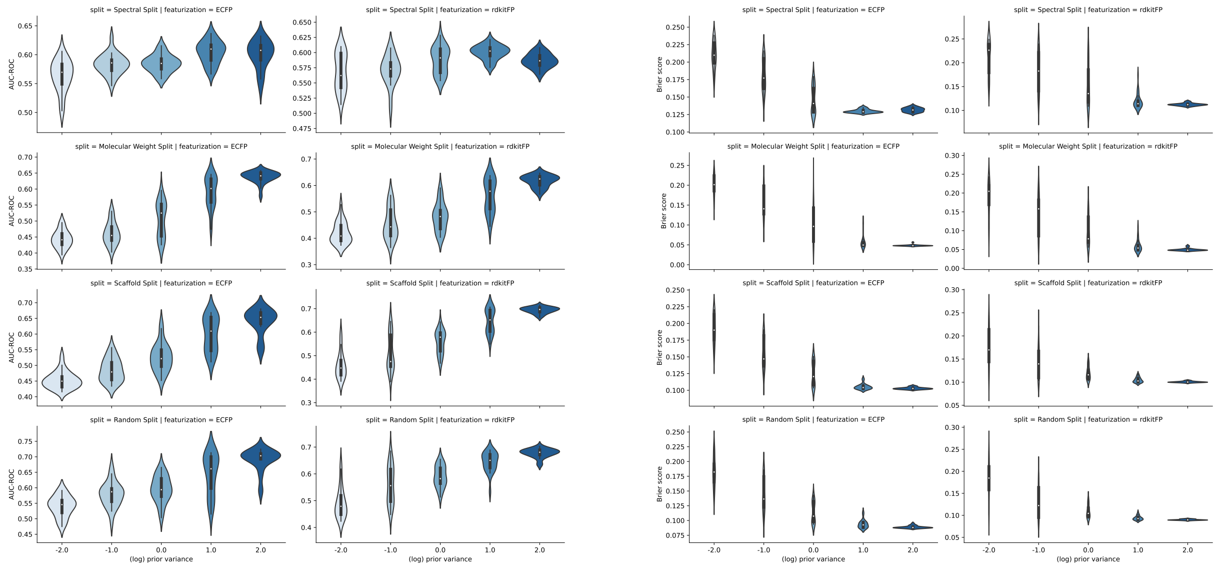
The models based on our probabilistic regularization scheme were trained using the implementation of the local linearization scheme presented in Rudner et al. (2022a;b) provided by the authors and using the exact same architecture, initialization, and optimization procedures as for the multi-layer perceptrons and deep ensembles—differing only in the objective function. Specifically, at each gradient step iteration, a sample of M molecules (where M is a hyperparameter) was drawn from a uniform distribution over the ZINC database (Irwin et al., 2020), providing a set of context points on which to evaluate the objective in Equation (9), using the Bernoulli likelihood to specify $\log p(\mathbf{y}_{\mathcal{D}} \mid f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta}))$. To construct a prior distribution over parametric function mappings $p_{f(\{\mathbf{X}, \mathbf{X}_c\}; \boldsymbol{\Theta})}$ that maximizes predictive uncertainty away from the training data, it was defined as a distribution over functions with a logit-space mean vector of approximately zero and minimal structure in the off-diagonal entries of its covariance matrix. We refer to our code repository for further implementational details. The full hyperparameter search space is presented in Table 13.

Table 13. Hyperparameters for Our Model

Model	Hyperparameter	Search Space
Q-SAVI	learning rate	1×10^{-4}, 1×10^{-3}
	number of layers	2, 4, 6
	embedding dimension	32, 64
	prior variance	1×10^{-2}, 1×10^{-1}, 1, 1×10^{1}, 1×10^{2}
	context points per sample	16, 128

B.3. Ablation Studies

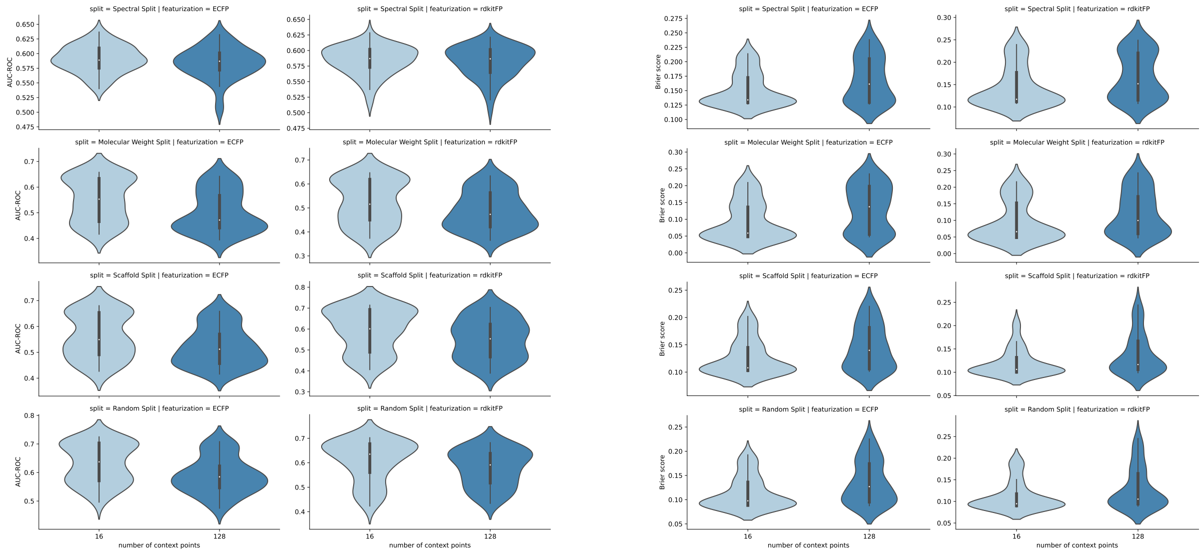
To understand the impact of different hyperparameters on the performance of our proposed probabilistic regularization scheme, we conducted a series of ablation experiments. In these experiments, we systematically varied the hyperparameters relevant to evaluating the objective in Equation (9)—namely the prior variance and the number of sampled context points—while keeping others fixed, and measured their effects on the test set AUC-ROC and BRIER SCORE. The resulting ablation plots are presented in Figures 7 and 8 and show that larger prior covariances are strongly correlated with more robust test-set performances across splits—while the effect of larger context point samples is less pronounced.



(a) Ablation plot showing the effect of the prior covariance on the test set AUC-ROC.

(b) Ablation plot showing the effect of the prior covariance on the test set BRIER SCORE.

Figure 7. Effect of (log) prior variance on the test set performance metrics.



(a) Ablation plot showing the effect of the number of context points on the test set AUC-ROC.

(b) Ablation plot showing the effect of the number of context points on the test set BRIER SCORE.

Figure 8. Effect of the number of context points on the test set performance metrics.

C. Additional Experimental Details for the Merck Molecular Activity Challenge Data

In order to assess the practical utility of our method in real-world production settings, an evaluation on the Merck Molecular Activity Challenge datasets (Ma et al., 2015) was conducted. This data consists of 15 datasets from real-world production environments with time-split training and test sets. As the compound structures are only provided in the form of anonymized atom-pair descriptors, it is not possible to use a uniform subsample of a large chemical database as a context point distribution. Instead, the evaluation focused on the three most covariate- and label-shifted datasets, see Figure 9, using a uniform distribution over molecules from the remaining datasets as a context point distribution. To select these datasets, the multiset version of the standard Jaccard/Tanimoto index

$$k_{\text{jac-multiset}}(\mathbf{x}, \mathbf{y}) = \frac{\sum_i \min(\mathbf{x}_i, \mathbf{y}_i)}{\sum_i \max(\mathbf{x}_i, \mathbf{y}_i)}$$

was used to evaluate the MMD statistic between two sets of count vectors and quantify covariate shift. Label shift between the regression targets of every training and test set was quantified through the two-sample Kolmogorov-Smirnov test statistic.

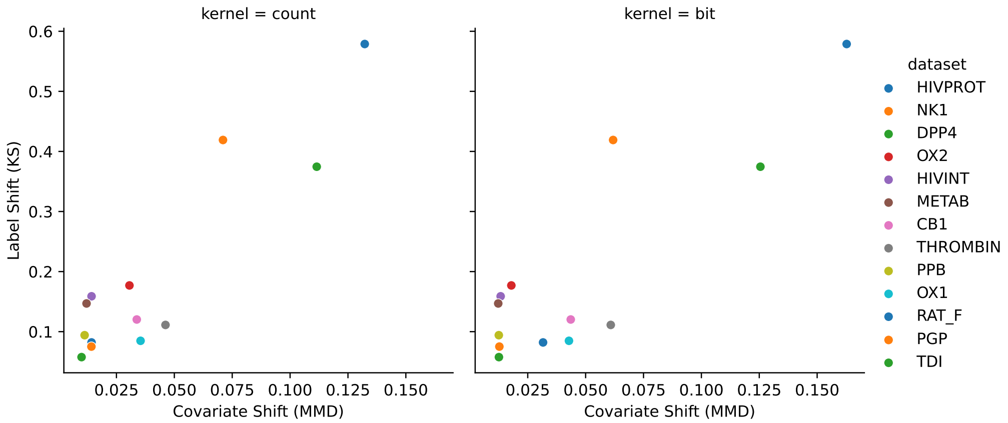


Figure 9. Scatterplot illustrating the covariate and label shifts in the Merck Molecular Activity Challenge datasets

As shown in Figure 10, the direct overlap between the HIVPROT, NK1, and DPP4 datasets with the remaining data is minimal, warranting its use as a general and diverse context point distribution. Using this evaluation setup, 10% of the training sets was randomly split off as a validation set for hyperparameter optimization and, where applicable, early stopping. Model-specific details are outlined below, including implementational details and hyperparameter ranges for regularized linear regressions (Appendix C.1), random forest regressors (Appendix C.2), and an adapted version of our probabilistic regularization scheme (Appendix C.3).

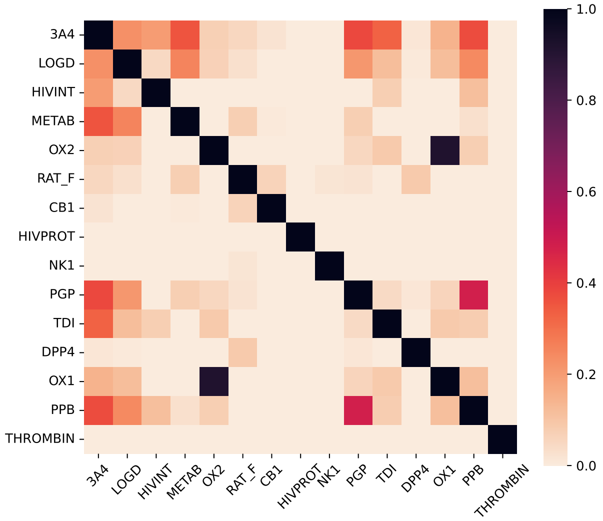


Figure 10. Heatmap illustrating the pairwise overlap between different datasets from the Merck Molecular Activity Challenge, defined as the proportion of molecules from the smaller dataset that are found in the larger dataset, i.e., $|\mathbf{X}_1 \cup \mathbf{X}_2| / \min(|\mathbf{X}_1|, |\mathbf{X}_2|)$

C.1. Regularized Linear Regression

The **regularized linear regression models** were trained using the scikit-learn library (Pedregosa et al., 2011). The LIBLINEAR solver (Fan et al., 2008) was employed with a maximum of 1,000 iterations and a stopping tolerance of 1×10^{-4}. The models were independently fitted for all specified hyperparameter combinations presented in Table 14. The combination yielding the lowest validation set mean squared error was selected to evaluate the model on the held-out test set.

Table 14. Hyperparameters for $L-1$ and L_2-Regularized Linear Regression

Model	**Hyperparameter**	**Search Space**
Linear Regression	regularization type	$\ell 1, \ell 2$
	regularization strength	100 values spaced log-linearly in $[1 \times 10^{-4}, 1 \times 10^{4}]$

C.2. Random Forest Regressors

The **random forest regression models** were trained using the scikit-learn library (Pedregosa et al., 2011). The models consisted of 100 decision trees with the GINI splitting criterion. Each model was independently fitted for all specified hyperparameter combinations shown in Table 15. The combination with the lowest validation set mean squared error was selected to evaluate the model on the held-out test set.

Table 15. Hyperparameters for Random Forest Regressor

Model	**Hyperparameter**	**Search Space**
Random Forest	maximum depth	50 values spaced linearly in [5, 500]
	min. samples per split	5, 15, 50, 100
	min. samples per leaf	1, 5, 10, 30, 100

C.3. Q-SAVI

The regression variant of our probabilistic regularization scheme was set up identically to the classification variant described in Appendix B.2.8, the only difference being the likelihood function used to evaluate Equation (9). Instead of specifying $\log p(\mathbf{y}_{\mathcal{D}} \,|\, f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta}))$ as a Bernoulli likelihood, a homoscedastic multivariate Normal likelihood with a unit diagonal covariance matrix was used. While a more expressive approach of either optimizing the covariance as a hyperparameter or letting the network predict point-wise means and variances to use in combination with a heteroscedastic likelihood function is possible, this straightforward method was found to be sufficient in this context. The full hyperparameter search space is presented in Table 16 and is identical to that in Table 13.

Table 16. Hyperparameters for Our Model

Model	**Hyperparameter**	**Search Space**
Ours	learning rate	1×10^{-4}, 1×10^{-3}
	number of layers	2, 4, 6
	embedding dimension	32, 64
	prior variance	1×10^{-2}, 1×10^{-1}, 1, 1×10^{1}, 1×10^{2}
	context points per sample	16, 128