DiscoBAX: Discovery of Optimal Intervention Sets in Genomic Experiment Design

Clare Lyle [*1 2] **Arash Mehrjou** [*†3] **Pascal Notin** [*1]
Andrew Jesson [1] **Stefan Bauer** [4 5] **Yarin Gal** [1] **Patrick Schwab** [†3]

Abstract

The discovery of therapeutics to treat genetically-driven pathologies relies on identifying genes involved in the underlying disease mechanisms. Existing approaches search over the billions of potential interventions to maximize the expected influence on the target phenotype. However, to reduce the risk of failure in future stages of trials, practical experiment design aims to find a set of interventions that maximally change a target phenotype via diverse mechanisms. We propose DiscoBAX, a sample-efficient method for maximizing the rate of significant discoveries per experiment while simultaneously probing for a wide range of diverse mechanisms during a genomic experiment campaign. We provide theoretical guarantees of approximate optimality under standard assumptions, and conduct a comprehensive experimental evaluation covering both synthetic as well as real-world experimental design tasks. DiscoBAX outperforms existing state-of-the-art methods for experimental design, selecting effective and diverse perturbations in biological systems.

1. Introduction

Genomic experiments probing the function of genes under realistic cellular conditions are the cornerstone of modern early-stage drug target discovery and validation; moreover, they are used to identify effective modulators of one or more disease-relevant cellular processes. These experiments, for example using Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR) (Jehuda et al., 2018) perturbations, are both time and resource-intensive (Dickson & Gagnon, 2004; 2009; DiMasi et al., 2016; Berdigaliyev & Aljofan, 2020). Therefore, an exhaustive search of the billions of potential experimental protocols covering all possible experimental conditions, cell states, cell types, and perturbations (Trapnell, 2015; Hasin et al., 2017; Worzfeld et al., 2017; Chappell et al., 2018; MacLean et al., 2018; Chappell et al., 2018) is infeasible even for the world's largest biomedical research institutes.

To mitigate the chances of failure in subsequent stages of the drug design pipeline, it is desirable for the subset of precursors selected in the target identification stage to operate on diverse underlying biological mechanisms (Nica et al., 2022). That way, if a promising candidate based on in-vitro experiments triggers undesirable outcomes when tested in-vivo (e.g., unexpected side effects), other lead precursors relying on different pathways might be suitable replacements that are not subject to the same issues. This two-phase maximization problem diverges from standard formulations of Bayesian optimization or active learning. In particular, the noisy measurements obtained by the experimenter don't correspond to the objective of interest, but are only correlated with this outcome via some unknown mechanism. Thus even in the limit of infinite intermediate phenotype measurements, it is not possible to identify the maximum of the objective function.

Our first contribution is formalizing this problem in order to identify properties of an optimal solution. Mathematically, finding a diverse set of precursors corresponds to identifying and sampling from the different modes of the black-box objective function mapping intervention representations to the corresponding effects on the disease phenotype (§ 2). Existing machine learning methods for iterative experimental design (e.g., active learning, Bayesian optimization) have the potential to aid in efficiently exploring this vast biological intervention space. However, to our knowledge, there is no method geared toward identifying the modes of the underlying black-box objective function to identify candidate interventions that are both effective and diverse (§ 6).

To this end, we introduce DiscoBAX - a sample-efficient

[*]Equal contribution, [†] Senior authorship [1]University of Oxford [2]Google DeepMind [3]GlaxoSmithKline [4]Helmholtz AI [5]Technical University of Munich. Correspondence to: Clare Lyle <clarelyle@deepmind.com>, Arash Mehrjou <arash@distantvantagepoint.com>, Pascal Notin <pascal.notin@cs.ox.ac.uk>.

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

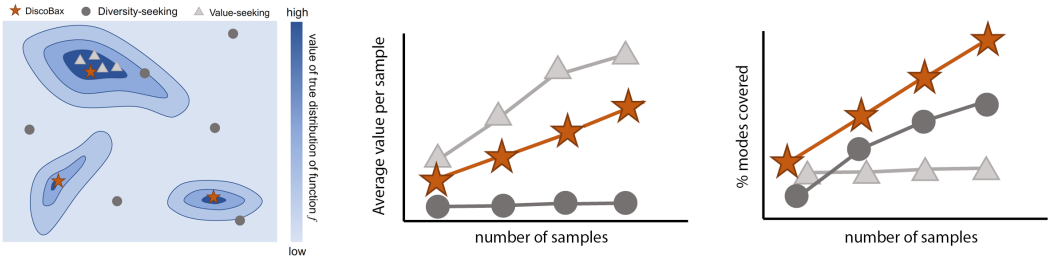

Figure 1. We compare DiscoBAX (orange star) to existing diversity-seeking (dark grey circle) and value-seeking (light grey triangle) batch active learning policies. DiscoBAX aims to recover a maximally diverse set of interventions with values above a pre-defined threshold from a given underlying distribution. This aim contrasts with value-seeking strategies focusing on maximizing value and diversity-seeking strategies focusing on maximizing coverage. We expect DiscoBAX to design genomic experiments yielding high value findings that maximize mode coverage. As discussed in § 1, the diversity of selected interventions is highly desirable to increase the chances that at least some of these interventions will succeed in subsequent stages of the drug discovery pipeline.

Bayesian Algorithm eXecution (BAX) method for discovering genomic intervention sets with both high expected change in the target phenotype and high diversity to maximize chances of success in the following stages of drug development (Figure 1), which we formalize as set-valued maximization problem (Equation 4). After providing theoretical guarantees on the approximate optimality of the presented approach under standard conditions, we perform a comprehensive experimental evaluation in both synthetic and real-world datasets. These experiments show that DiscoBAX outperforms existing state-of-the-art active learning and Bayesian optimization methods in designing genomic experiments that maximize the yield of findings that could lead to the discovery of new potentially treatable disease mechanisms. The implementation of DiscoBAX and the code to reproduce the experimental results are publicly available in `https://github.com/amehrjou/DiscoBAX`.

Our contributions are as follows:

- We give a formalization of the gene target identification problem (§ 3) and discuss limitations of existing methods in addressing this problem (§ 6).
- We develop DiscoBAX - a sample-efficient BAX method for maximizing the rate of significant discoveries per experiment while simultaneously probing for a wide range of diverse mechanisms during a genomic experiment campaign (§ 4).
- Leveraging insights from the mathematical structure of our formalization, we provide theoretical guarantees that substantiate the optimality properties of DiscoBAX (§ 4 and Appendix C).
- We conduct a comprehensive experimental evaluation covering both synthetic as well as real-world experimental design tasks that demonstrate that DiscoBAX outperforms existing state-of-the-art methods for experimental design in this setting (§ 5).

2. Background and Notation

Genomic experimentation is an early stage in drug discovery where geneticists assess the effect of genomic interventions on moving a set of disease-relevant phenotypes to determine suitable drug targets.

To formalize this process, we assume a black-box function, $f : \mathcal{G} \to \mathbb{R}$, that maps each gene, $\mathrm{g} \in \mathcal{G}$, to the value, $f(\mathrm{g})$, corresponding to the magnitude of phenotypic change under gene knock out. The set, $\mathcal{G}$, is finite, $|\mathcal{G}| = m < \infty$, because there are a limited number of protein-encoding genes in the human genome ($\approx 20,000$) (Pertea et al., 2018), and can be represented by either the set of integers or one-hot vectors with dimension m. However, biologically informed embeddings, $\mathbf{X} : \mathcal{G} \to \mathcal{X}$, are often preferred to represent genes for their potential to capture genetic, functional relationships. We assume that gene embeddings, $\mathbf{X}(g) = \mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$, are sets of d-dimensional real vectors, with m distinct members, $|\mathcal{X}| = m$, thus, we use $f(\mathrm{g})$ and $f(\mathbf{x})$ interchangeably, where $\mathbf{x}$ is the embedding of the gene g.

In drug development, a candidate target must meet several criteria to proceed to subsequent stages in the development pipeline. For example, engaging the target – down- or up-regulating the gene – must move the phenotype *significantly* in the desired direction. Such genes are called "top-movers" of the phenotype. We can define the K top-movers for a given phenotype as members of the set, $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$, corresponding to the K largest values of $\{f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_m)\}$. However, each evaluation of the phenotype change, f, requires a CRISPR-Cas9 knockout experiment in the lab, which makes exhaustive experimentation infeasible even for the most resourceful institutions. Hence in practice, the experimentation budget is limited to $T \ll m$ experiments. Instead of choosing the K

top-movers (requiring phenotype change knowledge, $f(\mathbf{x})$, for all inputs $\mathbf{x} \in \mathcal{X}$), a more practical approach is to form the subset, $\mathcal{X}_c \subseteq \mathcal{X}$, of genes that when knocked out lead to a change in the phenotype, $f(\mathbf{x})$, larger than a selected threshold value, c, i.e. $\mathcal{X}_c \coloneqq \{\mathbf{x} \in \mathcal{X} : f(\mathbf{x}) \geq c\}$.

A critical aspect distinguishing the drug discovery pipeline from the standard Bayesian Optimization setting is that we do not seek to identify a single point which maximizes the unknown function f; rather, we wish to identify a computable property of f using a limited number of evaluations of f. To do so we will leverage Bayesian Algorithm Execution (BAX), proposed by Neiswanger et al. (2021), which is designed precisely to identify the output, $O_{\mathcal{A}} \coloneqq O_{\mathcal{A}}(f)$, of an algorithm, $\mathcal{A}$, run on a function, f, by evaluating the function on a budgeted set of inputs, $\{\mathbf{x}_i\}_{i=1}^{T} \in \mathcal{X}$. Estimating a computable property, i.e. the output of the algorithm $\mathcal{A}$, is done by positing a probabilistic model for f for estimating $O_{\mathcal{A}}$. Data is acquired by searching for the value $\mathbf{x} \in \mathcal{X}$ that maximizes the mutual information, $I(\mathrm{Y}_{\mathbf{x}}; O_{\mathcal{A}} \mid \mathcal{D}_t)$, between the function output, $\mathrm{Y}_{\mathbf{x}}$, and the algorithm output, $O_{\mathcal{A}}$. BAX assumes that functional output instances, $\mathrm{y}_{\mathbf{x}}$, of the function, f, can be observed for each acquired $\mathbf{x}$. The acquisition of data is sequential, where the information gain maximization procedure leads to a dataset of observations, $\mathcal{D}_t \coloneqq \{(\mathbf{x}_i, \mathrm{y}_{\mathbf{x}_i})\}_{i=1}^{t-1}$, at step $t \in [T]$. BAX can be used in conjunction with a number of algorithms, such as determining the superlevel set (i.e. $\mathcal{X}_c$), computing integrals, or finding local optima of f. Given that genomic experimentation seeks to find a diverse set of genes corresponding to the modes of f, the BAX framework is well suited to our task.

Concretely, BAX acquisition functions select points by maximizing the expected information gain (EIG) obtained from each point about the output of the algorithm. Crucial to the applicability of BAX to our problem setting is the tractability of accurate approximators of the EIG for algorithms which, like the one we will propose, return a subset of their inputs. The exact computation of the EIG for arbitrary algorithms is not generally tractable; however, Neiswanger et al. (2021) present an approximation that only requires the computation of the entropy of the distribution over function values conditioned on algorithm outputs.

$$\mathrm{EIG}_t^v(\mathbf{x}, \mathcal{D}_t) = H(f_{\mathrm{ip}}(\mathbf{x}) | \mathcal{D}_t) - \mathbb{E}_{p(S|\mathcal{D}_t)}[H(f_{\mathrm{ip}}(\mathbf{x}) | S, \mathcal{D}_t)]. \tag{1}$$

When the model P is a Gaussian Process (GP), both quantities are straightforward to compute: the first is the entropy of the GP's predictive distribution at $\mathbf{x}$, and we can estimate the second by conditioning a posterior on the values of elements in the set S. Monte Carlo approximation of this quantity is possible when the model P does not permit a closed form.

3. Problem Setting

A primary challenge in the drug discovery pipeline is the discrepancy in outcomes between *in vitro* experimental data and *in vivo* diseases. Where *in vitro* experimental data can quantify the effect of a gene knockout on a specific aspect of a cellular phenotype in a petri dish, *in vivo* interactions between the drug and the organism may lead to weaker effect sizes or toxicity. The drug discovery pipeline consists of stages, starting by testing a set of candidate interventions and then proceeding by selecting a subset of promising candidates to pass on for further development. For example, one might test a broad range of gene knockouts on cell cultures and then select a subset of promising gene candidates to evaluate in animal models. These trials can be expensive, so it is desirable to weed out potentially ineffective or toxic candidates before this phase. To do so, researchers can leverage heuristic score functions that predict the "drug-likeness" or likelihood of toxicity of a compound (Jiménez-Luna et al., 2020). Considering a diverse set of candidate interventions, where each intervention applies to a different mechanism in the disease phenotype, is also of use as it increases the likelihood of at least one candidate succeeding in the subsequent phase.

We formalize this setting as an optimization problem in which the optimizer has access to a measurement which is correlated with the outcome of interest; however, some assumed noise model distorts this quantity before yielding the primary objective function value. We formalize our search space (i.e., the set of available genes, though in principle this could be any set) $\mathcal{G} = \{\mathrm{g}_1, \ldots, \mathrm{g}_m\}$, for which we have some phenotype measurement f_{ip}. We will primarily refer to f_{ip} as a function from *features* to phenotype changes, but it is equivalent to expressing f_{ip} as a function on genes $\mathcal{G}$. The subscript 'ip' stands for *intermediate phenotype* as it is not the actual clinical measurement caused by the gene knockout. Instead, it is a measurement known to correlate with a disease pathology and is tractable in the lab setting (see Appendix A for detailed formalization). In this paper, we will assume the phenotype change is a real number $f_{\mathrm{ip}}(\mathbf{x}) \in \mathbb{R}$; however, given suitable modeling assumptions, it is possible to extend our approach to vector-valued phenotype readouts. We also define a function called the *disease outcome*, f_{out}, which is composed of f_{ip} and factors outside the biological pathway, such as toxicity of a molecule that engages with a target gene. The noise component, η, encapsulates all these extra factors.

In practice, η will depend on the nature of the biological systems under consideration, and could take on a variety of forms of varying degrees of complexity. Here, we illustrate two tractable formulations of the relationship between the disease outcome, f_{out}, and the *in vitro* phenotype, f_{ip}.

1. **Multiplicative Bernoulli noise:**

$$f_{\text{out}}(\mathbf{x};\eta) = f_{\text{ip}}(\mathbf{x})\eta(\mathbf{x}) \tag{2}$$

where $\eta(\mathbf{x}) \in \{0,1\}, \forall \mathbf{x} \in \mathcal{G}$, and η is sampled from a Gaussian process classification model. This setting presents a simplified model of drug toxicity: η corresponds to a binary indicator of whether or not the drug is revealed to exhibit unwanted side effects in future trials. The multiplicative noise model assumes that the downstream performance of an intervention is monotone with respect to its effect on the phenotype, conditional on the compound not exhibiting toxicity in future trials. In our experiments, we assume η exhibits correlation structure over inputs corresponding to a GP classification model, and construct the kernel $K_{\mathcal{X}}$ of this GP to depend on some notion of distance in the embedding space $\mathcal{X}$.

2. **Additive Gaussian noise:**

$$f_{\text{out}}(\mathbf{x};\eta) = f_{\text{ip}}(\mathbf{x}) + \eta(\mathbf{x}) \quad \eta \sim \text{GP}(\mathbf{0}, K_{\mathcal{X}}) \tag{3}$$

where $\eta : \mathcal{G} \to \mathbb{R}$ is drawn from a Gaussian process model with kernel $K_{\mathcal{X}}$. In this case, we assume that the unforeseen effects of the input $\mathbf{x}$ are sufficiently numerous to resemble a Gaussian perturbation of the measured in vitro phenotype $f_{\text{ip}}(\mathbf{x})$.

Notice that in the above models, noise is an umbrella term for everything that affects the fitness of a target but is not part of the biological pathway from the gene to the phenotype change. Therefore, the choice of noise distribution and how it affects the outcome is a modelling assumption that is intended to capture coarse inductive biases known to the researcher. We additionally seek out a *set* of interventions $S \subset \mathcal{G}$ of some fixed size $|S| = k$ whose elements cause the maximum expected change (for some noise distribution) in the disease outcome. In other words, we seek an intervention that best moves the disease phenotype, which will be the best candidate drug. This goal is distinct from either sampling the super-level-sets of f_{ip} or finding the set S with the best average performance. Instead, we explicitly seek to identify a set of points whose toxicity or unintended side effects will be minimally correlated, maximizing the odds that at least one will succeed in the subsequent trials. We thus obtain a set-valued maximization problem

$$\max_{S \subseteq \mathcal{X}} \mathbb{E}_{\eta}\left[\max_{\mathbf{x} \in S} f_{\text{out}}(\mathbf{x};\eta)\right]. \tag{4}$$

This compact formula is critical to attain our overarching objective: identifying interventions with both a large impact on the phenotype of interest and with high diversity to increase the chance of success of some of them in the subsequent steps of the drug discovery pipeline. An illustrative example is provided in Figure 4 in Appendix B to provide further intuition into this formula.

The general formulation of this problem is NP-hard (Goel et al., 2010); therefore, we propose a tractable algorithm that provides a constant-factor approximation of the optimal solution by leveraging the submodular structure of the objective under suitable modeling assumptions. Given such an algorithm, our task is the active learning problem of optimally querying the function, f_{ip}, given a limited number of trials, T, to accurately estimate the algorithm's output on the ground-truth dataset.

Importantly, this formulation allows us to decouple modeling the measured phenotype, f_{ip}, from modeling the noise η. For example, we might make the modeling assumption that we sample f_{ip} from a GP with some kernel k_1 and that η is a Bernoulli random variable indicating the safety of the compound.

4. Method

Various methods exist for efficiently optimizing black-box functions; however, our problem setting violates several assumptions underlying these approaches. In particular, while we assume access to intermediate readouts f_{ip}, the actual optimization target of interest f_{out} is not observable. Further, we seek to find a *set* of interventions that maximize its expected value under some modeling assumptions. These two properties render a broad range of prior art inapplicable. Active sampling methods do not prioritize high-value regions of the input space. Bayesian optimization methods assume access to the ground-truth function outputs (or a noisy observation thereof). And Bayesian algorithm execution approaches based on level-set sampling may not sufficiently decorrelate the hidden noise in the outcome.

We propose an intervention set selection algorithm in a Bayesian algorithm execution procedure that leverages the modeling assumptions characterized in the previous section. This method, Subset Discovery via Bayesian Algorithm Execution (DiscoBAX), consists of two distinct parts. (1) a subset-selection algorithm obtaining a $1 - 1/e$-factor approximation of the set that maximizes equation 3, and (2) an outer BAX loop that queries the phenotype readings to maximize the information gain about the output of this algorithm. In Section 4.1, we present the idealized form of DiscoBAX and show that it attains an approximately optimal solution. Our approach is easily adaptable to incorporate approximate posterior sampling methods, enabling its use with deep neural networks on high-dimensional datasets. We outline this practical implementation in Section 4.2.

4.1. Algorithm

Subset maximization: we first address the problem of identifying a subset $S \subset \mathcal{X}$ such that $|S| = k$ which maximizes the value $\mathbb{E}_\eta[\max_{\mathbf{x}\in S} f_{\text{out}}(\mathbf{x};\eta)]$ As mentioned previously, the exact maximization of this objective is intractable. To construct a tractable approximation, we propose a submodular surrogate objective, under which the value of an intervention is lower-bounded by zero $f^*_{\text{out}}(\mathbf{x};\eta) = \max(f_{\text{out}}(\mathbf{x};\eta), 0)$. This choice is motivated by the intuition that any intervention with a negative expected value on the phenotype is equally useless as it will not be considered in later experiment iterations, and so we do not need to distinguish between harmful interventions. The resulting function $f(S) = \mathbb{E}_\eta[\max_{\mathbf{x}\in S} f^*_{\text{out}}(\mathbf{x};\eta)]$ will be submodular, and thus Algorithm 1, the greedy algorithm, will provide a $1-1/e$ approximation of the optimal solution (Nemhauser et al., 1978).

Observation 1. *The score function $f : \mathcal{P}(\mathcal{G}) \to \mathbb{R}$ defined by*

$$f(S) = \mathbb{E}_\eta\left[\max_{\mathbf{x}\in S}\left(\max(0, f_{out}(\mathbf{x};\eta)\right)\right] \tag{5}$$

is non-negative, monotone, and submodular.

We provide proof of this result in Appendix C. In practice, we can estimate the expected value in this objective using Monte Carlo (MC) samples over the noise distribution η. Where MC sampling is too expensive, a heuristic that uses a threshold to remove points whose values under η are too highly correlated can also obtain comparable results with a reduced computational burden.

Algorithm 1 SubsetSelect

Require: integer $k > 0$, set $\mathcal{X}$, noise distribution $P(\eta)$, sampled readouts $\widehat{f}_{\text{ip}} : \mathcal{X} \to \mathbb{R}$
  $S \leftarrow \emptyset$
  if multiplicative noise **then**
    $\widehat{f}_{\text{out}}(\mathbf{x};\eta) := \widehat{f}_{\text{ip}}(\mathbf{x})\eta(\mathbf{x})$
  end if
  if additive noise **then**
    $\widehat{f}_{\text{out}}(\mathbf{x};\eta) := \widehat{f}_{\text{ip}}(\mathbf{x}) + \eta(\mathbf{x})$
  end if
  for $i < k$ **do**

$$S \leftarrow S \cup \{\arg\max_{x\in\mathcal{X}\setminus S} \mathbb{E}_\eta[\max_{y\in S\cup\{x\}} \widehat{f}_{\text{out}}(\mathbf{x};\eta)]\}$$

  end for
output S

Algorithm 2 DiscoBAX

Require: finite sample set $\mathcal{X}$, budget T, Monte Carlo parameter $\ell \in \mathbb{N}$
  $\mathcal{D} \leftarrow \emptyset$
  for $i < T$ **do**
    sample $\{\widehat{f}_{\text{ip}}\}_{j=1}^{\ell} \sim P(f_{\text{ip}}|\mathcal{D})$
    $S_j \leftarrow \text{SubsetSelect}(\widehat{f_{\text{ip},j}}), \forall j = 1, \ldots, \ell$
    $\mathbf{x}_i \leftarrow \arg\max_{\mathbf{x}\in\mathcal{X}} \text{EIG}^v(\mathbf{x}, S^\ell_{j=1})$
    query $f_{\text{ip}}(\mathbf{x}_i)$
    $\mathcal{D} = \mathcal{D} \cup \{(\mathbf{x}_i, f_{\text{ip}}(\mathbf{x}_i)\}$
  end for
output $\mathcal{D}$

Active sampling: because we do not assume prior knowledge of the phenotype function f_{ip}, we require a means of selecting potential interventions for querying its value at a specified input $\mathbf{x}$. In practice, running these experiments may incur a cost, and so it is desirable to minimize the number of queries necessary to obtain an accurate estimate of the optimal intervention set. BAX (Neiswanger et al., 2021) presents an effective active sampling approach to approximate the output of an algorithm using a minimal number of queries to the dataset of interest. In our setting, this allows us to approximate the output of Algorithm 1 over the set $(\mathcal{X}, f_{\text{ip}}(\mathcal{X}))$ without incurring the cost of evaluating the effect of every knockout intervention in $\mathcal{G}$. Concretely, this procedure takes as input some probabilistic model P which defines a distribution over phenotype readings f_{ip} conditioned on the data $\mathcal{D}_t$ seen so far and from which it is possible to draw samples. We consider two noise models in Algorithm 1, but the algorithm can be extended to arbitrary noise models by setting $\widehat{f}_{\text{out}}(x;\eta)$ to be a suitable function of the noise η.

A remark on the efficiency of subset maximization & active sampling. We emphasize that subset selection is a function called within each active sampling cycle. Hence, the above observation about submodularity refers specifically to Algorithm 1 rather than its incorporation in Algorithm 2; without access to the ground truth value of f, it is not possible to provide deterministic guarantees on the output of the algorithm. If sample efficiency is not a concern, Algorithm 1 can be run on the set of all inputs to obtain the $(1-1/e)$-optimal solution.

We outline this procedure in Algorithm 2, and refer to Section 2 for additional details. In the batch acquisition setting, we form batches of size B at each cycle by selecting the B points with the highest EIG values.

4.2. Practical implementation in high dimensions

When working with high-dimensional input features, we typically leverage Bayesian Neural Networks in lieu of Gaus-

sian Processes. We sample from the parameter distribution via Monte Carlo dropout (MCD) (Gal & Ghahramani, 2016), and rely on Monte Carlo simulation to estimate the quantities introduced in Algorithm 2. In particular, the entropy of the posterior distribution is obtained as follows:

$$H(\mathrm{y}_{\mathbf{x}}|\mathcal{D}_t) = \mathbb{E}_{p(\mathrm{y}_{\mathbf{x}}|\mathcal{D}_t)}\left[\log p(\mathrm{y}_{\mathbf{x}}|\mathcal{D}_t)\right] \sim \frac{1}{M}\sum_{s=1}^{M} \log\ p(y_x^s|\mathcal{D}_t, f_s) \tag{6}$$

where the samples $\{y_x^s = f_s(x)\}_{i=1}^M$ are obtained by sampling from the distribution over model parameters with MCD to obtain the parameter samples $\{f_s\}_{i=1}^M$.

Remarks on optimality– Notice that the theoretical guarantees of optimality are provided for the inner loop (Algorithm 1) of DiscoBAX. The sample-efficiency of the entire algorithm is supported by empirical evidence from a wide range of synthetic and real-world experiments. For further empirical analysis of the sample-efficiency of the BAX procedure, we refer to Neiswanger et al. (2021).

5. Experiments

In the experimental evaluation of DiscoBAX, we specifically seek to answer the following questions: 1) Does DiscoBAX allow us to reach a better trade-off between recovery of the top interventions and their diversity (Tables 1 and 2 to 6)? 2) Is the method sample-efficient, i.e., identifies global optima in fewer experiments relative to random sampling or naive optimization baselines (Figure 3 and 7)? 3) Is the performance of DiscoBAX sensitive to various hyperparameter choices (Appendix D.3)? To address these questions, we first focus on experiments involving synthetic datasets (§ 5.1) in which we know the underlying ground truth objective function. We then conduct experiments across several large-scale experimental assays from the GeneDisco benchmark (Mehrjou et al., 2021) that cover a diverse set of disease phenotypes.

5.1. Synthetic Dataset

We begin with a concrete example to illustrate the distinction between the behavior DiscoBAX and existing methods. The dataset we consider is a one-dimensional regression task on a mixture-of-Gaussians density function f_{mog}. We construct f_{mog} such that it exhibits several local optima at a variety of values, necessitating a careful trade-off between exploration and exploitation to optimize the DiscoBAX objective. Crucially, exploitation in this setting requires not only an accurate estimation of the global optimum but also an accurate estimation of the local optima. We provide evaluations on additional datasets in Appendix D.1.1. We consider the following baseline acquisition functions which select the optimal point $\mathbf{x}^*$ to query at each iteration, letting $\mu(\mathbf{x})$ denote the posterior mean over $f_{\mathrm{ip}}(\mathbf{x})$ and $\sigma^2(\mathbf{x})$ its variance. We evaluate random sampling, a UCB-like acquisition function, BAX on super-level set and top-k algorithms, Thompson sampling, and uncertainty maximization baselines. Full details are provided in Appendix D.1.1.

In Figure 2, we visualize the solutions found by each approach after 30 iterations. We further evaluate the score of each method, computed as $\mathbb{E}_\eta \max_{\mathbf{x}\in S} f_{\mathrm{ip}}(\mathbf{x})\eta(\mathbf{x})$, where η is drawn from a Bernoulli distribution whose logits are determined by an affine transformation of a sample from a GP with zero mean and radial basis function covariance kernel. This construction ensures a high correlation between the values of nearby inputs and reward sets S whose elements are distant from each other. To select S, we use the learned posterior mean μ from each acquisition strategy as input to Algorithm 1 and set S to be equal to its output. We observe that most baselines over-exploit the high-value local optima, leading to inaccuracies on the lower optima: Algorithm 1 is unable to select the optimal subset elements from the lower-value modes and the model score suffers. The advantage of DiscoBAX is better visible by looking at the 5 minor modes on the right side of the function. DiscoBAX has distributed its experimental budget almost evenly among those modes and has discovered all of them while the other methods either miss some of the modes (no violet star under a mode) or waste extra budget on some modes (more than one violet star under a mode.)

5.2. GeneDisco Datasets

Datasets & baselines. The GeneDisco benchmark (Mehrjou et al., 2021) is comprised of five large-scale genome-wide CRISPR assays and compares the relative strengths of nine active learning algorithms (eg., Margin sampling, Coreset) for optimal experimental design. The objective of the different methods is to select the set of interventions (ie., genetic knockouts) with the largest impact on the corresponding disease phenotype. We include all baselines from the GeneDisco benchmark, as well as eight additional approaches: UCB, qUCB, qEI, qPOI, Thompson sampling, Top-K BAX, Levelset BAX, and DiscoBAX.

Metrics & approach. We define the set of optimal interventions as those in the top percentile of the experimentally-measured phenotype (referred to as 'Top-K interventions'). We use Top-K recall to assess the ability of a method to identify the best interventions. To quantify the diversity across the set of optimal interventions, we first cluster interventions in a lower-dimensional subspace (details provided in Appendix D.2.1). We then measure the proportion of clusters that are recalled (i.e., any of its members are selected) by a given algorithm over the different experiment cycles. The geometric mean between Top-K recall and the diversity metric defines the overall score for a method. For

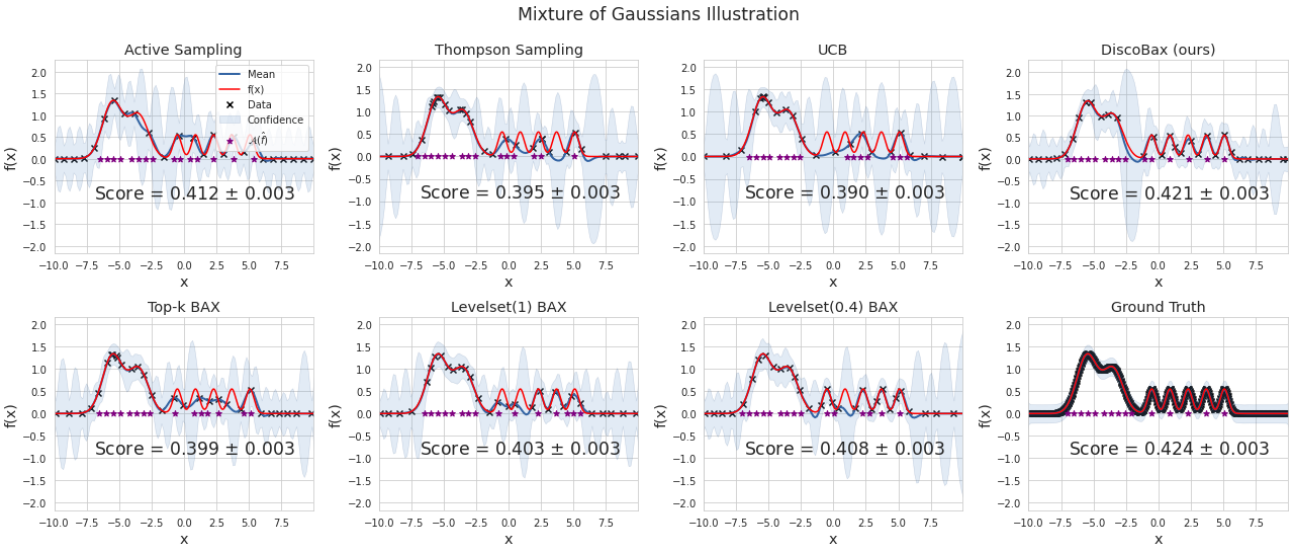

Figure 2. Illustration of failure modes of benchmark acquisition functions in our problem setting: existing methods struggle to accurately capture both the high- and low-valued local optima. We use a batch size equal to one for all methods.

all methods and datasets, we perform 25 consecutive batch acquisition cycles (with batch size 32). All experiments are repeated 10 times with different random seeds.

Results & discussion. We observe that, across the different datasets, DiscoBAX enables to identify a more diverse set of optimal interventions relative to baselines (Table 1). It does so in a sample-efficient manner as it achieves higher diversity throughout the different acquisition cycles (Fig.3). Note that sample-efficiency is an empirical observation here not a theoretical property of the algorithm since it is possible to construct adversarial datasets where a BAX method will attain no better performance than random sampling. Interestingly, it tends to recall a higher share of optimal interventions on several assays as well, which may be the result of very steep extrema in the corresponding datasets. We also find the performance of DiscoBAX to be relatively insensitive to the choice of hyperparameters (Appendix D.3). Lastly, we note that when the input feature space (ie., the intervention representation) does not correlate much with the disease phenotype of interest, the model being learned tends to perform poorly and we observe no lift between the different methods and random sampling (eg., the SARS-CoV-2 assay from (Zhu et al., 2021) – see Appendix D.2.2).

6. Related work

Prior works have studied the application of genomic discovery and method development for diverse target generation. While sharing a philosophical connection to our contribution, the mathematical formalization of the problems these methods seek to solve exhibit subtle but important distinctions from the formulation presented in this paper.

Bayesian optimization Bayesian optimization (BO) is concerned with finding the global optimum of a function with the fewest number of function evaluations (Snoek et al., 2012; Shahriari et al., 2015). Since this target function is often expensive-to-evaluate, one typically uses a Gaussian process as a surrogate function (Srinivas et al.). The candidates for function evaluation are then determined through a so-called acquisition function, which is often expressed as the expected utility over the surrogate model. Typical utility functions include the expected improvement (Močkus, 1975, EI) and probability of improvement (Kushner, 1964, PI). Recent work includes variational approaches Song et al. (2022) which yield a tractable acquisition function whose limiting behavior is equivalent to PI. Bayesian optimization has been applied to biological problem settings such as small molecule optimization (Griffiths & Hernández-Lobato, 2017; Korovina et al., 2020; Notin et al., 2021) or protein design (Moss et al., 2020). While BO bears some resemblance to our problem formulation, these methods cannot naively be applied to our setting as the noisy function observations we receive are not noisy samples of the objective function we seek to maximize.

Active learning While many probabilistic models like Gaussian processes provide principled uncertainty estimates (Rasmussen, 2003), modern neural network architectures often rely on heuristics or only provide approximations approaches (Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017). Active learning based approaches use the uncertainty estimates for maximizing expected information

Table 1. **Performance comparison on GeneDisco CRISPR assays** We report the aggregated performance of DiscoBAX and other methods on all assays from the GeneDisco benchmark. All other baselines and the breakdown per assay are provided in Appendix D.2.2.

Method	Category	Top-K recall	Diversity score	Overall score
Random	-	29.3% (1.4%)	4.9% (0.3%)	12.0% (0.6%)
Thompson Sampling	Bandits	27.5% (1.5%)	4.8% (0.4%)	11.5% (0.7%)
UCB	Bayesian Optim.	33.5% (2.0%)	5.9% (0.5%)	14.1% (1.0%)
Coreset	Active learning	39.3% (1.9%)	5.5% (0.3%)	14.7% (0.8%)
Levelset BAX	BAX	35.4% (2.2%)	6.3% (0.4%)	15.0% (0.9%)
Top-K BAX	BAX	38.8% (2.3%)	6.8% (0.6%)	16.2% (1.2%)
DiscoBAX (ours)	BAX	**44.1% (2.2%)**	**7.8% (0.5%)**	**18.6% (1.1%)**

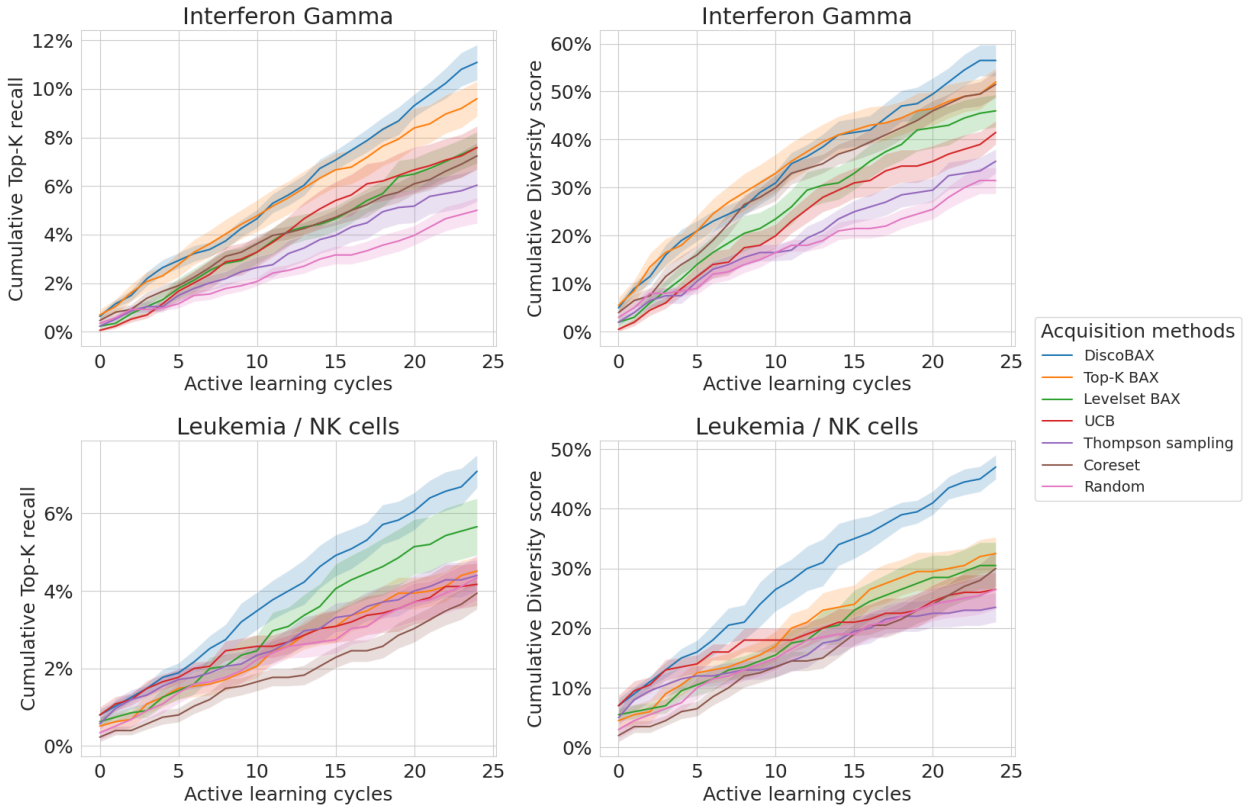

Figure 3. **Top-K recall and Diversity score Vs acquisition cycles** The two top plots are for the Interferon γ assay (Schmidt et al., 2021), and the two bottom plots are based on the Leukemia assay (Zhuang et al., 2019).

gains of model parameters (Houlsby et al., 2011; Kirsch et al., 2019). Note that active learning is agnostic to the value of the function being estimated, and so does not perform any exploration-exploitation trade-off.

Bandits The upper confidence bounds seen in BO originate in the bandit setting (Lai & Robbins, 1985), in which one can extend the widely-used UCB algorithm to Gaussian processes (Grünewälder et al., 2010; Srinivas et al.). While both bandits and BO seek to find the maximum of a function, the two problem settings leverage different notions of optimality. BO seeks to *identify* the argmax, whereas bandits seek to *minimize* the number of sub-optimal queries. Related to bandits and BO, some efforts are made to formulate active learning as a reinforcement learning problem (Slade & Branson, 2022; Casanova et al., 2020; Konyushkova et al., 2017; Pang et al., 2018). As with BO, the assumption that the learner has access to noisy samples of the objective function is baked into bandit algorithms, making these approaches also unsuitable for our setting.

Optimal experiment design (OED) is a broad umbrella

whose scope includes Bayesian Optimization: rather than simply maximizing a parametric function, the task is to adaptively identify an optimal set of experiments to efficiently reach some goal (Robbins, 1952; Chernoff, 1959). Applying machine learning to automate hypothesis generation and testing goes back multiple decades (King et al., 2004). Optimal experiment design is amenable to Bayesian optimization (Greenhill et al., 2020) and reinforcement learning approaches (Kandasamy et al., 2019). While many OED approaches are unsuitable for our setting for the same reasons as BO and bandits, our method benefits from the application of Bayesian Algorithm Execution (BAX) (Neiswanger et al., 2021), which we leverage as an acquisition function to identify candidate points with a high value of information.

7. Conclusion

This work has presented a first step towards the development of optimal experiment design techniques targeted at the multi-stage drug discovery process. We have introduced a mathematical formalization of the drug discovery problem that captures the uncertainty inherent in the transition from in vitro to in vivo experiments. We proposed a novel algorithm based on Bayesian Algorithm Execution and illustrated its utility on many illustrative synthetic datasets. We have further evaluated this class of methods against the real-world large-scale assays from the GeneDisco benchmark, where they help identify diverse top interventions better than existing baselines.

A variety of exciting directions present themselves from the foundation laid by this paper. Future work could see the extension of the current framework to explicitly account for the fact that experimental cycles happen in batches, generalizing the iterative sampling approach considered here. Further, we assume in this work that distant representations of interventions imply different underlying biological mechanisms – developing a causal formulation of the problem, and correspondingly identifying feature representations which capture this causal structure, would allow us to tell apart causally connected pathways more cleanly. Finally, it is standard practice to measure several potential intermediate phenotypes of interest to capture different aspects of interest, which requires an extension of our approach to the setting of multiple objectives.

Acknowledgements

CL was funded by an Open Philanthropy AI Fellowship when this work was completed. PN is supported by GSK and the UK Engineering and Physical Sciences Research Council (ESPRC ICASE award no.18000077).

References

Berdigaliyev, N. and Aljofan, M. An overview of drug discovery and development. *Future Medicinal Chemistry*, 12(10):939–947, 2020.

Casanova, A., Pinheiro, P. O., Rostamzadeh, N., and Pal, C. J. Reinforced active learning for image segmentation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkgC6TNFvr.

Chappell, L., Russell, A. J., and Voet, T. Single-cell (multi) omics technologies. *Annual Review of Genomics and Human Genetics*, 19:15–41, 2018.

Chernoff, H. Sequential design of experiments. *The Annals of Mathematical Statistics*, 30(3):755–770, 1959.

Dempster, J. M., Rossen, J., Kazachkova, M., Pan, J., Kugener, G., Root, D. E., and Tsherniak, A. Extracting biological insights from the project achilles genome-scale crispr screens in cancer cell lines. *bioRxiv*, 2019. doi: 10.1101/720243. URL https://www.biorxiv.org/content/early/2019/07/31/720243.

Dickson, M. and Gagnon, J. P. Key factors in the rising cost of new drug discovery and development. *Nature Reviews Drug Discovery*, 3(5):417–429, 2004.

Dickson, M. and Gagnon, J. P. The cost of new drug discovery and development. *Discovery Medicine*, 4(22): 172–179, 2009.

DiMasi, J. A., Grabowski, H. G., and Hansen, R. W. Innovation in the pharmaceutical industry: new estimates of r&d costs. *Journal of Health Economics*, 47:20–33, 2016.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Goel, A., Guha, S., and Munagala, K. How to probe for an extreme value. *ACM Trans. Algorithms*, 7 (1), dec 2010. ISSN 1549-6325. doi: 10.1145/1868237.1868250. URL https://doi.org/10.1145/1868237.1868250.

Greenhill, S., Rana, S., Gupta, S., Vellanki, P., and Venkatesh, S. Bayesian optimization for adaptive experimental design: A review. *IEEE Access*, 8:13937–13948, 2020. doi: 10.1109/ACCESS.2020.2966228.

Griffiths, R.-R. and Hernández-Lobato, J. M. Constrained bayesian optimization for automatic chemical design. *arXiv preprint arXiv:1709.05501*, 2017.

Grünewälder, S., Audibert, J.-Y., Opper, M., and Shawe-Taylor, J. Regret bounds for gaussian process bandit problems. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 273–280. JMLR Workshop and Conference Proceedings, 2010.

Hasin, Y., Seldin, M., and Lusis, A. Multi-omics approaches to disease. *Genome Biology*, 18(1):1–15, 2017.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Jehuda, R. B., Shemer, Y., and Binah, O. Genome editing in induced pluripotent stem cells using crispr/cas9. *Stem Cell Reviews and Reports*, 14(3):323–336, 2018.

Jiménez-Luna, J., Grisoni, F., and Schneider, G. Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, 2(10):573–584, 2020.

Kandasamy, K., Neiswanger, W., Zhang, R., Krishnamurthy, A., Schneider, J., and Poczos, B. Myopic posterior sampling for adaptive goal oriented design of experiments. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3222–3232. PMLR, 09–15 Jun 2019.

King, R. D., Whelan, K. E., Jones, F. M., Reiser, P. G., Bryant, C. H., Muggleton, S. H., Kell, D. B., and Oliver, S. G. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971): 247–252, 2004.

Kirsch, A., Van Amersfoort, J., and Gal, Y. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32, 2019.

Konyushkova, K., Sznitman, R., and Fua, P. Learning active learning from data. *Advances in neural information processing systems*, 30, 2017.

Korovina, K., Xu, S., Kandasamy, K., Neiswanger, W., Poczos, B., Schneider, J., and Xing, E. Chembo: Bayesian optimization of small organic molecules with synthesizable recommendations. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3393–3403. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/korovina20a.html.

Kushner, H. J. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, mar 1964. doi: 10.1115/1.3653121. URL https://doi.org/10.1115%2F1.3653121.

Lai, T. and Robbins, H. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985. ISSN 0196-8858. doi: https://doi.org/10.1016/0196-8858(85)90002-8. URL https://www.sciencedirect.com/science/article/pii/0196885885900028.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

MacLean, A. L., Hong, T., and Nie, Q. Exploring intermediate cell states through the lens of single cells. *Current Opinion in Systems Biology*, 9:32–41, 2018. ISSN 2452-3100. doi: https://doi.org/10.1016/j.coisb.2018.02.009. URL https://www.sciencedirect.com/science/article/pii/S2452310017302238. Mathematic Modelling.

Mehrjou, A., Soleymani, A., Jesson, A., Notin, P., Gal, Y., Bauer, S., and Schwab, P. Genedisco: A benchmark for experimental design in drug discovery. *arXiv preprint arXiv:2110.11875*, 2021.

Močkus, J. On bayesian methods for seeking the extremum. In *Optimization techniques IFIP technical conference*, pp. 400–404. Springer, 1975.

Moss, H. B., Beck, D., Gonzalez, J., Leslie, D. S., and Rayson, P. BOSS: Bayesian Optimization over String Spaces, October 2020. URL https://arxiv.org/abs/2010.00979v1.

Neiswanger, W., Wang, K. A., and Ermon, S. Bayesian algorithm execution: Estimating computable properties of black-box functions using mutual information. In *International Conference on Machine Learning*, pp. 8005–8015. PMLR, 2021.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14:265–294, 1978.

Nica, A. C., Jain, M., Bengio, E., Liu, C.-H., Korablyov, M., Bronstein, M. M., and Bengio, Y. Evaluating generalization in gflownets for molecule design. In *ICLR2022 Machine Learning for Drug Discovery*, 2022.

Notin, P., Hernández-Lobato, J. M., and Gal, Y. Improving black-box optimization in VAE latent space using decoder uncertainty. In *Advances in Neural Information Processing Systems*, volume 34, pp. 802–814. Curran Associates, Inc., 2021.

Pang, K., Dong, M., Wu, Y., and Hospedales, T. Meta-learning transferable active learning policies by deep reinforcement learning. *arXiv preprint arXiv:1806.04798*, 2018.

Pertea, M., Shumate, A., Pertea, G., Varabyou, A., Breitwieser, F. P., Chang, Y.-C., Madugundu, A. K., Pandey, A., and Salzberg, S. L. Chess: a new human gene catalog curated from thousands of large-scale rna sequencing experiments reveals extensive transcriptional noise. *Genome Biology*, 19(1):1–14, 2018.

Rasmussen, C. E. Gaussian processes in machine learning. In *Summer school on machine learning*, pp. 63–71. Springer, 2003.

Robbins, H. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

Sanchez, C. G., Acker, C. M., Gray, A., Varadarajan, M., Song, C., Cochran, N. R., Paula, S., Lindeman, A., An, S., McAllister, G., et al. Genome-wide crispr screen identifies protein pathways modulating tau protein levels in neurons. *Communications biology*, 4(1):1–14, 2021.

Schmidt, R., Steinhart, Z., Layeghi, M., Freimer, J. W., Nguyen, V. Q., Blaeschke, F., and Marson, A. Crispr activation and interference screens in primary human t cells decode cytokine regulation. *bioRxiv*, 2021. doi: 10.1101/2021.05.11.443701. URL https://www.biorxiv.org/content/early/2021/05/12/2021.05.11.443701.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104 (1):148–175, 2015.

Slade, E. and Branson, K. M. Deep reinforced active learning for multi-class image classification. *arXiv preprint arXiv:2206.13391*, 2022.

Snoek, J., Larochelle, H., and Adams, R. P. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

Song, J., Yu, L., Neiswanger, W., and Ermon, S. A general recipe for likelihood-free Bayesian optimization. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20384–20404. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/song22b.html.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design.

Trapnell, C. Defining cell types and states with single-cell genomics. *Genome Research*, 25(10):1491–1498, 2015.

Worzfeld, T., Pogge von Strandmann, E., Huber, M., Adhikary, T., Wagner, U., Reinartz, S., and Müller, R. The unique molecular and cellular microenvironment of ovarian cancer. *Frontiers in Oncology*, 7:24, 2017.

Zhu, Y., Feng, F., Hu, G., Wang, Y., Yu, Y., Zhu, Y., Xu, W., Cai, X., Sun, Z., Han, W., et al. A genome-wide crispr screen identifies host factors that regulate sars-cov-2 entry. *Nature communications*, 12(1):1–11, 2021.

Zhuang, X., Veltri, D. P., and Long, E. O. Genome-wide crispr screen reveals cancer cell resistance to nk cells induced by nk-derived ifn-γ. *Frontiers in Immunology*, 10: 2879, 2019. ISSN 1664-3224. doi: 10.3389/fimmu.2019.02879. URL https://www.frontiersin.org/article/10.3389/fimmu.2019.02879.

A. Biology background

Here we provide the mathematical formalization of the engaged processes in the CRISPR-based gene knockout experiments from gene embeddings to assay readouts. We take a comprehensive approach for clarity but not all notations below are used in this work.

- **Genes:** Let $\{g_1, g_2, \ldots, g_m\}$ with $g_i \in \mathcal{G}$ be all available genes for intervention.
- **Disease phenotype:** Several phenotype measurements are possible for every disease. Let $d \in D = \{d_1, d_2, \ldots, d_l\}$ be such a measurement from the list of l possible readouts.
- **Intermediate phenotype functions:** Instead of the actual disease phenotype, intermediate readouts are used to measure the effect of a gene intervention on the disease phenotype. These readouts should be correlated with the downstream outcomes, but may present a simplified view of the disease action; for example, they might include the expression of certain proteins in a cancerous cell culture which are known to correlate with tumour growth rate (the disease phenotype). We let $ip \in IP = \{ip_1, ip_2, \ldots, ip_p | ip : D \to \mathbb{R}\}$ be the set of maps from disease phenotype to real numbers that are the intermediate readouts for the effect of each gene intervention.
- **Knock-out function:** $\psi : G^m \to \mathcal{P}(G)$ shows which genes to intervene on. It takes the set of all available genes as input and returns the subset of genes to get knocked out.
- **Disease mechanism function:** $f : G \times \mathcal{P}(G) \to D^l$. This function takes all available genes and also the intervened subset and returns how the effect of the intervention on disease phenotype.
- **Knock-out representation** $\phi_{ko} : \mathcal{P}(G) \to \mathbb{R}^{d_{ko}}$ takes the subset of genes to knock out and returns a real-valued vector as the representation of this intervention.
- **Learnable mechanism:** To make the disease mechanism function amenable to learning algorithms, we use the intervention representation in the input and intermediate phenotype read-out in the output and work with $\{F_j : F_j = ip_j \circ f \circ \phi_{ko}^{-1} \text{for} 1 \leq j \leq p\}$ where $F_j : \mathbb{R}^d_{ko} \to \mathbb{R}$ is the effect of a knock-out represented by the knock-out representation ϕ_{ko} in the input on the jth intermediate phenotype read-out in the output.

It is natural to work with real-valued functions with real-value domain which are more friendly to function estimation algorithms. For example, one can use MSE error as a metric to learn the intervention-to-assay mechanism from the available labeled datasets $D = \{(x_i, y_i)\}_{i=1}^n$ using the objective function

$$\hat{F}_j = \frac{1}{n} \sum_{(x_i, y_i)} \arg\min_F \|F(x_i) - y_i)\|. \tag{7}$$

for every j that gives $\{\hat{F}_1, \hat{F}_2, \ldots, \hat{F}_p\}$. Notice that $\hat{y} = \hat{F}_j(x)$ is the best predictor of the intermediate disease phenotype (screen, assay) for the gene intervention $\psi(G^m)$ represented by $x = \phi_{ko}(\psi(G^m))$.

B. Deeper insights into the DiscoBAX algorithm

B.1. Insights of Equation (4):

In this section, we design a simplistic scenario to provide more insight into the proposed objective function Equation (4) and how it serves two purposes, i.e., choosing a set of interventions with high phenotype values and high diversity. For convenience, in Figure 4, we show a simple scenario where 2 out of 3 genes are to be chosen, i.e., $|\mathcal{S}| = 2$ and $|\mathcal{X}| = 3$. There are three ways of choosing a pair of genes out of three options. We aim to show which pair is favoured by Equation (4). Without loss of generality, assume $\mathbf{x}_1$ is chosen. Due to the probabilistic model of f_{out}, all $y_i = f_{\text{out}}(x_i), i = 1, 2, 3$ are random variables whose probability densities (P_i) are plotted next to each gene. It is observed that P_1 and P_3 are concentrated at larger values (higher regions of the vertical axis) compared to P_2 that puts much of its mass at lower values. Hence, in most realization, y_1 takes a large value (star) and $y_1 \approx \max(y_1, \cdot)$ as the second argument is sampled from distributions concentrated at lower values (P_2) or almost equally large values (P_3). The second argument becomes important in the rare events when y_1 takes a small value (cross). In this case, the output of $\max(y_1, \cdot)$ is no longer determined by its first argument and is, with high probability, influenced by the second argument which takes on a large value if realized from

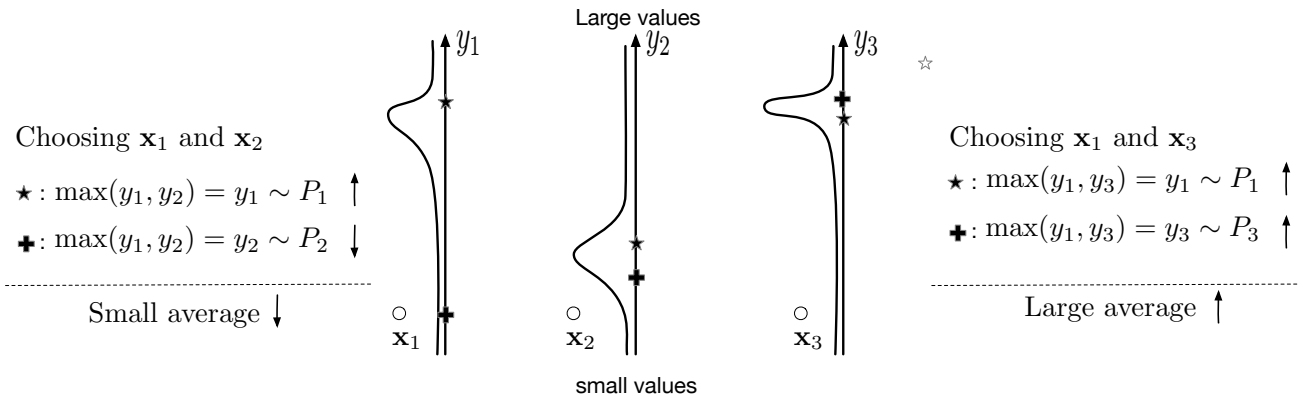

Figure 4. A simple visualization to gain insight into Equation (4).

P_3 compared with P_2 as the former is concentrated at larger values. Hence, choosing the pair $(\mathbf{x}_1, \mathbf{x}_3)$ produces a larger average than $(\mathbf{x}_1, \mathbf{x}_2)$ and is therefore favourable by Equation (4). Moreover, it is implicitly assumed in the above reasoning that P_1, P_2 and P_3 are not highly correlated. Otherwise, a small value of y_1 led to a small value of y_3 as well. Hence, Equation (4) chooses the genes which produce large values and the mechanisms that are modeled as the random effect are as independent as possible. This choice of genes increases the chance that if one gene fails to proceed to further steps of the drug discovery pipeline for some reason such as safety, tractability, etc, the other chosen genes will preserve high chances of success as they are likely to be involved in mechanisms different from those that cause the failure of the previous gene.

B.2. Insights of Figure 1:

This illustrates the motivation and goal of this research which is finding mathematical formulation and practical implementation of an algorithm that meets the actual needs of initial stages of drug discovery pipeline that neither value-seeking nor diversity-seeking methods can fulfill. The phenotypic effect of genetic perturbation can follow a complex function with many modes. We are mainly interested in genes which cause large changes in the measured phenotype as those are the genes that engage more with the disease and can be a potential target for a drug compound. However, as the figure shows, the value-seeking methods stop after finding one mode of the function (the light gray triangles which are concentrated in one of the modes but do not cover the other modes which have equally large values). This is risky since the genes that are associated with that mode are probably correlated in the sense that if one of them fails in the further steps of the drug discovery pipeline, the other may also fail with high likelihood. On the other end of the spectrum, although a diversity-seeking algorithm proposes uncorrelated genes that are unlikely to fail together (the dark gray circles which cover a large domain but miss the modes), it is highly inefficient and a large number of chosen genes may not be highly involved in the disease mechanism. Hence, the nature of the problem requires a middle-ground method that seeks the modes of the underlying function but covers as many does as possible (the red stars that efficiently cover all modes but not in-between spaces) so that if the genes associated with one mode fails, those associated with the other modes have chance to proceed in the pipeline.

C. Sub-modularity of S

Observation 2. *The score function $S : \mathcal{P}(\mathcal{G}) \to \mathbb{R}$ defined by*

$$S(G) = \mathbb{E}_{f_{out}}[\max_{g \in G} \max(0, f_{out}(g))] \tag{8}$$

is monotone submodular.

Proof. We first show monotonicity.

$$\begin{aligned}
S(G \cup \{g\}) &= \mathbb{E}_{f_{\text{out}}}[\max_{g' \in G \cup \{g\}} \max(0, f_{\text{out}}(g'))] \\
&= \mathbb{E}_{\eta}[\max_{g' \in G \cup \{g\}} \max(0, f_{\text{out}}(g', \eta))] \\
&\leq \mathbb{E}_{\eta} \max_{G}[\max(0, f_{\text{out}}(g', \eta)) + \max(0, f_{\text{out}}(g, \eta))] \\
&= \mathbb{E}_{\eta} \max_{G}[\max(0, f_{\text{out}}(g', \eta))] + \mathbb{E}_{\eta}[\max(0, f_{\text{out}}(g, \eta))] \\
&= S(G) + S(\{g\})
\end{aligned}$$

The proof for submodularity follows similarly. Letting $X \subseteq Y$ we have that S is submodular if for any point g we have $S(X \cup \{g\}) - S(X) \geq S(Y \cup \{g\}) - S(Y)$. First, recall:

$$S(X \cup \{x\}) - S(X) = \mathbb{E}_{f_{\text{out}}}[\max_{g' \in X \cup \{g\}} \max(0, f_{\text{out}}(g'))] - \mathbb{E}_{f_{\text{out}}}[\max_{g' \in X} \max(0, f_{\text{out}}(g'))] \tag{9}$$

We consider a single realization of the outcome f_{out}, and will show that the inequality holds for this outcome. If the maximum of f_{out} over Y is negative, then the result is trivial. Otherwise, there are three cases to consider: first, if g maximizes f_{out} over the set $Y \cup \{g\}$, then we have

$$\max_{g' \in X \cup \{g\}} \max(0, f_{\text{out}}(g')) - \max_{g' \in X} \max(0, f_{\text{out}}(g')) = f_{\text{out}}(g) - \max_{g' \in X} \max(0, f_{\text{out}}(g')) \tag{10}$$

$$\geq f_{\text{out}}(g) - \max_{g' \in Y} \max(0, f_{\text{out}}(g')) \tag{11}$$

as $X \subseteq Y$. Next, if g does not maximize f_{out} in X, then the difference on both sides of the inequality will be zero. Finally, if g maximizes f_{out} in X but not in Y, we have the following:

$$\max_{g' \in X \cup \{g\}} \max(0, f_{\text{out}}(g')) - \max_{g' \in X} \max(0, f_{\text{out}}(g')) = f_{\text{out}}(g) - \max_{g' \in X} \max(0, f_{\text{out}}(g')) \tag{12}$$

$$> 0 = \max_{g' \in Y \cup \{g\}} \max(0, f_{\text{out}}(g')) - \max_{g' \in Y} \max(0, f_{\text{out}}(g')) \tag{13}$$

Since the inequality holds for each random realization of f_{out}, it applies to the expectation, and so we have

$$\mathbb{E}_{\eta}[\max_{g' \in X \cup \{g\}} \max(0, f_{\text{out}}(g'))] - \mathbb{E}_{\eta}[\max_{g' \in X} \max(0, f_{\text{out}}(g'))] \geq \mathbb{E}_{\eta}[\max_{g' \in Y \cup \{g\}} \max(0, f_{\text{out}}(g'))] \tag{14}$$

$$- \mathbb{E}_{\eta}[\max_{g' \in Y} \max(0, f_{\text{out}}(g'))] \tag{15}$$

□

Corollary 1. *The greedy algorithm which iteratively selects points maximizing $S(G)$ is a $1 - 1/e$ approximation of the optimal.*

D. Detailed experimental results

For reproducibility, the entire codebase for all experiments in this work can be found in the supplementary material attachment. Experiments with DiscoBAX in this section were conducted with Bernoulli noise. We present results with Gaussian noise in the next section (GeneDisco experiments).

D.1. Synthetic dataset experiments

D.1.1. Sample complexity

Our objective in this section is to validate a number of properties of the proposed method in interpretable synthetic datasets.

- **Sample complexity:** our method requires fewer samples to reach a global optimum relative to random sampling or naive uncertainty maximization methods.

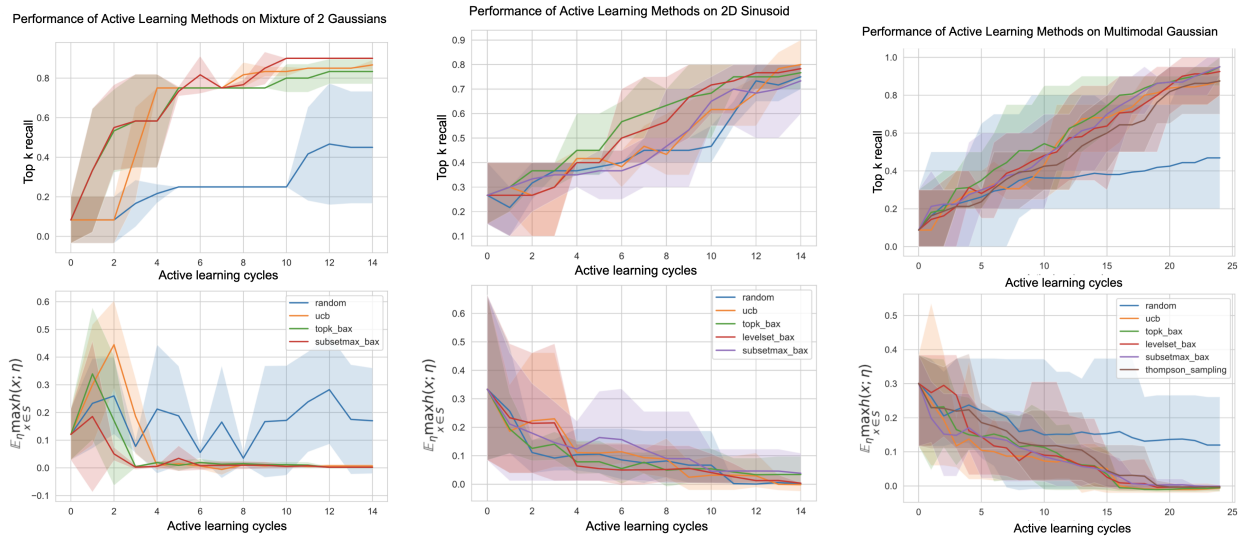

Figure 5. Top-k recall and expected maximal intervention value on: a) a mixture of two RBF kernels; b) a one-dimensional linear combination of sinusoids with multiple local optima; c) a mixture of four RBF kernels of varying scales.

- **Diversity of candidate set:** unlike standard Bayesian optimization methods, our approach identifies a set of points which approximately maximize the function while also maintaining diversity with respect to a pre-chosen metric, improving the robustness of the candidate set to uncertainty in the mapping between observable and terminal outcomes.

In these experiments and in Figure 2 we consider a number of baselines, including the following.

- **Random:** $\mathbf{x}^* \sim \text{Unif}(\mathcal{X} \setminus \mathcal{D}_t)$.
- **UCB:** naive upper-confidence sampling approach, letting $c \in \mathbb{R}$ be some constant: $\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} \mu(\mathbf{x}) + c\sqrt{\sigma^2(\mathbf{x})}$.
- **BAX** acquisition (Algorithm 2) for algorithm $\mathcal{A} \in \{\text{Top-k, Levelset, Disco}\}$.
- **Thompson sampling:** acquisition based on maximum of sampled function from a Bayesian posterior. $\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} \widehat{f}_{\text{ip}}(\mathbf{x}) \quad \widehat{f}_{\text{ip}} \sim P(f_{\text{ip}}|\mathcal{D}_{\text{train}})$.
- **Active sampling:** maximize uncertainty over the input set $\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} \sigma^2(\mathbf{x})$.

We consider the following synthetic datasets, where for all synthetic experiments we use a batch size equal to one.

Mixture-of-Gaussians: pdf of a mixture of gaussians with means [-0.5, 0.5], variances 0.1 and relative weights [0.3, 0.7]. $x \in [-1, 1]$.

Multimodal mixture: given domain $[-7, 7]$, outputs the (scaled) density of a mixture of Gaussians with means $\{-4, -2, 0, 3\}$, variances $\{0.3, 0.35, 0.3, 0.35\}$, and weights $\{0.6, 0.45, 0.5, 0.4\}$. **2-d sinusoid:** $f(x) = \sin\left[\frac{1}{2}\begin{pmatrix} 0.25 & -\frac{1}{\pi} \\ 0.1 & .02 \end{pmatrix}\mathbf{x}\right], \mathbf{x} \in \mathbb{R}^2, -\pi < \mathbf{x} < \pi$

D.1.2. Additional Empirical Evaluations on Synthetic Dataset

We include an evaluation of the Expected Improvement (EI) acquisition function (Fig. 6) on the same task as was previously illustrated in Figure 2. Because we use an acquisition batch size of one in these experiments, the parallel acquisition strategy qEI coincides with the incremental expected improvement acquisition function. Concretely, the expected improvement acquisition function performs the following maximization, given some pool $\mathcal{D}$ of already sampled points:

$$\max_{x \notin \mathcal{D}} \mathbb{E}_{P(f(x)|\mathcal{D})} \max(f(x) - f(x^*), 0) \tag{16}$$

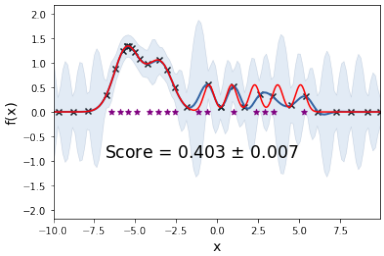

Figure 6. Evaluation of the EI acquisition function on the regression problem discussed previously.

where x^* is the element of $\mathcal{D}$ which maximizes f and $P(\cdot|\mathcal{D})$ denotes the posterior over function values $f(x)$ for some fixed x.

D.2. GeneDisco experiments

D.2.1. Clustering of optimal interventions

In the GeneDisco experiments (§ 5.2), we define a diversity metric based on the recall of Top-K clusters. These clusters are obtained for each assay as follows. All experiments we carried out in § 5.2 leverage the Achilles dataset (Dempster et al., 2019) from GeneDisco to represent the different interventions. This dataset characterizes each gene with an 808-dimensional vector. We first select the optimal interventions as the ones in the top percentile of disease phenotype for a given assay. We then project the Achilles representations of each intervention into a lower-dimensional subspace of dimension 20 with PCA. We then fit a Gaussian Mixture Model (GMM) with 20 mixtures to obtain the different clusters, selecting the best result out of 20 random initializations.

D.2.2. Detailed performance analysis

We provide below detailed results across the five CRISPR assays from the GeneDisco benchmark: the Interferon γ and Interleukin 2 assays based on (Schmidt et al., 2021), the Leukemia assay with NK cells from (Zhuang et al., 2019), the SARS-CoV-2 assay from (Zhu et al., 2021) and the Tau protein assay from (Sanchez et al., 2021). All interventions for the five assays were represented based on the Achilles dataset (Dempster et al., 2019). For the active learning baselines already present in GeneDisco we used the same hyperparameters as in (Mehrjou et al., 2021). For the additional baselines introduced in this work, we use standard/default hyperparameters everywhere (see our codebase in supplementary material for all details) except as specified in Appendix D.3. We used DiscoBAX with Gaussian noise (with length scale for the underlying Radial Basis Function kernel equal to 1) in the results below, but obtain comparable performance with Bernoulli noise. To prevent model overfitting during the various active learning cycles, we closely followed experimental protocol in Mehrjou et al. (2021) and selected similar model architectures and hyperparameters.

We observe in Tables 2 to 6 that DiscoBAX outperforms all other 13 baselines we compare against on 3 out of the 5 datasets included in the GeneDisco benchmark, performs on par (significant overlap of confidence intervals) with the best methods on the 4th one (Tau protein) and is only outperformed by "random selection" on the last one (SARS-coV2). As discussed in section 5.2 and as noted in Mehrjou et al. (2021), the fact that random outperforms all other 13 methods on that dataset seems to indicate an issue with the data (eg., the feature space does not correlate with the disease phenotype, high label noise) rather than an algorithmic issue. Critically, none of the other baselines performs consistently high on all 5 assays: for instance, 'random' performs relatively poorly on all other 4 assays and the other methods that are on par with DiscoBAX on the Tau protein assay (eg., BADGE, Coreset) have inconsistent performance on other assays. Aggregated performance across assays is reported in Table 1 and demonstrates the overall higher performance of DiscoBAX over other baselines. The superior sample efficiency of the scheme is also apparent Fig. 7 as DiscoBAX exhibits superior recall and diversity score throughout the different learning cycles.

D.3. GeneDisco experiments - hyperparameter selection

For the three BAX alogirthms (Top-K BAX, Levelset BAX and DiscoBAX), we optimize the main hyperparameters of each method (ie., respectively the K parameter the level threshold and the number of Sets in SetSelect). To mitigate the risk of

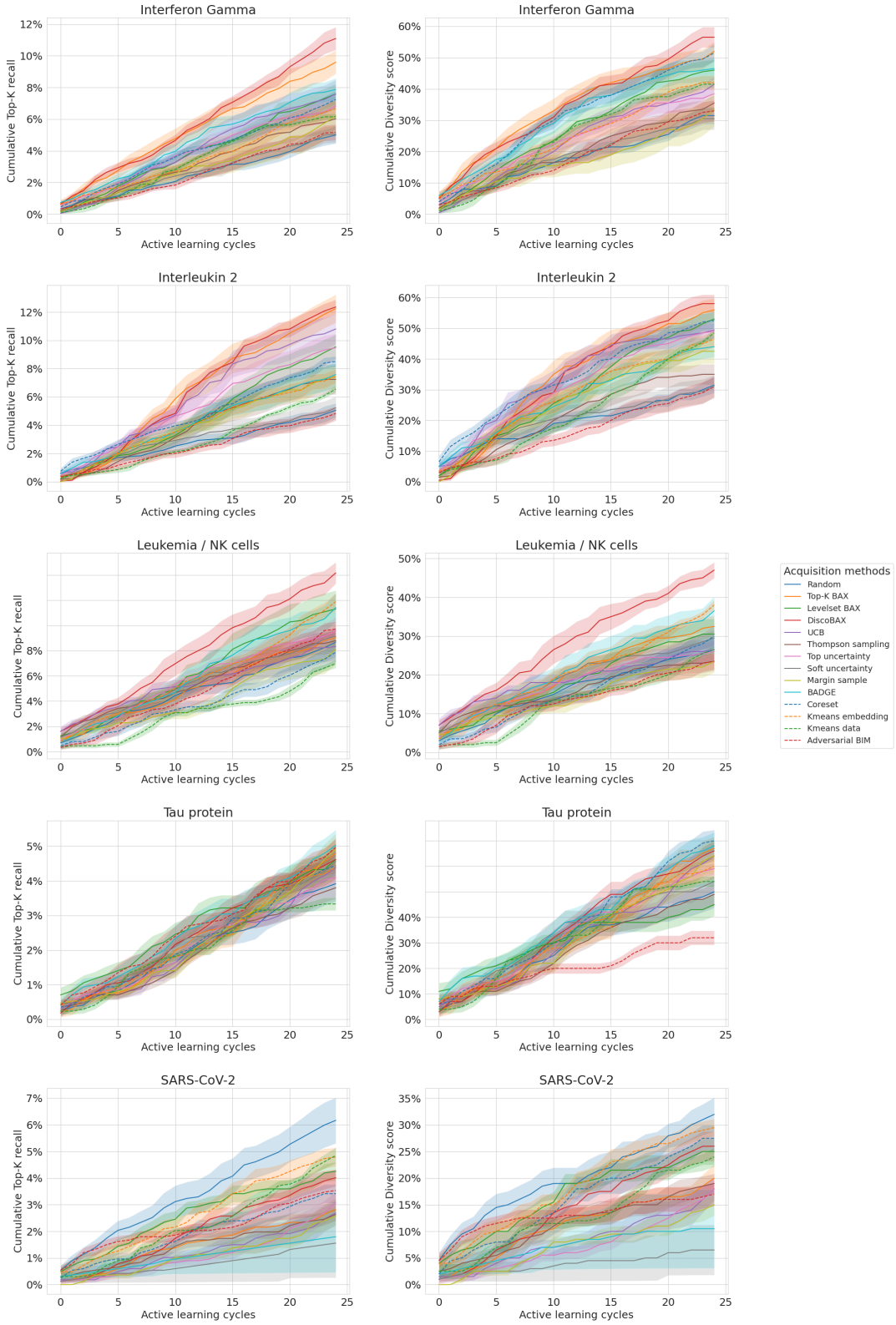

Figure 7. **Top-K recall and Diversity score Vs acquisition cycles for all GeneDisco CRISPR assays**

Table 2. **Detailed performance comparison on GeneDisco - Interferon γ assay**

Dataset	Method	Top-K recall	Diversity score	Overall score
Interferon γ	Adversarial BIM	33% (3.6%)	5.2% (0.5%)	13.1% (1.4%)
	BADGE	46.5% (3.9%)	7.9% (0.7%)	19.1% (1.7%)
	Coreset	51.5% (3%)	7.2% (0.6%)	19.3% (1.3%)
	DiscoBAX (ours)	**56.5% (3.4%)**	**11.1%** (0.8%)	**25% (1.6%)**
	Kmeans Data	41.5% (1.3%)	6.1% (0.2%)	16% (0.5%)
	Kmeans Embedding	42% (2.4%)	6.7% (0.6%)	16.7% (1.2%)
	Levelset BAX	46% (3.6%)	7.6% (0.7%)	18.7% (1.5%)
	Margin sample	33.5% (5.8%)	6.2% (1.1%)	14.4% (2.6%)
	qEI	45% (3.6%)	7.9% (0.4%)	18.9% (1.1%)
	qPOI	44% (3.1%)	8.1% (0.4%)	18.9% (1.1%)
	qUCB	43.7% (3.7%)	7.8% (0.4%)	18.5% (1.2%)
	Random	31.5% (2.9%)	5% (0.6%)	12.5% (1.3%)
	Soft Uncertainty	30.5% (3.7%)	5.1% (0.6%)	12.4% (1.5%)
	Thompson Sampling	35.5% (2.6%)	6% (0.7%)	14.6% (1.4%)
	Top-K BAX	52% (3.1%)	9.6% (0.8%)	22.3% (1.5%)
	Top Uncertainty	38.5% (3%)	6.8% (0.7%)	16.2% (1.4%)
	UCB	41.5% (2.6%)	7.6% (0.9%)	17.7% (1.6%)

Table 3. **Detailed performance comparison on GeneDisco - Interleukin 2 assay**

Dataset	Method	Top-K recall	Diversity score	Overall score
Interleukin 2	Adversarial BIM	31% (3.6%)	4.8% (0.5%)	12.2% (1.4%)
	BADGE	44% (3.6%)	7.6% (1%)	18.3% (1.9%)
	Coreset	52.5% (2.9%)	8.5% (0.4%)	21.1% (1.1%)
	DiscoBAX (ours)	**58% (3.1%)**	**12.4% (0.5%)**	**26.8% (1.3%)**
	Kmeans Data	48.5% (1.7%)	6.6% (0.3%)	17.8% (0.7%)
	Kmeans Embedding	46.5% (2.8%)	7.5% (0.5%)	18.6% (1.2%)
	Levelset BAX	53% (3%)	9.5% (0.9%)	22.5% (1.6%)
	Margin sample	42.5% (4.2%)	7.4% (0.9%)	17.8% (2%)
	qEI	52.5% (2.9%)	11.4% (0.9%)	24.5% (1.6%)
	qPOI	54% (2.8%)	11.9% (0.9%)	25.3% (1.6%)
	qUCB	52.5% (4.7%)	11.3% (1%)	24.4% (2.2%)
	Random	31.5% (2.7%)	5.1% (0.5%)	12.6% (1.2%)
	Soft Uncertainty	31% (4%)	5.2% (0.8%)	12.7% (1.8%)
	Thompson Sampling	35% (3.5%)	7.2% (1.1%)	15.9% (2%)
	Top-K BAX	56% (3.9%)	12.2% (1%)	26.2% (2%)
	Top Uncertainty	49% (2.8%)	9.5% (1.1%)	21.6% (1.7%)
	UCB	49.5% (2.8%)	10.8% (1.1%)	23.1% (1.8%)

overfitting, we select our hyperparameters based on a single assay (the 'Tau protein' assay), and use the obtained optimal values in experiments for all assays. We perform a grid search for each hyperparameter, repeating each experiment over 5 seeds. We find that on that dataset, optimal values for the hyperparameters are respectively k=5, Levelset=1.0 and S=10.

Table 4. **Detailed performance comparison on GeneDisco - SARS-CoV-2 assay**

Dataset	Method	Top-K recall	Diversity score	Overall score
SARS-CoV-2	Adversarial BIM	17% (2.4%)	3.5% (0.8%)	7.7% (1.4%)
	BADGE	10.5% (7.8%)	1.8% (1.4%)	4.3% (3.3%)
	Coreset	27.5% (2.6%)	3.4% (0.4%)	9.7% (1%)
	DiscoBAX (ours)	26% (3%)	4% (0.3%)	10.2% (1%)
	Kmeans Data	24% (1.9%)	4.9% (0.3%)	10.8% (0.8%)
	Kmeans Embedding	29.5% (1.7%)	4.8% (0.4%)	11.9% (0.8%)
	Levelset BAX	25% (2.6%)	4.3% (0.4%)	10.3% (1%)
	Margin sample	15% (2.1%)	2.6% (0.4%)	6.2% (0.9%)
	qEI	23.5% (3.3%)	4.1% (0.8%)	9.8% (1.6%)
	qPOI	21.7% (3%)	3.5% (0.4%)	8.7% (1%)
	qUCB	21.5% (2.9%)	4.4% (0.6%)	9.7% (1.3%)
	Random	**32% (3.3%**	**6.2% (0.9%)**	**14% (1.7%)**
	Soft Uncertainty	6.5% (4.9%)	1.6% (1.4%)	3.2% (2.6%)
	Thompson Sampling	19% (1.9%)	2.6% (0.3%)	7.1% (0.7%)
	Top-K BAX	20% (2.8%)	2.8% (0.4%)	7.5% (1.1%)
	Top Uncertainty	18% (2.3%)	3.1% (0.5%)	7.4% (1%)
	UCB	18% (2.4%)	2.7% (0.5%)	7% (1.1%)

Table 5. **Detailed performance comparison on GeneDisco - Leukemia/NK assay**

Dataset	Method	Top-K recall	Diversity score	Overall score
Leukemia/NK	Adversarial BIM	23.5% (2.2%)	4.9% (0.3%)	10.7% (0.8%)
	BADGE	36.5% (3.9%)	5.7% (0.6%)	14.4% (1.5%)
	Coreset	30% (3.2%)	3.9% (0.4%)	10.9% (1.2%)
	DiscoBAX (ours)	**47% (2.1%)**	**7.1% (0.4%)**	**18.2% (1%)**
	Kmeans Data	26.5% (1.1%)	3.5% (0.2%)	9.6% (0.4%)
	Kmeans Embedding	38% (1.3%)	5.9% (0.4%)	15% (0.7%)
	Levelset BAX	30.5% (4.1%)	5.7% (0.8%)	13.1% (1.8%)
	Margin sample	23.5% (3.1%)	4.1% (0.6%)	9.8% (1.4%)
	qEI	26.5% (3.2%)	4.3% (0.6%)	10.7% (1.4%)
	qPOI	31% (1.5%)	4.8% (0.6%)	12.2% (0.9%)
	qUCB	33% (2.9%)	5.4% (0.7%)	13.4% (1.4%)
	Random	26.5% (3.5%)	4.3% (0.6%)	10.7% (1.5%)
	Soft Uncertainty	29.5% (2.3%)	4.6% (0.4%)	11.6% (0.9%)
	Thompson Sampling	23.5% (2.6%)	4.4% (0.4%)	10.2% (1.1%)
	Top-K BAX	32.5% (2.9%)	4.5% (0.4%)	12.1% (1.1%)
	Top Uncertainty	26% (3.1%)	4.8% (0.6%)	11.2% (1.3%)
	UCB	26.5% (3%)	4.2% (0.6%)	10.5% (1.3%)

Table 6. **Detailed performance comparison on GeneDisco - Tau protein assay**

Dataset	Method	Top-K recall	Diversity score	Overall score
Tau protein	Adversarial BIM	16% (1.5%)	**5% (0.3%)**	8.9% (0.6%)
	BADGE	34% (2.8%)	**5% (0.5%)**	**13.1% (1.1%)**
	Coreset	**35% (2.2%)**	4.4% (0.3%)	12.5% (0.9%)
	DiscoBAX (ours)	33% (2.1%)	4.6% (0.4%)	12.3% (0.9%)
	Kmeans Data	27% (1.1%)	3.3% (0.2%)	9.5% (0.5%)
	Kmeans Embedding	30% (2.6%)	4.4% (0.4%)	11.5% (1%)
	Levelset BAX	22.5% (2.4%)	4.6% (0.5%)	10.2% (1.1%)
	Margin sample	32% (3.3%)	4.9% (0.5%)	12.5% (1.2%)
	qEI	32.1% (2.8%)	4.3% (0.6%)	11.7% (1.3%)
	qPOI	31% (2.5%)	4.5% (0.5%)	11.8% (1.1%)
	qUCB	31.2% (2.6%)	4.4% (0.4%)	11.7% (1%)
	Random	25% (3.3%)	3.9% (0.5%)	9.9% (1.3%)
	Soft Uncertainty	27% (2.4%)	4.6% (0.4%)	11.1% (0.9%)
	Thompson Sampling	24.5% (2.4%)	3.8% (0.3%)	9.7% (0.9%)
	Top-K BAX	33.5% (2.4%)	4.8% (0.4%)	12.7% (1%)
	Top Uncertainty	29.5% (1.2%)	4.1% (0.2%)	11% (0.5%)
	UCB	32% (2.7%)	4.4% (0.3%)	11.8% (1%)

Table 7. **GeneDisco experiment - Hyperparameter selection**

Method	Hyperparameter value	Top-K recall	Diversity score	Overall score
Top-K BAX	2	32% (3.6%)	4% (0.5%)	11.3% (1.3%)
	3	32% (3.3%)	4.3% (0.5%)	11.8% (1.1%)
	5	33% (2.4%)	4.4% (0.4%)	**12.1% (0.9%)**
	10	30% (3.2%)	4.2% (0.4%)	11.2% (1.3%)
Levelset BAX	0.8	19% (2.1%)	3.4% (0.3%)	8% (0.8%)
	1	30% (4.3%)	5.4% (0.7%)	**12.7% (1.8%)**
	1.2	21% (1.3%)	4.1% (0.6%)	9.3% (0.9%)
	1.5	29% (0.7%)	5.4% (0.5%)	12.5% (0.6%)
DiscoBAX	2	36% (5.3%)	4.8% (0.8%)	13.1% (2%)
	3	32% (2.6%)	4.1% (0.5%)	11.4% (1.1%)
	5	37.5% (2.7%)	5.4% (0.4%)	14.2% (1%)
	10	38% (1.8%)	5.5% (0.3%)	**14.5% (0.8%)**