ChemBO: Bayesian Optimization of Small Organic Molecules with Synthesizable Recommendations

Ksenia Korovina[1]   **Sailun Xu**[1]   **Kirthevasan Kandasamy**[2]   **Willie Neiswanger**[1]

Barnabás Póczos[1]   **Jeff Schneider**[1]   **Eric P. Xing**[1]

[1]Carnegie Mellon University   [2]U. C. Berkeley

Abstract

In applications such as molecule design or drug discovery, it is desirable to have an algorithm which recommends new candidate molecules based on the results of past tests. These molecules first need to be synthesized and then tested for objective properties. We describe ChemBO, a Bayesian optimization framework for generating and optimizing organic molecules for desired molecular properties. While most existing data-driven methods for this problem do not account for sample efficiency or fail to enforce realistic constraints on synthesizability, our approach explores the synthesis graph in a sample-efficient way and produces synthesizable candidates. We implement ChemBO as a Gaussian process model and explore existing molecular kernels for it. Moreover, we propose a novel optimal-transport based distance and kernel that accounts for graphical information explicitly. In our experiments, we demonstrate the efficacy of the proposed approach on several molecular optimization problems.

1   Introduction

In many applications, such as drug discovery and materials optimization, one is interested in designing chemical molecules with desirable properties (Elton et al., 2019). For instance, in drug discovery, one wishes to find molecules with high solubility in blood and high potency, but low toxicity. Recently, we have seen a surge of interest in adoption of machine learning techniques for such tasks, due to their effectiveness in modeling structure-property relations of molecules, and due

to limitations of traditional computational chemistry methods in effectively exploring the large and complex space of chemical molecules. For instance, the number of drug-like molecules is estimated to be between 10^{23} and 10^{60} (Polishchuk et al., 2013), among which only around 10^8 have been synthesized. While there have been several strategies for this problem, such as generative modeling, reinforcement learning, and more (Gómez-Bombarelli et al., 2018, Jin et al., 2018a, Popova et al., 2017, You et al., 2018, Oliynyk et al., 2016), one promising approach is to treat this task as a black-box optimization problem (e.g. Ling et al. (2017), Griffiths and Hernández-Lobato (2017)). Here, we assume existence of a function $f : \mathcal{X} \to \mathbb{R}$ defined on the chemical space $\mathcal{X}$, where $f(x)$ is a measure of goodness of molecule x for the relevant application. The goal is to find the optimum of this function $\text{argmax}_{x \in \mathcal{X}} f(x)$. In real world settings, f is typically derived from the results of laboratory experiments. The algorithm would then use results of the past experiments, i.e the $f(x)$ values, to recommend new molecules. Since such experiments are expensive, it is imperative to find the maximum in as few evaluations as possible.

In this work, we contribute to this line of research by developing ChemBO, a Bayesian optimization (BO) framework for generating and optimizing molecules, focusing on small[1] organic molecules for drug discovery. In doing so, we wish to emulate a real world setting, where an algorithm would recommend new candidate molecules. These molecules *first need to be synthesized*, and then tested for necessary properties. Ideally, the algorithm would not only ensure that the recommended molecule is chemically valid and synthesizable, but also provide a recipe for synthesis and take into consideration the reagents and resources available. Even in cases where the recommended molecules are synthesized manually, providing a recipe can be a helpful guide to the chemist and greatly reduce the amount of

---

[1]In contrast with biologics (large molecules), which are protein based.

manual work required. Combining sequential decision making and synthesis, ChemBO is a first step towards automated molecular optimization. To summarize, our contributions are:

1. We develop a Gaussian process (GP) model of structure-property relations in molecules. For the GP kernel, we use prior work on molecular fingerprints (Ralaivola et al., 2005, Hinselmann et al., 2010) and additionally design a new optimal transport based similarity measure between molecules by treating them as graphs.

2. We use a synthesis graph to navigate the chemical space. On each iteration of BO, ChemBO recommends the molecule on this synthesis graph that is deemed to be the most promising by the GP, i.e. the molecule with the highest *acquisition* value (Brochu et al., 2010). This approach not only ensures that each recommended molecule is chemically valid, but also provides a synthesis recipe[2].

3. In our experiments, we demonstrate that ChemBO outperforms simpler alternatives for synthesizeable optimization, which do not use a probabilistic model to guide search. The final values for the popular QED and penalized partition coefficient benchmarks achieved by ChemBO are competitive with state-of-the-art methods, while using significantly less data and function evaluations. Our code is released open source at https://github.com/ks-korovina/chembo.

2 Related Work

Optimization: SMILES strings (Anderson E, 1987), which describe the structure of molecules as a string, are a common representation used in machine learning techniques for molecular optimization (Gómez-Bombarelli et al., 2018, Kang and Cho, 2018). Other recent methods adopt graph representation for molecules, using generative models or reinforcement learning to construct a molecular graph, and optimize the property in question while attempting to maintain validity (Jin et al., 2018a, You et al., 2018, Guimaraes et al., 2017, Zhou et al., 2018, Jin et al., 2018b). In learning representations for molecules, they draw on methods that process graph data directly, such as graph neural networks (You et al., 2018, Jin et al., 2018b) and covariant compositional networks (Hy et al., 2018). However, drug/materials optimization is a *stateless* optimization problem, where there is no explicit need to deal with states and solve credit assignment. This can require a large number of samples (Jiang et al., 2017), and is not desirable in settings where each evaluation might involve several laboratory experiments. BO methods, which are par-

ticularly well suited for optimization problems with expensive evaluations, are sparsely represented in the field. Gómez-Bombarelli et al. (2018), Jin et al. (2018a), Kusner et al. (2017) learn a Euclidean representation for molecules and perform BO on this space, while Griffiths and Hernández-Lobato (2017) extend that work to account for validity constraints.

Synthesizable recommendations: In much of the above work, synthesizability of recommendations remains one of the most important concerns. The common approach to tackle this problem is to consider a proxy synthesizability score, by either imposing search constraints on the objective (Griffiths and Hernández-Lobato, 2017) or incorporating the score into f along with the other properties (Gómez-Bombarelli et al., 2018). However, synthesizability scores are not always reliable. For example, Gómez-Bombarelli et al. (2018) found that their autoencoder produced a large number of molecules with unrealistically large carbon rings when using the SA synthesizability score (Ertl and Schuffenhauer, 2009) as the reward function. More critically, it ignores practical challenges in a laboratory environment. First, a chemist may not have the reagents and/or the process conditions available to synthesize the molecule. Second, figuring out the synthesis plan for a single molecule, let alone the hundreds of them recommended during the optimization routine, can be quite challenging.

We leverage a large and separate direction of research which use ML techniques to predict outcomes of chemical reactions (Schwaller et al., 2018a, Wei et al., 2016, Coley et al., 2019, Chen and Baldi, 2009, Law et al., 2009) (see Engkvist et al. (2018) for a more complete list). The first methods for such *synthesis prediction* tasks were template based, in that they either select relevant rules from a fixed library, or rank enumerated outcomes of applying these rules. One of the first examples in the ML community was Wei et al. (2016), which predicted reaction type and then used SMARTS transformations to construct candidate outcome graphs. Due to rigidity of template-based approaches, template-free methods have become increasingly popular (Jin et al., 2017, Schwaller et al., 2018b). One such method in this class, and the one we adopt in this work to explore the chemical space, is Rexgen (Coley et al., 2019). It proceeds in two stages: first, reactive sites are predicted using a Weisfeiler-Lehman network with global attention (Lei et al., 2017) on the graph representation of reaction inputs; next, possible configurations of connectivity changes in reactive sites are scored with a Weisfeiler-Lehman Difference network (Jin et al., 2017).

Joint optimization and synthesis: Our approach in this paper, which marries both directions of work, can be viewed in two ways. On one hand, it performs

---

[2]We will qualify this statement later in Section 3.3.

Ksenia Korovina[1] **Sailun Xu**[1] **Kirthevasan Kandasamy**[2] **Willie Neiswanger**[1]

optimization while ensuring the recommendations are synthesizable. On the other hand, as we will explain shortly, it explores synthesis paths to discover promising candidate molecules via a data-driven guide. This approach is the core novelty of our work. As far as we are aware, there is only one work in this direction: concurrently with us, Bradshaw et al. (2019) pursued a similar goal of performing optimization with synthesis guarantees. However, their methodology and outcomes are very different from ours, in that they adopt a generative model on subsets of molecules and train it jointly with a property predictor directly on that latent space, all of which may require many samples. As a result, while their method produces useful representations for subsets of molecules, unlike ChemBO, it is not designed for sample-efficient goal-directed optimization tasks.

Kernels on molecules: For our GP based BO approach, we need to define a kernel between molecules. While there has been prior work on defining kernels and similarity metrics between graphs (Wallis et al., 2001, Kondor and Lafferty, 2002, White et al., 2019, Sutherland, 2015), most do not account for more complex properties of molecules in addition to graphical structure. There have been a variety of neural network based graph similarity measures proposed for molecules (Liu et al., 2018, Torng and Altman, 2018, Kearnes et al., 2016). However, these approaches are computationally expensive, which can be challenging in our GP based approach, where the similarity needs to be computed for several pairs of molecules during each iteration of the BO routine. A common class of graph based kernels used in chemoinformatics are based on molecular fingerprints (Ralaivola et al., 2005, Hinselmann et al., 2010). In ChemBO, we use one such molecular fingerprint kernel in our GP. However, molecular fingerprints essentially featurize the graph attributes and might not capture all necessary graphical information. For this reason, we develop a novel graph based similarity measure between molecules which is computed via an optimal transport program. It is most similar to Kandasamy et al. (2018) who use an optimal transport based kernel for neural architecture search. In our experiments, we found that while the performance of the molecular fingerprint kernel and our dissimilarity measure can depend on the objective, they

generally outperform naive strategies which do not use a probabilistic model to inform recommendations.

3 Method

3.1 ChemBO as a Gaussian Process based Bayesian Optimization Algorithm

In this work, we design a Gaussian process (GP) based Bayesian optimization (BO) procedure. The reader can find a detailed review of GP-based BO in (Snoek et al., 2012, Kandasamy et al., 2019a), and specific implementation details (such as the choice of acquisition function) in Appendix B. Here we focus on the two central decisions of designing a GP based BO solution for molecular optimization: *choosing a GP kernel* to specify a GP model, and *designing a method to optimize the acquisition function*. In Section 3.2, we specify GP models, specifically choices for the kernel $\kappa(x, x')$ between two molecules x and x'. Next, in Section 3.3, we describe a method to optimize the acquisition φ_t over the chemical space $\mathcal{X}$. As mentioned previously, when doing so, we will strive to ensure that the recommendations are synthesizable and provide a synthesis recipe. We note that while there are several options for the kernel and the acquisition optimization strategy in conventional domains, such as Euclidean spaces, both tasks are nontrivial in the chemical space and constitute the major contributions of this work. We outline the ChemBO procedure in Algorithm 1.

3.2 Kernel

A natural option would be to simply use one of the existing molecular kernels. Indeed, molecular fingerprint based kernels are known to work well for several applications, and we use that of Ralaivola et al. (2005) in ChemBO. However, they may not be able to capture all graphical information, which motivates us to develop a new similarity measure described below.

An optimal transport based kernel: We will describe a dissimilarity measure $d : \mathcal{X}^2 \rightarrow \mathbb{R}_+$ between molecules. Given such a measure, $\kappa = e^{-\beta d}$ where $\beta > 0$, is a similarity measure which can be used as a kernel. The graphical structure of a molecule determines many of its chemical properties, and as such, our measure will view molecules as graphs. We will define this dissimilarity measure via a matching scheme which attemps to match the atoms in one molecule to another. The matching will only permit matching identical atoms, i.e. carbon atoms can only be matched to carbon atoms, but we will incur penalties for matching atoms with different bond types.

Molecules as graphs: For what follows, it will be convenient to view a molecule M as a graph $M = (A, B)$, which is defined by a set of atoms A (vertices) and a set of bonds B (edges). A bond $(u, v) \in B$

Algorithm 1 ChemBO

1: **Input:** Number of steps T, Initial evaluations D_0
2: **for** $t = 1, \ldots, T$ **do**
3:     Infer posterior $\text{GP}(\mu_t(x), \kappa_t(x, x')|D_{t-1})$
4:     $x_t \leftarrow \text{argmax}_{x \in \mathcal{X}} \varphi_t(x)$ ▷ ACQUISITION-OPT
5:     $f(x_t) \leftarrow$ EVALUATE x_t
6:     $D_t \leftarrow D_{t-1} \cup \{x_t, f(x_t)\}$
    return $x^* \leftarrow \text{argmax}_{x_t \in \{x_1, \ldots, x_T\}} f(x_t)$

is an unordered pair of atoms $u, v \in A$. Each atom $a \in A$ has a label, denoted $\ell_\mathsf{a}(a)$, as does each bond $b \in B$, denoted $\ell_\mathsf{b}(b)$. For example, $\ell_\mathsf{a}(a)$ could take values such as C, H, or O, indicating carbon, hydrogen, or oxygen atoms, while $\ell_\mathsf{b}(b)$ could take values such as SINGLE, DOUBLE, or AROMATIC, indicating single, double, or aromatic bonds. We will also assign weights $w_\mathsf{a}(a) > 0$ for all atoms $a \in A$ of a molecule – our matching scheme will attempt to match the weights in one molecule to another, in order to compute a dissimilarity measure. We will discuss choices for w_a shortly.

Description of the measure: Given two molecules $M_1 = (A_1, B_1)$, $M_2 = (A_2, B_2)$ with n_1, n_2 atoms respectively, let $U \in \mathbb{R}_+^{n_1 \times n_2}$ denote the matching matrix, i.e. $U(i, j)$ is the weight matched between $i \in M_1$ and $j \in M_2$. The dissimilarity measure is the solution of the following program.

$$\underset{U}{\text{minimise}} \quad \varphi_{\mathrm{at}}(U) + \varphi_{\mathrm{st}}(U) + \varphi_{\mathrm{nm}}(U) \qquad (1)$$

$$\text{s.t.} \sum_{j \in A_2} U(i, j) \leq w_\mathsf{a}(i), \ \sum_{i \in A_1} U(i, j) \leq w_\mathsf{a}(j), \ \forall i, j$$

Here, the first term is the atom type penalty φ_{at} which only permits matching similar atoms, i.e. C atoms can only be matched to other C atoms and not H or O atoms. Accordingly, it is defined as

$$\varphi_{\mathrm{at}}(U) = \langle C_{\mathrm{at}}, U \rangle = \sum_{i \in A_1} \sum_{j \in A_2} C_{\mathrm{at}}(i, j) U(i, j),$$

where $C_{\mathrm{at}}(i, j) = 0$ if $\ell_\mathsf{a}(i) = \ell_\mathsf{a}(j)$ and ∞ otherwise. The second term is the bond type penalty term, which, similar to φ_{at}, is given by $\varphi_{\mathrm{st}}(U) = \langle C_{\mathrm{st}}, U \rangle$, where $C_{\mathrm{st}}(i, j)$ is the penalty for matching unit weight from atom $i \in A_1$ to atom $j \in A_2$. We let $C_{\mathrm{st}}(i, j)$ to be the fraction of dissimilar bonds in the union of all bonds. If the atom type and bond type penalties are too large or infinite, we can choose to not match the atoms from one molecule to another. However, we will incur a penalty via the non-matching penalty term φ_{nm}. We set this term to be the sum of weights unassigned in both graphs, i.e.

$$\varphi_{\mathrm{nm}}(U) = \sum_{i \in A_1} \left(w_\mathsf{a}(i) - \sum_{j \in A_2} U(i, j) \right)$$
$$+ \sum_{j \in A_1} \left(w_\mathsf{a}(j) - \sum_{i \in A_1} U(i, j) \right).$$

For two molecules M_1, M_2, we will denote the resulting dissimilarity measure, i.e. the solution of (1), by d.

Design choices: Let us first consider choices for the *weights* $\{w_\mathsf{a}(a)\}_{a \in A}$ in the matching scheme. A natural option here is to let $w_\mathsf{a}(a)$ be the atomic mass of

atom a, which assigns more importance to larger and heavier atoms, which heavily influence the 3D structure of the molecule. Indeed, the molecular mass (sum of atomic masses) is commonly used as an indicator of how drug-like a molecule is in many metrics, including the QED (Bickerton et al., 2012). However, lighter atoms may able to influence other important drug-like properties. For example, the existence of hydroxyl groups (-OH), is strongly correlated with solubility in water, since it can function as an electron donor. Hydrogen (atomic mass 1.008 Au) plays a crucial role in this behaviour, and, setting w_a as above would downplay its significance when compared to, say, carbon (atomic mass 12.011 Au). In such cases, it is more appropriate to treat all atoms types equally, setting $w_\mathsf{a}(a) = 1$ for all atoms.

In addition to d, we also consider a *normalized* version of this dissimilarity,

$$\bar{d}(M_1, M_2) = d(M_1, M_2)/(w_\mathsf{m}(M_1) + w_\mathsf{m}(M_2))$$

where $w_\mathsf{m}(M) = \sum_{a \in A} w_\mathsf{a}(a)$ is the total weight of a molecule $M = (A, B)$. Our experience suggested that using d had a tendency to exaggerate the dissimilarity between larger molecules, simply because a larger amount of atom weights needed to be matched. That said, the size of the molecule affects its drug-like properties (such as its ability to bind with the target), and d accounts for the differences between small and large molecules better than its normalized counterpart.

Combining OT kernels: The two options for the weights $\{w_\mathsf{a}(a)\}_{a \in A}$ and the two options for normalization give rise to four different combinations for our dissimilarity measure. Instead of attempting to find a single best combination, we use an *exponential sum* kernel of the form $\kappa = e^{-\sum_i \beta_i d_i}$, where $\{d_i\}_i$ are the measures obtained for each combination. An ensemble approach of this form allows us to account for all of the factors discussed above when comparing molecules. The β_i terms, which affect the relative importance of each measure, are treated as kernel hyperparameters which can be fitted using maximum likelihood or posterior sampling. It is worth mentioning that while the above form is similar to many popular kernels, it is not known if it is in fact a valid positive definite kernel. However, there are many ways to circumvent this issue in practice; in this work, we project the $n \times n$ matrix of $\kappa(\cdot, \cdot)$ values to the positive-definite cone (Sutherland, 2015, Kandasamy et al., 2018). In Appendix A, we show that (1) can be solved via an optimal transport program (Villani, 2008) and discuss some shortcomings in the proposed dissimilarity measure.

A simple test: Finally, we perform a simple experiment to demonstrate that this dissimilarity metric

Ksenia Korovina[1]   **Sailun Xu**[1]   **Kirthevasan Kandasamy**[2]   **Willie Neiswanger**[1]

aligns with drug-like properties. In Figure 1, we provide the following scatter plot for molecules sampled from the ChEMBL dataset. Each point in the figure is for pair of networks. The x-axis is the dissimilarity measure and the y-axis is the difference in the QED drug likeliness score (Bickerton et al., 2012) and Synthetic accessibility score (Ertl and Schuffenhauer, 2009). We use 100 molecules, giving rise to 5000 pairs. We see that when the measure is small, the difference in the QED score is close to 0. As the measure increases, the points are more scattered. One should expect that for a meaningful distance measure, while molecules that are far apart could have either similar or different properties (as there could be several distinct "clusters"), molecules that are close by should have similar properties, and our measure satisfies this requirement. Additionally, in Appendix A, we provide some interesting T-SNE visualizations for our measure.

3.3 Exploring the Space of Synthesizable Molecules and Optimizing the Acquisition

Our proposal for acquisition optimize involves randomly exploring the space of synthesizable molecules and picking the one with the highest acquisition—this can be viewed as performing a random walk on a *synthesis graph*[3]. For this, consider a setting in a laboratory or an automated experimentation apparatus, where we have access to a limited library of reagents $\mathcal{S}$ and process conditions $\mathcal{Q}$. We will assume that we have access to an oracle SYNTHESIZE which can take as input a set of compounds and process conditions and tell us the set of molecules M produced if these compounds are reacted in the given conditions. In the event, a reaction cannot be effected, it will output NULL. Our procedure for optimizing the acquisition function, described in Algorithm 2, operates as follows. As input, it takes $\mathcal{S}$, $\mathcal{P}$, the number of evaluations n and a set D of evaluations where we have already conducted experiments. First it randomly samples a few molecules S and a few process conditions Q from $\mathcal{S}$ and $\mathcal{Q}$ respectively. It passes them to SYNTHESIZE to generate a set of outputs M. If the synthesis was successful, i.e. if we could generate new molecules that were not evaluated before, they are added to the pool $\mathcal{S}$. It repeats this for n successful steps. At the end, we return the maximizer $\text{argmax}_{x \in \mathcal{S}} \varphi(x)$ of the acquisition φ.

The above procedure relies crucially on the SYNTHESIZE oracle, which can perfectly predict the outcomes of reactions. Alas, no perfect such oracle exists[4]. While

---

[3]A synthesis graph is a directed graph where each node is a molecule, and the parents of this node are the reagents, which when combined, produce the child molecule.

[4]If it did, the entire field of organic chemistry might be expressed as a massive graph search problem.

Algorithm 2 ACQUISITION-OPT: Random Walk Explorer

1: **Input:** n, $\mathcal{S}$, $\mathcal{P}$, D ▷ Steps n, Initial molecules $\mathcal{S}$ and conditions $\mathcal{P}$, Past evaluations D
2: $k = 0$
3: **while** $k \leq n$ **do**
4:    $S \leftarrow$ RAND-SELECT($\mathcal{S}$)   ▷ Select a subset of molecules as reaction inputs
5:    $Q \leftarrow$ RAND-SELECT($\mathcal{Q}$)   ▷ Select a subset of process conditions
6:    $M \leftarrow$ SYNTHESIZE(S, Q)   ▷ Predict product
7:    **if** $M \neq$ NULL **and** $M \backslash D \neq \varnothing$ **then**   ▷ $M \backslash D$ is set difference.
8:       $k \leftarrow k + 1$
9:       $\mathcal{S} \leftarrow \mathcal{S} \cup M \backslash D$ ▷ Add outcomes to the pool
    return $\text{argmax}_{x \in \mathcal{S}} \varphi(x)$

---

outputs of reactions are well known for simple cases, it is impossible to predict outcomes with complex molecules, and in some cases, the outputs may not even be deterministic. Fortunately however, there have been several advances in computational chemistry to predict outcomes of chemical reactions, which can be used in place of the oracle. In our work we use Rexgen (Coley et al., 2019). It should be emphasized that since such predictors are not perfect, so in practice, ChemBO could end up recommending unsynthesizable molecules and/or incorrect synthesis recipes. An additional concern is that the random walk in Algorithm 2 could take long and circuitous paths to arrive at a molecule. Consequently, the synthesis recipe arrived at via Algorithm 2 may not be the most efficient way to synthesize a given molecule. Despite these concerns, we contend that our approach is far more likely to yield synthesizable recommendations than existing approaches. Moreover, an incorrect and/or inefficient recipe can still be a useful guide to a chemist, who might choose to modify it rather than develop from scratch.

4 Experiments

Optimization objectives: We evaluate our methods on two of the most common molecular property functions found in the literature: the QED score (Quantitative Estimate of Drug likeliness) (Bickerton et al., 2012), and Pen-logP score (penalized octanol-water partition coefficient). The former is computed using the procedure described in Bickerton et al. (2012), while the latter is computed using the following formula: Pen-logP$(m) = $ logP$(m) - $ SA-Score$(m) - $ ring-penalty(m), where logP is the octanol-water partition coefficient (Miller et al., 1985), SA-Score is the synthetic accessibility score (Ertl and Schuffenhauer, 2009), and ring penalty is the number of long cycles. The partition coefficent measures solubility in water, SA-score is a negative proxy for synthesizability (lower is easier), and large rings might indicate that molecules
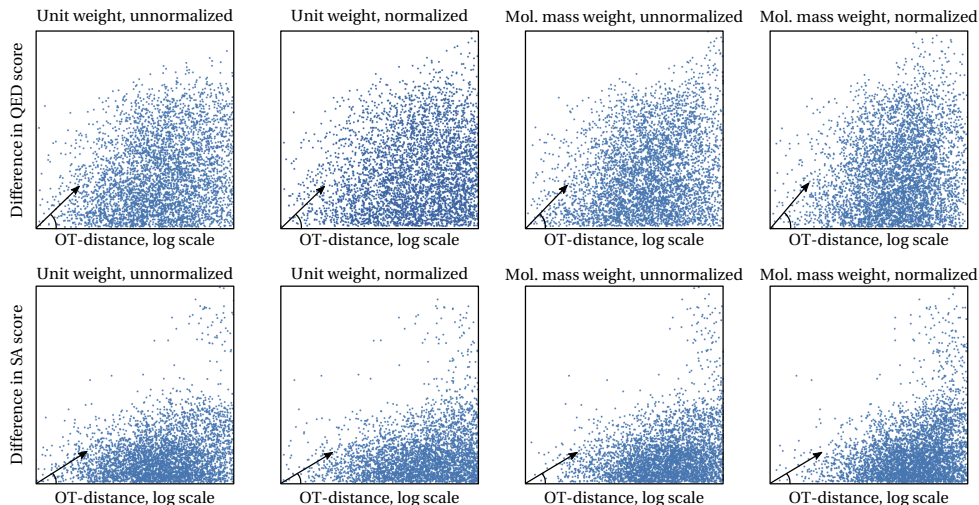
Figure 1: Each point in the scatter plot indicates the dissimilarity measure between the molecules (x axis) and the difference in the QED score and SA score (y axis). The four images are for the four different combinations of the distance. See text for interpretation.

are not stable once synthesized. Note that the range of penalized logP is unbounded, and QED is constrained to values between 0 and 1. In implementing Pen-logP, we followed the exact implementation of this metric in (Jin et al., 2018a). We mention that these metrics may not be the most relevant to actual drug discovery applications – for instance, they do not account for how well the molecule binds with the given target of interest. However, their use in literature makes them good comparative benchmarks.

Methods: We compare three instantiations of ChemBO: 1. using a molecular fingerprint kernel (fingerprint), 2. using the dissimilarity metric described in Section 3.2 (ot-dist), and 3. using a linear combination of fingerprint and ot-dist (sum-kernel). The fingerprint based kernel computes Tanimoto similarity between topological (path-based) fingerprints of given molecules (Landrum et al., 2006). The sum-kernel is a kernel given by $k(x, y) = \alpha_1 \cdot \mathsf{fingerprint}(x, y) + \alpha_2 \cdot \mathsf{ot\text{-}dist}(x, y)$, where $\alpha_i \in [0, 10]$ are kernel parameters fitted at training time. In addition, we also also compare to the random walk explorer (rand) in Algorithm 2, which operates exactly as described except returns the maximum of the function f in step 9 (instead of the acquisition). This can be viewed as a simple random search baseline which attempts to optimize in the space of synthesizable molecules. We wish to reiterate that to our best knowledge other work do not enforce a hard constraint on synthesizability, nor do they require that a recipe for synthesis be provided. Hence, they are not directly comparable to our method. However, we quote results on the best QED and Pen-logP values from their papers for comparison. Moreover, we include an additional virtual screening baseline, which is allowed

to randomly sample and evaluate molecules from the entire dataset, instead of just the compounds reachable by synthesis from the starting pool.

Experimental set up: As stated previously, we wish to emulate a setting where a chemist has to work with the reagents and process conditions available to her. We choose 20 randomly chosen molecules from the openly available ChEMBL database as our initial set of reagents. The maximum QED score of the initial pool was 0.858 (when QED > 0.9, it is typically considered high). As the process conditions for the random explorer, we use all the process conditions available in Rexgen. We bootstrap all three methods listed above by evaluating the metric (QED or Pen-logP) on this initial set, and then execute the methods for 80 iterations, totaling 100 evaluations of f. We describe additional details on our BO implementation in Appendix C.

4.1 Results & Discussion

Main Results: In Figure 2, we plot the number of iterations against the optimal found value by each method over 80 function evaluations for both QED and Pen-logP. We provide the final optimal values for each method in Table 1. The results were obtained by averaging over 5 independent runs. ChemBO methods, fingerprint, ot-dist and sum-kernel, all outperform the naive random walk strategy on both tasks, validating the use of model based Bayesian strategies for this task. ot-dist does better than fingerprint on the QED score while vice versa on Pen-logP, and sum-kernel provides a good adaptive trade-off between them that works well for both benchmarks, and also has lower variance.

Optimal Molecules & Synthesis Recipes: Figure 3 illustrate some optimal molecules found for the QED and

Ksenia Korovina[1]   **Sailun Xu**[1]   **Kirthevasan Kandasamy**[2]   **Willie Neiswanger**[1]
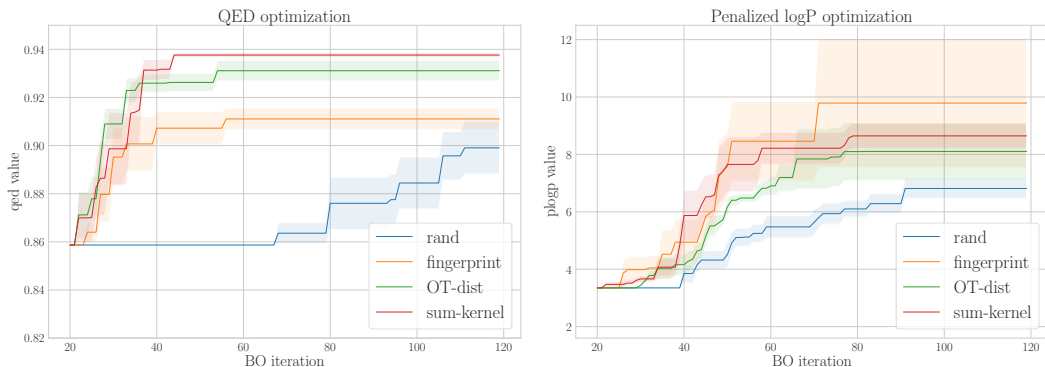
Figure 2: Results comparing the three methods described in the beginning of Section 4. We plot the number of iterations (after initialization) against the highest found QED (left) and Pen-LogP (right) values by each method. Higher is better in both cases. All curves were produced by averaging over 5 independent runs. The shaded regions indicate one standard error.

Pen-logP objectives by ChemBO. For the most part, optimal QED molecules were found by ot-dist, while optimal Pen-LogP molecules by fingerprint. Interestingly, molecules with high QED scores tend to be simpler than those with high Pen-logP scores. In Appendix C, we visualize and discuss the synthesis recipes for some of the optimal molecules.

	rand	fingerprint	ot-dist	sum-kernel
QED	0.90 ± 0.01	0.91 ± 0.01	0.93 ± 0.01	$\mathbf{0.94 \pm 0.01}$
P.logP	6.81 ± 0.34	$\mathbf{9.79 \pm 2.26}$	8.10 ± 1.01	8.65 ± 0.43

Table 1: Final value for QED and Pen-logP over 80 eval-s

Reliability of synthesis paths: A thorough validation of the synthesis paths proposed by ChemBO would require performing actual synthesis in lab conditions. However, we can perform the following sanity checks. Using synthetic accessibility score Ertl and Schuffenhauer (2009) as a proxy for ease of synthesis of the resulting molecule, we can evaluate plausibility of the end result. The results presented in Table 2 show that the end molecule is within a reasonable range from averages in curated datasets, and the minimum score over synthesis path is well below these values.

ChEMBL	ZINC250k	Avg SA score	Min path SA
2.73 ± 0.65	3.1 ± 0.77	3.77 ± 1.46	2.5 ± 0.44

Table 2: Synthetic accessibility scores over 50 samples/runs over the datasets, optimal results from ChemBO, and average minimum over produced synthesis paths.

Novel Molecules: As ChemBO runs, we compute the fraction of molecules that do not appear in the entire ChEMBL dataset. For ot-dist optimizing QED, on average 95.64% molecules are novel, for fingerprint 96.84%; and for Pen-logP 78% and 87.67%, respectively. This indicates that ChemBO is able to explore the chemical space well, despite the constraints on synthesizability.

Comparison with existing work: In Table 3, we compare ChemBO to state-of-the-art methods adopting reinforcement learning or generative modeling techniques (Jin et al., 2018a, You et al., 2018, Guimaraes et al., 2017, Zhou et al., 2018). We use the same evaluation strategy as in these works, reporting top scores across several runs. It is interesting to compare the number of QED/Pen-LogP evaluations required by some of these methods. Guimaraes et al. (2017) is trained with supervision on a random subset of $5K$ molecules from the ZINC dataset (Irwin and Shoichet, 2005), and hence uses at least $5K$ evaluations. VAE in Jin et al. (2018a) is trained on full ZINC dataset ($\approx 250k$ molecules) in an unsupervised manner, and then $25k$ evaluations to train a GP and optimize the given objective. Both You et al. (2018) and Zhou et al. (2018) train RL policies using all the $250K$ molecules in the ZINC dataset and incorporate the penalized logP or QED score as part of the reward, hence making at least that many evaluations. In contrast, in our ChemBO experiments, we ran 100 BO iterations using two different kernels for 5 trials, totalling 1000 function evaluations. It should be emphasised that the above methods are not designed to keep the number of QED/Pen-logP evaluations to a minimum, and in fact, are tools developed for very different settings. Yet, it speaks to the efficiency of ChemBO, that we were able to obtain better or comparable values than the above work in significantly fewer evaluations, particularly given our more stringent conditions on synthesizability.

Virtual screening baseline: There are two possible ways to translate the virtual screening experiment into a computational simulation. Firstly, we can assume that a fixed number of compounds is available to an experimenter, and we can either synthesize from them, or use them directly for screening. This baseline is already part of the results above, since we spend the

	ORGAN (Guimaraes et al., 2017)	JT-VAE (Jin et al., 2018a)	GCPN (You et al., 2018)	MolDQN (Zhou et al., 2018)	ChemBO (ours)
QED	0.896	0.925	0.948	0.948	0.941
Pen-logP	3.63	5.30	7.98	11.84	18.39
# evaluations	$\geq 5K$	$275K$	$\geq 25K$	$\geq 25K$	100

Table 3: The best QED and Pen-LogP scores reported from prior work. For ChemBO, we use the best value obtained across the 5 trials for both fingerprint and ot-dist. Note that not all methods treat this as an optimization problem and do not impose conditions on synthesizability as we do.

(a) QED 0.92083 (b) QED 0.92145 (c) QED 0.94023 (d) QED 0.94087

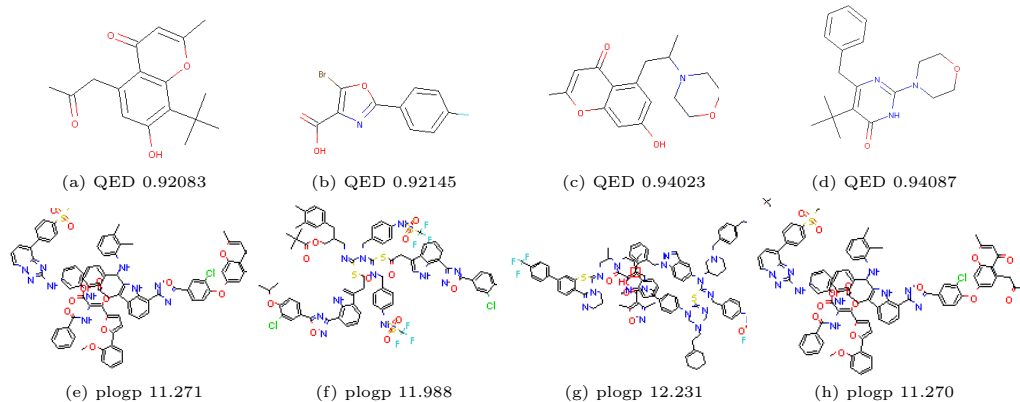(e) plogp 11.271 (f) plogp 11.988 (g) plogp 12.231 (h) plogp 11.270

Figure 3: A random sample of optimal molecules and values found by ChemBO. In the top row, we show those with the highest QED scores, and in the bottom row we show the same for Pen-logP.

initial BO budget on finding the maximum value of the initial pool ("screening" it), and after that all optimizers have to improve upon that value. In the second version, we compare virtual screening and ChemBO/rand for the same number of evaluations. Now we start with a pool, and then sample compounds *outside* of that pool from the rest of the dataset. This corresponds to a situation where the experimenter purchases the compounds randomly in addition to the ones she has; in theory, this could lead to a larger optimum due to accessing more dataset molecules than in our setup, i.e. a larger search space. The results obtained by simulating such an experiment are shown in Figure 4. Even with using more samples, these values are worse than the numbers in Table 1.

QED	penalized logp
0.922 ± 0.013	5.34 ± 0.973

Table 4: Virtual screening baseline: means and standard deviations over 10 replications.

5   Conclusion

In real world use cases for computational and statistical methods for molecular optimization, an algorithm recommends a molecule, which is synthesized, tested, and the results returned to the algorithm. These results are then used by the algorithm to inform future recommendations. In order to achieve full automation, computational methods should strive to ensure that such recommendations are synthesizable and provide a recipe to do so. ChemBO, which uses BO techniques to design recommendations, is a first step towards

this ambitious goal. Our experiments indicate that model-based Bayesian methods can outperform naive alternatives for this problem. We study kernels for ChemBO and find that the ot-dist kernel we propose can outperform standard kernels in some tasks, and that combining it with other kernels (such as fingerprint) can be a lower-variance alternative that performs well across benchmarks. In addition, on two benchmark objectives, we are able to get competitive or better scores than existing work, while using significantly less evaluations of the objective. While our approach is constrained by limitations of current synthesis predictors, it can still be a very useful guide to a practitioner.

Improving the reliability of synthesis predictors and developing smarter methods to explore the chemical space are interesting avenues for future research, which will improve the efficacy of our framework. Another direction is to use ChemBO (and other methods) to optimize for the ability to bind with a given target. Separately, it would also be interesting to view the optimization budget not in terms of the number of compounds tested, but rather in terms of the number of additional synthesis steps required (it is plausible that synthesis is the bottleneck, not the cost of testing). These goals bring up some new interesting methodological questions for Bayesian optimization, and the potential for new modeling paradigms Kandasamy et al. (2019b), Neiswanger et al. (2019). Finally, it would be interesting to extend and test our framework on biologics and other molecular optimization problems in drug discovery and materials science.

Ksenia Korovina[1]   Sailun Xu[1]   Kirthevasan Kandasamy[2]   Willie Neiswanger[1]

References

Daniel C. Elton, Zois Boukouvalas, Mark D. Fuge, and Peter W. Chung. Deep learning for molecular generation and optimization - a review of the state of the art, 2019.

Pavel G Polishchuk, Timur I Madzhidov, and Alexandre Varnek. Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of computer-aided molecular design*, 27(8):675–679, 2013.

Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 2018. doi: 10.1021/acscentsci.7b00572.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. *International Conference on Machine Learning, 2018*, 2018a.

Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de-novo drug design. *Science Advances, 2018, vol. 4, no. 7, eaap7885*, 2017. doi: 10.1126/sciadv.aap7885.

Jiaxuan You, Bowen Liu, Rex Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in Neural Information Processing Systems, 2018*, 2018.

Anton O Oliynyk, Erin Antono, Taylor D Sparks, Leila Ghadbeigi, Michael W Gaultois, Bryce Meredig, and Arthur Mar. High-throughput machine-learning-driven synthesis of full-heusler compounds. *Chemistry of Materials*, 28(20):7324–7331, 2016.

Julia Ling, Maxwell Hutchinson, Erin Antono, Sean Paradiso, and Bryce Meredig. High-dimensional materials and process optimization using data-driven experimental design with well-calibrated uncertainty estimates. *Integrating Materials and Manufacturing Innovation*, 6(3): 207–217, 2017.

Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design. *Advances in Neural Information Processing Systems, 2017*, 2017.

Liva Ralaivola, Sanjay J Swamidass, Hiroto Saigo, and Pierre Baldi. Graph kernels for chemical informatics. *Neural networks*, 18(8):1093–1110, 2005.

Georg Hinselmann, Nikolas Fechner, Andreas Jahn, Matthias Eckert, and Andreas Zell. Graph kernels for chemical compounds using topological and three-dimensional local atom pair environments. *Neurocomputing*, 74(1-3):219–229, 2010.

Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

Weininger D Anderson E, Veith GD. Smiles: A line notation and computerized interpreter for chemical structures. Technical Report EPA/600/M-87/021, EPA, Environmental Research Laboratory-Duluth, 1987.

Seokho Kang and Kyunghyun Cho. Conditional molecular design with deep generative models. *Journal of Chemical Information and Modeling 59(1): 43-52, 2019*, 2018. doi: 10.1021/acs.jcim.8b00263.

Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Car-

los Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *Preprint at arXiv:1705.10843*, 2017.

Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *arXiv preprint arXiv:1810.08678*, 2018.

Wengong Jin, Kevin Yang, Regina Barzilay, and Tommi Jaakkola. Learning multimodal graph-to-graph translation for molecular optimization. *International Conference on Learning Representations 2019*, 2018b.

Truong Son Hy, Shubhendu Trivedi, Horace Pan, Brandon M Anderson, and Risi Kondor. Predicting molecular properties with covariant compositional networks. *The Journal of chemical physics*, 148(24):241745, 2018.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1704–1713. JMLR. org, 2017.

Matt J. Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. *ICML'17: Proceedings of the 34th International Conference on Machine Learning*, 2017.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):8, 2009.

Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Costas Bekas, and Alpha A Lee. Molecular transformer for chemical reaction prediction and uncertainty estimation. *Machine Learning for Molecules and Materials workshop, NeurIPS 2018*, 2018a.

Jennifer N Wei, David Duvenaud, and Alán Aspuru-Guzik. Neural networks for the prediction of organic chemistry reactions. *ACS central science*, 2(10):725–732, 2016.

Connor W. Coley, Wengong Jin, Luke Rogers, Timothy F. Jamison, Tommi S. Jaakkola, William H. Green, Regina Barzilay, and Klavs F. Jensen. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem. Sci.*, 2019.

Jonathan H Chen and Pierre Baldi. No electron left behind: a rule-based expert system to predict chemical reactions and reaction mechanisms. *Journal of chemical information and modeling*, 49(9):2034–2043, 2009.

James Law, Zsolt Zsoldos, Aniko Simon, Darryl Reid, Yang Liu, Sing Yoong Khew, A Peter Johnson, Sarah Major, Robert A Wade, and Howard Y Ando. Route designer: a retrosynthetic analysis tool utilizing automated retrosynthetic rule generation. *Journal of chemical information and modeling*, 49(3):593–602, 2009.

Ola Engkvist, Per-Ola Norrby, Nidhal Selmi, Yu-hong Lam, Zhengwei Peng, Edward C Sherer, Willi Amberg, Thomas Erhard, and Lynette A Smyth. Computational prediction of chemical reactions: current status and outlook. *Drug discovery today*, 23(6):1203–1218, 2018.

Wengong Jin, Connor W. Coley, Regina Barzilay, and Tommi Jaakkola. Predicting organic reaction outcomes with weisfeiler-lehman network. *Conference on Neural Information Processing Systems*, 2017.

Philippe Schwaller, Theophile Gaudin, David Lanyi, Costas Bekas, and Teodoro Laino. "found in translation": predicting outcomes of complex organic chemistry reactions

using neural sequence-to-sequence models. *Chemical science*, 9(28):6091–6098, 2018b.

Tao Lei, Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Deriving neural architectures from sequence and graph kernels. *ICML*, 2017.

John Bradshaw, Brooks Paige, Matt J Kusner, Marwin HS Segler, and José Miguel Hernández-Lobato. A model to search for synthesizable molecules. *Advances in Neural Information Processing Systems, 2019*, 2019.

Walter D Wallis, Peter Shoubridge, M Kraetz, and D Ray. Graph distances using graph union. *Pattern Recognition Letters*, 22(6-7):701–704, 2001.

Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML*, volume 2, pages 315–322, 2002.

Colin White, Willie Neiswanger, and Yash Savani. Bananas: Bayesian optimization with neural architectures for neural architecture search. *arXiv preprint arXiv:1910.11858*, 2019.

Dougal J Sutherland. *Scalable, Active and Flexible Learning on Distributions*. PhD thesis, PhD thesis, Carnegie Mellon University Pittsburgh, PA, 2015.

Ke Liu, Xiangyan Sun, Lei Jia, Jun Ma, Haoming Xing, Junqiu Wu, Hua Gao, Yax Sun, Florian Boulnois, and Jie Fan. Chemi-net: a graph convolutional network for accurate drug property prediction. *International Journal of Molecular Sciences*, 2018.

Wen Torng and Russ B Altman. Graph convolutional neural networks for predicting drug-target interactions. *bioRxiv*, page 473074, 2018.

Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabas Poczos, and Eric Xing. Neural Architecture Search with Bayesian Optimisation and Optimal Transport. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems*, 2012.

Kirthevasan Kandasamy, Karun Raju Vysyaraju, Willie Neiswanger, Biswajit Paria, Christopher R. Collins, Jeff Schneider, Barnabas Poczos, and Eric P. Xing. Tuning hyperparameters without grad students: Scalable and robust bayesian optimisation with dragonfly, 2019a.

G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90, 2012.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Michele M Miller, Stanley P Wasik, Guo Lan Huang, Wan Ying Shiu, and Donald Mackay. Relationships between octanol-water partition coefficient and aqueous solubility. *Environmental science & technology*, 19(6): 522–529, 1985.

Greg Landrum et al. Rdkit: Open-source cheminformatics, 2006.

John J Irwin and Brian K Shoichet. Zinc- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, 45(1): 177–182, 2005.

Kirthevasan Kandasamy, Willie Neiswanger, Reed Zhang, Akshay Krishnamurthy, Jeff Schneider, and Barnabás Póczos. Myopic posterior sampling for adaptive goal oriented design of experiments. In *International Conference on Machine Learning*, pages 3222–3232, 2019b.

Willie Neiswanger, Kirthevasan Kandasamy, Barnabas Poczos, Jeff Schneider, and Eric Xing. Probo: a framework for using probabilistic programming in bayesian optimization. *arXiv preprint arXiv:1901.11515*, 2019.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.