

ChemBO: Bayesian Optimization of Small Organic Molecules with Synthesizable Recommendations

Ksenia Korovina

Carnegie Mellon University
kkorovin@cs.cmu.edu

Sailun Xu

Carnegie Mellon University
sailunx@andrew.cmu.edu

Kirthivasan Kandasamy

Carnegie Mellon University
kandasamy@cs.cmu.edu

Willie Neiswanger

Carnegie Mellon University
willie@cs.cmu.edu

Barnabás Póczos

Carnegie Mellon University
bapocz@cs.cmu.edu

Jeff Schneider

Carnegie Mellon University
schneide@cs.cmu.edu

Eric P. Xing

Carnegie Mellon University
epxing@cs.cmu.edu

Abstract

We describe ChemBO, a Bayesian Optimization framework for generating and optimizing organic molecules for desired molecular properties. This framework is useful in applications such as drug discovery, where an algorithm recommends new candidate molecules; these molecules first need to be synthesized and then tested for drug-like properties. The algorithm uses the results of past tests to recommend new ones so as to find good molecules efficiently. Most existing data-driven methods for this problem do not account for sample efficiency and/or fail to enforce realistic constraints on synthesizability. In this work, we explore existing kernels for molecules in the literature as well as propose a novel kernel which views a molecule as a graph. In ChemBO, we implement these kernels in a Gaussian process model. Then we explore the chemical space by traversing possible paths of molecular synthesis. Consequently, our approach provides a proposal synthesis path every time it recommends a new molecule to test, a crucial advantage when compared to existing methods. 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 [1]. 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 the adoption of machine learning techniques for such tasks, due to their effectiveness in modeling structure-property relations of molecules, as well as due to limitations of traditional methods in computational chemistry methods in effectively exploring the large and complex chemical space (space of chemical molecules). For instance, the number of drug-like molecules is estimated to be between 10^{23} and 10^{60} [2], 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 [3–7], one promising approach is to treat this task as a black-box optimization problem (e.g. [8, 9]). Here, we assume the existence of a function $f : \mathcal{X} \rightarrow \mathbb{R}$ 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 $\operatorname{argmax}_{x \in \mathcal{X}} f(x)$. 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 conducting 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, specifically focusing on small¹ 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. Moreover, this recipe would take into consideration the reagents and resources available. While these requirements might seem stringent, they are necessary if we are to achieve the moonshot goal of fully autonomous robotic systems that can design new molecules and materials. Moreover, even in cases where the recommended molecules are synthesized manually, providing a recipe that adheres to these constraints can be a helpful guide to the chemist and greatly reduce the amount of manual work required. ChemBO is a first step towards achieving this goal. Our contributions in this work are:

1. We use a Gaussian process (GP) to model structure-property relations in molecules. For the GP kernel, we use prior work on molecular fingerprints [10, 11] 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 [12]. This approach not only ensures that each recommended molecule is chemically valid, but also provides a synthesis recipe².
3. In our experiments, we demonstrate that ChemBO outperforms simpler alternatives for synthesizable optimization which do not use a probabilistic model to guide search. The final values for the popular QED [13] and penalized partition coefficient [9] 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>.

Related Work

Optimization: SMILES strings [14], which describe the structure of molecules as a string, are a common representation used in machine learning techniques for molecular optimization [3, 15]. One of the main reason for its adoption is that SMILES strings allows one to use existing NLP machinery largely unchanged. Recently, graph representations for molecules have become popular. Most recent methods adopting this representation use generative models or reinforcement learning to construct a molecular graph, and optimize the property in question while attempting to maintain validity [4, 6, 16–18]. In learning representations for molecules, they draw on the methods that process graph data directly, such as graph neural networks [6, 18] and covariant compositional networks [19]. 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 [20], and is not desirable in settings where each evaluation might involve several laboratory experiments. BO methods, which are particularly well suited for optimization problems with expensive evaluations, are sparsely represented in the field. Gómez-Bombarelli et al. [3], Jin et al. [4], Kusner et al. [21] learn a Euclidean representation for molecules and perform BO on this space, while Griffiths and Hernández-Lobato [9] 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 constraints on this score [9] or incorporating this into f along with the other properties [3]. However, synthesizability scores are not always reliable. For example, Gómez-Bombarelli et al. [3] found that their autoencoder produced a large number of molecules with unrealistically large carbon rings when using the SA synthesizability score [22] 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

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

²We will qualify this statement later in Section 2.3.

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. While one could consider using retrosynthesis techniques [23] for the latter, they may not always be reliable, and moreover, one can run into the same availability problems mentioned above.

We leverage a large and separate direction of research which use ML techniques to predict outcomes of chemical reactions [23–27] (see Engkvist et al. [28] 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. [25], 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 [29, 30]. One such method in this class, and the one we adopt in this work to explore the chemical space, is Rexgen [26]. It proceeds in two stages: first, reactive sites are predicted using a Weisfeiler-Lehman network with global attention [31] on the graph representation of reaction inputs; next, possible configurations of connectivity changes in reactive sites are scored with a Weisfeiler-Lehman Difference network [29].

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 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. [32] 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 cannot be applied 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 for defining kernels and similarity metrics between graphs [33–35], we cannot directly adopt them in our work since molecules have more complex properties in addition to graphical structure. There have been a variety of neural network based graph similarity measures proposed for molecules [36–38]. 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 [10, 11], which have been found to outperform conventional graph kernels on some tasks [39]. 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. [40] 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.

2 Method

We now describe our proposed framework. We begin with a brief review of Bayesian optimization.

2.1 A Review of Gaussian Processes and Bayesian Optimization

A Gaussian process (GP), written $GP(\mu, \kappa)$, is a distribution over functions, commonly used as a nonparametric prior for a function f . It is specified by a mean function $\mu : \mathcal{X} \rightarrow \mathbb{R}$ and covariance function (kernel) $\kappa : \mathcal{X}^2 \rightarrow \mathbb{R}$. Given observations $D_n = \{(x_i, y_i)\}_{i=1}^n$, where $y_i = f(x_i) + \epsilon_i$ and $\epsilon_i \sim \mathcal{N}(0, \eta^2)$ for all i , the posterior $f|D_n$ is also GP with some mean μ_n and variance κ_n , both of which can be computed in closed form [41]. In empirical Bayes methods, the GP hyperparameters such as the parameters of the kernel κ and noise variance η^2 , are typically fitted by maximum

likelihood or posterior sampling techniques [42, 43]; moreover, it is customary to assume that the prior mean is a constant function, where this constant value is fitted along with the other parameters.

Assume we are tasked with optimizing a function $f : \mathcal{X} \rightarrow \mathbb{R}$ over some domain \mathcal{X} by making repeated evaluations to f . BO methods impose a prior belief distribution on f , and use the posterior given the results of the past evaluations to decide future points for evaluation. They are usually more computationally expensive than other optimization methods, but can be significantly more sample efficient in practice. This makes them ideal for applications such as drug discovery and materials optimization, where each real world experiment could take several hours and incur significant monetary cost. BO is an adaptive sequential procedure, which usually proceeds as follows at time step t : we first build a surrogate *acquisition* function $\varphi_t(x)$ derived from the current posterior for f ; we then optimize $\varphi_t(x)$ to obtain the next point for evaluation $x_t = \arg \max \varphi_t$, and query the optimal point $y_t = f(x_t)$. After the query, the posterior is updated and we proceed to the next step. The acquisition φ_t trades off between exploration to gain more information about f , exploiting the information already gained to find the maximum of f . There are many choices for the acquisition, including EI [44], UCB [45, 46], TTEI [47] and more [48–50].

GPs are one of the most popular options for the prior in BO. To design a GP based BO solution for molecular optimization, we solve two important challenges in this work. First, in Section 2.2, we specify GP models, specifically choices for the kernel $\kappa(x, x')$ between two molecules x and x' . Next, in Section 2.3, 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 mention that while there are several options for the kernel and optimizing the acquisition in conventional domains, such as Euclidean spaces, both tasks are nontrivial in the chemical space and constitute the major contributions of this work.

2.2 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. [10] in ChemBO. However, they may not be able to capture all graphical information, which motivates us to develop a new similarity measure described below.

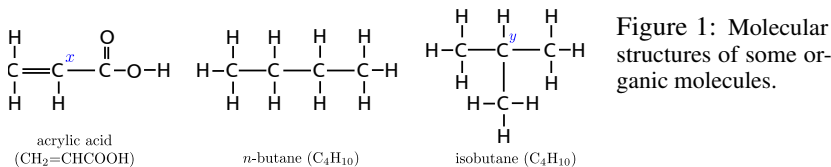
An Optimal Transport Based Kernel

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. For example, both n-butane and isobutane have the same number of C and H atoms (C_4H_{10}), but have different chemical properties due to different structure. We will define this dissimilarity measure via a matching scheme which attempts 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 $(u, v) \in B$ is an unordered pair of atoms $u, v \in A$. Each atom $a \in A$ has a label, denoted $\ell_a(a)$, as does each bond $b \in B$, denoted $\ell_b(b)$. For example, $\ell_a(a)$ could take values such as C, H, or O, indicating carbon, hydrogen, or oxygen atoms, while $\ell_b(b)$ could take values such as SINGLE, DOUBLE, or AROMATIC, indicating single, double, or aromatic bonds. In Figure 1, atoms are represented by letters indicating their label; bonds are represented by lines where single lines indicate single bonds and double lines indicate double bonds. We will also assign weights $w_a(a) > 0$ 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.

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



$$\text{subject to } \sum_{j \in A_2} U(i, j) \leq w_a(i), \sum_{i \in A_1} U(i, j) \leq w_a(j), \forall i, j$$

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

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

where $C_{\text{at}}(i, j) = 0$ if $\ell_a(i) = \ell_a(j)$ and ∞ otherwise. The second term is the bond type penalty term, which, similar to φ_{at} , is given by $\varphi_{\text{st}}(U) = \langle C_{\text{st}}, U \rangle$, where $C_{\text{st}}(i, j)$ is the penalty for matching unit weight from atom $i \in A_1$ to atom $j \in A_2$. We let $C_{\text{st}}(i, j)$ to be the fraction of dissimilar bonds in the union of all bonds; for example, in Figure 1, $C_{\text{st}}(x, y) = 3/5$, since, between them they have one C-H bond, three C-C bonds, and one C=C bond, of which the C-H bond and one C-C bond are common. 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.

$$\varphi_{\text{nm}}(U) = \sum_{i \in A_1} (w_a(i) - \sum_{j \in A_2} U(i, j)) + \sum_{j \in A_1} (w_a(j) - \sum_{i \in A_1} U(i, j)).$$

For two given 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* $\{w_a(a)\}_{a \in A}$ in the matching scheme. A natural option here is to let $w_a(a)$ 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 [13]. However, lighter atoms may be 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 $w_a(a) = 1$ for all atoms.

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

$$\bar{d}(M_1, M_2) = d(M_1, M_2) / (w_m(M_1) + w_m(M_2))$$

where $w_m(M) = \sum_{a \in A} w_a(a)$ is the total weight of a molecule $M = (A, B)$. Our experience suggested that using \bar{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 $\{w_a(a)\}_{a \in A}$ 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 [35, 40]. In Appendix A, we

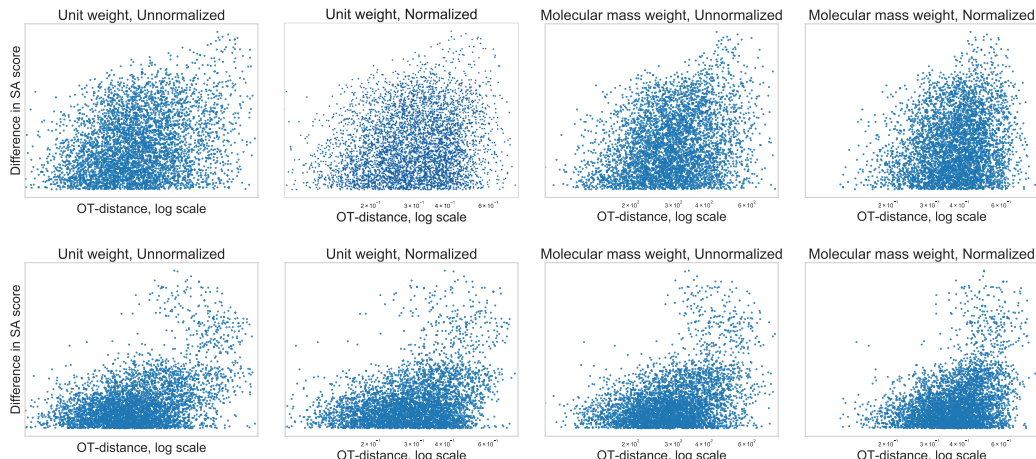


Figure 2: Each point in the scatter plot indicates the dissimilarity measure between the molecules (x axis) and the difference in the QED score and SA score (y axis). The four figures are for the four different combinations of the distance. See text for interpretation.

show that (1) can be solved via an optimal transport program [51] and discuss some shortcomings in the proposed dissimilarity measure.

A Simple Test: Finally, we perform a simple experiment to demonstrate that this dissimilarity metric correlates with drug-like properties. In Figure 2, 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 [13] and Synthetic accessibility score [22]. We used 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 while molecules that are far apart could perform similarly or differently, similar molecules should perform similarly, and our measure satisfies this requirement. This demonstrates that it can be incorporated in a BO framework to optimize a molecule for its drug likeliness. In Appendix A, we provide some interesting T-SNE visualizations for this measure.

2.3 Exploring the Space of Synthesizable Molecules and Optimizing the Acquisition

Our proposal to optimize the acquisition randomly explores the space of synthesizable molecules and picks 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, described in Algorithm 1, 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 $\operatorname{argmax}_{m \in \mathcal{S}} \varphi(m)$ of the acquisition φ .

The above procedure relies crucially on the SYNTHESIZE oracle, which can perfectly predict the outcomes of reactions. Alas, no such oracle exists⁴. While 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

³A 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.

⁴If it did, the entire field of organic chemistry would be just a massive graph search problem.

Algorithm 1 Random Walk Explorer

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

computational chemistry to predict outcomes of chemical reactions, which can be used in place of the oracle. In our work we use Rexgen [26]. 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 1 could take long and circuitous paths to arrive at a molecule – consequently, the synthesis recipe arrived at via Algorithm 1 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. Developing synthesis predictors is an active area of research [24, 30], and as such methods become more reliable, so will the efficacy of our framework. Moreover, an incorrect and/or inefficient recipe can still be a useful guide to a chemist (who might choose to modify it), and in most cases is better than expecting the chemist to develop a recipe of her own from scratch.

Finally, it is worth asking if one could consider smarter approaches to exploring the space, instead of random search. While this is an interesting question worth investigating, it is not clear if a relatively simple solution exists. For instance, there are numerous examples where reacting two molecules produces an output that is unlike either of the reagents with very different chemical properties. Hence, say, a greedy approach which attempts to expand on the frontier of those with a high acquisition value may not necessarily do better than random search. Next, we provide experimental results.

3 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) [13], and Pen-logP (penalized octanol-water partition coefficient). The former is computed using the procedure described in Bickerton et al. [13], while the latter is computed using the following formula: $\text{Pen-logP}(m) = \log\text{P}(m) - \text{SA-Score}(m) - \text{ring-penalty}(m)$, where $\log\text{P}$ is the octanol-water partition coefficient [52], SA-Score is the synthetic accessibility score [22], and ring penalty is the number of long cycles. The partition coefficient measures solubility in water, SA-score is a negative proxy for synthesizability, and large rings might indicate that molecules are not stable once synthesized. Note that the range of penalized logP is unbounded, and QED is constrained to values between 0 and 1. For Pen-logP, we followed the exact implementation of this metric in [4] in our implementation. We mention that these metrics may not be perfectly aligned for a given drug discovery problem – after all, they do not account for how well the molecule binds with the given target of interest. However, they provide good benchmark to compare different optimization methods.

Methods: We compare two instantiations of ChemBO: one using a molecular fingerprint kernel (fingerprint) and the other using the dissimilarity metric described in Section 2.2 (ot-dist). The fingerprint based kernel computes Tanimoto similarity between topological (path-based) fingerprints of given molecules [53]. In addition, we also compare to the random walk explorer (rand) in Algorithm 1, 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

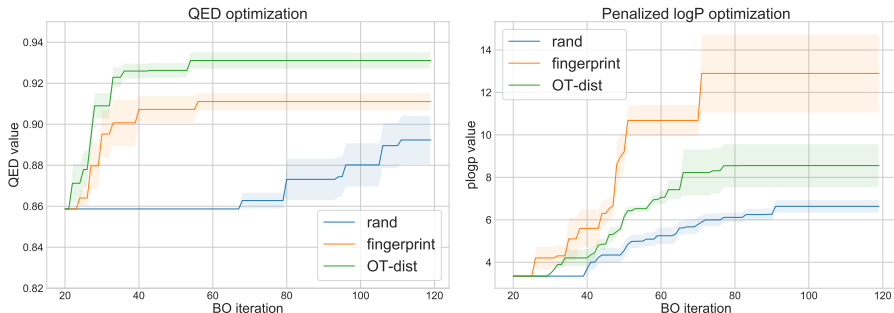


Figure 3: Results comparing the three methods described in the beginning of Section 3. We plot the number of iterations (after initialization) against the highest found QED (left) and Pen-LogP (right) values by each method. Higher is better in both cases. All curves were produced by averaging over 5 independent runs. The shaded regions indicate one standard error.

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 B.

	rand	fingerprint	ot-dist
QED	0.892 \pm 0.012	0.911 \pm 0.005	0.931 \pm 0.004
Pen-logP	6.635 \pm 0.320	12.895 \pm 1.863	8.227 \pm 1.105

Table 1: Final optimal value for QED and Pen-logP over 80 evaluations.

Results & Discussion: In Figure 3, 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. The ChemBO methods, fingerprint and ot-dist, both 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, indicating that the choice of the kernel can be important for the application. It is worth mentioning that, generally speaking, the QED score is considered a more holistic view of drug likeliness than Pen-logP [13].

Optimal Molecules & Synthesis Recipes: Figure 4 illustrate some optimal molecules found for the QED and 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 B, we visualize and discuss the synthesis recipes for some of the optimal molecules.

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 [22] as a proxy for ease of synthesis (higher is better) of the resulting molecule, we can evaluate plausibility of the end result; additionally, we compute the minimum SA score over the synthesis graph as a proxy for plausibility of the entire synthesis recipe. The results presented in Table 2 shows that the end result is on average *easier* to synthesize than the molecules in the ChEMBL and ZINC datasets, and the minimum score is within reasonable range.

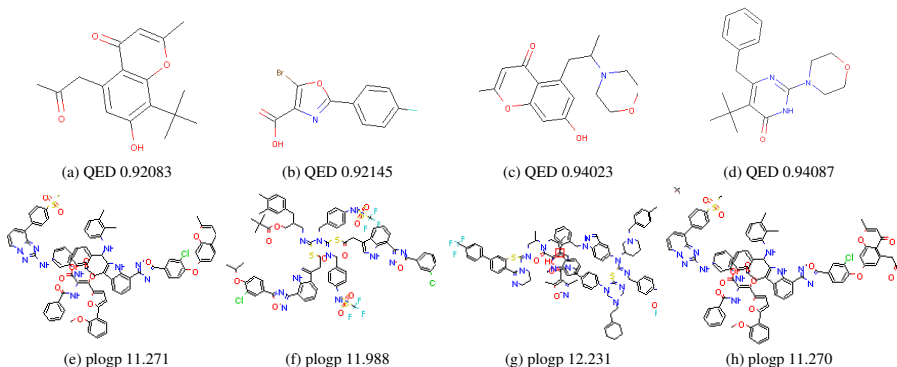


Figure 4: A random sample of optimal molecules and values found by ChemBO. In the top row, we show those with the highest QED scores, and in the bottom row we show it for Pen-logP.

ChEMBL	ZINC250k	Avg SA score	Min SA over path
2.733 ± 0.650	3.096 ± 0.771	3.772 ± 1.455	2.498 ± 0.442

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: During the execution of ChemBO, 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 [4, 6, 16, 17]. 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. [16] is trained with supervision on a random subset of 5K molecules from the ZINC dataset [54], and hence uses at least 5K evaluations. VAE in [4] 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 [6] and [17] 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.

	ORGAN [16]	JT-VAE [4]	GCPN [6]	MolDQN [17]	ChemBO
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.

Virtual screening baseline: There are two possible ways to translate the virtual screening experiment into a computational simulation. In the first version, we assume that a fixed number of compounds is available to an experimenter (same as the starting pool), and we can either synthesize from them, or use them directly for screening. This baseline is already accounted for in the results above, since we spend the initial BO budget on evaluating 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 keep compare

virtual screening and ChemBO/rand for the same number evaluations. Now we start with the 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 that he/she has; in theory, this could lead to a larger optimum since you have access to molecules you do not have in our set up, i.e. a large search space. The results obtained by simulating such experiment are shown in Figure 4: even using more samples, these values are worse than the numbers in Table 1.

QED	penalized logp
0.922 ± 0.0128	5.34 ± 0.973

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

4 Conclusion

In real world use cases for computational and statistical methods for molecular optimization, an algorithm recommends a molecule, which is synthesized, tested on, and the results submitted to the algorithm. These results are 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. In addition, on two benchmark objectives, we are able to get better or competitive scores than existing work using significantly less evaluations of the objective. While our approach is invariably 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 synthesis steps from what we already have – it might be that synthesis is the bottleneck, not testing the compound. This paradigm also brings up some new interesting questions for Bayesian optimization. Finally, it would be interesting to extend and test our framework to biologics and other molecular optimization problem in drug discovery and materials science.

Acknowledgments

We would like to thank Christopher R. Collins for reviewing the initial draft of this manuscript.

References

- [1] Daniel C. Elton, Zoïs Boukouvalas, Mark D. Fuge, and Peter W. Chung. Deep learning for molecular generation and optimization - a review of the state of the art, 2019.
- [2] 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.
- [3] 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.
- [4] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation, 2018.
- [5] 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.
- [6] Jiaxuan You, Bowen Liu, Rex Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation, 2018.
- [7] 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.
- [8] 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.
- [9] Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design, 2017.
- [10] Liva Ralaivola, Sanjay J Swamidass, Hiroto Saigo, and Pierre Baldi. Graph kernels for chemical informatics. *Neural networks*, 18(8):1093–1110, 2005.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models, 2017.
- [17] 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.
- [18] Wengong Jin, Kevin Yang, Regina Barzilay, and Tommi Jaakkola. Learning multimodal graph-to-graph translation for molecular optimization, 2018.
- [19] 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.
- [20] 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.
- [21] Matt J. Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder, 2017.
- [22] 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.
- [23] 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.
- [24] Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Costas Bekas, and Alpha A Lee. Molecular transformer for chemical reaction prediction and uncertainty estimation. *arXiv preprint arXiv:1811.02633*, 2018.

- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] Wengong Jin, Connor W. Coley, Regina Barzilay, and Tommi Jaakkola. Predicting organic reaction outcomes with weisfeiler-lehman network, 2017.
- [30] 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, 2018.
- [31] Tao Lei, Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Deriving neural architectures from sequence and graph kernels, 2017.
- [32] John Bradshaw, Brooks Paige, Matt J Kusner, Marwin HS Segler, and José Miguel Hernández-Lobato. A model to search for synthesizable molecules. *arXiv preprint arXiv:1906.05221*, 2019.
- [33] Walter D Wallis, Peter Shoubridge, M Kraetz, and D Ray. Graph distances using graph union. *Pattern Recognition Letters*, 22(6-7):701–704, 2001.
- [34] Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML*, volume 2, pages 315–322, 2002.
- [35] Dougal J Sutherland. *Scalable, Active and Flexible Learning on Distributions*. PhD thesis, PhD thesis, Carnegie Mellon University Pittsburgh, PA, 2015.
- [36] 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. *arXiv preprint arXiv:1803.06236*, 2018.
- [37] Wen Torng and Russ B Altman. Graph convolutional neural networks for predicting drug-target interactions. *bioRxiv*, page 473074, 2018.
- [38] 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.
- [39] Hong Yang Sun. *Learning over molecules: Representations and kernels*. PhD thesis, 2014.
- [40] Kirthivasan 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.
- [41] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. Adaptive computation and machine learning series. University Press Group Limited, 2006.
- [42] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems*, 2012.
- [43] Kirthivasan 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, 2019.
- [44] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 1998.
- [45] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 2003.
- [46] Niranjana Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *International Conference on Machine Learning*, 2010.
- [47] Chao Qin, Diego Klabjan, and Daniel Russo. Improving the Expected Improvement Algorithm. In *Advances in Neural Information Processing Systems*, 2017.
- [48] W. R. Thompson. On the Likelihood that one Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, 1933.
- [49] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive Entropy Search for Efficient Global Optimization of Black-box Functions. In *Neural Information Processing Systems*, 2014.
- [50] Zi Wang and Stefanie Jegelka. Max-value Entropy Search for Efficient Bayesian Optimization. In *International Conference on Machine Learning-Volume 70*, 2017.
- [51] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

- [52] 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.
- [53] Greg Landrum et al. Rdkit: Open-source cheminformatics, 2006.
- [54] 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.
- [55] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Appendix

A Some Additional Details on the Dissimilarity Measure

A.1 Solving (1)

In this section, we describe how the linear program for computing the dissimilarity measure (1), can be solved using an optimal transport (OT) program [51]. This reformulation is similar to that of Kandasamy et al. [40], who use OT to describe a distance between neural network architectures.

Say we are 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 $j \in M_2$. We now define a sequence of variables which form the parameters of our OT program. First, let $w_m(M_i) = \sum_{a \in A_i} w_a(a)$ is the total weight of a molecule $M_i = (A_i, B_i)$ for $i = 1, 2$. Denote $y_1 = [\{w_a(a)\}_{a \in A_1}, w_m(M_2)] \in \mathbb{R}^{n_1+1}$ and $y_2 = [\{w_a(a)\}_{a \in A_2}, w_m(M_1)] \in \mathbb{R}^{n_2+1}$. Next, let $C = C_{\text{at}} + C_{\text{st}} \in \mathbb{R}^{n_1 \times n_2}$ and $C' = [C \mathbf{1}_{n_1}; \mathbf{1}_{n_2}^\top 0] \in \mathbb{R}^{(n_1+1) \times (n_2+1)}$; i.e. C' has $C_{\text{at}} + C_{\text{st}}$ in its first $n_1 \times n_2$ block, representing the atom type and bond type penalties in (1), while the 1's in the last row and column capture the non-matching penalty. We finally let $U' \in \mathbb{R}^{(n_1+1) \times (n_2+1)}$ be our optimization variable where the first $n_1 \times n_2$ block will correspond to the optimization variable U in the original program. It is easy to see that (1) is equal to the following linear program, which is an optimal transport program.

$$\begin{aligned} & \text{minimise} \quad \langle U', C' \rangle \\ & \text{subject to} \quad U' \mathbf{1}_{n_2+1} = y_1, \quad U'^\top \mathbf{1}_{n_1+1} = y_2. \end{aligned}$$

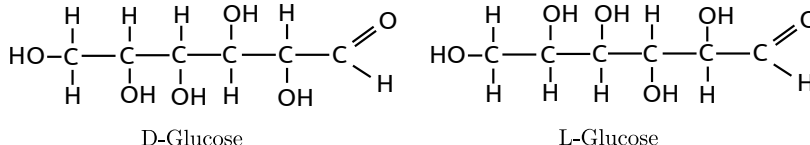
We refer the reader to Theorem 2 in Kandasamy et al. [40], who formally prove this result in a similar setting.

A.2 T-SNE visualizations for the OT distance

We perform another experiment to verify the validity of the proposed optimal transport dissimilarity measure. We use the four different base combinations of settings for the OT distance to compute distances between 200 randomly sampled molecules, and use these distances to compute 2-dimensional t-SNE embeddings [55]. These embeddings aim to preserve distances, so that visual closeness translates into OT-distance closeness. We also color the points by values of QED (drug-likeness) and synthetic accessibility scores. The results are shown in Figure 11. We see that despite the fact that the chemical space has complicated dependencies between molecule structure and properties, dependencies in the induced embedding space are relatively continuous. We can also observe clusters of molecules with similar values. In Figure 12, we compare the planar embeddings produced by other possible distances: ℓ_2 distance between pairs of fingerprints and inverted Tanimoto similarity measure between molecules (referred to as fingerprint kernel in the main part of the paper), one may say OT-dist looks slightly better (e.g. low versus high values are more separated in the plots).

A.3 Some Known Limitations

Stereoisomers: Since our dissimilarity measure is based on the graph representation, it will not be able to distinguish between stereoisomers, i.e. molecules which have the same formula and bonded atoms, but different 3D orientation. For example, pictured below are D-Glucose and L-Glucose. Since, they have the same graph representation, our dissimilarity measure will be 0 between both molecules. However, they have different 3D structures (being mirror images of each other), which can give rise to different physical properties. For instance, D-Glucose can be digested by the human body while L-Glucose cannot.



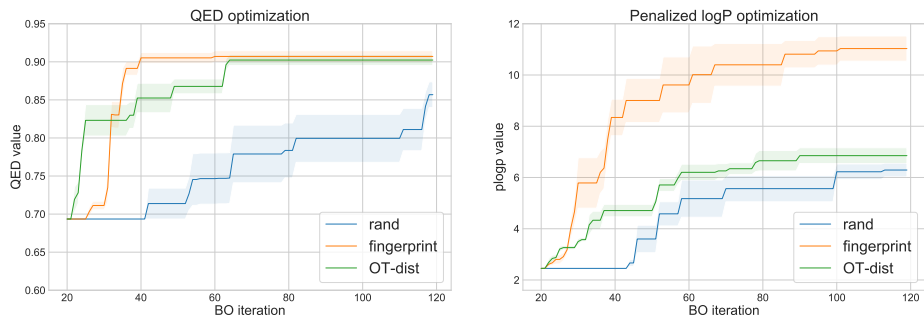


Figure 5: Results comparing the three methods described in the beginning of Section 3. We plot the number of iterations (after initialization) against the highest found QED (left) and Pen-LogP (right) values by each method. Higher is better in both cases. All curves were produced by averaging over 5 independent runs. The shaded regions indicate one standard error.

It is worth noting that many graph convolution based approaches for modeling molecules face this challenge. One way to circumvent this issue is to combine our kernel with other features which account for 3D structure in a sum or product kernel.

B Additional Experimental Results

Experiments with low starting value

To verify that ChemBO successfully optimizes the objective regardless of the quality of initial pool, we conduct an experiment on pools of 20 molecules randomly selected from subset of ChEMBL dataset that has value of the objective function capped by 0.7 for QED and 3 for penalized LogP function (approximately 60% percentiles in ChEMBL). The results below show that ChemBO performs well in such cases, too, and does so better than baseline with the same regularities as before (the fingerprint kernel performs worse than ot-dist kernel on QED and better on penalized LogP task).

Synthesis Paths

We visualize the synthesis paths for some of the optimal molecules in Figures 7-10. The boxed molecules are from the initial pool of 20 reagents. In this figure, when arrows from two or more parent molecules point to a child molecule, it means that the child molecule was obtained by reacting the parent molecules.

It is worth mentioning some caveats here. First, we see a few cases of complex molecules being combined to produce a simpler molecule – the most striking example being the one in Figure 8 where two complex molecules are combined to produce Methane (CH_4)⁵. It is more likely that simpler molecules will be available as reagents in a realistic setting. This is an artefact of our initial pool, and we believe that such cases can be avoided by carefully selecting an initial pool. Second, note that in all synthesis paths shown, there are molecules with large rings. Large rings are not necessarily stable, and hence such molecules are hard to synthesize. We believe this could be due to Rexgen, and, as mentioned in the main text, when such synthesis predictors become more accurate and reliable, so will the efficacy of our proposed framework.

The red boxes in the molecules are because RDkit’s 2D layout algorithm overlays two atoms – which is likely to happen with large molecules.

Some statistics on the ChEMBL Dataset: In Figure 6, we plot the distribution of QED and Pen-logP on the ChEMBL dataset. These values help us understand the success of optimization procedures relative to the average over the dataset from which the starting pool was drawn: the histograms show that the optimized values lie in the highest percentiles of the original dataset.

⁵In reality, Methane was probably just meant to be a by-product of a reaction meant to produce some other molecule.

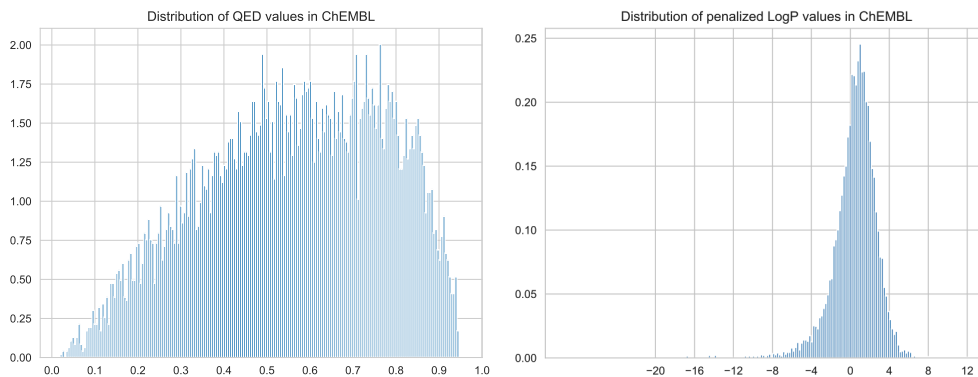


Figure 6: ChEMBL dataset statistics: normalized histograms of QED score and penalized logP score.

Some Implementation Details: For the BO methods, we fit GP hyperparameters by maximizing the marginal likelihood. As the acquisition, we adopt the ensemble method described in [43] using the EI, UCB, and TTEI acquisitions instead of sticking to a single acquisition. To optimize the acquisition, we ran the explorer for 20 iterations on BO iteration, but added the new molecules to our initial pool \mathcal{S} for the next iteration, so that we can search across a large pool during the entire optimization routine.

Figure 7: Synthesis path for molecule with penalized logP 11.988. The boxed molecules are from the initial pool of 20 reagents.

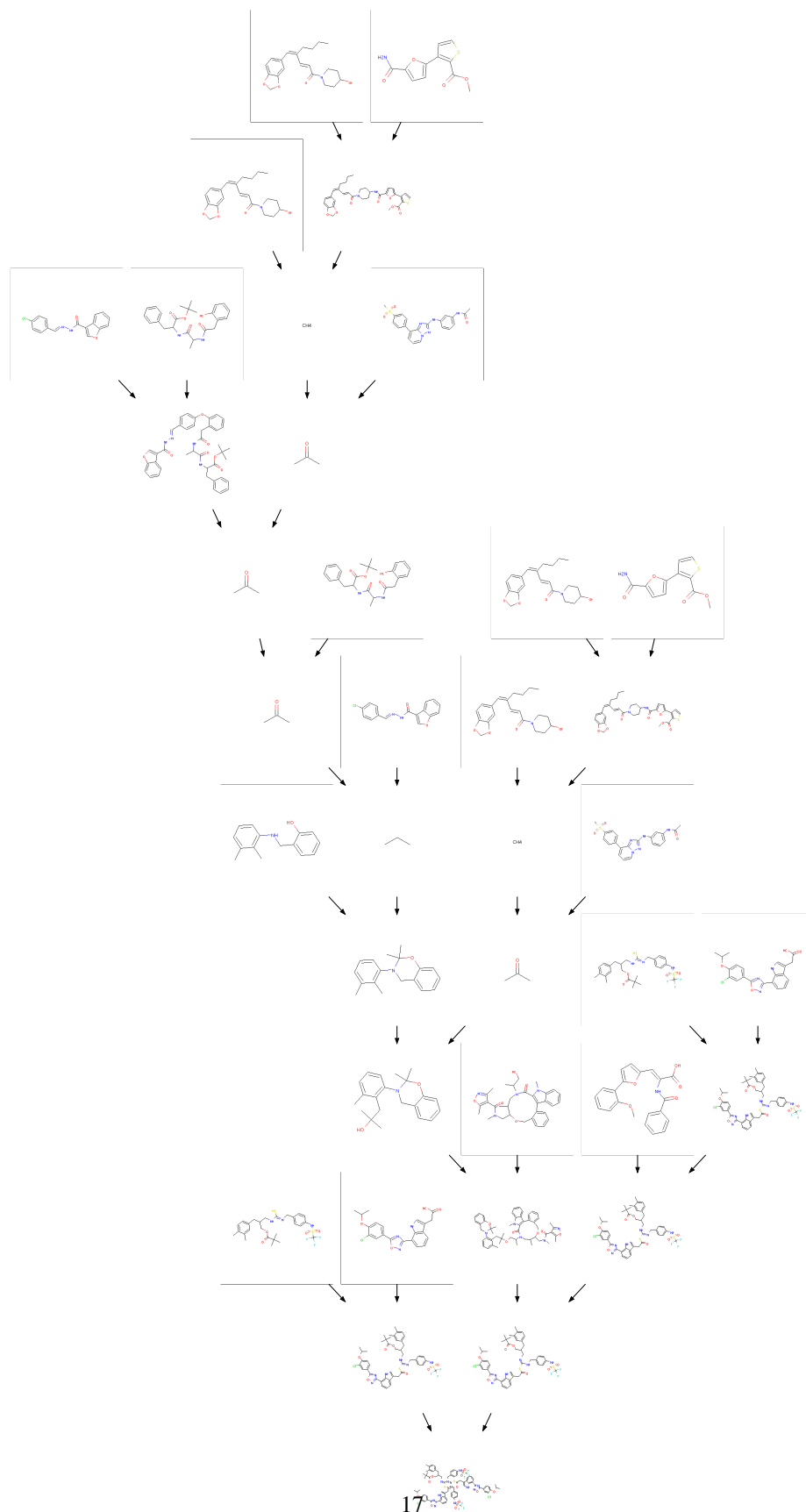


Figure 8: Synthesis path for molecule with QED 0.92. The boxed molecules are from the initial pool of 20 reagents.

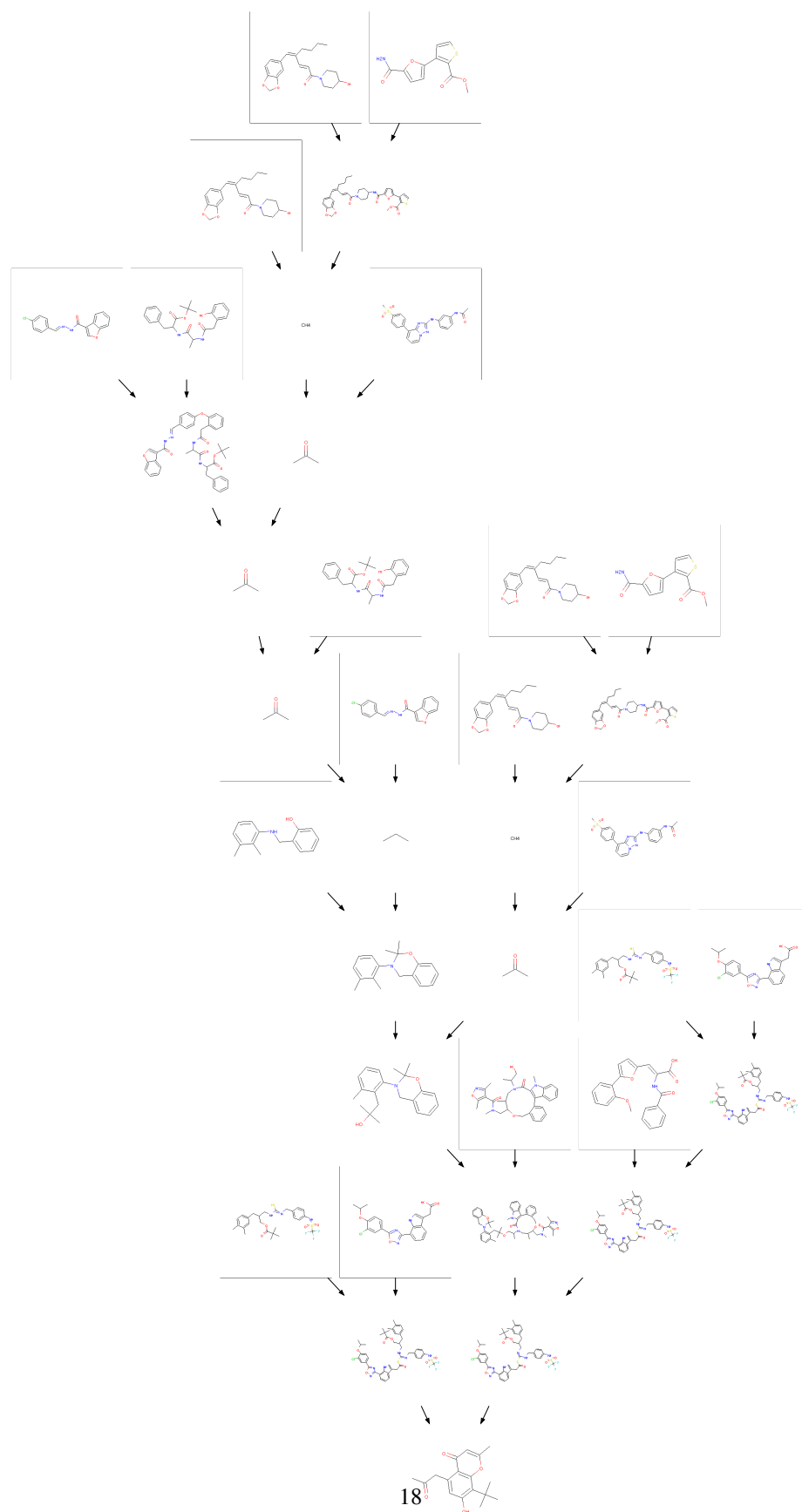


Figure 9: Synthesis path for molecule with penalized logP 8.306. The boxed molecules are from the initial pool of 20 reagents.

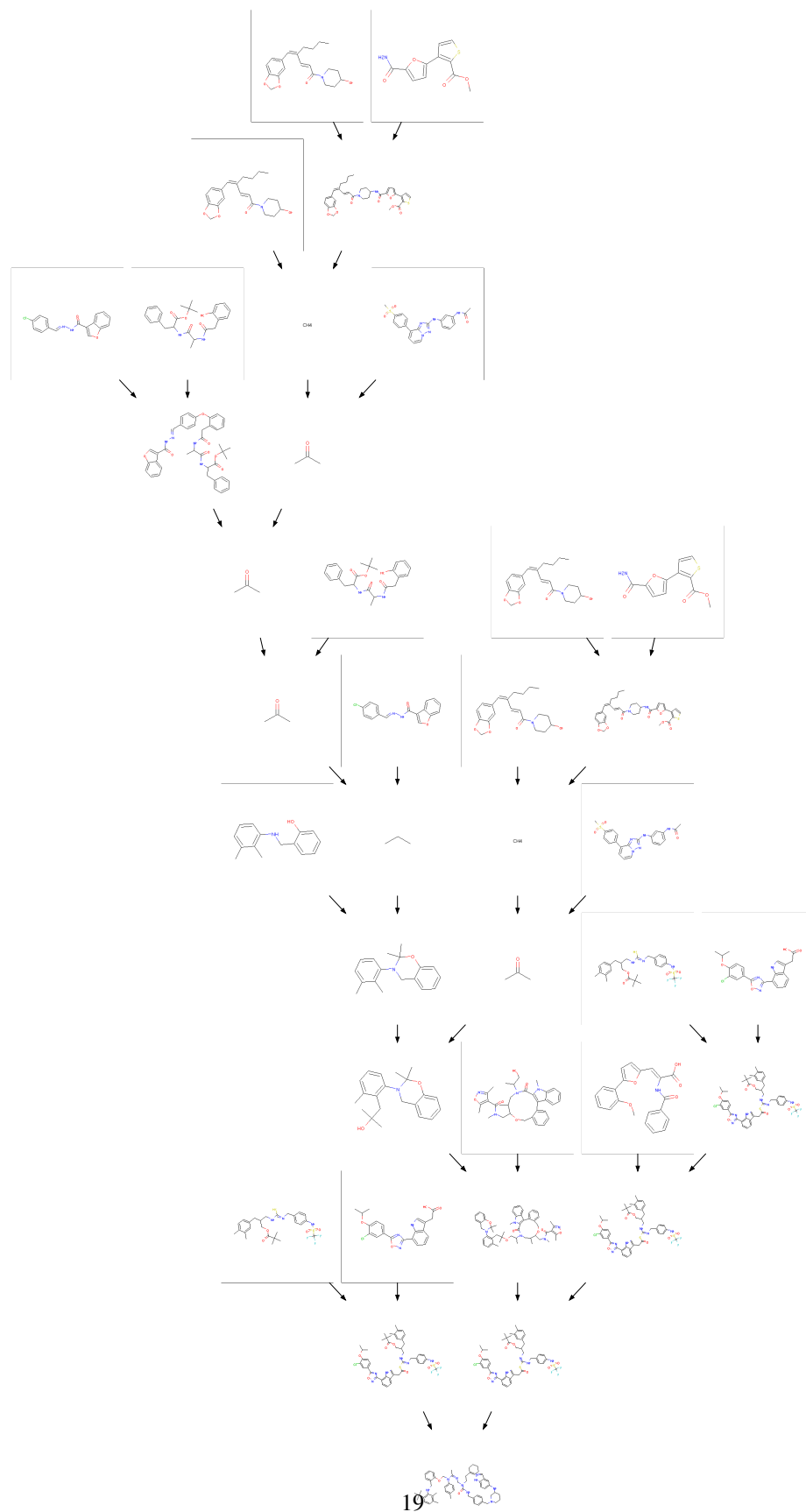
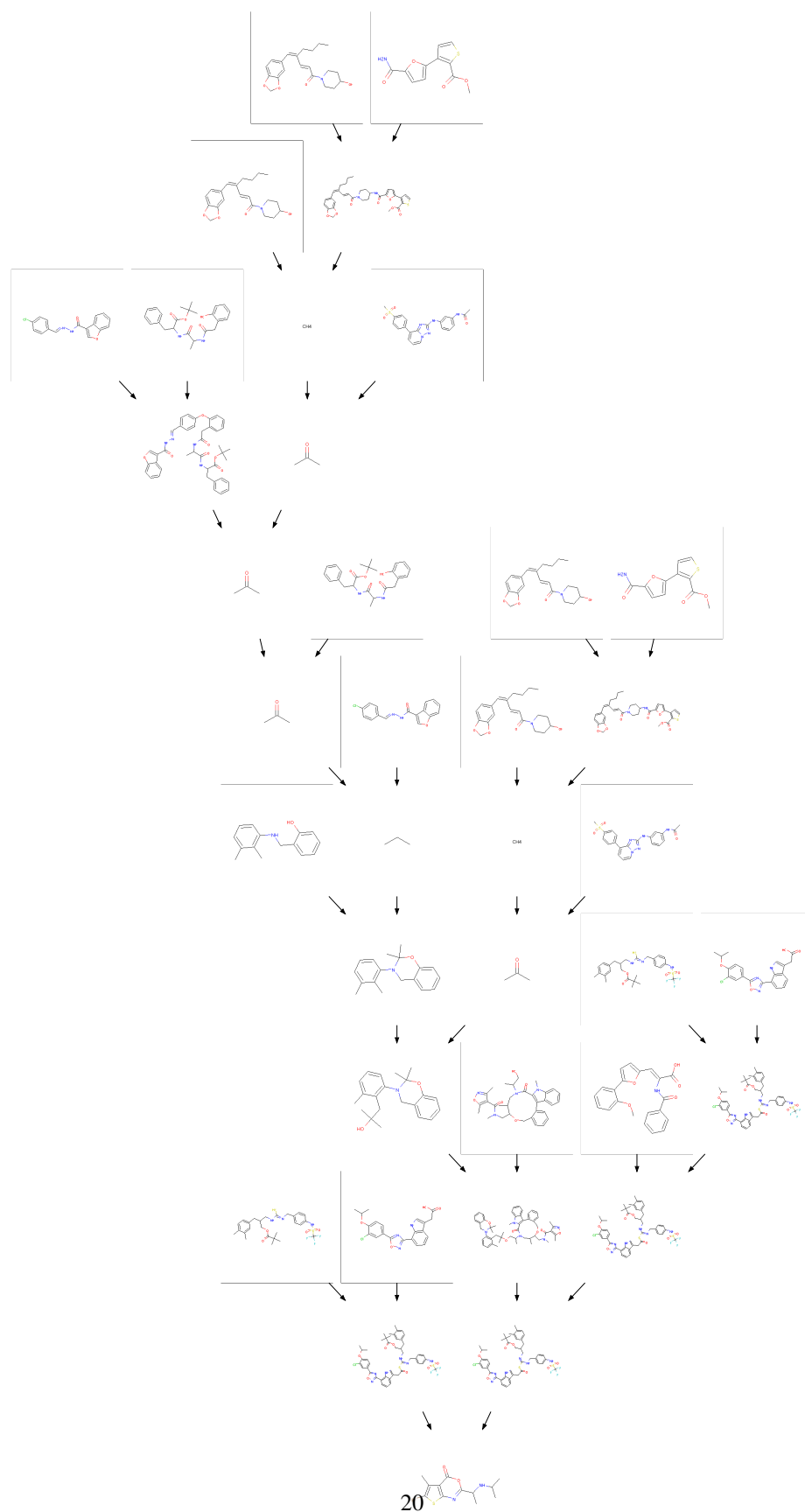


Figure 10: Synthesis path for molecule with QED 0.93. The boxed molecules are from the initial pool of 20 reagents.



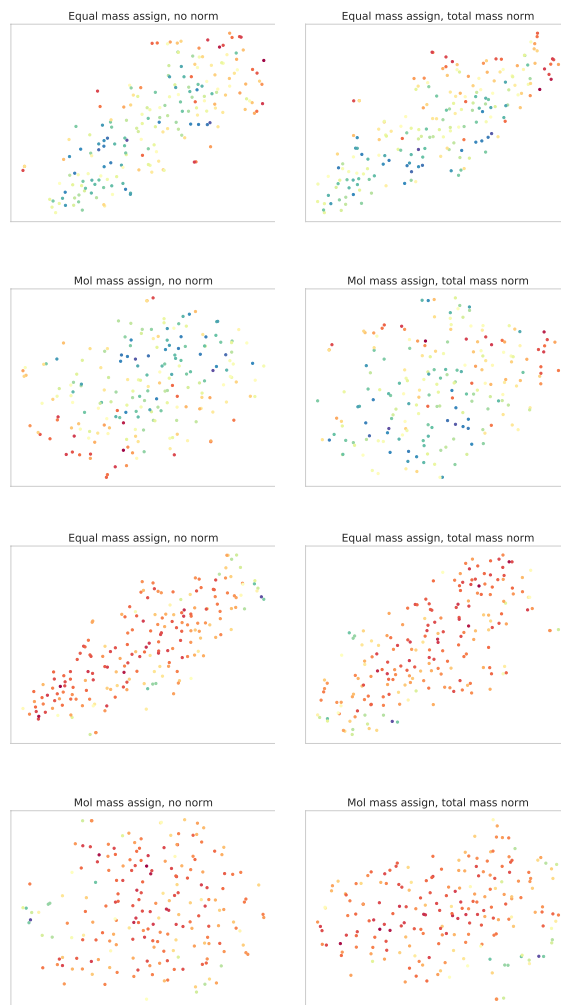


Figure 11: t-SNE visualization of OT distance $ot\text{-}dist$ for different parameter configurations, first four color-coded by QED value, last four by SA score.

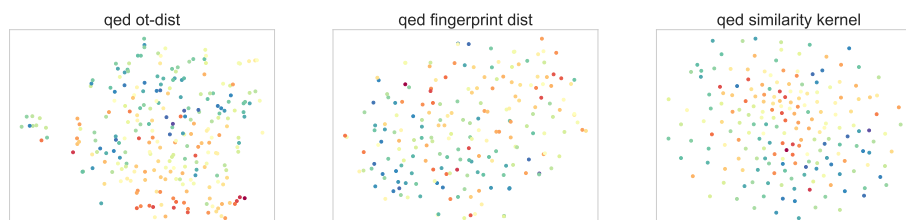


Figure 12: Comparison of t-SNE embeddings produced based on three molecular distances: $ot\text{-}dist$, ℓ_2 distance between fingerprint vectors, and inverted similarity kernel between fingerprints.