REINFORCEMENT LEARNING USING A MOLECULAR FRAGMENT BASED APPROACH FOR REACTION DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep learning methods have recently been applied to both predictive and generative tasks in the molecular space. While molecular generation and prediction of an associated property are now reasonably common, studies on reaction outcome due to the generated molecules remain less explored. Chemical reactions present a complex scenario as they involve multiple molecules and breaking/forming of bonds. In reaction discovery, one aims to maximise yield and/or selectivity, which depends on a multitude of factors, including partner reactants and reaction conditions. We propose a multi-pronged approach that combines policy gradient reinforcement learning with a recurrent neural network-based deep generative model to identify prospective new reactants, whose yield/selectivity is estimated by a pre-trained regressor. Using SMILES (simplified molecular-input line-entry system) as the raw representation, our approach involves attaching a user-defined core fragment to the generated molecules for reaction-specific learning. On three distinct reaction types (alcohol deoxyflourination, imine-thiol coupling, asymmetric hydrogenation of imines or alkenes), we obtain notable improvements in yield and/or enantioselectivity. The generated molecules are diverse, while remaining synthetically accessible.

1 INTRODUCTION

Discovering new reactions is central to the progress in chemistry and other related disciplines, including in the synthesis of drug molecules and materials (Blakemore et al., 2018; Campos et al., 2019). Newer reactions are expected to offer improved efficiencies besides paving way to novel target compounds (Kanda et al., 2020). Despite the availability of a repertoire of known reactions in the toolkit, realising complex target molecules such as anti-infective (McCauley et al., 2010) or anti-cancer drugs (Nicolaou et al., 1994; Hu et al., 2021), often presents a formidable task. Common strategies in developing new reactions rely heavily on intuition, initial planning, and an accompanying series of empirically driven trial and error attempts to identify potential substrates/reactants/optimal reaction conditions etc. While going through these steps, known as reaction optimization and expansion of substrate scope, a wealth of scattered data is generated. Access to such reaction data can make data-driven reaction discovery a feasible endeavour (Wu et al., 2018; Friederich et al., 2020). The development of novel machine learning (ML) models, built on such sparse datasets, can bring about a paradigm shift toward sustainable practices in reaction discovery (Jorner et al., 2021).

In a conventional ML approach in a regression setting, the model learns within the boundaries of the output values as seen in the training data. Extrapolative tasks using a trained model are more challenging as they entail predictions outside the envelope of the training data (Hatakeyama-Sato and Oyaizu, 2021). In a typical reaction development scenario, one would strive to maximise the yield and/or selectivity while maintaining reasonable experimental cost/hazard/time etc (Kutchukian et al., 2016; Gallarati et al., 2021). To achieve these goals and to minimise the arduous exploration of the large design space, an inverse molecular design strategy could be considered, wherein an ML model can identify promising candidates with a desired target property such as the yield of the reaction. The design strategy should ideally generate new molecules that are likely to offer yields higher than the training set reactions.

The numerous applications of deep learning (DL) in chemical space has led to interesting modelling paradigms (LeCun et al., 2015; Walters and Barzilay, 2020). DL methods are naturally poised for generative as well as inverse design applications (Sanchez-Lengeling et al., 2017). Reinforcement learning (RL) is known to be an effective approach towards goal-directed drug design (Olivecrona et al., 2017; Popova et al., 2018). It would be of high significance to apply the inverse design concept, akin to those employed in the identification of drug-like molecules, into the realm of reaction discovery. While these methods are employed for fine-tuning a chemical/biological property of molecules, it has seldom been extended to chemical reactions.

Whereas most chemical reactions would require the reacting partners to possess one or more functional groups, it is not necessarily a hard constraint in property optimization tasks. Additionally, in a chemical reaction, the participating molecule(s) undergo bond breaking and bond forming to form the product of the reaction. The problem would demand more diligence given that chemical reactions are inherently more complex due to the concurrent or sequential participation of different molecules (e.g., substrates, catalyst, base, additive, solvents, besides several other environmental factors) to yield the final product. The extent of conversion of reactants to the desired product, expressed in terms of %yield, depends on a number of factors. The exploration of such a complex, vast, and high-dimensional reaction space makes it an inherently interesting pursuit. It assumes additional importance, when one aims to maximise the reaction yield on small data settings.

In this study, we demonstrate an end-to-end application of a reinforcement learning (RL) framework for reaction discovery. The REINFORCE (Williams, 1992) policy gradient algorithm in conjunction with a recurrent neural network (RNN) based deep generative model is used for identifying prospective new molecules. The reaction yield due to such novel molecules are then predicted using a trained transfer learning (TL) model. The key aspect here is the deployment of an RL framework to navigate the generation of new molecules towards higher yield/selectivity regimes for three diverse reaction types as shown in Figure 1 (a deoxyfluorination reaction of alcohol (Reaction_a), a chiral phosphoric acid catalysed coupling between imines and thiols (Reaction_b), and an asymmetric hydrogenation of imines or alkenes using axially chiral catalysts (Reaction_c)). We have employed a fragment-based approach, wherein a user-defined core fragment is combined with the generated strings to facilitate reaction specific learning (Figure 1). Notable improvements in the mean yield of Reaction_a (by 28%) and mean enantioselectivity of Reaction_b (by 27%) as compared to the available experimental data could be achieved. For Reaction_c, although improvement is modest, the generated molecules showed visible diversity. The proposed modular framework can help speed up the design-make-test-analyse cycle, thus addressing one of the major bottlenecks in reaction discovery.

The manuscript is organised in seven subsections. The reviewing of previous works in Section 2 is followed by formulation of the problem and its solution in Section 3. Next, we describe experimental details and our results in Sections 4 and 5 respectively. The concluding remarks are in Section 6.

2 Related work

2.1 Predictive model

Recent years witnessed increased activities towards developing ML-based predictive models suitable for chemical space. Early works on predicting reactivity primarily involved the usage of quantum mechanically derived descriptors (Ahneman et al., 2018; Zahrt et al., 2019). Such feature extraction methods being computationally expensive, structure and connectivity based featurizations were explored (Sandfort et al., 2020; Schwaller et al., 2021). Models, especially those based on SMILES, demonstrated improved performance and generalizability (Schwaller et al., 2019; Kwon et al., 2022). Most of these approaches were mainly tested on high throughput datasets (HTE) wherein every possible combination between the key reacting components were evaluated. It is important to note that in a real-world scenario encountered in reaction development, sparsely distributed smaller datasets are more likely. To predict yield/selectivity, a regression model can be built using a transfer learning (TL) protocol. In TL, models are first pre-trained on a generic database of chemical structures, then fine-tuned on the target task. The use of TL has made predictive modelling in low-data regimes more affordable (Wang et al., 2022; Kim et al., 2021; Singh and Sunoj, 2022). In this work, we have adopted a TL-based regression model and fine-tuned separately on the three different reaction datasets of interest.

2.2 GENERATIVE MODEL

Deep generative models have been used for exploring chemical space. These models make use of DNN to learn from the encoding of a collection of training set molecules. Recurrent neural networks (RNNs) (Gupta et al., 2017; Bjerrum and Threlfall, 2017; Segler et al., 2017), variational autoencoders (Blaschke et al., 2017), generative adversarial networks (Prykhodko et al., 2019; Wang et al., 2021), and adversarial autoencoders (Putin et al., 2018) are a few examples of commonly found generative approaches. It has been shown that alternative methods, such as building molecules from substructures (Jin et al., 2018) as well as learning to produce graphs (Li et al., 2018), do not significantly outperform SMILES-based RNN models (Brown et al., 2019; Polykovskiy et al., 2020). Recently, Skinnider et al. have used a chemical language model (CLM) based on RNNs that could effectively learn the sequential distribution of SMILES strings from a relatively small sized dataset (Skinnider et al., 2021). In this work, we have used this CLM for generative applications tailored to chemical space exploration. While these generative models hold promise, a direct deployment for molecular generation under specified property constraints is a non-trivial task. Interestingly, reinforcement learning (RL) has been known as an effective approach towards goal-directed drug design. In this work, we have employed a multi-pronged approach merging the CLM with RL as shown in Figure 1 (Sanchez-Lengeling et al., 2017; Neil et al., 2018; Ståhl et al., 2019)

2.3 REINFORCEMENT LEARNING

Olivecrona et al. integrated RNNs with RL to produce targeted molecules with user-defined scoring function (REINVENT) (Olivecrona et al., 2017). Recently, Popova et al. proposed an RL-guided optimization using stack-augmented gated recurrent units (GRUs) on properties like logP, quantitative estimate of drug likeliness (QED) and synthetic accessibility (SA) (Popova et al., 2018). The RL methods based on Deep Q-Networks such as MolDQN are also proven to be successful for molecule optimization (Zhou et al., 2018). These existing methods built on property optimization as a key strategy are rarely extended to chemical reactions. In view of this lacuna and the fact that reaction space optimization is an inherently challenging problem, we became interested in developing suitable models that focus on maximisation of yield/*%ee* of reactions of high current importance. In our RL framework, the model is first pre-trained on a generic dataset containing a large number of diverse molecules. Another key aspect of training involves the fine-tuning with certain applied constraints such as a policy function $J(\theta)$ set to maximise the yield. This is to ensure a guided generation towards a region of higher interest in the chemical space.

2.4 NOVELTY ELEMENTS OF OUR METHOD

Despite the several applications of ML for exploring chemical space as described above, most of them focused on property prediction/optimization tasks, not in chemical reactions (See section h in the additional information). Prediction or optimization of a molecular property (say, logP) concerns only a molecule of interest, not a concatenated set of reactants. It would also be informative to contrast the typical accuracies reported for property prediction and reaction outcome. One of the best known RMSEs for yield prediction of a catalytic reaction is 7.5 %yield (Ahneman et al., 2018) whereas for logP it is as good as 0.47 log units (Ulrich et al., 2021). Further, the known regression models for yield predictions work only within the boundaries of the yields as seen in the training set. These together suggest that extrapolative tasks for improving yield/*%ee* of reactions are yet to be tackled.

It shall also be noted that the direct adaptation of a property optimization RL protocol to reaction discovery would not be advisable. For most of the property optimization tasks, random forest (RF) regressors built using molecular fingerprints were used. However, we consider the use of RF model with caution, in that the predicted output represents a two-level average (average over all samples falling in a given leaf node of a decision tree and average over all the predicted output values over all decision trees for a given sample). This limits the upper bound to 93.7 %yield, which is less than the actual maximum (99) for Reaction_a (see Table 8 in additional information). Further, RF regressor exhibited notable overfitting for unevenly distributed output values (such as in Reaction_c). Importantly, the TL based regressor that we employed exhibits very little overfitting, making it more generalizable for unseen samples.

In addition, a molecule should possess the requisite functional group(s) for serving as a substrate in a chemical reaction. Finding molecules through model conditioning such that it ensures certain substructure or functional groups is a non-trivial task. Thus, we have incorporated reaction specific 'core' fragments in our RL optimisation loop to improve reaction optimization tasks. Repetitive molecular generation, which prevents optimum exploration, is another common issue with generative models. Our RL framework incorporates a uniqueness factor β^k in the reward to address this. All these intuition based step by step customizations of our model makes it a novel approach for reaction discovery.

3 OUR METHOD

Our approach combines an auto-regressive RNN for generation, a TL-based regressor for predicting yield/*%ee*, and a policy based REINFORCE algorithm for optimising the generation.

Herein, we used an RNN-based CLM for generation (Skinnider et al., 2021). The RNN was trained on half million molecules composed of elements $\in$ {H, C, N, O, F, P, S, Cl, Br, I} from the ChEMBL dataset (Gaulton et al., 2011). Each molecule was represented first using the corresponding canonical SMILES as generated using the RDKIT program (Landrum, 2016). SMILES represents a molecule as a string of symbols for atoms and a few special characters for opening and closing of a ring and branches. A typical SMILES notation and an overview of molecule generation is shown in Figure 2. These SMILES strings serve as the input data for the ML model. During the training of the RNN, the model learns to produce the probability distribution of the next character, given a prefix string. A fully trained generator (G) can be used to generate new molecules (see additional information).

The newly generated sample from G is sent through the trained predictor (P_R) to evaluate their yield or *%ee*, as applicable. In this case, the source task is a language model (LM) trained to predict the next character in a sequence of SMILES. One million molecules from the ChEMBL database were used in the pre-training of a general domain LM. Concatenated SMILES of the individual reaction partners (catalyst, substrates, additive/solvent) serve as the input for the target task regressor that predicts the reaction outcome. The output of P_R is employed in the RL workflow to formulate the reward function. The three independent P_R employed here offered accurate predictions of yield/*%ee* for the three reactions (see additional information).

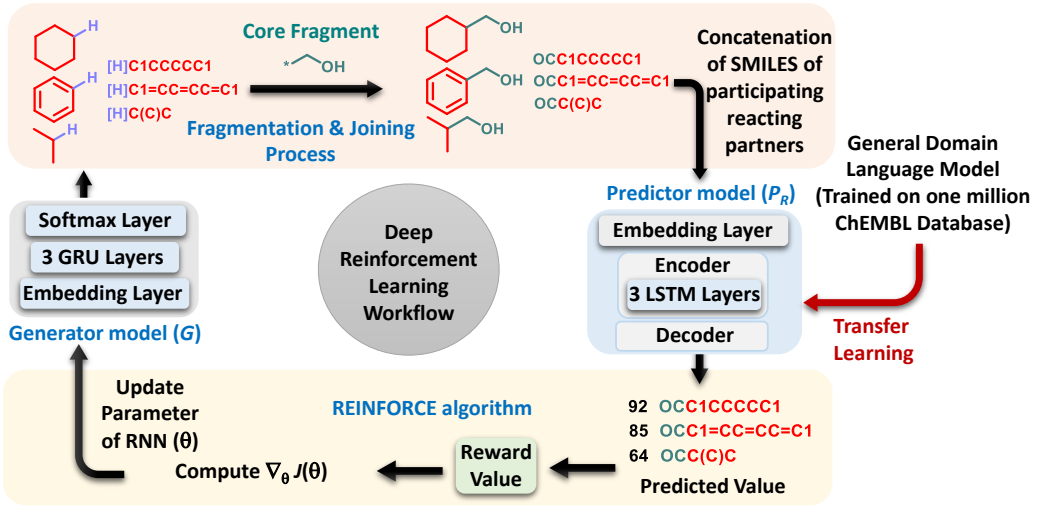


Figure 1: Overview of building blocks of our reinforcement learning approach (Rxn-B)

3.1 FRAGMENT BASED RL-ENABLED YIELD/*%ee* OPTIMIZATION

The generative and predictive models so trained are then used in the design of our RL workflow aimed at optimising the reaction outcome. The parameters of the generator (θ) are used as policy parameters and the predictor provides the reaction outcome value. The state space S is defined as the set of all terminal and non-terminal strings with length $l \in [0, T]$, for some maximum string

length T. An action a_t represents the addition of one character, given a prefix string, through the sampling process (Figure 2). Therefore, the generation of a full string s of length t is a t-action process. The goal of RL training is to identify the optimal policy, or the action with the maximum expected reward from a given state. Rewards are given only to the terminal state (s_T), implying that rewards for intermediate state are set to zero [$r(s_t) = 0$, when $t < T$] as they represent incomplete or partially generated strings.

These generated strings represent only a fragment backbone, which should then be joined to a user-specified core fragment (Figure 1). For this purpose, the generated string is "fragmented", at the first H-containing atom as located within the string, which is then joined to the core fragment to obtain the full molecule pertinent to the reaction in question. To evaluate the performance of such a generated molecule, the strings of the other reactants in the reaction are concatenated with it and the predictor provides an output value for this reaction (Figure 2). This output value is then used for the reward calculation.

Quantifying a reward merely using the predicted outcome value as obtained from the predictor (or some function of it) could be misleading. An optimal model should not only maximise the reaction outcome but also provide diverse, novel, and more importantly realistic molecules. If the reaction outcome value is set as the only objective, the model may be prone to repeated generation of a single molecule that fetches a high reward, and hence might lead to suboptimal exploration of the chemical space (See Table 3, additional information). To alleviate this, the reward value associated with the predicted yield/selectivity was multiplied by a uniqueness factor β^k such that it penalises duplication within a generated set by scaling the reward (see section (i), additional information). Here, β is a tunable parameter ($\beta \in [0, 1]$) for every k^{th} iteration when a given string occurs in the set. The reward for a generated string $r(s_T)$ can therefore be formalised as follows,

$$r(s_T) = f(P_R(s'_T)); \; where \; s'_T = F_J(s_T) \tag{1}$$

Here, the core fragment and other reacting partners are added to the string of state s_T using the F_J function to create a modified string s'_T. F_J represents the combined function of a series of operations, as shown within the dashed line in Figure 2. In eqn. (1), P_R is a predictive model that evaluates the modified state s'_T, f is the reward function, s_0 the initial state, and $s*$ the subset of terminal states [$s* = s_T \in S$]. The objective of the model is to maximise the expected return $J(\theta)$. The REINFORCE algorithm is used to estimate gradients of $\partial_\theta J(\theta)$ (Popova et al., 2018). P_θ represents the probability of sampling the terminal sequence (s_T) given policy parameter (θ).

$$J(\theta) = E(r(s_T) \mid s_0, \theta) \tag{2}$$

$$\partial_\theta J(\theta) = \sum_{s_T \in S^*} [\partial_\theta P_\theta(s_T)] r(s_T) \tag{3}$$

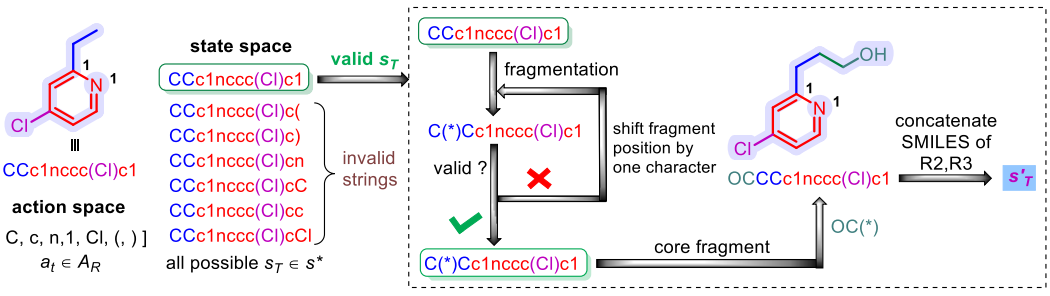


Figure 2: Details of SMILES notation for a representative molecule and an overview of molecule generation. The state space (s_T) and action space (a_t) for a set of actions A_R are shown. A valid state s_T is transformed into s'_T through the subsequent operations such as fragmentation, joining with the core fragment, and concatenation with other reaction partners (denoted as R2 and R3).

The training process consists of two phases. In the first phase, both the generator and predictor are trained separately. In the second phase, the pre-trained generative and predictive models are

used together to generate novel molecules in the targeted reaction space. The results of these two approaches, i.e., unbiased generation (UB) and biassed generation (B) are then compared. In UB, token selection for creating a new SMILES uses only the trained generator and is devoid of RL. In B, the trained generator is further optimised by the RL with the applied constraints.

4 EXPERIMENTS

4.1 DATASET DETAILS

In order to demonstrate the potential of our RL workflow, three different reactions of varying data size and distribution in their output values are considered. Given the increasing interest in fluorine containing compounds as potential drug candidates, (Purser et al., 2008; Yerien et al., 2016) we have chosen a deoxyfluorination reaction (Figure 3; Reaction_a). In this reaction, an alcohol gets converted to a fluorinated compound by the action of a sulfonyl fluoride (SF) that serves as a source of fluorine in the presence of a base (B). Deoxyfluorination of alcohols is a valuable reaction (Neil et al., 2018). Here, the data consists of 37 alcohols, 5 SFs, and 4 Bs, together making 740 reactions. The goal is to explore the alcohol, which is the key substrate undergoing the reaction. We have considered [OC(*)] as the core fragment. Other reactants, such as the SF (perfluorobutanesulfonyl fluoride) and B (phosphazene BTPP: P1-t-Bu-tris(tetramethylene)) are kept fixed (logical basis of keeping them fixed is provided in the additional information). We deployed the RL framework for targeted generation of high-yielding alcohols in the case of Reaction_a.

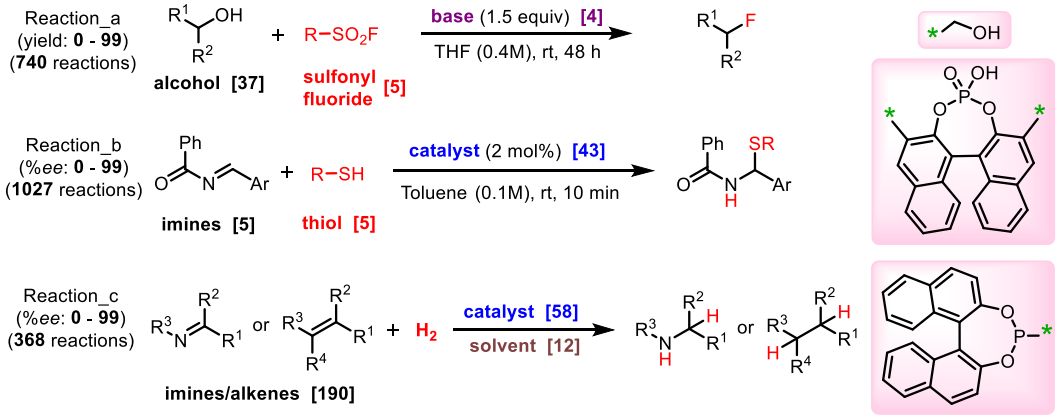


Figure 3: Important details of the three reactions considered in this study and the respective core fragments shown in the highlighted box.

The concept is extended to two catalytic reactions chosen from the domain of asymmetric reactions. In these asymmetric transformations, one of the enantiomers (stereoisomers with a non-superimposable object and mirror image relationship) is produced in excess. Two such reactions are considered; (a) a chiral phosphoric acid (CPA) catalysed coupling reaction between imines and thiols (Zahrt et al., 2019) (Figure 3; Reaction_b), and (b) an asymmetric hydrogenation reaction of imines and alkenes using axially chiral catalysts (Singh et al., 2020) (Figure 3; Reaction_c). These reactions have high practical utility, including that in industrial applications, for enantioselective synthesis of pharmaceuticals and agrochemicals. Reaction_b consists of 1027 reactions (43 CPA catalysts, 5 imines, and 5 thiols). The nature of the 3,3′ substituents on the CPA catalyst are known to impact the enantioselectivity (Rueping et al., 2011). Based on this key insight, we have considered [OP1(Oc2c(c3c4ccccc4cc(*)c3O1)c5ccccc5cc2(*))=O] as the core fragment (Figure 3). Reactants such as the imine and thiol are kept fixed during exploration of the 3,3′ substituents to be attached to the CPA core. The goal is to identify suitable 3,3′ substituents that would help maximise the formation of the desired enantiomer of the product. In other words, the substituents that provide maximum enantioselectivity are most desired. Reaction_c contains only 368 examples (190 substrates, 12 solvents, and 58 chiral catalysts), forming the smallest dataset used in this work. Further, the output values are visibility imbalanced, with fewer samples in the low enantiomeric excess

region. [P(*)1Oc2ccc3ccccc3c2c(c(O1)cc4)c5c4cccc5] is taken as the core fragment here, with a fixed set of other reactant such as the alkene (dimethylitaconate) and solvent (dichloromethane).

4.2 BENCHMARK MODELS

There have been two earlier studies that explored the chemical space by generating new substrate molecules for reaction discovery (Singh and Sunoj, 2022; Popova et al., 2018). The first one, used an RNN-based deep generative model, augmented with an NLP-based regression model to identify high yielding substrates. Since no RL was used in their study, the results serve as a suitable baseline for an unbiased model (UB). The second study employed the ReLeaSE (Reinforcement Learning for Structural Evolution) model, that contains two generative and predictive DNNs jointly trained using the RL technique, to create new focused chemical libraries. A biassed generation of new molecules towards a desired physical and/or biological property was accomplished. Instead of following a similar protocol, it was necessary for us to modify the ReLeaSE model by including (a) the fragment based approach, (b) TL-based surrogate regressor, and (c) a uniqueness factor (β^k) (see section (k), additional information). ReLeaSE-UB and ReLeaSE-B in Table 1 corresponds to these modified models.

5 RESULTS AND DISCUSSION

A comparison of performance of our Rxn-B with other models is provided in Table 1. In Reaction_a, the goal is to find new high-yielding alcohols as obtained from a given core fragment (Figure 3). Not considering the core fragment in the generation process might result in the absence of the required functional group in the generated molecules, thus rendering them unsuitable for the reaction (see Table 5, additional information). In addition to maximising the yield or enantioselectivity, it is important that the generated molecules are sufficiently diverse and remain amenable to easy synthesis. The quality of the generated molecules can be assessed by using some simple evaluation matrices such as validity (V), uniqueness (U), and novelty (N) (see succinct mathematical representation of these in the additional information). Ease of synthesis can also be evaluated by using the synthetic accessibility score (SAS). Here, V is the fraction of valid molecules among all the generated molecules, U is the number of unique molecules among the valid molecules, and N represents the percentage of unique molecules not present in the training set.

All evaluation metrices indicate better performance of our unbiased model (Rxn-UB) than the baseline ReLeaSE-UB model for Reaction_a. A notable improvement in V, U and N of the generated alcohol molecules shows that the model has effectively learned the semantics and long-term dependencies of the SMILES string. Although the *VUN* values for Rxn-UB model are lower than the TL-UB baseline model, the increase in $\bar{y}$ from 56.5 to 63.8, is an important aspect. With respect to the second baseline ReLeaSE-UB the *VUN* values obtained from Rxn-UB are better. When *VUN* and $\bar{y}$ are considered together, the Rxn-UB performs better than both ReLeaSE-UB and TL-UB. The improved exploration ability of the Rxn-UB has prompted us to perform additional experiments with biassed generation (B).

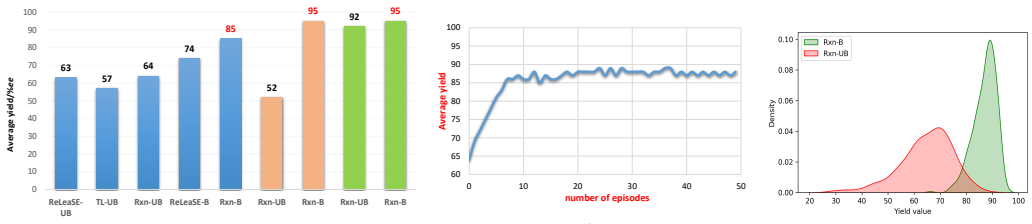
The biassed generation using the Rxn-B model has led to significantly improved performance over the baseline ReLeaSE-B model. Rxn-B appears to explore the valid reaction space of the vast high dimensional latent space, as evidenced from the high percentage of validity (94.6%). The model is also successful in addressing the issue of repetitive generation, as revealed by the uniqueness of 89.4% and is able to explore new molecules with a 89.4% novelty. A significantly higher $\bar{y}$ of %yield for Rxn-B model (85.1) over that obtained from ReLeaSE-B (74.2) is noticed. A systematic improvement in $\bar{y}$ of %yield over the RL training campaigns can be noted from Figure 4b. A notable increase in the $\bar{y}$ value for Rxn-B is also evident as compared to the TL-UB and Rxn-UB models. Higher yields could be achieved ($\bar{y}$, 85.1 versus 56.5) with a modest lowering of the *VUN* values (appox. 6 units). Rxn-B model shows significant improvement over previous baselines in generating high yielding alcohols. It is evident from Figure 4c, how the Rxn-B model is able to guide the generation of alcohols towards the higher yield regions under the applied constraint of yield maximisation. The experimentally known $\bar{y}$ of the yield across all alcohols is 57.2, whereas the biassed Rxn-B and unbiased Rxn-UB models have outperformed this with $\bar{y}$ values of 85.1 and 63.8 respectively. Furthermore, the Rxn-B algorithm is more time-efficient than ReLeaSE-B, requiring only 80 minutes to reach the maximum yield as compared to 720 minutes for the latter. Inspired

Reaction_a					
Method	V	U	N	$\bar{y}$	t
ReLeaSE-UB	60.0	60.4	60.4	63.0	*na*
TL-UB	98.0	95.6	95.6	56.5	*na*
Rxn-UB	91.8	91.8	91.4	63.8	*na*
ReLeaSE-B	90.0	80.2	64.6	74.2	720
Rxn-B	94.6	89.4	89.4	85.1	80

Method	Reaction_b					Reaction_c				
	V	U	N	$\bar{y}$	t	V	U	N	$\bar{y}$	t
ReLeaSE-UB	62.4	62.4	61.8	51.7	*na*	63.6	63.6	61.8	91.6	*na*
Rxn-UB	88.4	88.4	87.6	52.2	*na*	86.6	86.6	86	91.8	*na*
ReLeaSE-B	77.6	77.6	76.6	58.6	960	84.6	84.6	79.2	93.8	840
Rxn-B	99.6	96.0	96.0	95.2	80	96.4	93	92.4	95.1	80

Table 1: Performance comparison of our model (abbreviated as Rxn-B) with other baseline models ReLeaSE (Popova et al., 2018) and TL-UB (Singh and Sunoj, 2022). The mean value of the predicted yield/*%ee* is denoted as $\bar{y}$. The *VUN* values are in %. t denotes the time required for RL training in minutes. Since unbiased models (UB) do not involve any RL training, their run time is negligible ($<$ 10 minutes), thus denoted by na.

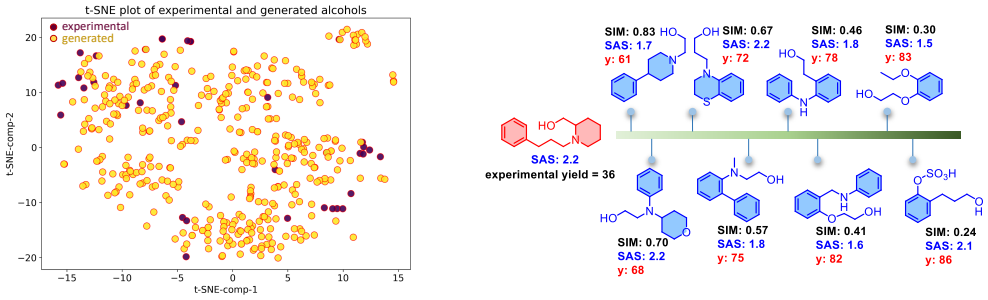
by the promising results on deoxyfluorination reaction, we have extended this concept to two more reaction datasets, aimed at higher *%ee* using our Rxn-B model. Additional experiments to check if the RL optimization is exploiting the surrogate regressor revealed that neither a model-specific nor a data-specific bias exists (section (j) additional material).



(a) (b) (c)

Figure 4: (a) Comparison of mean yield ($\bar{y}$) as obtained using different models. The actual $\bar{y}$ for the experimentally known examples are; Reaction_a = 57.2, Reaction_b = 68.4, and Reaction_c = 94.6 (calculated using the same reacting partner as used in developing the RL framework). Note that the $\bar{y}$ for all 740 fluorination reactions of alcohols is 42, but with a fixed combination of other reactants SF5 and B4 it is 57.2, (b) Plot showing the improvement in mean yield value over the RL training episodes for Reaction_a, (c) A comparison of the $\bar{y}$ values for Reaction_a as obtained from biassed (Rxn-B) and unbiased (Rxn-UB) generation.

With 99.6% validity, 96.0% uniqueness, and 96.0% novelty of the generated molecules, Rxn-B has offered a notable improvement for Reaction_b as well. The increase in the $\bar{y}$ value of *%ee* by 43 units (Rxn-B = 95.2, Rxn-UB = 52.2) indicates the effectiveness of the model in biassing the generation of the chiral phosphoric acids (CPAs) capable giving of higher *%ee*. This encouraging result in exploring CPAs requiring two arms (as opposed to just one appendage of the core fragment in Reaction_a and Reaction_c; Figure 3) again implies higher general utility of our model in reaction discovery. In the case of Reaction_c, Rxn-B is able to improve the *VUN* indices. For instance, $\bar{y}$ value of the Rxn-B model is about 4 units higher than the Rxn-UB model. Notably, the chiral ligands generated by Rxn-B are much more diverse than the previously known ligands (Figure. 12c). Thus, Rxn-B can be considered successful in exploring the region outside of the training set. These findings demonstrate that our proposed RL framework (Rxn-B) can function as a highly effective tool for exploring the reaction space in the direction of high yield/*%ee*.



(a) (b)

Figure 5: (a) The t-SNE plot showing the chemical space spanned by the generated (yellow dots) and the known (dark pink dots) alcohols for Reaction_a, (b) representative examples of the generated alcohols (blue) with a gradual increase in yield in comparison to a known alcohol shown in the left (red)

5.1 FAITHFULNESS OF RL WORKFLOW

Given these promising results, we wanted to visualise the spread of the explored chemical space viz-a-viz the training set containing 37 known alcohols. A dimensionality reduction technique, t-SNE (t-distributed Stochastic Neighbor Embedding) is used to convert the high dimensional space to a lower dimension (van der Maaten and Hinton, 2008). The t-SNE plot for Reaction_a, created using the 166-bit MACCS (Molecular Access System) keys for the generated and training set molecules, is shown in Figure 5a. The diversity of the Rxn-B generated alcohols, can be understood from the spread of the yellow colored dots. The Tanimoto similarity score (a common method to compare the similarity between molecules) (Chevillard and Kolb, 2015) between these two sets of alcohol molecules is 0.3, indicating that the generated molecules are very diverse as compared to the known alcohol molecules in the training set. These are encouraging results obtained through Rxn-B on a non-trivial extrapolative task in the molecular space focused on chemical reactions.

A few representative molecules from the generated alcohols and the corresponding predicted yields (y), synthetic accessibility (SAS), and similarity (SIM) scores are shown in Figure 5b. The SIM score for these alcohols is calculated with respect to a real alcohol with an experimental yield of 36 (shown in the left using red color). Given that the SAS score indicates the ease of synthesis in a 1 (easy) to 10 (difficult) scale, values from 1.5 to 2.2 obtained for these generated alcohols are encouraging. Similar analyses on Reaction_b and Reaction_c can be found in additional information, section (l), (m), (n).

6 CONCLUSION

Our current contribution builds on the use of RL for reaction discovery that guides toward high yield/*%ee*. The Rxn-B model contains an RNN-based deep generative model which generates novel molecules, and the predicted values obtained from a TL-based predictor is then used in the reward function. The Rxn-B showed notable improvement over the baseline models. Rxn-B exhibited stable learning across all the three reaction datasets considered in this study and have discovered higher yielding substrates or higher *%ee* chiral catalysts. Training of these models required only around an hour of compute time, much shorter than other baseline models. The proposed workflow would help in planning synthesis of important molecules by quickly identifying high-yielding substrates, thus minimising tedious empirical trial and error attempts.

Our experiments showed that the Rxn-B strategy is effective in exploring one major partner (substrate/catalyst) in a reaction where the yield/*%ee* is maximised with other components kept fixed. Although other partners may have only a lower impact on the reaction outcome, it would be interesting to develop an RL algorithm for joint optimisation of multiple components of importance to the reaction. In future, we intend to consider multi-objective tasks to facilitate a more effective exploration of the reaction space.

REFERENCES

Derek T. Ahneman, Jesús G. Estrada, Shishi Lin, Spencer D. Dreher, and Abigail G. Doyle. Predicting reaction performance in c–n cross-coupling using machine learning. *Science*, 360(6385): 186–190, 2018.

Esben Jannik Bjerrum and Richard Threlfall. Molecular generation with recurrent neural networks (rnns), 2017.

David C. Blakemore, Luis Castro, Ian Churcher, David C. Rees, Andrew W. Thomas, David M. Wilson, and Anthony Wood. Organic synthesis provides opportunities to transform drug discovery. *Nature Chemistry*, 10(4):383–394, 2018.

Thomas Blaschke, Marcus Olivecrona, Ola Engkvist, Jürgen Bajorath, and Hongming Chen. Application of generative autoencoder inide novo/imolecular design. *Molecular Informatics*, 37(1-2): 1700123, 2017.

Nathan Brown, Marco Fiscato, Marwin H.S. Segler, and Alain C. Vaucher. GuacaMol: Benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 59(3): 1096–1108, 2019.

Kevin R. Campos, Paul J. Coleman, Juan C. Alvarez, Spencer D. Dreher, Robert M. Garbaccio, Nicholas K. Terrett, Richard D. Tillyer, Matthew D. Truppo, and Emma R. Parmee. The importance of synthetic chemistry in the pharmaceutical industry. *Science*, 363(6424):eaat0805, 2019.

F. Chevillard and P. Kolb. SCUBIDOO: A large yet screenable and easily searchable database of computationally created chemical compounds optimized toward high likelihood of synthetic tractability. *Journal of Chemical Information and Modeling*, 55(9):1824–1835, 2015.

Pascal Friederich, Gabriel dos Passos Gomes, Riccardo De Bin, Alán Aspuru-Guzik, and David Balcells. Machine learning dihydrogen activation in the chemical space surrounding vaska's complex. *Chemical Science*, 11(18):4584–4601, 2020.

Simone Gallarati, Raimon Fabregat, Rubén Laplaza, Sinjini Bhattacharjee, Matthew D. Wodrich, and Clemence Corminboeuf. Reaction-based machine learning representations for predicting the enantioselectivity of organocatalysts. *Chemical Science*, 12(20):6879–6889, 2021.

A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, and J. P. Overington. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(D1):D1100–D1107, 2011.

Anvita Gupta, Alex T. Müller, Berend J. H. Huisman, Jens A. Fuchs, Petra Schneider, and Gisbert Schneider. Generative recurrent networks for ide novo/i drug design. *Molecular Informatics*, 37 (1-2):1700111, 2017.

Kan Hatakeyama-Sato and Kenichi Oyaizu. Generative models for extrapolation prediction in materials informatics. *ACS Omega*, 6(22):14566–14574, 2021.

Ya-Jian Hu, Chen-Chen Gu, Xin-Feng Wang, Long Min, and Chuang-Chuang Li. Asymmetric total synthesis of taxol. *Journal of the American Chemical Society*, 143(42):17862–17870, 2021.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2323–2332. PMLR, 2018.

Kjell Jorner, Tore Brinck, Per-Ola Norrby, and David Buttar. Machine learning meets mechanistic modelling for accurate prediction of experimental activation energies. *Chemical Science*, 12(3): 1163–1175, 2021.

Yuzuru Kanda, Hugh Nakamura, Shigenobu Umemiya, Ravi Kumar Puthukanoori, Venkata Ramana Murthy Appala, Gopi Krishna Gaddamanugu, Bheema Rao Paraselli, and Phil S. Baran. Two-phase synthesis of taxol. *Journal of the American Chemical Society*, 142(23):10526–10533, 2020.

Yongtae Kim, Youngsoo Kim, Charles Yang, Kundo Park, Grace X. Gu, and Seunghwa Ryu. Deep learning framework for material design space exploration using active transfer learning and data augmentation. *npj Computational Materials*, 7(1), 2021.

Peter S. Kutchukian, James F. Dropinski, Kevin D. Dykstra, Bing Li, Daniel A. DiRocco, Eric C. Streckfuss, Louis-Charles Campeau, Tim Cernak, Petr Vachal, Ian W. Davies, Shane W. Krska, and Spencer D. Dreher. Chemistry informer libraries: a chemoinformatics enabled approach to evaluate and advance synthetic methods. *Chem. Sci.*, 7(4):2604–2613, 2016.

Youngchun Kwon, Dongseon Lee, Youn-Suk Choi, and Seokho Kang. Uncertainty-aware prediction of chemical reaction yields with graph neural networks. *Journal of Cheminformatics*, 14(1), 2022.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

Yibo Li, Liangren Zhang, and Zhenming Liu. Multi-objective de novo drug design with conditional graph generative model. *Journal of Cheminformatics*, 10(1), 2018.

John A. McCauley, Charles J. McIntyre, Michael T. Rudd, Kevin T. Nguyen, Joseph J. Romano, John W. Butcher, Kevin F. Gilbert, Kimberly J. Bush, M. Katharine Holloway, John Swestock, Bang-Lin Wan, Steven S. Carroll, Jillian M. DiMuzio, Donald J. Graham, Steven W. Ludmerer, Shi-Shan Mao, Mark W. Stahlhut, Christine M. Fandozzi, Nicole Trainor, David B. Olsen, Joseph P. Vacca, and Nigel J. Liverton. Discovery of vaniprevir (MK-7009), a macrocyclic hepatitis c virus NS3/4a protease inhibitor. *Journal of Medicinal Chemistry*, 53(6):2443–2463, 2010.

Daniel Neil, Marwin H. S. Segler, Laura Guasch, Mohamed Ahmed, Dean Plumbley, Matthew Sellwood, and Nathan Brown. Exploring deep recurrent models with reinforcement learning for molecule design. In *ICLR*, 2018.

K. C. Nicolaou, Z. Yang, J. J. Liu, H. Ueno, P. G. Nantermet, R. K. Guy, C. F. Claiborne, J. Renaud, E. A. Couladouros, K. Paulvannan, and E. J. Sorensen. Total synthesis of taxol. *Nature*, 367 (6464):630–634, 1994.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(1), 2017.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alán Aspuru-Guzik, and Alex Zhavoronkov. Molecular sets (MOSES): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11, 2020.

Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science Advances*, 4(7), 2018.

Oleksii Prykhodko, Simon Viet Johansson, Panagiotis-Christos Kotsias, Josep Arús-Pous, Esben Jannik Bjerrum, Ola Engkvist, and Hongming Chen. A de novo molecular generation method using latent vector based generative adversarial network. *Journal of Cheminformatics*, 11(1), 2019.

Sophie Purser, Peter R. Moore, Steve Swallow, and Véronique Gouverneur. Fluorine in medicinal chemistry. *Chem. Soc. Rev.*, 37(2):320–330, 2008.

Evgeny Putin, Arip Asadulaev, Yan Ivanenkov, Vladimir Aladinskiy, Benjamin Sanchez-Lengeling, Alán Aspuru-Guzik, and Alex Zhavoronkov. Reinforced adversarial neural computer foride novo/imolecular design. *Journal of Chemical Information and Modeling*, 58(6):1194–1204, 2018.

Magnus Rueping, Boris J. Nachtsheim, Winai Ieawsuwan, and Iuliana Atodiresei. Modulating the acidity: Highly acidic brønsted acids in asymmetric catalysis. *Angewandte Chemie International Edition*, 50(30):6706–6720, 2011.

Benjamin Sanchez-Lengeling, Carlos Outeiral, Gabriel L. Guimaraes, and Alan Aspuru-Guzik. Optimizing distributions over molecular space. an objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC). August 2017.

Frederik Sandfort, Felix Strieth-Kalthoff, Marius Kühnemund, Christian Beecks, and Frank Glorius. A structure-based platform for predicting chemical reactivity. *Chem*, 6(6):1379–1390, 2020.

Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A. Hunter, Costas Bekas, and Alpha A. Lee. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, 5(9):1572–1583, 2019.

Philippe Schwaller, Alain C Vaucher, Teodoro Laino, and Jean-Louis Reymond. Prediction of chemical reaction yields using deep learning. *Machine Learning: Science and Technology*, 2 (1):015016, 2021.

Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4(1): 120–131, 2017.

Sukriti Singh and Raghavan B. Sunoj. A transfer learning approach for reaction discovery in small data situations using generative model. *iScience*, 25(7):104661, 2022.

Sukriti Singh, Monika Pareek, Avtar Changotra, Sayan Banerjee, Bangaru Bhaskararao, P. Balamurugan, and Raghavan B. Sunoj. A unified machine-learning protocol for asymmetric catalysis as a proof of concept demonstration using asymmetric hydrogenation. *Proceedings of the National Academy of Sciences*, 117(3):1339–1345, 2020.

Michael A. Skinnider, R. Greg Stacey, David S. Wishart, and Leonard J. Foster. Chemical language models enable navigation in sparsely populated chemical space. *Nature Machine Intelligence*, 3 (9):759–770, 2021.

Niclas Ståhl, Göran Falkman, Alexander Karlsson, Gunnar Mathiason, and Jonas Boström. Deep reinforcement learning for multiparameter optimization in ide novo/i drug design. *Journal of Chemical Information and Modeling*, 59(7):3166–3176, 2019.

Nadin Ulrich, Kai-Uwe Goss, and Andrea Ebert. Exploring the octanol–water partition coefficient dataset using deep learning techniques and data augmentation. *Communications Chemistry*, 4(1), 2021.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

W. Patrick Walters and Regina Barzilay. Applications of deep learning in molecule generation and molecular property prediction. *Accounts of Chemical Research*, 54(2):263–270, 2020.

Feng Wang, Xiaochen Feng, Xiao Guo, Lei Xu, Liangxu Xie, and Shan Chang. Improving de novo molecule generation by embedding LSTM and attention mechanism in CycleGAN. *Frontiers in Genetics*, 12, 2021.

Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: a benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018.

Damian E. Yerien, Sergio Bonesi, and Al Postigo. Fluorination methods in drug discovery. *Organic &amp Biomolecular Chemistry*, 14(36):8398–8427, 2016.

Andrew F. Zahrt, Jeremy J. Henle, Brennan T. Rose, Yang Wang, William T. Darrow, and Scott E. Denmark. Prediction of higher-selectivity catalysts by computer-driven workflow and machine learning. *Science*, 363(6424), 2019.

ADDITIONAL INFORMATION

(a) Experimental settings and details of model architecture

Generator: The architecture of the language model (LM) consists of three hidden layers with 512 GRU (Gated Recurrent Units) in each layer. It also contains an embedding layer of dimension 128 and no dropout layers. Models were trained with a batch size of 128 using the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ having a learning rate of 0.001. 10% of the molecules from the training set were kept aside as a validation set and used to carry out early stopping with a patience of 50,000 mini-batches.

SMILES strings are used as input for this model. The encoding of the data renders it suitable for RNN models. The initial step in data encoding is tokenization wherein each character representing an atom or a bond is transformed into a char type (token). A vocabulary of length A is determined, which is made up of all distinct tokens found in the training data along with start-of-string [SOS], end-of-string [EOS], and padding token characters [<PAD>]. In the sequence modelling situations, the training of RNN is done by maximum likelihood estimation of the next token (a_t) in the target sequence, given the tokens for the previous steps (a_{t-1}). This can be expressed in terms of a cost function $J_G(\theta)$, as shown in Eq.(4). Each token array is sent to the NN as an input at each stage of the training process, to receive the probability distribution of all possible tokens as the output. Then, the next character is sampled from this predicted probability distribution and is compared to a real token. Through this loop, the model learns to produce the probability distribution of what the next character is likely to be, with an aim to maximise the likelihood assigned to the real token:

$$J_G(\theta) = -\sum_{t=1}^{T} \log P(a_t \mid a_{t-1}, a_{t-2}, ..., a_1) \tag{4}$$

The goal of training is to minimise the cost function $J_G(\theta)$. When applied to a batch of samples, the cost function is minimised with respect to the NN parameters (θ). After a number of iterations, the model learns to assign the highest probability to the correct token while taking the rest of sequences as the input. By learning the grammar and semantics contained in the SMILES representations, the RNN will be able to produce syntactically valid molecules (Popova et al., 2018).

Once the RNN has been fully trained, it can be utilised to generate new molecules. The next token in the sequence is sampled using the probability distribution that the RNN has learned, and the sampled token is then used as the input for the next time-step. The [SOS] token serves as the first input, and at each subsequent time step, an output token (a_t) is sampled from the predicted probability distribution $P(A_t)$ over vocabulary A. The sequence generation ends once the [EOS] token is sampled. Finally, generative RNN produces complete SMILES in the form of sequential strings by predicting their constitution token by token.

Predictor: The ULMFiT (Universal Language Model Fine Tuning) is used in the domain of natural language processing (NLP), that works in a transfer learning setting, served as an unified regression model. AWD-LSTM (ASGD Weight-Dropped LSTM), a model architecture with built-in optimization and regularisation capabilities, is used. An embedding layer, an encoder consisting of 3 LSTM layers, and a decoder layer together formed the general domain LM architecture. The model has an embedding vector of length of 400 for the first LSTM layer, making the input size 400. The number of hidden units is 1152 whereas output size is 400, the same as the embedding input layer. A fully connected layer is then used to decode the output hidden state of the final LSTM layer. Finally, the probability of each token in the vocabulary is assigned using a softmax function. For the target task regressor, the LM was modified by introducing two linear blocks to the decoder unit (i.e., after the final LSTM layer of the encoder). The first layer of the decoder unit has an input size of 1200 and an output size of 50. A feature vector of size 1200 is generated by concatenating the final hidden state (400) with the maximum-pooled (400) and mean-pooled (400) representations of each hidden state in the final LSTM layer. In the final linear layer, the dimension is further reduced to 1 for the regression task.

RL framework details: In our workflow, value of the discount factor (γ) is set to 0.99, by which the rewards will be discounted within one trajectory. The uniqueness factor (β, set to 0.75) will penalise the model for repeated generation of same molecules. The batch size per episode is set to 2000. Performance comparisons between different models are made on 500 newly generated molecules in

terms of their validity, novelty, uniqueness, and mean value of the predicted output (yield/*%ee*). It is important to note that with the exception of fine-tuning the predictor, the same hyperparameter settings for the RL framework were used across all three data sets.

(b) Performances of TL-predictor

Using a pre-trained general domain LM, we have separately fine-tuned the target task regressor for three different reactions. The train/validation/test splits consists of 70, 10, 20 % of randomly distributed samples. The performances for fine tuned TL-regressor are tabulated in Table 2. This model gave excellent test performance as evident from the RMSEs (root mean square error) of 7.28 (in % yield for Reaction_a), 8.96 (in *%ee* for Reaction_b), and 7.50 (in *%ee* for Reaction_c).

Dataset	Train RMSE	Validation RMSE	Test RMSE	Test R^2
Reaction_a	6.55	7.85	7.28	0.94
Reaction_b	10.50	9.92	8.96	0.89
Reaction_c	6.55	7.52	7.50	0.75

Table 2: Performance of TL-based surrogate regressor

(c) Evaluation metrics

Validity is a basic and important benchmarking metric of the generated molecules. In most cases, the validity score is calculated by dividing the total number of valid molecules by the total number of molecules generated. Suppose, the model has sampled n molecules forming a set Y, then the validity score (Y_{valid}) can be expressed as,

$$Y = \{y_1, y_1, y_1, ..., y_n\}; \quad Y_{valid} = \frac{1}{n}\sum_{i=1}^{n} valid(y_i); \quad Y_{valid} \in [0, 1] \tag{5}$$

Where the function valid (y_i) returns 1 for a valid molecule and 0 otherwise. Uniqueness is used as a measure of model robustness. While a model can produce a high number of valid molecules, the frequency of generation of same molecules over and again may become an issue. The uniqueness score (Y_{unique}) for the l valid generated molecules can be written as,

$$Y_{unique} = \frac{1}{n} \mid \bigcup_{i=1}^{l} \{y_i\} \mid; \quad Y_{unique} \in [0, 1] \tag{6}$$

Where, U refers to the union operator applied to y_i consisting of l valid molecules. Novelty is expressed using the novelty score, which is computed by dividing the number of novel molecules by the total number of sampled molecules. Ideally, a robust model should discover new molecules that are unseen in the training. If Y is the set of m valid and unique generated molecules obtained from a training set of S molecules, the novelty score (Y_{novel}) can be defined as,

$$Y_{novel} = \frac{\mid Y \cap S \mid}{n} \quad Y_{novel} \in [0, 1] \tag{7}$$

(d) Importance of uniqueness factor (β^k)

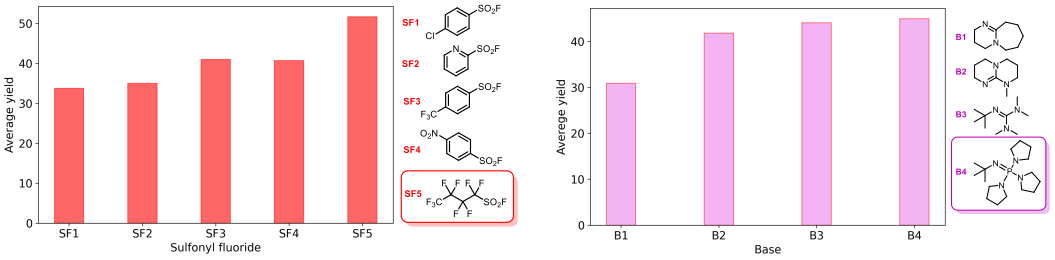
We conducted an additional experiment to assess the importance of β^k in the reward scheme for Reaction_a. The results shown in Table 3 show a drastic loss of uniqueness and novelty of the generated molecules as compared to those in the original Rxn-B (Table 1) model. This suggests that our Rxn-B model is able to generate a good number of unique molecules, by successfully preventing repetitive generation.

Dataset	Method	V	U	N	$\bar{y}$
Reaction_a	Rxn-B	99.8	7.20	7.20	88.7

Table 3: Performance of Rxn-B devoid of β^k, *VUN* values are in %.

(e) Justification for using the other reaction partners as held fixed for every generated molecules in the regression

In the case of Reaction_a, the RL model focuses on generating new alcohol molecules, which are subjected to fluorination by using a sulfonyl fluoride (SF) and a base (B) as the other reaction partners. While one can combine each generated alcohol with any of the five SFs and four Bs, for ease of comparing the yields between various alcohols it would be good to keep the combination of SF and B fixed. A careful analysis of the reported experimental data suggests that SF5 and B4 returns a higher average yield (Figures 6a and 6b). In this analysis, every reaction that uses only a given SF (say, SF1) as the fluorinating agent is chosen and the mean yield is calculated over the output values of such reactions. A similar process is followed for all other sulfonyl fluorides and their average yields are then plotted in Figure 6a. The best mean yield value is found to be for SF5. The base with the highest average yield value is similarly identified, which is found to be B4 (Figure 6b).



(a) (b)

Figure 6: Mean experimental yield for each (a) sulfonyl fluoride (SF), (b) base (B). The SF and B which are held fixed are shown inside the highlighted box.

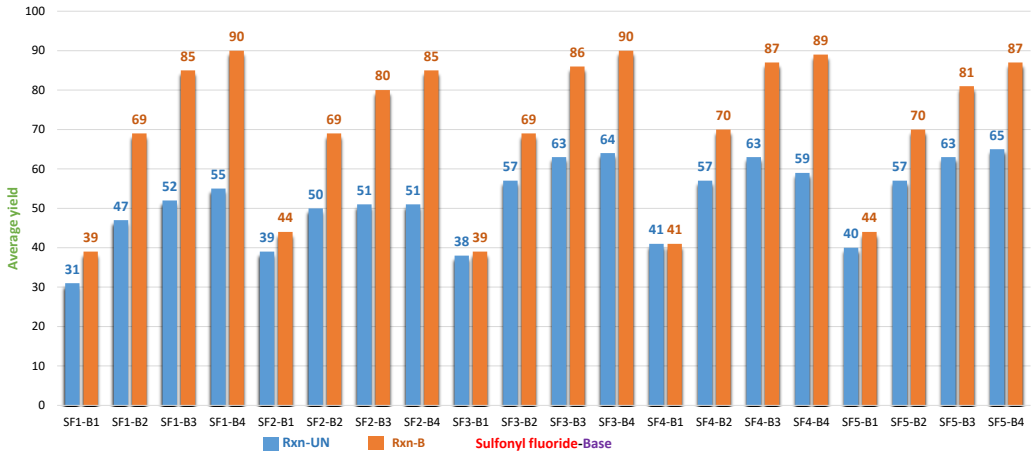


Figure 7: The mean yield values for deoxyfluorination reactions across all sulfonyl fluoride and base combinations in the case of Reaction_a.

In the case of Reaction_b, IM5 (imine) and TH2 (thiol) are kept as the fixed components. These compounds offer the highest mean *%ee* (Figure 8A & B). Figure 8C plots the mean of the predicted *%ee* for the generated ligands, with each possible combination of IM and TH. Regarding Reaction_c, making a plot is not feasible as there are a large number of imine/alkene reacting partners (190).

(f) Reproducibility of model performance

We have conducted additional experiments, maintaining the same settings of the RL model, for Reaction_a, to assure the reproducibility of results. The average value of all performances is shown in Table 4. It is likely that certain minor differences, between different runs, may become apparent in ML models that are sensitive to the initialization. In the present case, the RL generator or RL training would have similar, but non-identical initialization. It shall be noted that the sampling of

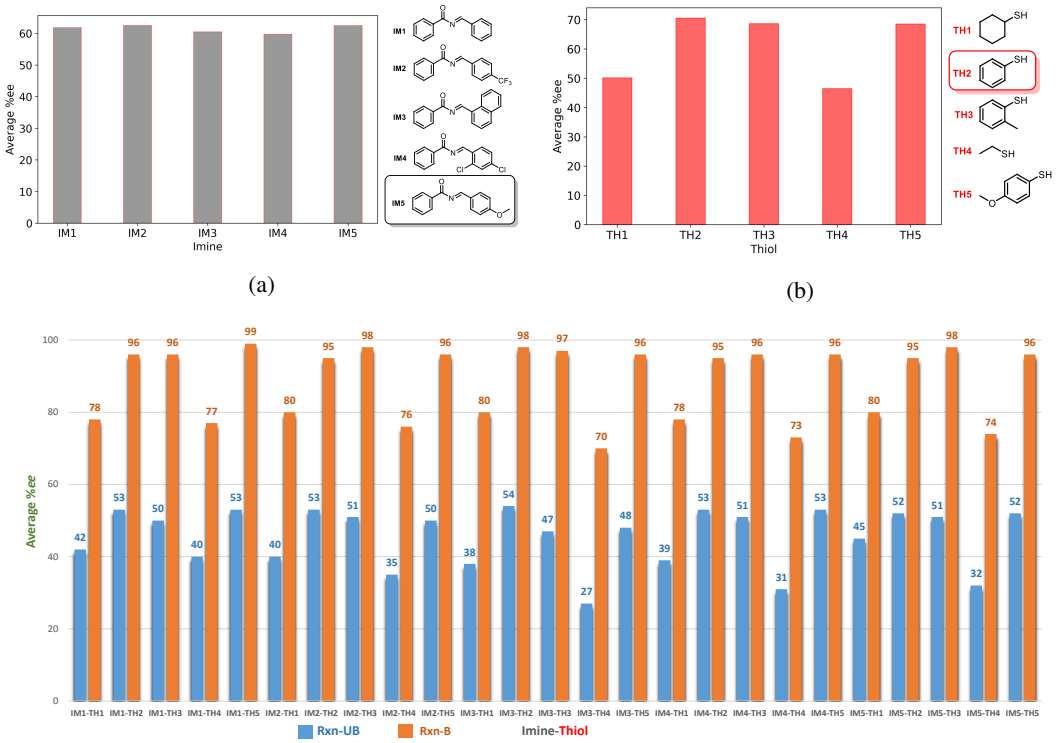


Figure 8: The mean experimental *%ee* values for each (a) imine and (b) thiol (fixed IM and TH are shown in the box); (c) mean predicted *%ee* for the generated ligands with every pair of imines and thiols in the case of Reaction_b.

characters based on probability distribution during the initialization of trained generators is random. This process will therefore not always generate the same set of samples. In the RL training, it explores different regions of the latent space in different runs. Although more experiments might lower the standard deviation in *VUN*, the most important quantity, i.e., $\bar{y}$, already has very small standard division.

We provide our code, datasets, and pre-trained models through the zip file in the Supplementary Materials.

Experiment No.	Method	V	U	N	$\bar{y}$
1	Rxn-B	94.6	89.4	89	85.1
2	Rxn-B	94.6	94	94	87.1
3	Rxn-B	98.2	90.6	91	87.1
4	Rxn-B	96.4	92.8	93	86.9
5	Rxn-B	95.6	89.2	89	86.3
avg. ± std. dev.	Rxn-B	95.9 ± 1.3	91.2 ± 1.9	91.2 ± 1.9	86.5 ± 0.8

Table 4: Performances for different runs in case of Reaction_a

(g) Importance of core fragment constraints

To assess the importance of the core fragment, additional experiments were performed. If the core fragment or substructure [OC(*) is the core fragment for Reaction_a] is not considered throughout the generation process, they may not have the required functional group, rendering them unsuitable for the reaction. Here, we modified the reward function to make it dependent on the presence of the necessary functional groups. If the substructure is present in the generated molecule, we rewarded

the generation process; if not, we penalised the model. The results are shown in Table 5, where a sharp decline of novelty and uniqueness relative to the original model performance (Table 1) strongly suggests that reaction space exploration without the core fragment is not very useful. In the case of Reactions_b and , the core fragment is the axially chiral phospohric acids, which are rarely found in the ChemBL molecules. Sampling such molecules from the pre-trained model is unlikely, thus not included in Table 5.

Dataset	Method	V	U	N	$\bar{y}$
Reaction_a	Rxn-B	100	0.20	0.20	68.7

Table 5: Performance of Rxn-B devoid of core fragment and corresponding *VUN* values in %

(h) A comparison between property prediction and reaction outcome

Consider the following two alcohols: C(CCCO)c1ccccc1, c1(CCC(C)O)ccccc1. Both of them are present in the Reaction_a dataset. These molecules are very similar in structure, one is a linear primary alcohol and the other a branched secondary alcohol. Due to this similarity, they have very similar individual properties, especially those routinely considered in literature for property optimisations. The logP values are nearly identical (2.00 and 2.0016) so are the QED (Quantitative Estimation of Druglikeness) values (0.65 and 0.69). On the other hand, the yields of fluorination reaction under identical conditions are 94% and 65% respectively. Such observations ascertain that reaction optimisation requires a different level of control on molecular structure, making it a relatively more complex problem statement.

(i) Reward function

As our target is to maximise the reaction outcome such as yield/*%ee*, a monotonic function of the predicted outcome as the reward function would be suitable. In this study, we employed a linear step function, as shown in Figure 9. The actual reward function includes a multiplication by β^k.

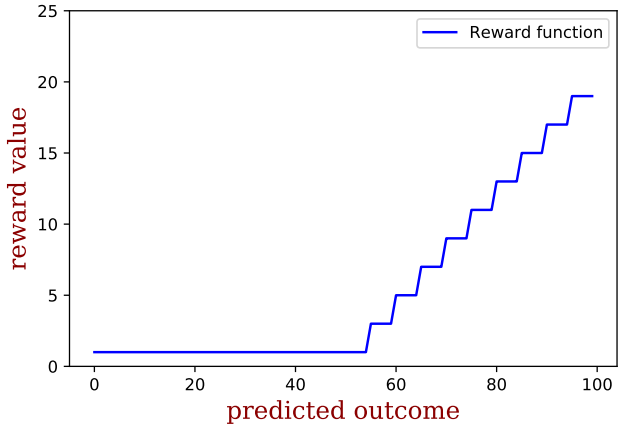


Figure 9: A plot of step reward function used for this study

(j) Tests to examine whether exploitation of surrogate regressor occurs

We randomly divided the initial dataset into two sets, split1 and split2, for Reaction_a, each containing 370 samples. Three fine tuned surrogate models P_{R1}, P_{R2}, and P_{R3} were built. P_{R1} is then trained on split1 and used as the surrogate model for RL training. P_{R2} was trained on the same split (split1), but with a different seed (to evaluate any model-specific bias was induced by the surrogate model). P_{R3} was separately trained on split2 (to examine the likelihood of data-specific bias induced by the surrogate model). The test and train performances for these models are shown in Table 6. The results of RL training as shown in Figure 10a, demonstrate that the optimization algorithm did not exploit the surrogate regressor.

Moreover, we have examined whether any model-specific bias is present (P_{R2} and $P_{R2}\prime$) in the surrogate regressor trained on the original 740 reactions for Reaction_a. These surrogate models

Dataset	Surrogate Regressor	Split Type	Train RMSE	Test RMSE	Test R^2
Reaction_a	P_{R1}	Split1	8.37	9.38	0.88
Reaction_a	P_{R2}	Split1	7.75	11.02	0.84
Reaction_a	P_{R3}	Split2	6.11	9.15	0.91

Table 6: Performances for different splits having 370 samples in case of Reaction_a

are trained with different seed values. The performances of various models are provided in Table 7. It is also evident from Figure 10b that predicted yield did not significantly differ across different model controlled surrogate regressors P_{R1}, P_{R2}, or $P_{R2}\prime$. For Reaction_b and Reaction_c, model controlled experiments showed similar trends (Figure 10), indicating that the optimization algorithm did not exploit the surrogate regressor model.

Dataset	Surrogate Regressor	Train RMSE	Test RMSE	Test R^2
Reaction_a	P_{R1}	6.55	7.29	0.94
Reaction_a	P_{R2}	5.82	7.67	0.93
Reaction_a	$P_{R2}\prime$	5.89	7.44	0.93
Reaction_b	P_{R1}	10.50	8.96	0.89
Reaction_b	P_{R2}	10.76	8.50	0.90
Reaction_b	$P_{R2}\prime$	11.20	7.70	0.92
Reaction_c	P_{R1}	6.55	7.50	0.75
Reaction_c	P_{R2}	6.56	7.51	0.75
Reaction_c	P_{R3}	6.70	7.68	0.74

Table 7: Performances of different runs for Reaction_a, Reaction_b, and Reaction_c

(k) Influence of including different functionalities step-by-step to the baseline model

The performances of unmodified ReLeaSE models (ReLeaSE-B(UM)) are shown in Table 8 in the case of Reaction_a. Note that ReLeaSE-B(UM) (i) lacks constraints on generating molecules with a user defined functional group (i.e., core fragment in our terminology), (ii) employs a random forest regressor (found to be less suitable for our task (Table 8), and (iii) found to reach a state of repetitive generation. This indicates that the ReLeaSE model finds it challenging to identify new molecule containing alcohol functional group. To make it applicable to a reaction discovery, as in the present case, the ReLeaSE model should accordingly be customised. The results of our Rxn-B method shown in Table 9 convey a gradual improvement in performance with step-by-step inclusion of each of the tailored functionalities such as (a) the fragment based approach, (b) TL-based surrogate regressor, and (c) a uniqueness factor (β^k), as compared to the unmodified ReLeaSE baseline models (Figure 11).

Dataset	Surrogate Regressor	Train RMSE	Test RMSE	Test R^2
Reaction_a	RF	7.74	11.00	0.86
Reaction_b	RF	7.80	8.63	0.90
Reaction_b	RF	9.83	12.93	0.40

Table 8: Performances for different runs in the case of Reaction_a, Reaction_b, Reaction_c with a random forest (RF) surrogate regressor

The notations (UM) and (M) respectively denote the unmodified ReLeaSE and a partially modified (inclusion of only the core fragment).

(l) Synthetic accessibility score (SAS)

SA scores of the generated molecule sets were calculated, and the corresponding mean values are 2.1, 4.9 and 3.7 for Reaction_a, Reaction_b and Reaction_c respectively (Figure 11).

(m) Diversity of generated molecules

The diversity of generated CPAs can be seen from Figures 12a and 12c respectively for Reaction_b and Reaction_c. Some representative CPAs generated by our Rxn-B model for Reaction_b and the

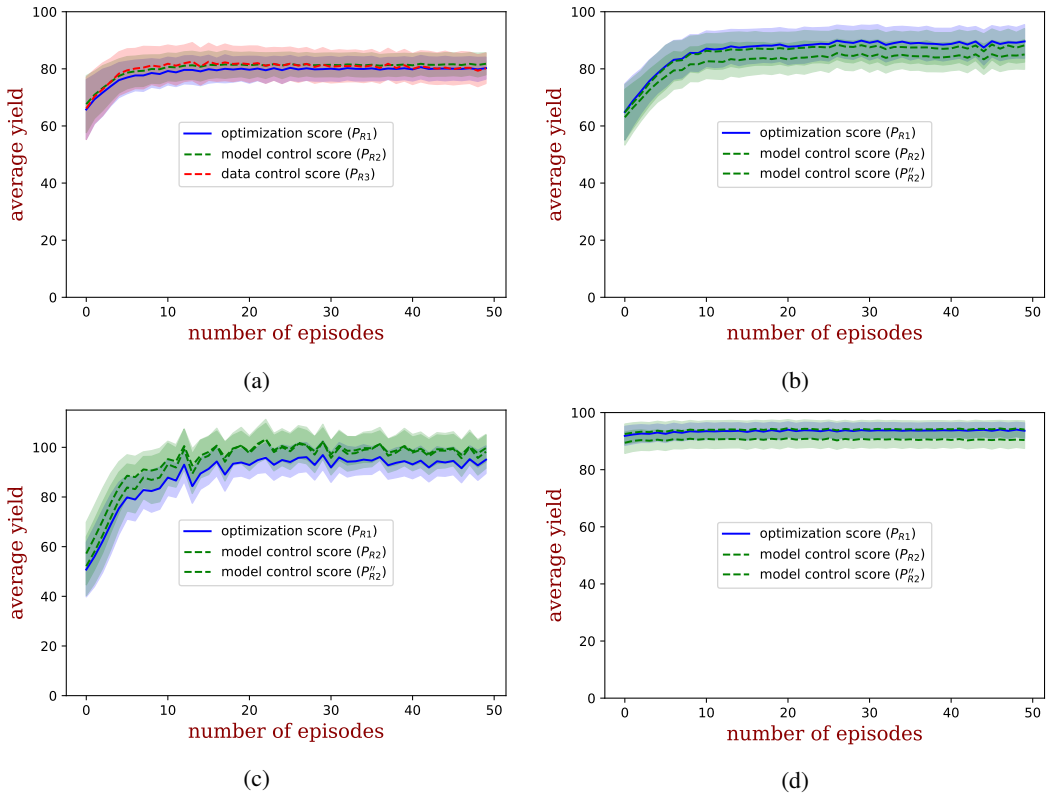


Figure 10: (a) Plot shows a comparison of average yield value obtained with different regressors trained on a split dataset for Reaction_a (370 samples). Comparison of mean yield($\bar{y}$) as obtained using different surrogate regression models trained using the respective full datasets for (b) Reaction_a, (c) Reaction_b, and (d) Reaction_c. The shaded regions represent standard deviations while the bold line represents the mean value.

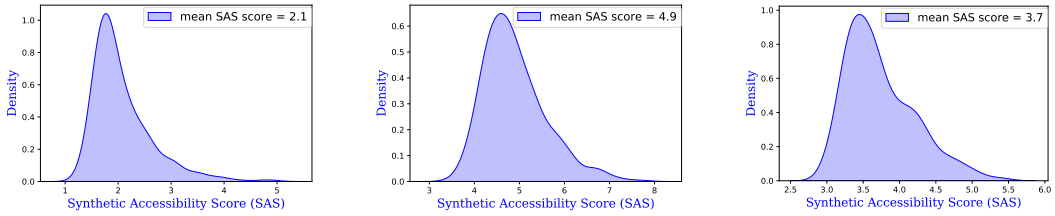
Method	Core fragment	Uniqueness Factor	Surrogate Regressor	V	U	N	$\bar{y}$
ReLeaSE-B(UM)	No	No	RF	82.0	26.0	26.0	58.9
ReLeaSE-B(M)	Yes	No	RF	87.0	66.0	42.0	67.0
ReLeaSE-B	Yes	Yes	TL	90.0	80.2	64.6	74.2
Rxn-B	Yes	Yes	TL	95.8	94.8	94.8	89.0

Table 9: Performances for different runs in case of Reaction_a, Reaction_b, Reaction_c with RF surrogate regressor

corresponding *%ee* values are shown in Figure 12b. It can be seen that Rxn-B has identified new CPAs with higher *%ee* that are structurally similar to the experimentally known reference (shown in red, Figure 12b). The model has generated a novel CPA with a predicted *%ee* of 99, much higher than the reported *%ee* of 18 for the reference CPA. The similarity score (SIM score) of 0.85 between this and the reference CPA indicates the ability of Rxn-B in efficiently exploring neighbourhoods of the known CPA in the higher *%ee* regimes. The SIM score for generated CPAs for Reaction_c is calculated with respect to a experimentally known CPAs with an experimental *%ee* of 89 (shown in the left using red color, Figure 12d).

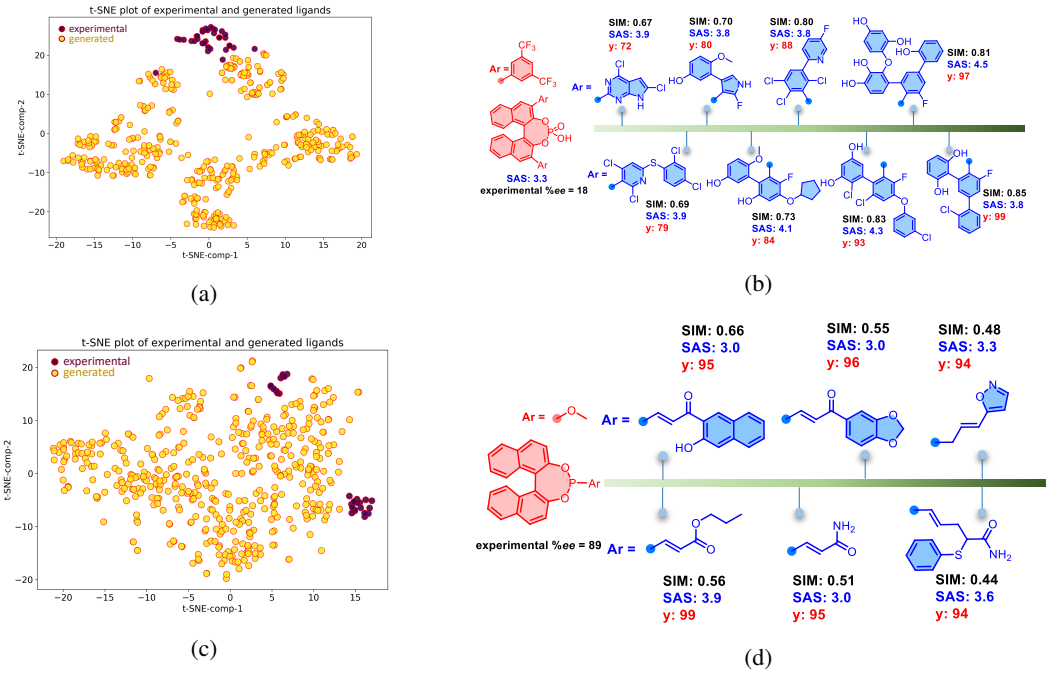
(n) Quantitative estimate of drug-likeness (QED) score

QED is a metric of drug likelihood, which ranges from 0 to 1, where compounds with higher drug-like properties are closer to 1. In the case of Reaction a, we have calculated the QED score for



(a) (b) (c)

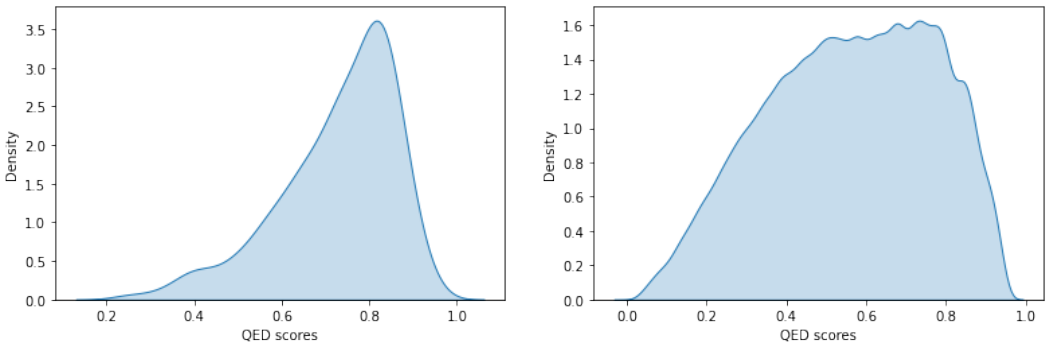
Figure 11: Plots of SA scores for the generated molecules in the case of (a) Reaction_a, (b) Reaction_b, and (c) Reaction_c.



(a) (b) (c) (d)

Figure 12: (a) The t-SNE plot showing the chemical space spanned by the generated (yellow dots) and the known (dark pink dots) alcohols for Reaction_b, (b) representative examples of the generated alcohols (blue) with a gradual increase in yield in comparison to a known alcohol shown in the left (red) for Reaction_b; (c) The t-SNE plot containing the known chiral ligands and the generated ligands for Reaction_c, (d) representative examples of the generated CPAs (blue) by the Rxn-B model for Reaction_c.

generated molecules (Figure 13a), the QED score is found to be 0.72 ± 0.13. This value is higher than the QED score obtained using the ChEMBL dataset (0.56 ± 0.20) shown in Figure 13b.



(a) (b)

Figure 13: Plot of QED score of (a) the generated molecules for Reaction_a, (b) the molecules present in ChEMBL dataset.