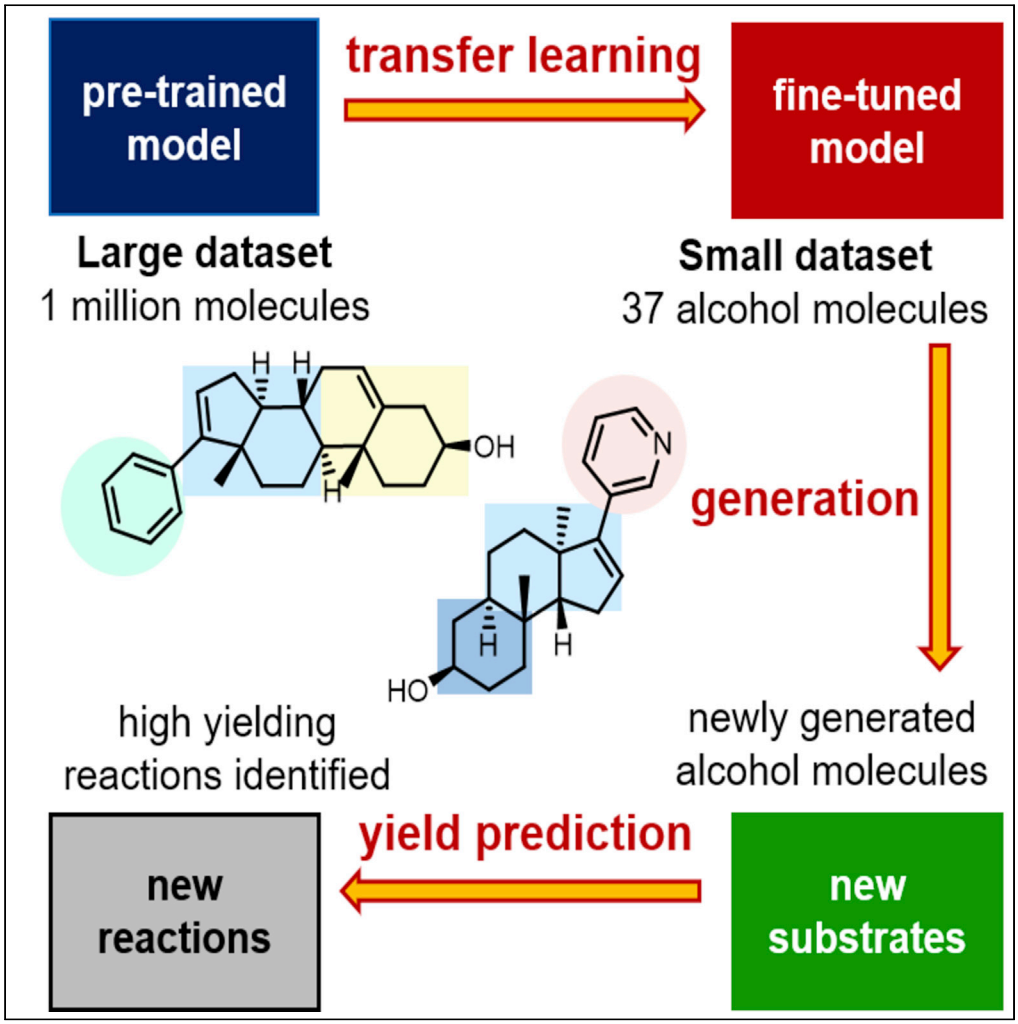
iScience

CellPress
OPEN ACCESS

Article

A transfer learning approach for reaction discovery in small data situations using generative model



Sukriti Singh, Raghavan B. Sunoj

sukriti243@gmail.com (S.S.)
sunoj@chem.iitb.ac.in (R.B.S.)

Highlights

Dual pronged transfer learning, both to generate and predict yields of new molecules

Demonstrated the utility for an important family of deoxyfluorination of alcohols

Applicable for practically more likely situations with relatively smaller data

Extendable to other reaction manifolds to facilitate expedited reaction discovery

Singh & Sunoj, iScience *25*, 104661
July 15, 2022
https://doi.org/10.1016/j.isci.2022.104661

iScience

CellPress
OPEN ACCESS

Article

A transfer learning approach for reaction discovery in small data situations using generative model

Sukriti Singh[1,*] and Raghavan B. Sunoj[1,2,3,*]

SUMMARY

Sustainable practices in chemical sciences can be better realized by adopting interdisciplinary approaches that combine the advantages of machine learning (ML) on the initially acquired small data in reaction discovery. Developing new reactions generally remains heuristic and even time and resource intensive. For instance, synthesis of fluorine-containing compounds, which constitute ~20% of the marketed drugs, relies on deoxyfluorination of abundantly available alcohols. Herein, we demonstrate the use of a recurrent neural network-based deep generative model built on a library of just 37 alcohols for effective learning and exploration of the chemical space. The proof-of-concept ML model is able to generate good quality, synthetically accessible, higher-yielding novel alcohol molecules. This protocol would have superior utility for deployment into a practical reaction discovery pipeline.

INTRODUCTION

The discovery of novel molecules using the currently available synthetic methods may demand time and resource intensive approaches. The high-dimensional nature of the chemical space consisting of the participating molecules such as the reactants as well as the reaction conditions renders such tasks nontrivial (Santanilla et al., 2015). In a typical situation encountered in reaction development, one would aim to maximize the yield and/or selectivity under affordable experimental conditions (Kutchukian et al., 2016; Schneider and Baringhaus, 2008; Singh et al., 2020; Gallarati et al., 2021). The high dimensional and vast chemical space can be explored and analyzed using machine learning (ML) methods to predict substrate specific reaction outcomes, such as the yield.

With the rapid advances in the field of deep learning (DL), there have been notable applications focused on addressing the challenges in molecular design (LeCun et al., 2015; Butler et al., 2018; Sun et al., 2019; Griffiths and Hernández-Lobato, 2020; Li et al., 2021). The use of DL for molecular discovery can broadly be categorized into reaction outcome prediction and molecule generation (Gomez-Bombarelli et al., 2018; Miljkovic et al., 2021; Walters and Barzilay, 2021). The predictive DL models, consisting of multiple hidden layers, have been used for predictions of molecular properties as well as the yield/selectivities of reactions (Schwaller et al., 2021; Senior et al., 2020; Singh and Sunoj, 2022). First, DL methods can take molecular structures to learn the data-driven feature representation with minimal feature engineering (Winter et al., 2019; Li et al., 2021; Atz et al., 2021; Mi et al., 2021; Lim and Jung, 2019). This approach can extract the underlying structural patterns and capture the relationship between the features and the output. Second, the generative models, where the model learns from the given dataset without the requirement of explicit design rules, and can produce novel molecules resembling those in the training data (Grisoni et al., 2020; Wang et al., 2021).

The application of DL in chemistry requires molecular structures to be presented in a machine-readable format. The simplified molecular input line entry system (SMILES) is one such commonly used representations for molecules. It is a text-based representation where the molecular graph is encoded compactly into a sequence of characters and thus closely resembles a natural language. This very similarity suggests that the SMILES strings could be utilized to build chemical language models. The generation of novel SMILES strings could therefore be considered as equivalent to natural language generation. A recurrent neural network (RNN) that is specifically designed to learn sequential data is a common tool for such tasks. In

[1]Department of Chemistry, Indian Institute of Technology Bombay, Mumbai 400076, India

[2]Centre for Machine Intelligence and Data Science, Indian Institute of Technology Bombay, Mumbai 400076, India

[3]Lead contact

*Correspondence: sukriti243@gmail.com (S.S.), sunoj@chem.iitb.ac.in (R.B.S.)

https://doi.org/10.1016/j.isci.2022.104661

the recent years, RNN has become as an important method for generative models, and have found a wide range of applications in natural language processing (NLP), images, speech, music, computer code generation and so on (Young et al., 2018; van den Oord et al., 2016; Graves et al., 2004; Bhoopchand et al., 2016; Eck and Schmidhuber, 2002; Wu et al., 2018; Horwood and Noutahi, 2020). The analogy between language and SMILES presents an intriguing opportunity to employ RNNs for molecule generation (Kotsias et al., 2020; Gupta et al., 2018; Dollar et al., 2021).

One of the key aspects in the use of deep generative models is to fine-tune the model so as to render it capable of generating molecules with desired properties. With the advent of large publicly available chemical databases, various techniques for molecular design have recently been developed to explore the chemical space (Sterling and Irwin, 2015; Gindulyte et al., 2016; Gaulton et al., 2017). These include variational autoencoders (VAE) (Blaschke et al., 2018; Popova et al., 2018), adversarial autoencoders (Putin et al., 2018), generative adversarial networks (GAN) (Prykhodko et al., 2019; Feng et al., 2021), reinforcement learning (RL) (Olivecrona et al., 2017), and transfer learning (TL) (Skinnider et al., 2021; Santana et al., 2021; Yuan et al., 2020), all employing RNN for sequence generation. The neural network-based methods are generally known to require large data for efficient learning of chemical knowledge. This very requirement might limit the application of deep generative models to many practical situations in the chemical space, where only smaller data are available.

The TL approach has shown promise in addressing problems in the small data regimes (Karpov et al., 2021). In TL, the RNN learns from a large dataset(s) and transfers the knowledge by way of fine-tuning it on a smaller dataset of interest. This protocol can therefore help generate molecules, similar to those in the available limited sized dataset of immediate utility. For example, Segler et al. developed a generative RNN model by training on 1.4 million molecules drawn from the ChEMBL library, which was subsequently fine-tuned on a smaller set of molecules known to be active against certain bacteria (Segler et al., 2018). This protocol was able to generate novel molecules for *de novo* drug discovery applications. It is of high timely significance to make use of RNNs on such a large library of molecules for reaction discovery.

It is important to note that fluorine-containing compounds continue to find an increasing number of applications in the pharmaceutical industry (Yerien et al., 2016). Currently, at least one fluorine atom is present in nearly 20% of the marketed drugs (Purser et al., 2008). Among the various methods for the synthesis of fluorinated compounds, deoxyfluorination is an attractive strategy that converts alcohols to the corresponding fluorides by the action of a fluorinating agent (Figure 1A) (Nielsen et al., 2018). This is one of the most popular fluorination methods, as the starting material such as alcohol is abundant, both in nature as well as in synthetic precursor libraries (Champagne et al., 2015; Singh and Shreeve, 2002; Campbell and Ritter, 2014). Owing to the importance of fluorinated compounds, extensive research has been carried out in this area and several fluorinating agents have now been developed (Sorlin et al., 2020; L'Heureux et al., 2010; Jia et al., 2021). Among them, sulfonyl fluorides are inexpensive, thermally stable, and highly modular. They can be easily synthesized and have also been fine-tuned for specific alcohols (Dong et al., 2014; Davies et al., 2017; Tribby et al., 2017). Given the significance of fluorination reactions and the potential of ML in reaction discovery, we consider it important to examine whether such a reaction space be explored using ML tools.

A large body of work on the applications of deep generative models for molecular design has been focused on drug discovery (Elton et al., 2019; Mendez-Lucio et al., 2020; Lim et al., 2020; Mehta et al., 2021). We wondered if the capabilities of these methods could suitably be tailored to fit the reaction discovery workflow. In this study, we develop a combination of RNN-based deep generative and predictive models for an important reaction, namely the deoxyfluorination. We wish to demonstrate how novel substrate molecules (alcohols) can be generated and subsequently predict the yield of fluorination for the generated alcohols. These methods may become a promising tool if new molecules can be designed, the efficacy of a given transformation examined, and potentially high yielding substrates thus be identified.

RESULTS AND DISCUSSION

The Doyle group has screened a large number of alcohols against a selected set of bases and sulfonyl fluorides (Figure 1A) (Nielsen et al., 2018). The diversity in this reaction space consisting of 37 alcohols, when subjected to 20 reaction conditions offered by 5 sulfonyl fluorides and 4 bases, leads to a total of 740 distinct reactions. Thus, the data from a complete reaction space, as obtained using high-throughput experimentation (HTE), is available.

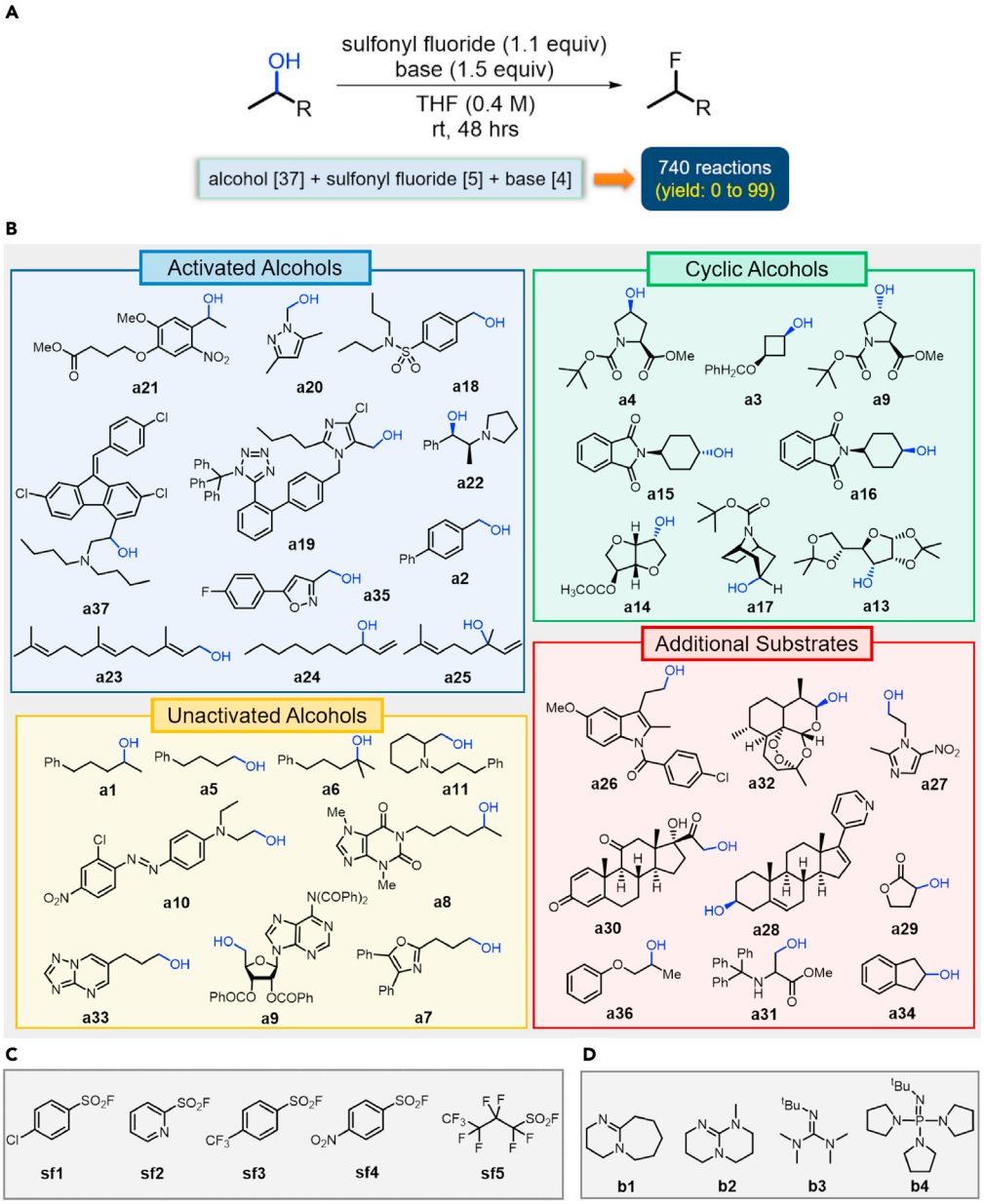


Figure 1. General scheme

(A–D) (A) A general representation of deoxyfluorination reaction and the substrate scope of (B) alcohols, (C) sulfonyl fluorides, and (D) bases.

The diversity of the reaction components involved in the deoxyfluorination can be appreciated from Figure 1. A range of sulfonyl fluorides, such as the commercially available PyFluor (**sf2**) and perfluorobutanesulfonyl fluoride (PBSF, **sf5**) were evaluated besides arylsulfonyl fluorides (**sf1**, **sf3**, **sf4**) that can be readily obtained from the corresponding sulfonyl chlorides (Figure 1C). These sulfonyl fluorides are expected to span a broad range of reactivity. The bases chosen in this study are sterically diverse, ranging from a compact base such as DBU (**b1**) to a bulky phosphazene BTPP (**b4**) (Figure 1D). Alcohols are the most common precursor for fluorination and a much higher level of diversity with the alcohols could as well be noted. In the case of the unactivated alcohols, the choices consist of primary, secondary, and tertiary alcohols. Substrates containing 4- and 5-membered cyclic alcohols as well as cyclohexanols were also considered. Activated substrates such as benzylic and allylic alcohols were also found in the repertoire. Furthering

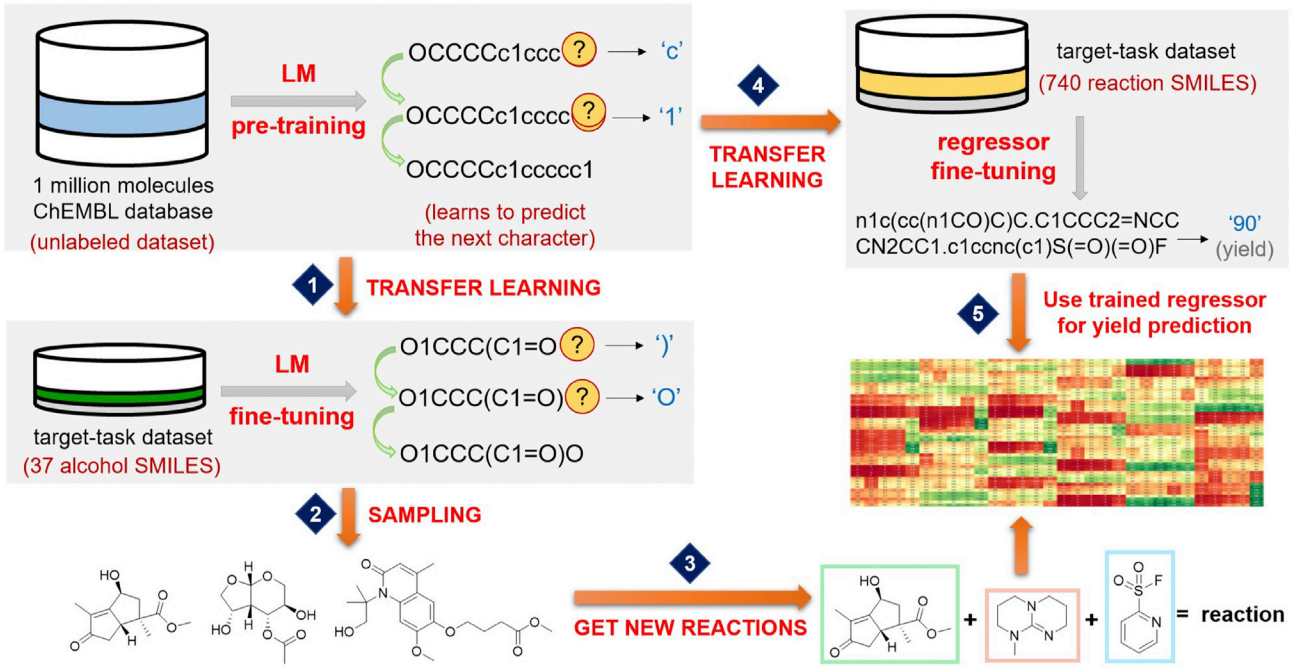


Figure 2. An overview of the transfer learning approach used for generating the focused library of novel substrates.

the spread of the substrate by including homobenzylic and homoallylic alcohols along with α/β-hydroxy carbonyls and hemiacetals are to be noted as well (Figure 1B).

Given the importance of fluorinated compounds that could be derived from the alcohol precursors listed in Figure 1, we herein employ the RNN-based deep generative models to extend the substrate scope for alcohols. An overview of our approach is shown in Figure 2. To generate a focused library of molecules, a two-step learning procedure is adopted. The model is first trained on a large and diverse dataset of molecules drawn from the ChEMBL database.[10c] Transfer learning is then used to fine-tune the model on a smaller data set containing the alcohols. The sampling is performed from the fine-tuned model to obtain new alcohol molecules. To identify the promising substrates, a regression model is subsequently trained to predict the yield of the fluorination reaction of the generated alcohols. For improved clarity, we have organized the discussions into three key sections as described in the following sections.

Training the language model

The first task is to train a language model (LM) designed to learn molecular representation. In the LM training, a probability distribution of the next word from a given sequence of words is predicted (Goldberg, 2016). LMs can capture both the grammar and semantics of the language. The similarity between natural language and SMILES representation of molecules suggests that one can also model molecules as LM. Given a sequence of SMILES strings, a SMILES-based chemical LM can learn to predict the next character (Figure 2). Two additional tokens, 'BOS' (beginning of string) and 'EOS' (end of string), are added respectively to the beginning and end of each SMILES string, to indicate the start and end of a SMILES sequence. In order to use the SMILES strings as an input for ML, the sequences are subsequently tokenized and encoded as one-hot vector representation. The RNN architecture used in the training of the LM is shown in Figure 3A (Figure S1) (Howard and Sebastian, 2018).

The random SMILES variant for any given molecule is generated following the Bjerrum SMILES enumeration procedure (Bjerrum, 2017). It allows for data augmentation, which is particularly useful for problems with relatively lower data size (Kimber et al., 2018). The data augmentation provides valid SMILES with the main difference that the starting atom and the direction of traversing the graph are chosen randomly. Given the small data size of only 37 alcohols, we have augmented the training data using randomized SMILES (Moret et al., 2020). We note that the data augmentation is beneficial for the LM fine-tuning and

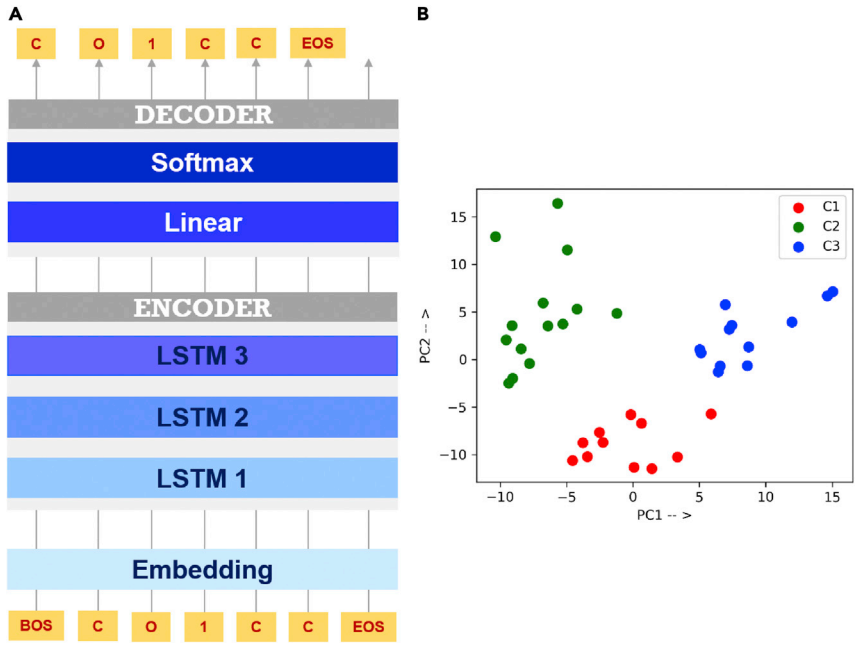


Figure 3. Language model training

(A) The RNN architecture used in the training of the LM.

(B) A plot of the first two principal components obtained from the PCA of the encoder output of the fine-tuned model comprising 37 alcohol molecules of interest. The clusters are identified using the k-means clustering method.

therefore for the generation task. Once the data and model architecture are ready, the LM can be trained, and new molecules can be generated as discussed in the following sections.

The generation of molecules utilizing deep generative models can be divided into two tasks; (a) learning the valid molecular representation from the SMILES strings, and (b) learning the task-specific representation to generate the desired functional molecules. The first task requires the model to be trained on a large and diverse dataset to efficiently learn the syntax of valid SMILES and general features of molecules therein. To accomplish this, we trained a chemical LM using SMILES strings of one million molecules from the ChEMBL database. For the second task, a small dataset containing the SMILES strings of 37 alcohols is used (Figure 1B). Following the TL approach, the LM is then fine-tuned on the above dataset to learn the task-specific features (Table S2). The primary objective here is to adapt the model to generate SMILES strings that are similar to the smaller dataset of alcohols of interest. Because the data size is of modest size for deep generative models, we extracted the encoder output to examine whether the fine-tuned model has actually learned relevant features (Figures 3A and S2). The encoder takes the input and summarizes the information (e.g., the context and temporal dependencies of the sequence) into a vector of fixed length. The encoded information could then be used by the decoder for various tasks (Sutskever et al., 2014). To gather insights into what information the model has encoded, we have analyzed the final hidden state of the encoder. For the purpose of visualization, principal component analysis (PCA) is used to project the high dimensional data onto a lower-dimensional space. Next, a k-means clustering is performed on the first two principal components, as obtained through the PCA. To our delight, three distinct clusters of alcohol molecules are identified, as elaborated below (Figure 3B).

A closer look at the identities of the alcohols present in each of these clusters provided some interesting details. For instance, cluster C1 (shown in red color) primarily consists of relatively simple and small sized open chain alcohols (e.g., **a1**, **a5**, **a11**, **a24** etc., as shown in Figure 1B). The activated alcohols (except allylic alcohols) are exclusively found in cluster C2 (green). Members of cluster C2, such as **a2**, **a18**, **a19**, **a26**, **a27** and so on, are found to contain aryl and heteroaryl groups. Most of the cyclic alcohols group into cluster C3 (blue, e.g. **a4**, **a13**, **a14**, **a15**). The remaining alcohols (e.g., **a29**, **a36**) are distributed among all the three clusters. These observations convey that our fine-tuned model has learned some substrate-specific details

from the SMILES representation that it was provided with. This is an encouraging insight that engenders additional confidence in the model for generating newer molecules.

Generating novel substrate molecules and chemical space analysis

Having a trained chemical LM at hand, one can use the same for the generation of SMILES of new molecules. This procedure involves sampling one character/token at a time, where the tokens are sampled from the probability distribution predicted by the model. We specify 'BOS' as the first token and progressively sample the next token until the 'EOS' token is sampled, or a predefined length constraint is reached. The generative behavior of the LM can be modified by changing the softmax temperature. Before applying the final softmax (Figure 3A), its inputs are divided by the sampling temperature. As a result, the output probabilities get adjusted and thus the degree of randomness of the generated SMILES could be controlled. The goodness of the fine-tuned model is evaluated by examining the validity, uniqueness, and novelty of the generated alcohol molecules (Table S3). Here, validity is the percentage of valid SMILES generated by the fine-tuned model, uniqueness measures the model's ability to generate unique SMILES that has not been already sampled, and novelty implies the percentage of unique molecules not present in the training set.

An important aspect of deep generative models is its use in sampling a larger chemical space. In the present context, the alcohols in the training set span structurally diverse regions of the chemical space and the generated alcohols are therefore expected to cover the corresponding regions. In addition, if the new samples get more spread out, beyond those in the training set alcohols, newer reactions even with higher yields can be identified. We sampled 500 SMILES strings from the fine-tuned model and could obtain good quality generated molecules, as indicated by the validity of 97%, uniqueness of 92%, and novelty of 54%. For the ease discussion, we have selected a total of 75 novel alcohols generated using the fine-tuned model (Table S4). To visualize the chemical space covered by the alcohols in the training set as well as those in the generated set, we have used the UMAP (Uniform Manifold Approximation and Projection) plot. UMAP is a dimensionality reduction technique that is designed to visualize complex data in lower dimensions (McInnes et al., 2020). For this purpose, the alcohol molecules of both the generated and training sets are first converted into 166-bit 2D structure fingerprints, known as MACCS (Molecular Access System) keys (Durant et al., 2002). The UMAP plot generated by using these MACCS keys is shown in Figure 4 (Figure S3).

The UMAP plot as shown in Figure 4 reveals that our model was able to thoroughly explore and populate the chemical space analogous to the training set. More interestingly, the extended regions of the chemical space are covered by filling in the gaps with novel alcohols. Because similar molecules cluster together in the UMAP plot, we have divided the chemical space in seven clusters to simplify the discussion. The identity of alcohols in the training set and some representative members from the generated set present in each cluster is shown in Figure 4. We note that the generated alcohols span a range from minor to more complex modifications. For instance, a minor replacement of the phenyl substituent in **a7** with a pyridyl, *p*-fluoroaryl, or -H leads to different generated alcohols as shown in Figure 5A. Our model is shown to combine different structural motifs from the training set to generate new alcohols (Figure 5B). More complex modifications are observed with **a28** and **a30**, where different numbers and combinations of 5, 6, and 7 membered rings provide various generated alcohols (Figure 5C). The notably different set of generated alcohols listed in Figure 5D can be regarded as an admixture of substructure motifs as well as complex modifications.

It shall be noted that the ChEMBL dataset, as utilized for pretraining, may also contain alcohol molecules. A comparison of the generated alcohols as obtained using the pretrained and those from the fine-tuned models are as well performed. We have sampled 500 SMILES strings from the pretrained model and have obtained a validity of 96%, with a uniqueness of 90%, and novelty of 96%. Thus, a total of 449 novel SMILES could be obtained (Table S12). As anticipated, these newly generated molecules were not only alcohols. Subsequent filtration has identified only 24 novel alcohol molecules, which is less than 6% of the generated molecules. On the other hand, >98% of the generated molecules from the fine-tuned model are found to be alcohols. It is important to note that the fine-tuned model can effectively generate a focused library of molecules, whereas the use of the pretrained model leads to the majority of undesired molecules thus entails extensive filtering to identify molecules of interest. Although the alcohols generated from the pretrained model are all valid, our main goal is to extend the substrate scope in such a way that the generated alcohols expediently undergo the fluorination reaction. Hence, they should preferably be similar to the ones known to yield the fluorinated product. Instead of generating a random set of alcohols, we are

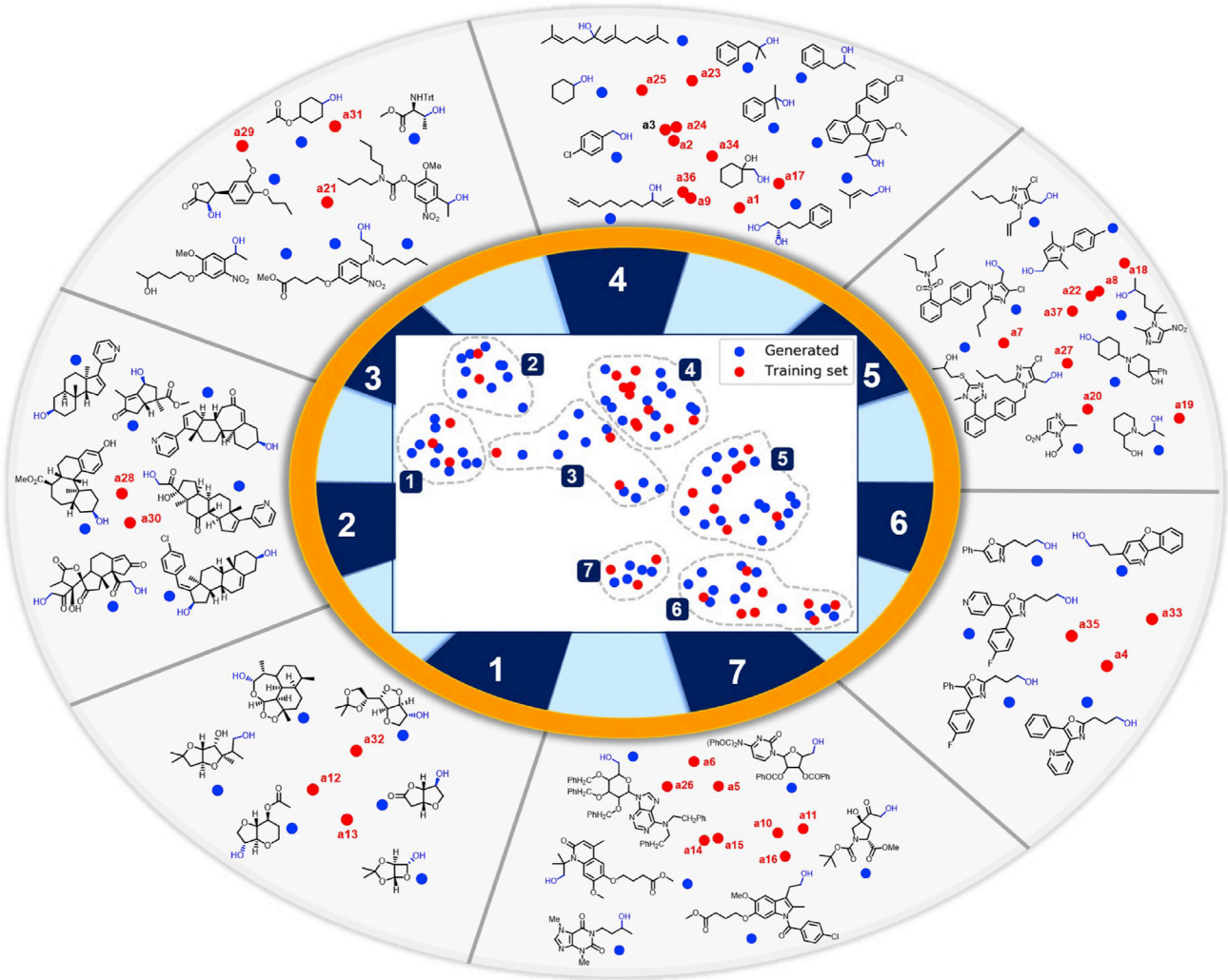


Figure 4. The UMAP plot of the chemical space of alcohols in the training set (represented as red dots) and those in the generated set (blue) are shown in the center of the diagram

A few representative structures of the generated alcohols are shown along the peripheral blocks. The identity of the training set alcohols ($\mathbf{a}_n$, where n = 1, 2, …, 37) can be found in Figure 1B.

more interested in the ones similar to the 37 alcohols employed in fine-tuning, necessitating the use of transfer learning. In addition, this study serves as a proof-of-concept demonstrated for the deoxyfluorination reaction, which can as well be extended to other classes of reactions, where the molecules of interest may not be present in the pretraining set.

To assess how similar or different the generated alcohols are to the training set, a nearest neighbor similarity analysis is carried out using the fingerprint-based similarity method (Tanimoto index) (Chevillard and Kolb, 2015). The MACCS key fingerprint is used to compute the Tanimoto similarity matrix. The mean neighbor similarity between the generated and training set alcohols is found to be 0.33, implying that the generated molecules are diverse (Figure S4). Another important aspect that needs to be considered at this point is the synthetic feasibility of the generated molecules. The ease of synthesis is examined using the synthetic accessibility score (SAS) (Ertl and Schuffenhauer, 2009; Gao and Coley, 2020). The mean SA score of the generated alcohols is found to be 3.06, which follows very closely to 2.75 for the training set candidates (Figure S5 and Table S5). These SA scores of the training and generated alcohols can be considered as the first level of assurance that the generated molecules are not difficult to synthesize (Table S6).

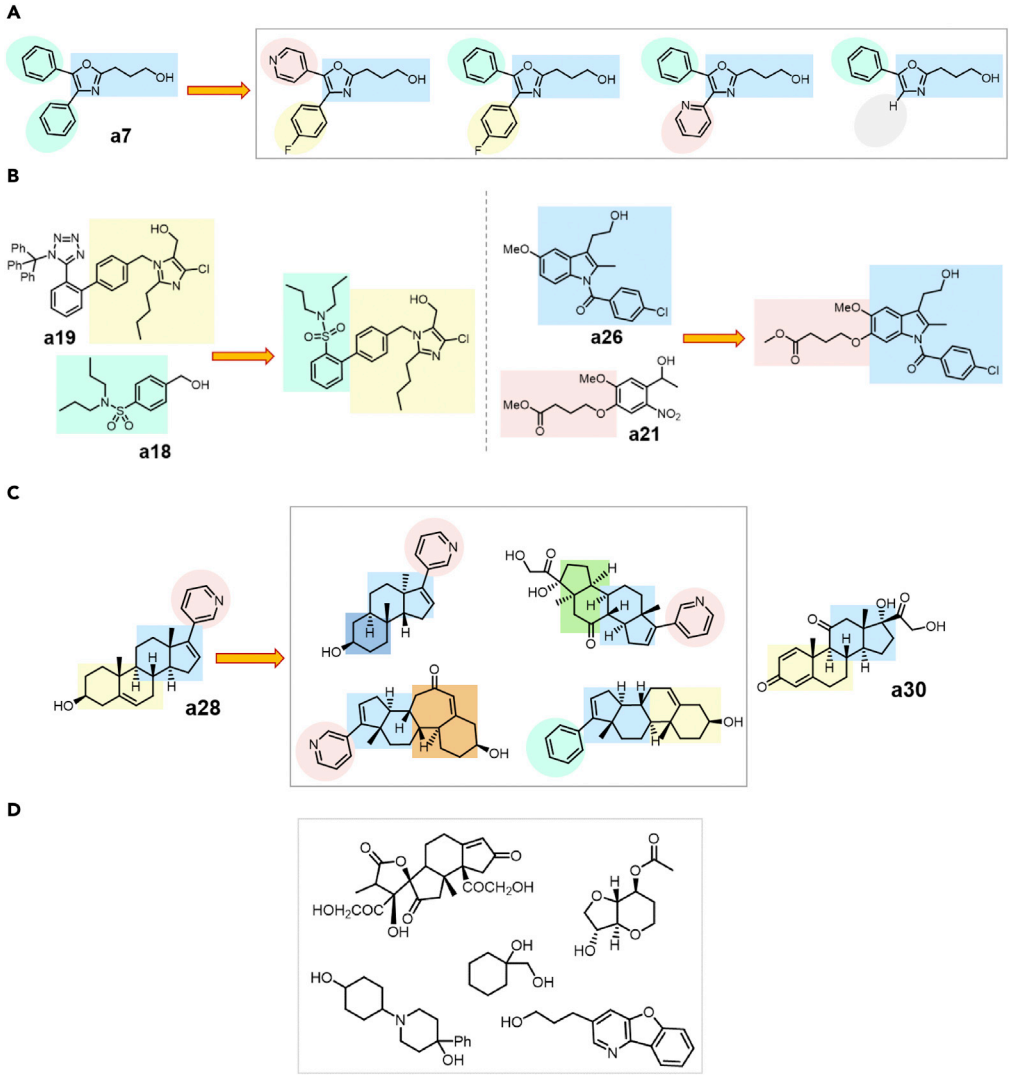


Figure 5. A representative set of generated alcohols

(A–C) Selected examples of the generated alcohols with (A) minimal changes, (B) as a combination of different structural motifs from the training set alcohols, and (C) complex modifications as compared to the training set alcohols. The generated alcohols are shown enclosed in gray color boxes.

(D) Notably different set of generated alcohols.

Training the regressor for yield prediction

After having demonstrated that our fine-tuned model has sufficiently been able to explore the chemical space, as evident from the diversity of the generated molecules, the next step is to identify the likely high yielding candidates for the deoxyfluorination reaction. For this purpose, we have used TL to build a regression model to predict the yield of reactions corresponding to the generated alcohols. As discussed in the earlier section (Figure 1), there are three reaction partners; alcohol, sulfonyl fluoride, and base. All three participating entities are known to affect the reaction outcome. Therefore, the SMILES strings of the individual components are merged together, as shown in Figure 6A for a representative reaction. The process of combining the strings, known as concatenation of SMILES, forms a composite representation of a given reaction. It is then encoded as a one-hot vector that serves as the input to the ML model, thus rendering it conducive for downstream tasks.

For the regression task, the LM architecture is slightly modified in the decoder region by introducing two linear layers (Figures 3A and S1). To predict the yield of reactions, the regressor is fine-tuned using the

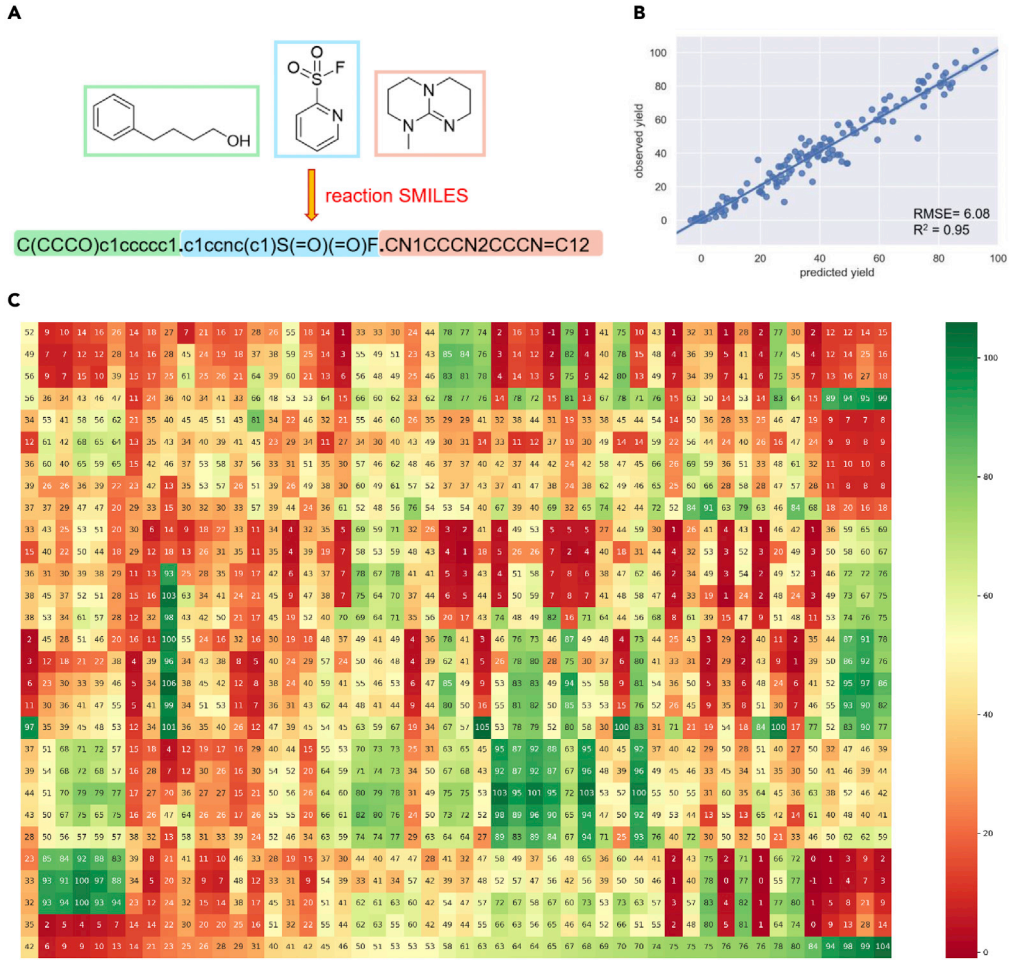


Figure 6. Yield prediction

(A) Reaction SMILES for a representative sample obtained through concatenation of individual SMILES of the alcohol, sulfonyl fluoride, and base.

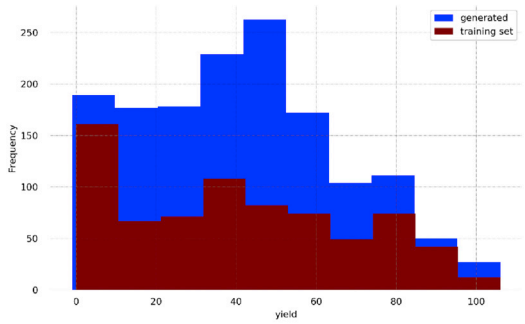
(B) The plot of predicted versus experimentally observed yields for the best test set (RMSE = 6.08). The average RMSE over all the 10 test sets is 7.05 $\pm$ 0.59.

(C) A heat map for the predicted yield of 1500 reactions corresponding to the 75 newly generated alcohols.

pretrained LM. Here, the labeled data consists of 740 deoxyfluorination reactions, whose yields are previously reported (Figure 1A). Prior to utilizing the trained regression model to predict the yield of the generated alcohols, it is important to identify how good the trained model is. To do this, the data is split randomly into 70:10:20 train-validation-test sets and different hyperparameters are tuned on the validation set. To examine if the composition of the validation set has any impact, hyperparameter tuning is performed on 3 random train-validation splits. The optimal set of hyperparameters is then used for prediction on the test set (Tables S7 and S8). The model is evaluated using root mean square error (RMSE) as the error metric. To ensure good technical quality, 10 different train-test splits are used and the final performance is reported in terms of the RMSE (in yield), averaged over the 10 independent runs. We could obtain an average test RMSE of 7.05 $\pm$ 0.59 (Table S9). The quality of predictions can alternatively be understood with the fact that out of 1480 predictions in all the 10 runs, 1309 (i.e., 88%) are within 10 units of the experimentally known values. The model performance can further be gleaned from the excellent R^2 as obtained by plotting the predicted versus observed yields for the best test set (Figure 6B).

Once the trained regression model is available, it can be utilized to predict the yield of reactions of the newly generated alcohols. The selected 75 alcohols are novel and have good synthetic accessibility and are similar to those in the training set alcohols (as discussed in the earlier section). The combination of

A



B

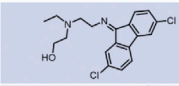
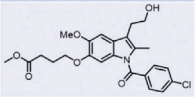
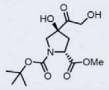
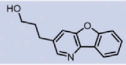
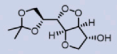
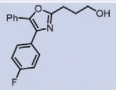
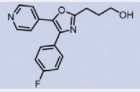
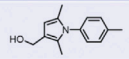
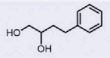
Generated Alcohols	Sulfonyl Fluorides	Bases	Predicted Yield	Generated Alcohols	Sulfonyl Fluorides	Bases	Predicted Yield
	sf5	b4	83		sf5	b3	99
	sf3	b2	75		sf5	b2	76
	sf4	b3	82		sf5	b2	84
	sf5	b2	99		sf3	b4	94
	sf1	b4	95		sf1	b4	93
	sf3	b3	78		sf5	b4	84

Figure 7. Identifying promising new reactions

(A) A plot for comparison of distribution of samples across various yield regimes.

(B) A combination of promising generated alcohols, sulfonyl fluorides, and bases with the corresponding predicted yield.

75 generated alcohols with 5 sulfonyl fluorides and 4 bases results in 1500 distinct reactions, whose yields are predicted. Therefore, we believe that the reactions are accurate and novel. A heatmap showing the predicted yields of all the 1500 reactions is provided in Figure 6C. We can focus primarily on the green pixels to identify those reactions, which are predicted to give high yields. It can be noted from the plot that the high yielding reactions are lesser in number as compared to those with lower yield. To impart additional clarity on these findings, we have compared the distribution of yields of the generated reactions to the previously reported experimental yields. An analysis of the prediction accuracies for the 740 known reactions used in building the regression model, spread across different class intervals (0 to 10, 11 to 20, …, 91–100) of the reaction yield, is carried out to examine whether the regression model is biased toward predicting the output values around the 50% yield. The average RMSE of our model is 7.05 $\pm$ 0.59 while that in the 40–50 and 50–60 yield windows are respectively found to be 8.84 and 7.19, revealing no distinctive improvements around the 50% yield value (Table S10). A histogram comparing the distribution of samples in various yield regimes is shown in Figure 7A. It is evident from the plot that both the original and generated set of reactions followed nearly the same distribution with a lesser number of samples in the high yield regions. The mean yield of the original and generated set of reactions is 40 and 41 respectively. The predicted yields with the generated alcohols are promising, as evident from 727 new reactions having yields higher than the average yield. If we assume above average yields of >60% as very good, 331 reactions with the generated alcohols fall in >60% yield regime as compared to 189 from the reported reactions. A representative set of promising new reactions with high yields is shown in Figure 7B (Table S11). The data containing 740 deoxyfluorination reactions were collected from the previously reported high-throughput experimentation (HTE). In some cases, the reported experimental yields are greater than 100, which were attributed to experimental error. All these yield values were used as is in our ML regression model. Thus, it is expected

that the predicted yields in a few instances also go marginally higher than 100% as discernible from Figure 6C.

CONCLUSION

Through this study, a proof-of-concept is demonstrated wherein an RNN-based deep generative model is developed on a small library of molecules to explore the chemical space to generate novel substrate molecules. This is augmented with an ensuing regression model to predict a complex quantity such as the yield of reaction to identify the likely high yielding substrates. Although the technique can find broader practical utility in reaction development, we have chosen an important class of deoxyfluorination of alcohol as a prototype reaction for conveying the potential of our approach. We have used transfer learning (TL) to fine-tune a chemical language model (LM) on just 37 alcohols in the training set to generate focused substrate libraries. Using the fine-tuned model, good quality new alcohol molecules (with validity = 97%, uniqueness = 92%, and novelty = 54%) are generated. The generated alcohols are found to cover the chemical space corresponding to the training set and beyond. The quality of the generated alcohols is assessed by the fact that they are sufficiently diverse and possess good synthetic accessibility scores indicating their ease of synthesis. To identify high yielding reactions of the generated alcohols, a regression model is built on previously reported 740 deoxyfluorination reactions. A well-trained model with an RMSE of 7.05 ± 0.59 is then used to predict the yield of this new set of reactions. We could identify 331 new reactions with yields greater than 60%, which is a considerable advancement with respect to the known reactions and the fact that fluorination is generally known to be a low yielding endeavor. We believe that this protocol would have superior practical utility when incorporated into the reaction discovery pipeline, as the approach begins with a relatively small number of substrates.

Limitations of the study

Although a large number of valid and unique SMILES could be generated, the novelty of the generated SMILES remains low. This implies that the model remembers some of the valid SMILES from the training set and can therefore lead to over-fitting, as evident from notable share of duplicates. This could be attributed to the small size of the data used in fine-tuning in the present study. Thus, it alludes to the need for investigating different fine-tuning strategies, which could be taken up in a future study for further improvements.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- METHOD DETAILS
  - Model architecture
  - Programming details
  - Language model fine-tuning
  - Analysis of the encoder output
  - Sampling from trained language model
  - Evaluation of fine-tuned models
  - Generated alcohols
  - Uniform manifold approximation and projection (UMAP)
  - Similarity analysis
  - Synthetic accessibility (SA) score
  - Fine-tuning of regressor
  - Yield of newly generated set of reactions
  - Generated alcohols from the pre-trained model

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2022.104661.

ACKNOWLEDGMENTS

SS acknowledges a fellowship through the research associate scheme of IIT Bombay.

AUTHOR CONTRIBUTIONS

SS and RBS designed research objectives and wrote the manuscript. SS carried out research and analyzed the data.

DECLARATION OF INTERESTS

Authors declare no conflicting financial interests.

Received: April 3, 2022
Revised: May 20, 2022
Accepted: June 16, 2022
Published: July 15, 2022

REFERENCES

Atz, K., Grisoni, F., and Schneider, G. (2021). Geometric deep learning on molecular representations. Nat. Mach. Intell. *3*, 1023–1032. https://doi.org/10.1038/s42256-021-00418-8.

Bhoopchand, A., Rocktaschel, T., Barr, E., and Riedel, S. (2016). Learning python code suggestion with a sparse pointer network. Preprint at arXiv. 1611.08307.

Bjerrum, E.J. (2017). SMILES enumeration as data augmentation for neural network modeling of molecules. Preprint at arXiv. 1703.07076.

Blaschke, T., Olivecrona, M., Engkvist, O., Bajorath, J., and Chen, H. (2018). Application of generative autoencoder in de novo molecular design. Mol. Inf. *37*, 1700123. https://doi.org/10.1002/minf.201700123.

Butler, K.T., Davies, D.W., Cartwright, H., Isayev, O., and Walsh, A. (2018). Machine learning for molecular and materials science. Nature *559*, 547–555. https://doi.org/10.1038/s41586-018-0337-2.

Campbell, M.G., and Ritter, T. (2014). Late-stage fluorination: from fundamentals to application. Org. Process Res. Dev. *18*, 474–480. https://doi.org/10.1021/op400349g.

Champagne, P.A., Desroches, J., Hamel, J.-D., Vandamme, M., and Paquin, J.-F. (2015). Monofluorination of organic compounds: 10 years of innovation. Chem. Rev. *115*, 9073–9174. https://doi.org/10.1021/cr500706a.

Chevillard, F., and Kolb, P. (2015). SCUBIDOO: a large yet screenable and easily searchable database of computationally created chemical compounds optimized toward high likelihood of synthetic tractability. J. Chem. Inf. Model. *55*, 1824–1835. https://doi.org/10.1021/acs.jcim.5b00203.

Davies, A.T., Curto, J.M., Bagley, S.W., and Willis, M.C. (2017). One-pot palladium-catalyzed synthesis of sulfonyl fluorides from aryl bromides. Chem. Sci. *8*, 1233–1237. https://doi.org/10.1039/c6sc03924c.

Dollar, O., Joshi, N., Beck, D.A.C., and Pfaendtner, J. (2021). Attention-based generative models for de novo molecular design. Chem. Sci. *12*, 8362–8372. https://doi.org/10.1039/d1sc01050f.

Dong, J., Krasnova, L., Finn, M.G., and Sharpless, K.B. (2014). Sulfur(VI) fluoride exchange (SuFEx): another good reaction for click chemistry. Angew. Chem. Int. Ed. *53*, 9430–9448. https://doi.org/10.1002/anie.201309399.

Dong, W., Moses, C., and Li, K. (2011). Efficient k-nearest neighbor graph construction for generic similarity measures. In Proceedings of the 20th International Conference on World Wide Web, pp. 577–586.

Durant, J.L., Leland, B.A., Henry, D.R., and Nourse, J.G. (2002). Reoptimization of MDL keys for use in drug discovery. J. Chem. Inf. Comput. Sci. *42*, 1273–1280. https://doi.org/10.1021/ci010132r.

Eck, D., and Schmidhuber, J. (2002). Finding temporal structure in music: blues improvisation with LSTM recurrent networks. In IEEE Proc. 12th IEEE Workshop Neural Networks for Signal Processing, pp. 747–756.

Elton, D.C., Boukouvalas, Z., Fuge, M.D., and Chung, P.W. (2019). Deep learning for molecular design-a review of the state of the art. Mol. Syst. Des. Eng. *4*, 828–849. https://doi.org/10.1039/c9me00039a.

Ertl, P., and Schuffenhauer, A.A. (2009). Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. J. Cheminf. *1*, 8. https://doi.org/10.1186/1758-2946-1-8.

Feng, W., Xiaochen, F., Xiao, G., Lei, X., Liangxu, X., and Shan, C. (2021). Improving de novo Molecule Generation by Embedding LSTM and Attention Mechanism in CycleGAN. Front. Genet. *12*, 709500. https://doi.org/10.3389/fgene.2021.709500.

Fortunato, M.E., Coley, C.W., Barnes, B.C., and Jensen, K.F. (2020). Data augmentation and pretraining for template-based retrosynthetic prediction in computer-aided synthesis planning. J. Chem. Inf. Model. *60*, 3398–3407. https://doi.org/10.1021/acs.jcim.0c00403.

Gallarati, S., Fabregat, R., Laplaza, R., Bhattacharjee, S., Wodrich, M.D., and Corminboeuf, C. (2021). Reaction-based machine learning representations for predicting the enantioselectivity of organocatalysts. Chem. Sci. *12*, 6879–6889. https://doi.org/10.1039/d1sc00482d.

Gao, W., and Coley, C.W. (2020). The synthesizability of molecules proposed by generative models. J. Chem. Inf. Model. *60*, 5714–5723. https://doi.org/10.1021/acs.jcim.0c00174.

Gaulton, A., Hersey, A., Nowotka, M., Bento, A.P., Cibrián-Uhalte, E., Atkinson, F., Mutowo, P., Papadatos, G., Smit, I., Bellis, L.J., et al. (2017). The ChEMBL database in 2017. Nucleic Acids Res. *45*, D945–D954. https://doi.org/10.1093/nar/gkw1074.

Gindulyte, A., Shoemaker, B.A., Yu, B., Fu, G., He, J., Zhang, J., Chen, J., Wang, J., Han, L., Thiessen, P.A., et al. (2016). PubChem substance and compound databases. Nucleic Acids Res. *44*, D1202–D1213. https://doi.org/10.1093/nar/gkv951.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. J. Artif. Intell. Res. *57*, 345–420. https://doi.org/10.1613/jair.4992.

Gomez-Bombarelli, R., Wei, J.N., Duvenaud, D., Hernández-Lobato, J.M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T.D., Adams, R.P., and Aspuru-Guzik, A. (2018). Automatic chemical design using a data-driven continuous representation of molecules. ACS Cent. Sci. *4*, 268–276. https://doi.org/10.1021/acscentsci.7b00572.

Graves, A., Eck, D., Beringer, N., and Schmidhuber, J. (2004). Biologically plausible speech recognition with LSTM neural nets. In International Workshop on Biologically Inspired Approaches to Advanced Information Technology, pp. 127–136.

Griffiths, R.-R., and Hernández-Lobato, J.M. (2020). Constrained Bayesian optimization for automatic chemical design using variational autoencoders. Chem. Sci. *11*, 577–586. https://doi.org/10.1039/c9sc04026a.

Grisoni, F., Moret, M., Lingwood, R., and Schneider, G. (2020). Bidirectional molecule generation with recurrent neural networks. J. Chem. Inf. Model. *60*, 1175–1183. https://doi.org/10.1021/acs.jcim.9b00943.

Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., and Baroni, M. (2018). Colorless green recurrent networks dream hierarchically. Preprint at arXiv. 1803.11138.

Gupta, A., Müller, A.T., Huisman, B.J.H., Fuchs, J.A., Schneider, P., and Schneider, G. (2018). Generative recurrent networks for de novo drug design. Mol. Inf. *37*, 1700111. https://doi.org/10.1002/minf.201700111.

Horwood, J., and Noutahi, E. (2020). Molecular design in synthetically accessible chemical space via deep reinforcement learning. ACS Omega *5*, 32984–32994. https://doi.org/10.1021/acsomega.0c04153.

Howard, J., and Gugger, S. (2020). Fastai: a layered API for deep learning. Information *11*, 108. https://doi.org/10.3390/info11020108.

Howard, J., and Sebastian, R. (2018). Universal language model fine-tuning for text classification. Preprint at arXiv. 1801.06146.

Jia, H., Häring, A.P., Berger, F., Zhang, L., and Ritter, T. (2021). Trifluoromethyl thianthrenium triflate: a readily available trifluoromethylating reagent with formal CF3+, CF3●, and CF3– reactivity. J. Am. Chem. Soc. *143*, 7623–7628. https://doi.org/10.1021/jacs.1c02606.

Karpov, K., Mitrofanov, A., Korolev, V., and Tkachenko, V. (2021). Size doesn't matter: predicting physico- or biochemical properties based on dozens of molecules. J. Phys. Chem. Lett. *12*, 9213–9219. https://doi.org/10.1021/acs.jpclett.1c02477.

Kimber, T.B., Engelke, S., Tetko, I.V., Bruno, E., and Godin, G. (2018). Synergy effect between convolutional neural networks and the multiplicity of SMILES for improvement of molecular prediction. Preprint at arXiv. 1812.04439.

Kotsias, P.C., Arús-Pous, J., Chen, H., Engkvist, O., Tyrchan, C., and Bjerrum, E.J. (2020). Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. Nat. Mach. Intell. *2*, 254–265. https://doi.org/10.1038/s42256-020-0174-5.

Kutchukian, P.S., Dropinski, J.F., Dykstra, K.D., Li, B., DiRocco, D.A., Streckfuss, E.C., Campeau, L.-C., Cernak, T., Vachal, P., Davies, I.W., et al. (2016). Chemistry informer libraries: a chemoinformatics enabled approach to evaluate and advance synthetic methods. Chem. Sci. *7*, 2604–2613. https://doi.org/10.1039/c5sc04751j.

L'Heureux, A., Beaulieu, F., Bennett, C., Bill, D.R., Clayton, S., LaFlamme, F., Mirmehrabi, M., Tadayon, S., Tovell, D., and Couturier, M. (2010). Aminodifluorosulfinium salts: selective fluorination reagents with enhanced thermal stability and ease of handling. J. Org. Chem. *75*, 3401–3411. https://doi.org/10.1021/jo100504x.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. Nature *521*, 436–444. https://doi.org/10.1038/nature14539.

Li, W., Ma, H., Li, S., and Ma, J. (2021). Computational and data driven molecular material design assisted by low scaling quantum mechanics calculations and machine learning. Chem. Sci. *12*, 14987–15006. https://doi.org/10.1039/d1sc02574k.

Lim, H., and Jung, Y.J. (2019). Delfos: deep learning model for prediction of solvation free energies in generic organic solvents. Chem. Sci. *10*, 8306–8315. https://doi.org/10.1039/c9sc02452b.

Lim, J., Hwang, S.-Y., Moon, S., Kim, S., and Kim, W.Y. (2020). Scaffold-based molecular design with a graph generative model. Chem. Sci. *11*, 1153–1164. https://doi.org/10.1039/c9sc04503a.

McInnes, L., Healy, J., and Melville, J. (2020). UMAP: uniform manifold approximation and projection for dimension reduction. Preprint at arXiv. 1802.03426.

Mehta, S., Laghuvarapu, S., Pathak, Y., Sethi, A., Alvala, M., and Priyakumar, U.D. (2021). MEMES: machine learning framework for enhanced MolEcular screening. Chem. Sci. *12*, 11710–11721. https://doi.org/10.1039/d1sc02783b.

Mendez-Lucio, O., Baillif, B., Clevert, D.-A., Rouquié, D., and Wichard, J. (2020). De novo generation of hit-like molecules from gene expression signatures using artificial intelligence. Nat. Commun. *11*, 10. https://doi.org/10.1038/s41467-019-13807-w.

Merity, S., Keskar, N.S., and Socher, R. (2017). Regularizing and optimizing LSTM language models. Preprint at arXiv. 1708.02182.

Mi, W., Chen, H., Zhu, D.A., Zhang, T., and Qian, F. (2021). Melting point prediction of organic molecules by deciphering the chemical structure into a natural language. Chem. Comm. *57*, 2633–2636. https://doi.org/10.1039/d0cc07384a.

Miljkovic, F., Rodríguez-Pérez, R., and Bajorath, J. (2021). Impact of artificial intelligence on compound discovery, design, and synthesis. ACS Omega *16*, 33293–33299. https://doi.org/10.1021/acsomega.1c05512.

Moret, M., Friedrich, L., Grisoni, F., Merk, D., and Schneider, G. (2020). Generative molecular design in low data regimes. Nat. Mach. Intell. *2*, 171–180. https://doi.org/10.1038/s42256-020-0160-y.

Nielsen, M.K., Ahneman, D.T., Riera, O., and Doyle, A.G. (2018). Deoxyfluorination with sulfonyl fluorides: navigating reaction space with machine learning. J. Am. Chem. Soc. *140*, 5004–5008. https://doi.org/10.1021/jacs.8b01523.

Olivecrona, M., Blaschke, T., Engkvist, O., and Chen, H. (2017). Molecular de-novo design through deep reinforcement learning. J. Cheminf. *9*, 48. https://doi.org/10.1186/s13321-017-0235-x.

Paszke, A., Gross, S., Chintala, S., Chana, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. In 31st Conf Neural Inf Process Syst.

Popova, M., Isayev, O., and Tropsha, A. (2018). Deep reinforcement learning for de novo drug design. Sci. Adv. *4*, eaap7885. https://doi.org/10.1126/sciadv.aap7885.

Prykhodko, O., Johansson, S.V., Kotsias, P.C., Arús-Pous, J., Bjerrum, E.J., Engkvist, O., and Chen, H. (2019). A de novo molecular generation method using latent vector based generative adversarial network. J. Cheminf. *11*, 74. https://doi.org/10.1186/s13321-019-0397-9.

Purser, S., Moore, P.R., Swallow, S., and Gouverneur, V. (2008). Fluorine in medicinal chemistry. Chem. Soc. Rev. *37*, 320–330. https://doi.org/10.1039/b610213c.

Putin, E., Asadulaev, A., Ivanenkov, Y., Aladinskiy, V., Sanchez-Lengeling, B., Aspuru-Guzik, A., and Zhavoronkov, A. (2018). Reinforced adversarial neural computer for de novo molecular design. J. Chem. Inf. Model. *58*, 1194–1204. https://doi.org/10.1021/acs.jcim.7b00690.

Santana, M.V.S., and Silva, F.P., Jr. (2021). De novo design and bioactivity prediction of SARS-CoV-2 main protease inhibitors using recurrent neural network-based transfer learning. BMC Chem. *15*, 8. https://doi.org/10.1186/s13065-021-00737-2.

Santanilla, A.B., Regalado, E.L., Pereira, T., Shevlin, M., Bateman, K., Campeau, L.-C., Schneeweis, J., Berritt, S., Shi, Z.-C., Nantermet, P., et al. (2015). Organic chemistry. Nanomole-scale high-throughput chemistry for the synthesis of complex molecules. Science *347*, 49–53. https://doi.org/10.1126/science.1259203.

Schneider, G., and Baringhaus, K.-H. (2008). Molecular Design: Concepts and Applications (John Wiley & Sons).

Schwaller, P., Vaucher, A.C., Laino, T., and Reymond, J.L. (2021). Prediction of chemical reaction yields using deep learning. Mach. Learn.: Sci. Technol. *2*, 015016. https://doi.org/10.1088/2632-2153/abc81d.

Segler, M.H.S., Kogej, T., Tyrchan, C., and Waller, M.P. (2018). Generating focused molecule libraries for drug discovery with recurrent neural networks. ACS Cent. Sci. *4*, 120–131. https://doi.org/10.1021/acscentsci.7b00512.

Senior, A.W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Zidek, A., Nelson, A.W.R., Bridgland, A., et al. (2020). Improved protein structure prediction using potentials from deep learning. Nature *577*, 706–710. https://doi.org/10.1038/s41586-019-1923-7.

Singh, R.P., and Shreeve, J.M. (2002). Recent advances in nucleophilic fluorination reactions of organic compounds using deoxofluor and DAST. Synthesis *34*, 2561–2578. https://doi.org/10.1002/chin.200309231.

Singh, S., and Sunoj, R.B. (2022). A transfer learning protocol for chemical catalysis using a recurrent neural network adapted from natural language processing. Digital Discovery *3*, 303–312. https://doi.org/10.1039/D1DD00052G.

Singh, S., Pareek, M., Changotra, A., Banerjee, S., Bhaskararao, B., Balamurugan, P., and Sunoj, R.B. (2020). A unified machine-learning protocol for asymmetric catalysis as a proof of concept demonstration using asymmetric hydrogenation. Proc. Nat. Acad. Sci. USA *117*, 1339–1345. https://doi.org/10.1073/pnas.1916392117.

Skinnider, M.A., Stacey, R.G., Wishart, D.S., and Foster, L.J. (2021). Chemical language models enable navigation in sparsely populated chemical space. Nat. Mach. Intell. *3*, 759–770. https://doi.org/10.1038/s42256-021-00368-1.

Sorlin, A.M., Usman, F.O., English, C.K., and Nguyen, H.M. (2020). Advances in nucleophilic allylic fluorination. ACS Catal. *10*, 11980–12010. https://doi.org/10.1021/acscatal.0c03493.

Sterling, T., and Irwin, J.J. (2015). ZINC 15 – Ligand discovery for everyone. J. Chem. Inf. Model. *55*, 2324–2337. https://doi.org/10.1021/acs.jcim.5b00559.

Sun, W., Zheng, Y., Yang, K., Zhang, Q., Shah, A.A., Wu, Z., Sun, Y., Feng, L., Chen, D., Xiao, Z., et al. (2019). Machine learning–assisted molecular design and efficiency prediction for high-performance organic photovoltaic materials. Sci. Adv. *5*, eaay4275. https://doi.org/10.1126/sciadv.aay4275.

Sutskever, I., Vinyals, O., and Le, Q.V. (2014). Sequence to sequence learning with neural networks. Preprint at arXiv. 1409.3215.

Tetko, I.V., Karpov, P., Van Deursen, R., and Godin, G. (2020). State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. Nat. Commun. *11*, 5575. https://doi.org/10.1038/s41467-020-19266-y.

Tribby, A.L., Rodríguez, I., Shariffudin, S., and Ball, N.D. (2017). Pd-catalyzed conversion of aryl iodides to sulfonyl fluorides using SO2 surrogate DABSO and selectfluor. J. Org. Chem. *82*, 2294–2299. https://doi.org/10.1021/acs.joc.7b00051.

van den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. (2016). Pixel recurrent neural networks. In International Conference on Machine Learning.

Walters, W.P., and Barzilay, R. (2021). Applications of deep learning in molecule generation and molecular property prediction. Acc. Chem. Res. *54*, 263–270. https://doi.org/10.1021/acs.accounts.0c00699.

Wang, J., Hsieh, C.Y., Wang, M., Wang, X., Wu, Z., Jiang, D., Liao, B., Zhang, X., Yang, B., He, Q., et al. (2021). Multi-constraint molecular generation based on conditional transformer, knowledge distillation and reinforcement learning. Nat. Mach. Intell. *3*, 914–922. https://doi.org/10.1038/s42256-021-00403-1.

Winter, R., Montanari, F., Steffen, A., Briem, H., Noé, F., and Clevert, D.-A. (2019). Efficient multi-objective molecular optimization in a continuous latent space. Chem. Sci. *10*, 8016–8024. https://doi.org/10.1039/c9sc01928f.

Wu, Z., Ramsundar, B., Feinberg, E.N., Gomes, J., Geniesse, C., Pappu, A.S., Leswing, K., and Pande, V. (2018). MoleculeNet: a benchmark for molecular machine learning. Chem. Sci. *9*, 513–530. https://doi.org/10.1039/c7sc02664a.

Yerien, D.E., Bonesi, S., and Postigo, A. (2016). Fluorination methods in drug discovery. Org. Biomol. Chem. *14*, 8398–8427. https://doi.org/10.1039/c6ob00764c.

Young, T., Hazarika, D., Poria, S., and Cambria, E. (2018). Recent trends in deep learning based natural language processing [review article]. IEEE Comput. Intell. Mag. *13*, 55–75. https://doi.org/10.1109/mci.2018.2840738.

Yuan, Q., Santana-Bonilla, A., Zwijnenburg, M.A., and Jelfs, K.E. (2020). Molecular generation targeting desired electronic properties via deep generative models. Nanoscale *12*, 6744–6758. https://doi.org/10.1039/c9nr10687a.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Data	GitHub	https://github.com/Sunojlab/TL-for-generation/tree/main/Data
Software and algorithms		
Code	GitHub	https://github.com/Sunojlab/TL-for-generation

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Raghavan B. Sunoj (sunoj@chem.iitb.ac.in) and Sukriti Singh (sukriti243@gmail.com).

Materials availability

All other data supporting the findings of this study are available within the article and the supplemental information.

Data and code availability

Data is available through GitHub (https://github.com/Sunojlab/TL-for-generation)

Code is available through GitHub (https://github.com/Sunojlab/TL-for-generation)

Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

METHOD DETAILS

Model architecture

Most of the NLP (natural language processing) models are typically trained from scratch and would require large data to afford reasonable performance. In addition, for a supervised learning problem, there may not be enough labelled data to start with. These challenges can be overcome by tapping the potential of transfer learning, wherein the knowledge acquired while learning one task (known as the source task) is retained and utilized later to solve some other related tasks (target task). Universal Language Model Fine-tuning (ULMFiT) is one such transfer learning method that can be applied for any NLP task. Here, the source task is a language model (LM) trained to predict the next word in a sentence and the target task can be any classification/regression problem. In this study, we have used ULMFiT first to train a LM for generative purpose followed by a regressor to predict the yield of reaction.

Language model architecture

For training the LM, we have used AWD-LSTM model architecture, a regular long short-term memory (LSTM) that has incorporated optimization and regularization capabilities (Merity et al., 2017). The architecture consists of an embedding layer, an encoder with three stacked LSTM layers, and a decoder layer (Figure S1A). Similar to the ULMFiT, an embedding size of 400 is used. The embedding layer converts each token into real-valued vector of length 400. The output of the embedding layer is then passed on to the encoder consisting of 3 LSTM layers with 1152 hidden activations. The hidden state of the last LSTM layer is then received as an input to the fully connected linear layer of the decoder. In the final step, a softmax function is used to calculate the probability of each token to be considered as the next token.

Regressor architecture

For training a regressor, the decoder of the LM is slightly modified to make it suitable for the regression task. In the decoder, two linear blocks with ReLU activation function are added (Figure S1B). The input to the decoder is the last hidden state of the encoder concatenated with max-pooled and mean-pooled representations of the two previous hidden states. This operation, known as concat pooling, thus provides

a feature vector of size 1200 (400*3), which is passed to the first linear layer of the decoder with 50 activations. It then forms the input to the second linear layer with one activation to predict the regression output.

Programming details

The model is implemented using PyTorch deep learning framework (Paszke et al., 2017) and fast.ai library (Howard and Gugger, 2020). All the calculations were run using the Google Colab Pro. It provides access to T4 and P100 GPUs with memory up to 25 GB.

Language model fine-tuning

The LM is first pre-trained on one million molecules from the ChEMBL database, thereby utilizing huge unlabeled data for the pre-training. The pre-training step helps the model in understanding the general syntax of SMILES (Gulordava et al., 2018). Although the pre-training step is compute- intensive, we need to do it only once and the pre-trained model could be re-used for other related tasks. Following the transfer learning (TL) approach, the knowledge learned from the pre-training could be used for the target-task with a smaller dataset. Thus, we fine-tuned the target-task LM comprising of alcohol SMILES using the pre-trained weights. At this stage, the model learns the task-specific features.

In TL, fine-tuning is a crucial step. With the ULMFiT, some NLP-specific fine-tuning methods such as gradual unfreezing, discriminative learning rates, and fit-one-cycle were introduced. Here, we have employed these techniques to fine-tune the LM. More details are provided in Table S2. A rigorous hyperparameter optimization is performed for fine-tuning the LM. For this purpose, the data is randomly partitioned into 80:20 train-test splits. The hyperparameters considered for optimization are number of epochs, learning rate and dropout rate. In addition, the effect of number of augmented SMILES on model performance is also taken into account. Accuracy, which is the ratio of number of correct predictions to total number of predictions, is used as the error metric to evaluate the model performance.

SMILES augmentation

To improve the performance of our model on small data, we have utilized a data augmentation approach. The data augmentation strategy was first put forward to tackle the problem of low-data by presenting different representations of the same entity. The successful application of data augmentation has been shown in many recent works (Fortunato et al., 2020; Tetko et al., 2020). Here, we have used the randomized SMILES (generated through different starting atom) as a technique for data augmentation. Although, the same chemical information is contained in the augmented SMILES, it helps in the learning of implicit features present in the data as the model constructs the same reaction with different SMILES strings. The SMILES augmentation of the training data is found to be very useful. We noticed that varying levels of SMILES augmentation, in the range of 0 to 150, assures improved performance. The results reflecting the impact of SMILES augmentation is provided in Table S2. Without augmentation, the number of samples in the training set is close to 33 and the corresponding test set accuracy is 0.58. With the successive increase in the number of training samples through augmentation, we noticed a significant improvement in the test set performance. The number of samples in the training set is well over 2000, when augmented with 110 SMILES per sample (although number of SMILES is varied from 0 to 150, the best results are obtained with 110). Through this approach, an impressive test accuracy of 0.86 is obtained.

Analysis of the encoder output

The ML being inherently quantitative, makes it difficult to understand the underlying mechanism of performance. It only reveals how well the model achieves its goals rather than how it was achieved. Additional measures are required to understand the inner workings of the model. In order to gather insights into what the model is actually learning, we extracted the output that the encoder passes to the decoder. The output size is same as the embedding size, i.e., 400 (see Figure S1A). First, 37 different alcohols are selected for this analysis. The 37 × 400 matrix is obtained from the encoder output. It is not possible to visualize the data containing 400 features (columns).

In these situations, principal component analysis (PCA) comes handy to gather a broad overview of the spread of the data through visualization. PCA takes as input a large feature matrix and projects it onto a lower-dimensional space, reduce them to few un-correlated important principal components (PC) that can be used for visualization. From this analysis, the first two PCs capturing most of the variance in the

data are selected. Next, a k-means clustering is performed on the first two principal components as obtained through the PCA. Based on the patterns in the data, clustering algorithm divides the entire data into groups or clusters. Clusters have the property that all the samples in the same cluster would be similar to each other and samples in different clusters would have different properties. Following this approach, we could notice some interesting clusters, details of which are presented in Figure S2A. Further details of how various alcohols are distributed between the three clusters can be gathered from Figures S2B–S2D.

Sampling from trained language model

Given the SMILES sequence of symbols S_i, the probability distribution of the next token can be estimated by using the following equation,

$$P_\theta(S_{t+1}|S_t, \ldots S_1) = \frac{\exp(y_t^k)}{\sum_{k'=1}^{k} \exp(y_t^{k'})}$$

where, y_t^k is the k^{th} element of the output vector y_t of the RNN at time step t. This distribution can be sampled to generate novel molecules. Suppose we sample a SMILES token S_{t+1} for the next time step t+1 and merge it to the existing sequence to create a new input sequence. The new input vector X_{t+1} can be fed into the RNN model to obtain the output vector y_{t+1}, which with the help of above equation gives $P_\theta(S_{t+2}|S_{t+1},\ldots S_1)$. Sampling from this distribution would generate SMILES token S_{t+2}. The newly generated token can again be appended to the previous SMILES sequence to get the input vector for the next step. This token-by-token sampling procedure is progressively continued until we have generated the desired number of characters or the end token (EOS) is sampled.

As discussed above, the sampling means the selection of next token based on the generated conditional probability distribution. There are various methods available for sampling. In this study, we have used the temperature sampling for SMILES generation.

Temperature sampling

We can obtain the class probabilities by the application of the softmax function to the logit vector $z=(z_1, \ldots,z_n)$, which is the output of the linear layer (Figure S1A). It gives the probability vector $q=(q_1, \ldots,q_n)$ by comparing z_i with all other logits (z_j) as given by the equation below,

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}, \text{ where T is the temperature parameter.}$$

When T = 1, the softmax is computed directly on the logits. T = 0.5 indicates that the model calculates the softmax on logits/0.5. With lower T values, the model becomes more confident and thus conservative, outputting only the most probable character. With higher T values, a softer probability distribution over tokens is obtained, resulting in diverse predictions and greater chance of mistakes. We have scanned a range of sampling temperatures (0.2, 0.5, 0.6, 0.7, 0.75, 0.8, 1.0, 1.2) to investigate its effect on goodness of the generated molecules. The results are summarized in Table S3.

Evaluation of fine-tuned models

We evaluated the fine-tuned model in terms of the validity, uniqueness, and novelty of the generated alcohols. Validity is the percentage of valid SMILES generated by the fine-tuned model, calculated by dividing the number of valid SMILES by the total number of SMILES sampled. Valid SMILES strings are confirmed using RDKiT. Uniqueness measures the model's ability to generate unique SMILES. It is calculated by dividing the number of unique SMILES by the total valid SMILES. Novelty implies the percentage of valid molecules not present in the training set, computed by dividing the number of novel SMILES by the number of valid SMILES. The effect of sampling temperature on validity, uniqueness, and novelty of the generated alcohols is presented in Table S3. For each sampling temperature, 100 SMILES are sampled in three independent runs. It can be noted from Table S3 that with T = 0.2, high validity of the generated samples are found, but with very low uniqueness and novelty. This can be attributed to high level of confidence in predicting the next SMILES character. Also, sampling with T > 0.7 yields greater number of valid, unique, and novel SMILES. Therefore, we have selected T = 1.0 as an optimal temperature for subsequent generation of new molecules.

Generated alcohols

With a sampling temperature of 1.0, we sampled 500 SMILES strings from the fine-tuned model and have obtained validity of 97%, uniqueness of 92%, and novelty of 54%. Thus, a total of 240 novel SMILES could be obtained. Out of 240 novel generated alcohols, a large percentage of molecules are found to be duplicates. Some of the generated substrates don't contain the desired alcohol functional group, while others have unusual stereogenic centers. An extensive post-filtering on such molecules is performed to finally select a subset of 75 alcohols, as presented in Table S4. In order to obtain more useful alcohols after applying all the filters (such as validity, uniqueness, novelty, as well as certain manual filters like desired functional groups etc), large number of molecules has to be generated. Since this work demonstrates a proof-of-concept, we have used only 75 new alcohols for further discussions.

Uniform manifold approximation and projection (UMAP)

UMAP is a dimensionality reduction technique that can be used to map a high-dimensional data to a lower-dimensional space. It starts with locating the nearest neighbors (n_neighbors) using the nearest neighbor descent algorithm (Dong et al., 2011). A graph is constructed by connecting the nearest neighbors. A hyperparameter, known as local_connectivity, is used to ensure that the points in the high-dimensional space are connected. Once we have learned the approximate manifold from the high-dimensional space in terms of the connected neighborhood graph, the next step is the projection to a lower dimensional space. Since we want a standard Euclidean distance instead of varying distances in the high dimension, a hyperparameter min_dist is used to define the minimum distance between the points. Subsequently, the algorithm identifies a good low-dimensional representation. UMAP achieves this by minimizing the cross-entropy loss function.

We have used the 166-bit MACCS (Molecular Access System) keys of the alcohols as features for the UMAP plot. The dimensionality is reduced from 166 to 2 for the purpose of visualization. The UMAP plot is shown in Figure S3. The values of hyperparameters used to generate the plot are: n_neighbors = 15, local_connectivity = 1, min_dist = 0.3.

Similarity analysis

To assess the similarity between the generated and training set alcohols, a nearest neighbor similarity analysis is carried out using the fingerprint-based similarity method (Tanimoto index). The MACCS key fingerprint is used to compute the Tanimoto similarity matrix (where a high similarity is indicated by a value closer to unity). Three different similarity matrices are calculated:

S1: The Tanimoto self-similarity between each training set alcohol to all the other alcohols in the same data set. The mean **S1** similarity is found to be 0.32.

S2: The Tanimoto self-similarity of each generated alcohol to all the other generated alcohols. The mean similarity **S2** is found to be 0.34.

S3: The Tanimoto similarity calculated for the alcohols in the training set and those in the generated set. The mean neighbor similarity between the generated and training set alcohols is found to be 0.33.

For visualization, the values in the similarity matrix are used to plot the histogram (Figure S4).

Synthetic accessibility (SA) score

One of the most important points to be considered is the synthesizability of the generated molecules. It can be assessed using the synthetic accessibility (SA) score. SA has a range from 1 to 10, where a high value implies more complex structures and thus difficult to synthesize. In this approach, the topological complexity of the molecule is assessed and complex structures are penalized. For instance, molecules with unusual ring systems, large number of stereocenters, larger molecular size and so on get more penalized. The plot shown in Figure S5 suggests that the distribution of SA scores of the generated alcohols follow very closely to those in the training set. More details are provided in Table S5. The SA score of the generated and training set alcohols is given in Table S6. The structural details of the generated alcohols can be found in Table S4, while that of training set alcohols is provided in Table S1.

Fine-tuning of regressor

The regressor is fine-tuned on the LM, pre-trained on 1 million SMILES strings. For the purpose of fine-tuning of the regressor, the model is first initialized with the pre-trained weights followed by training the full model at once. For fine-tuning, a rigorous hyperparameter optimization is performed. The data is split into 70:10:20 train-validation-test sets and different hyperparameters are tuned on the validation set. In order to examine if the composition of the validation set has any impact, hyperparameter tuning is performed on 3 random train-validation splits. The optimal set of hyperparameters is then used for prediction on the test set. In addition to number of epochs, learning rate, and dropout rate, we have also investigated the effect of SMILES augmentation. During the training, a gaussian noise (with mean zero and standard deviation σ_{g_noise}) is added to the labels (here, yield) of the augmented SMILES, which is as well tuned on the validation set. The model performance is evaluated using root mean square error (RMSE) as provided in Table S7. After optimizing the number of augmented SMILES and gaussian noise on the first validation split, the optimal values are directly (number of augmented SMILES = 100 and gaussian noise = 0.0) used on the other two splits as provided in Table S8 below.

From Tables S7 and S8, the optimal set of hyperparameters are; dropout_rate = 0.0, learning rate = 0.001, and number of epochs = 25 (average of 3 runs). The hyperparameters were chosen based on the balance between the train and validation losses. These optimal set of hyperparameters is then used for fine-tuning the target-task regressor on the pre-trained LM. We have considered 10 random 80:20 train-test splits. The final performance is reported in terms of RMSE, averaged over 10 independent runs. The results for individual runs are shown in Table S9.

Analysis of the prediction accuracies for the 740 known reactions used in building the regression model, spread across different class intervals (of yields from 0 to 10, 11–20, …, and 91–100) are given in Table S10. The average RMSE of our model is 7.05 while that in the 41–50 and 51–60 yield windows are respectively found to be 8.84 and 7.19, revealing no distinctive improvements around the 50% yield value. In addition, we have performed another analysis wherein all the true output values were arbitrarily held fixed at 50 for all samples. The RMSE between the ML predicted yields and the hypothetical value fixed at 50% was found to be as high as 28.42, again suggesting the regression model is not biased toward the 50% yield value.

Yield of newly generated set of reactions

The regressor trained on 740 known reactions is used to predict the yield of deoxyfluorination reactions for the generated set of alcohols. The predicted yield of all the 1500 new reactions is provided in Table S11. The yields in the range of 70–80, 81–90, and 91–100 are highlighted using different shades of green, with a darker shade indicating higher yields.

Generated alcohols from the pre-trained model

We have sampled same number of SMILES strings as with the fine-tuned model so as to make the comparison between the two modes of generation more direct. The structural details of the generated alcohols from pre-trained model are provided in Table S12. For further analysis of the chemical space, we have used the 166-bit MACCS (Molecular Access System) keys of the alcohols as features for the UMAP plot. The dimensionality is reduced from 166 to 2 for the purpose of visualization. The UMAP plot as shown in Figure S6, generated using n_neighbors = 15, local_connectivity = 1, min_dist = 0.3 values of hyperparameters. It can be noted from Figure S6 that the alcohols generated from the pre-trained model are mostly clustered around one region. On the other hand, the alcohols generated using the fine-tuned model cover a larger area of chemical space around the 37 training set alcohols.