Testing the limits of SMILES-based de novo molecular generation with curriculum and deep reinforcement learning

Deep reinforcement learning methods have been shown to be potentially powerful tools for de novo design. Recurrent-neural-network-based techniques are the most widely used methods in this space. In this work we examine the behaviour of recurrent-neural-network-based methods when there are few (or no) examples of molecules with the desired properties in the training data. We find that targeted molecular generation is usually possible, but the diversity of generated molecules is often reduced and it is not possible to control the composition of generated molecular sets. To help overcome these issues, we propose a new curriculum-learning-inspired recurrent iterative optimization procedure that enables the optimization of generated molecules for seen and unseen molecular profiles, and allows the user to control whether a molecular profile is explored or exploited. Using our method, we generate specific and diverse sets of molecules with up to 18 times more scaffolds than standard methods for the same sample size; however, our results also point to substantial limitations of one-dimensional molecular representations, as used in this space. We find that the success or failure of a given molecular optimization problem depends on the choice of simplified molecular-input line-entry system (SMILES).

Developing a novel drug is a complex and difficult problem beset with failure at many stages1–4. The efficiency of drug development could be improved by producing better early hits and generating novel molecules with specific properties that would improve cost, speed and effectiveness1.

Ideally, given a target and a required molecular profile, we would search for suitable molecules in all drug-like chemical space; however, given that there are an estimated $10^{60}$ synthetically accessible drug-like molecules, of which approximately $10^7$ have been synthesized, experimental methods are not sufficient for comprehensive sampling of chemical space.

Computational methods offer the promise of searching larger areas of chemical space and virtual screening is commonly used to search curated chemical libraries for potential hits5–11; however, the chemical space available for assessment is only a tiny proportion of the possible space12–14.

Instead of searching existing molecular datasets, computational de novo design models aim to create new sets of novel molecules15–18. Deep learning molecular generation tools have recently become more prevalent; they are also often paired with optimization pipelines to produce focused sets of molecules with improved performance19. Autoencoders20 are frequently used for generation and optimization21. Here the discrete representation of a molecule is converted to a continuous representation (encoded) from which its properties can be predicted and optimized. The resulting continuous representation is then converted back to a discrete molecular representation.
Bayesian optimization has been paired with variational autoencoders to explore chemical space for specific molecular profiles; however, the transition from latent space to discrete molecular representations is often non-trivial. Generative adversarial networks have been applied to de novo drug design. Together with reinforcement learning, these models have been shown to generate diverse libraries of realistic molecules with specific properties. Recent developments in state-of-the-art natural language processing tools have been implemented in de novo design with the introduction of transformer models for simplified molecular-input line-entry system (SMILES) generation. These tools have been reported with stable, and sometimes better performance than older architectures.

Like transformers, recurrent neural network (RNN)-based molecular generation models were inspired by natural language processing tools and have proved popular due to their ability to model long-term dependences in strings. Using the same training regime implemented in natural language processing, RNNs have successfully been applied to the generation of novel molecules while still using a simple network architecture. Unbiased RNNs have been shown to generate SMILES that cover large areas of chemical space (especially relative to training data). These models have also been shown to benefit from randomized SMILES with further performance improvements. An early example of this approach used RNNs to generate a molecular library through a SMILES language model, before fine-tuning the model on a smaller subset of molecules with desired properties.

Another popular method for molecular generation is reinforcement learning. In theory, reinforcement learning methods allow users to generate a set of molecules with specific properties without explicit examples of molecules that match the reward profile. Successful hit generation requires new molecules with novel combinations of properties, which often do not exist in currently available datasets; therefore, reinforcement learning methods must be able to extrapolate to be truly useful. It is currently unclear how well these models can do so. It is also important that reinforcement learning methods can: (1) generate diverse sets of molecules that explore the chemical space for a complex molecular profile; and (2) exploit chemical space to generate focused molecular libraries. Here, based on previous on-policy reinforcement learning models, we explore scenarios in which optimization is attempted with little or no representation in the training data, and investigate the extent of extrapolation. We manipulate the prevalence of specific properties, measured as a percentage of the entire training dataset, and test the limits of optimization of individual properties. We find that the reinforcement learning models tested can extrapolate beyond the training data, but often produce molecule sets with little diversity. We show that these models are frequently unable to generate molecules that satisfy complex molecular profiles. We go on to demonstrate a curriculum-learning-inspired optimization procedure that enables the generation of specific and diverse sets of molecules that satisfy complex and unseen molecular profiles. We also highlight the limitations of SMILES-based molecular generation tools.

Results and discussion
Deep reinforcement learning molecular generation models are powerful tools for optimizing molecular properties; however, their usefulness is dependent on their input training data. We show how these tools can optimize for specific property values, but only within a property-specific value range. We also show how these methods can generate molecular profiles that are not present in training data and how the representation of training data affects the composition of generated sets of molecules.

By evaluating the performance of deep reinforcement learning molecular generation methods with increasing proportions of training data that match a desired property profile, we show the effects of representation on generated molecules. To overcome the generated library restrictions caused by training data representation, we propose a curriculum-learning-inspired approach—the recurrent iterative optimization procedure (rIOP)—that allows for the optimization of under-represented properties, and also allows users to optimize generated molecules towards complex molecular profiles that are not possible with REINVENT while controlling the diversity of generated libraries. Our work highlights the strengths and limitations of using these tools on single-parameter optimization tasks, which are important to understand before the discussed methods can be used for multiparameter optimization in a drug discovery pipeline.

Control of generated libraries
A standard goal for de novo design deep reinforcement learning tools is to produce novel molecules with controlled distribution of a single property or many properties. Past studies using REINVENT and ReLeaSE have shown that it is possible to bias generated molecules towards specific properties such as the hydrophobicity, melting point or the predicted activity against the DRD2 receptor.

We tested the optimization performance of REINVENT by shifting the reward for cLogP and the number of hydrogen bond acceptors (HBAs). Figure 1a shows the property distribution of generated molecules for a cLogP reward range of between −15 and 20. It shows that it is possible to control the position of agent distributions with the reward range; however, optimization was unsuccessful in extreme cases (for example, a cLogP reward of between −15 and −10) and we observe no change in the agent distribution, that is, the training data distribution is reproduced.

The same behaviour was observed for HBA counts (Fig. 1b). Optimization anywhere in the 0–20 range was possible; however, optimizing the model to generate molecules with 20 HBAs or more was unsuccessful (red distribution). As with cLogP, a distribution similar to the training data was reproduced, which included several molecules with more than 30 HBAs. A complete breakdown of the generated molecular sets, reward ranges and molecule examples can be found in Supplementary Section 2.

We postulate that this failure occurs because, in the extreme case, fewer molecules generated by the prior model return any reward during reinforcement learning. If trained for an infinite time, the model will eventually randomly generate SMILES that will return a positive reward; nevertheless, poor representation can prevent effective optimization. This ineffective optimization then leads to the model repeatedly producing the same SMILES seen in training; thus, the training data is reproduced.

Effect of percentage representation
We have shown that optimization of under-represented properties is sometimes not possible using deep reinforcement learning molecular generation tools. To investigate how widespread this issue is, we tested the ability of the models to generate molecules with properties within, and outside of, the training data. We created several datasets in which the proportion of molecules corresponding to the desired reward profile varied.

Extended Data Table 1 also shows that the composition of each generated library depends on the representation of the desired property in the training data. For example, we can generate sets in which most molecules have a HBA count above eight (the reward threshold), but a higher percentage representation leads to molecules with higher HBA counts with the same reward function. The mean of the 0% representation is nine HBAs, compared with 20 for the 10% experiment. Directional changes across each experiment can be seen for all of the properties.
We postulate that the trends in the generated molecules mirror the training data. For example, for a specific reward property value (for example, HBA = 5) if, as the percentage representation of that property profile increases, the diversity of the training data increases, you will see an increase in the diversity of the generated library. Conversely, if all of the training data examples were very similar, you would observe a reduction in training data diversity and generated library diversity. Therefore, for most properties, we expect that a lower percentage representation would lead to a less diverse generated library.

These results show that it is possible to generate molecular profiles that are not seen during training and that the composition of each generated set depends on the prevalence of molecules in the training data with the desired molecular profile. Building on previous work39, we postulate that higher training data representations would often lead to greater diversity for generated molecules, as the model would often have a more diverse set of examples to learn chemical–structural relationships.

To improve the efficacy of deep reinforcement learning generation methods, we propose a new curriculum-learning-inspired approach, called rIOP. Our method allows deep reinforcement learning generation methods to maximize the diversity of generated molecules for seen and unseen molecular profiles during optimization. It also enables the model to generate molecules that perform more complex optimization tasks where standard methods fail.

Iterative optimization procedure to improve diversity. To demonstrate how rIOP can increase the diversity of simple optimization tasks, we undertook a topological polar surface (TPS) optimization task with single-model rIOP (SrIOP) and REINVENT’s standard implementation. Standard methods38 do generate molecules in this range; however, we expect that we will see an improvement in the diversity of generated molecules with SrIOP. Topological polar surface shift is a simple optimization task; we therefore only sampled the agent of the previous step when training the current prior (see the ‘rIOP’ section of the Methods).

Curriculum learning for generated library control
We have shown that it is possible to generate molecules with unseen molecular profiles in training data (Extended Data Table 1); however, the model’s ability to do this is limited at the extremes of the property distribution (Fig. 1). The composition of each generated set depends on the prevalence of molecules in the training data with the desired molecular profile. Building on previous work39, we postulate that higher training data representations would often lead to greater diversity for generated molecules, as the model would often have a more diverse set of examples to learn chemical–structural relationships.

To improve the efficacy of deep reinforcement learning generation methods, we propose a new curriculum-learning-inspired approach, called rIOP. Our method allows deep reinforcement learning generation methods to maximize the diversity of generated molecules for seen and unseen molecular profiles during optimization. It also enables the model to generate molecules that perform more complex optimization tasks where standard methods fail.

Iterative optimization procedure to improve diversity. To demonstrate how rIOP can increase the diversity of simple optimization tasks, we undertook a topological polar surface (TPS) optimization task with single-model rIOP (SrIOP) and REINVENT’s standard implementation. Standard methods38 do generate molecules in this range; however, we expect that we will see an improvement in the diversity of generated molecules with SrIOP. Topological polar surface shift is a simple optimization task; we therefore only sampled the agent of the previous step when training the current prior (see the ‘rIOP’ section of the Methods).
molecules, eliminating the user’s ability to control the composition of molecules. By contrast, REINVENT is only able to generate diverse libraries (with diversity filters) and diverse (with diversity filters enabled) libraries of molecules sampled from each intermediate (SrIOP 1–3) and final (SrIOP 4) agent trained during rIOP. We show that we can shift the distribution iteratively towards target property values. TPSA reward range for each step was: 100–150 (SrIOP 1); 150–200 (SrIOP 2); 200–250 (SrIOP 3); and 250–300 (SrIOP 4).

Figure 2 shows each step of the SrIOP procedure and the change in property distribution at each stage to match the reward function. To measure the diversity of each generated library, we calculated the number of unique Murcko scaffolds generated. With each SrIOP step, we see a reduction in the number of scaffolds generated. This is expected as we are moving toward the limit of the full TPSA property distribution, where there are fewer ways to achieve these property values. Extended Data Table 2 shows that, in our last step (SrIOP 4), we produce around 18 times more scaffolds with SrIOP (55 scaffolds) than with REINVENT (three scaffolds). Furthermore, SrIOP generates more molecules from the 500 sampled that match the reward profile (494 for SrIOP versus 476 for REINVENT). Extended Data Table 2 also shows the internal diversity of generated sets and the proportion of generated molecules that were present in the training dataset. Links to examples of generated molecules can be found in the Data availability section.

Iterative optimization procedure for diversity control. For de novo design tools to be effective, it should be possible to control the specificity of the generated molecules. We have shown how the representation of a desired profile during training can affect the composition—and hence the specificity—of generated sets of molecules. Our results also show how the use of our new method, SrIOP, can improve the diversity of molecules generated during simple optimization tasks.

To test this, we aimed to generate drug-like molecules (quantitative estimate of drug-likeness (QED) > 0.8) from training with a dataset containing no high-QED molecules (QED < 0.8). Figure 3 shows how it is possible to generate molecules with high QED values with and without diversity filters. Figure 3b also shows that a wider distribution of molecules is produced with a diversity filter enabled. Of the 500 molecules sampled, SrIOP generates 490 molecules that match the reward function, compared with only 317 using REINVENT. Of the 490 molecules, SrIOP generated 22 scaffolds. Standard methods produced more, with 130 scaffolds across those 317 molecules; however, with diversity filters enabled, we observed a considerable increase in SrIOP performance, with the scaffold diversity increasing from 22 to 297 in the 301 generated molecules. We also see a change using REINVENT; the model generates 234 scaffolds across 236 molecules. These results show how, using our SrIOP, we can generate specific (without diversity filters) and diverse (with diversity filters enabled) libraries of molecules. By contrast, REINVENT is only able to generate diverse molecules, eliminating the user’s ability to control the composition of the generated sets. Links to each generated dataset with example molecules is available in the Data availability section.

SrIOP gives the user more control over the specificity of the generated library. If they wish to exploit a property, SrIOP used without diversity filters will return a very narrow selection of molecules that match your reward profile. By contrast, if a diverse library is required, enabling diversity filters with SrIOP will produce one. In this example, we chose a simple optimization task as there are many ways to increase QED for a molecule. We therefore expected that the standard method would perform well. SrIOP still outperforms REINVENT in both specific (no diversity filter) and diverse (diversity filter used) set generation; however, we expect the difference in performance to increase for more complex optimization tasks.

Comparison to other curriculum learning methods
Curriculum learning has long been used as a tool to overcome complex machine learning problems in various applications; however, its use in deep reinforcement learning molecular generation tools is limited. There is one implementation of a similar method by the original authors of REINVENT, which applies a curriculum learning approach to solve complex optimization tasks. We refer to this approach as ReCL.

Iterative optimization procedure for complex tasks. One common use case of deep learning reinforcement learning models is to optimize for molecules similar to a target structure. In such scenarios, there may be few examples of the target structure in training data. To determine how useful SrIOP is in this practical situation, we have used it to generate molecules similar to target structures (Extended Data Fig. 1), with increasing difficulty.

Table 1a shows how, for a simple molecule (Extended Data Fig. 1a), it is possible to generate molecules identical to the target. Both SrIOP and ReCL perform well, with SrIOP generating more molecules with a Tanimoto similarity score of between 0.9 and 1.0 (486 molecules for SrIOP versus 448 for ReCL).

ReCL was less successful (Extended Data Fig. 2b, c) for a more complex molecule (Extended Data Fig. 1b): it could not generate any molecules with a high similarity to the target structure (Tanimoto similarity score of greater than 0.7). For SrIOP almost all (497 of 500 sampled) generated molecules have a high similarity to the target structure.

Another benefit of our method is the ability to control the diversity of the generated library.

Table 1 shows how, without a diversity filter enabled, all 497 molecules sampled have the same scaffold; however, with diversity filters enabled, SrIOP generates 142 scaffolds across 447 molecules. In this example, we highlight the ability of SrIOP to fulfill more complex reward functions where similar curriculum learning methods fail. We show how it can also be used to control the diversity of the generated library through the inclusion of diversity filters.

Limitations of SMILES-based molecular generators
We have investigated the performance of—and proposed new—SMILES-based deep learning molecular generation tools. These tools learn to generate novel one-dimensional SMILES representations of three-dimensional molecular structures (see Methods). SMILES-based tools are popular as they only require simple architectures and can be trained quickly. Yet SMILES do present challenges; namely, they do not detail the three-dimensional structure of a molecule beyond atomic connections, and there are several ways to represent the same molecule. Canonical SMILES provides a standard method for their generation; however, it has been shown that SMILES-based models trained on random SMILES show improved model coverage and a reduction in overfitting.

The lack of structural information and inherent redundancy in SMILES can cause SMILES-based models to struggle to fully understand the chemical and structural relationships between molecules. This
is because the similarity between two SMILES is not well correlated with the similarity between the chemical structures they represent. This limitation of SMILES-based methods and its effects can be seen in our study. For example, when we tried to generate molecules with increasingly complex structures, the performance of the model was heavily dependent on the strings used to represent each substructure. We found that the difference between each string representing a new target structure should be minimized to optimize performance.

We learned that the choice of SMILES directly affects the performance of the models. We aimed to generate molecules with a specific substructure, however, we used several alternate SMILES to represent each identical substructure during reinforcement learning. Approximately one-quarter of all of the substructure generation attempts failed across all molecules (that is, no molecules were generated with the final target substructure; Extended Data Table 3). Yet for each failed attempt, an alternative series of SMILES representing the same molecules was successful. We observed these successful and unsuccessful attempts for near-identical molecules. This highlights how the likelihood of success is more dependent on the choice of SMILES over molecular complexity. Figure 4 shows the total number of SMILES sampled from the final agent that included the desired substructure and the total number of distinct scaffolds present in successful attempts for two structurally similar molecules (A and B). For molecule A, both ReCL and SrIOP fail to generate the final target substructure at least once, and both methods fail in the final step (Supplementary Section 7). For molecule B, we were able to generate substructures regardless of the choice of SMILES, even though the intermediate and final substructures were almost identical compared with those of molecule A. This further highlights the issues caused by SMILES, as you would expect similar performance across both models given the structural similarity of the targets. Instead, the SMILES used has the largest effect on model performance.

The choice of SMILES also has a large effect on the diversity of the generated molecules. For molecule B using SrIOP series 3 (Fig. 4c), we generated more than 250 distinct scaffolds across the 500 molecules sampled, whereas all of the other SMILES series generated between 50 and 150 scaffolds with ReCL and SrIOP. Similar fluctuations in performance were observed for both ReCL and SrIOP across all of the molecules tested (Supplementary Fig. 13).

Molecular representations such as SELFIES and Deep-SMILES attempt to overcome some of the issues of SMILES in machine learning; however, higher dimensional representations that include structural information are likely to be a more powerful way to represent molecules.

**Conclusion**

We investigated how well deep reinforcement learning molecular design methods can search beyond the chemical space represented in training data, and the effects of the composition of the training data on generated molecular sets. The results show that it is possible to control the distribution of molecules generated by altering the reward function; however, we demonstrate how standard methods (REINVENT) can fail towards the edge of the training data distribution. We found that it is possible to generate molecules with properties that are not present in the training data; nevertheless, we showed that the representation of the desired molecular profile affects the distribution of the generated molecular library. We highlight the lack of control standard methods provide in terms of composition (particularly the diversity) of generated molecules and the limitations of SMILES-based molecular generation methods. To overcome some of these issues, we propose a new curriculum learning approach, rIOP, to help boost the diversity of generated molecules when few or no examples of the desired molecular profile are present in the training data. Using this method, we generate structures similar to a series of unseen target structures and outperform other curriculum learning approaches.
We describe a SrIOP and double-model rIOP (DrIOP), which enable a user to control the diversity of generated molecules for simple and complex optimization tasks. Using several SMILES representations of the same molecule when generating target structures, we show how the choice of SMILES directly affects the success and performance of SMILES-based tools. Our method, like any method based on SMILES or other one-dimensional representations, will therefore be hindered by the lack of direct structural information.

Methods

We assessed the performance of a popular on-policy SMILES generation model, REINVENT, to determine the limits of deep reinforcement learning tools in molecular design. Like earlier reinforcement learning molecular generation tools, REINVENT involves a two-step process. The first is to train a prior RNN to generate SMILES through supervised learning. This model is trained to correctly predict the next character of a SMILES string given a starting token or an incomplete string. The second is to fine-tune the prior model, producing an agent model that is able to generate a focused library through a reward–feedback loop. During this second step, the model learns a policy that maximizes the likelihood of generating a molecule with a favourable reward (Fig. 5). See ref. 40 for full details on the models.

For all experiments described in the following, the model was trained on subsets of 1.5 million drug-like molecules from ChEMBL. After the model was fully trained, we sampled 500 molecules unless otherwise stated. We chose to generate 500 molecules as this provided a large enough sample from which we could draw clear conclusions about the distribution of generated molecules.

Property characterization

Hydrogen bond acceptors and donors, molecular weight, TPSA and cLogP were calculated using the chemical descriptor module from RDKit synthetic accessibility, QED and Tanimoto similarity scores were also calculated using RDKit.

Reinforcement learning

A suitable reward function must be provided to successfully optimize for a property. For simplicity, we used the same step reward function throughout our study (see Supplementary Section 1 for examples). Any invalid SMILES did not return a reward, and all valid SMILES that met the reward criteria returned a reward of one.

Optimization success

We initially determined the success of optimization using the proportion of molecules in the generated library that fell within the reward range; however, after several properties were otherwise stated. We chose to generate 500 molecules as this provided a large enough sample from which we could draw clear conclusions about the distribution of generated molecules.

Property characterization

Hydrogen bond acceptors and donors, molecular weight, TPSA and cLogP were calculated using the chemical descriptor module from RDKit synthetic accessibility, QED and Tanimoto similarity scores were also calculated using RDKit.

Reinforcement learning

A suitable reward function must be provided to successfully optimize for a property. For simplicity, we used the same step reward function throughout our study (see Supplementary Section 1 for examples). Any invalid SMILES did not return a reward, and all valid SMILES that met the reward criteria returned a reward of one.

Optimization success

We initially determined the success of optimization using the proportion of molecules in the generated library that fell within the reward range; however, after several properties were otherwise stated. We chose to generate 500 molecules as this provided a large enough sample from which we could draw clear conclusions about the distribution of generated molecules.

Property characterization

Hydrogen bond acceptors and donors, molecular weight, TPSA and cLogP were calculated using the chemical descriptor module from RDKit synthetic accessibility, QED and Tanimoto similarity scores were also calculated using RDKit.

Reinforcement learning

A suitable reward function must be provided to successfully optimize for a property. For simplicity, we used the same step reward function throughout our study (see Supplementary Section 1 for examples). Any invalid SMILES did not return a reward, and all valid SMILES that met the reward criteria returned a reward of one.

Optimization success

We initially determined the success of optimization using the proportion of molecules in the generated library that fell within the reward range; however, after several properties were otherwise stated. We chose to generate 500 molecules as this provided a large enough sample from which we could draw clear conclusions about the distribution of generated molecules.
tested at increasing representations, it became clear that the difference between the proportion of optimized molecules in successful and unsuccessful optimization attempts was large enough so that a success threshold was not appropriate. Instead, optimization attempts that showed an increase in the proportion of optimized molecules were deemed successful. This was possible as all of the unsuccessful attempts resulted in zero optimized molecules.

Controlling generated libraries. To determine the degree to which the property distribution of the molecules generated by the agent (agent distribution) can be controlled, we set REINVENT the task of shifting the distribution of cLogP or HBA counts across their respective ranges. These properties or the property ranges tested may not be the most important in a drug discovery context; however, these experiments allow us to assess optimization performance. If it is possible to control the distribution of generated molecular sets, we expect to observe changes in the composition of these sets as the reward range changes. During reinforcement learning, all valid SMILES that match the reward function are given a reward of one. All other molecules, valid or not, receive zero reward.

Effects of percentage representation. In our study, our aim was to determine the effect that training data representation had on the generated molecules. To do this, we prepared several training datasets in which the proportion of molecules that matched the desired reward profile varied. We chose reward profiles (Supplementary Section 1) at the upper end of the full ChEMBL training data distribution such that at least 10% of the training data matched the reward function. Once the reward range was calculated, all molecules that matched the reward profile were removed from the full training dataset. Then smaller random samples equal in size to 0%, 2%, 5%, 7% and 10% of the entire training dataset were put back and used to train the model from scratch.

rIOP
We propose a novel, curriculum-learning-inspired rIOP. Curriculum learning is a method used to teach models how to complete difficult tasks through the gradual introduction of more complex examples during training\(^{48}\). For single-step optimization attempts, it is common for reinforcement learning methods to exploit molecular motifs found to return positive rewards, leading to generated sets with low diversity (specialization). We expect that the greater the difference between the reward profile and training data, the more prevalent this behaviour is. By splitting the optimization task into a series of smaller tasks, we reduced the difference between the molecules generated by the prior and the desired reward profile at each step, thus reducing the likelihood of early specialization. Repeating a prior–agent training loop with a series of small changes in the reward profile, we encouraged each agent model to shift its property distribution toward the final, desired distribution. The resulting agent was then used as the prior in the next step. Splitting the final optimization task into a series of increasingly complex subtasks allowed the model to satisfy increasingly difficult molecular profiles that directed it towards the final goal.

We demonstrate the use of rIOP implementations, which can be used in on-policy reinforcement learning training regimes. The first, SrIOP, only samples from the previous agent when training the current model. The second, DrIOP, samples from the previous two models. Unless stated otherwise, for DrIOP we sampled the current agent once every five times the previous agent was sampled.

Diversity filters. To control diversity, where appropriate, we incorporated the diversity filters described by Blaschke et al.\(^{50}\). With diversity filters enabled, the model will only give a positive reward for the first \(n\) molecules that satisfy the reward function for a given scaffold. Once \(n\) molecules that match the reward profile have been generated, molecules with this scaffold are no longer rewarded. This prevents the model from entering a local optimization minimum by producing many molecules with the same scaffold and small structural differences to satisfy the reward function.

rIOP for diversity control. To demonstrate how rIOP can be used to control the diversity of generated molecules, we conducted two experiments. First, we generated molecules with a reward for TPSA between 250 and 300 using SrIOP and REINVENT’s standard implementation. Second, we created a set of molecules in which our aim was to maximize QED. We used no molecules with QED greater than 0.8 during training, then iteratively increased the QED reward profile at each step. Diversity filters were also used to further improve the diversity of generated molecules.

rIOP for complex optimization tasks. To showcase rIOP’s ability to complete complex optimization tasks, we generated molecules similar to target structures with no relevant examples in the training data. We removed all molecules with a Tanimoto similarity score of greater than 0.4 to each target from the training data and then increased the Tanimoto similarity reward threshold by 0.1 at each step.

Limitations of SMILES. To examine the limitations of SMILES-based molecular generators, we attempted to generate molecules that included a target substructure using multiple different SMILES strings to represent the intermediate substructures. We used ReCL and SrIOP to generate molecules with a series of increasingly complex substructures. For each molecule, we enumerated five alternate SMILES for each target intermediate substructure. The intermediate SMILES across each series were kept constant.

Reporting summary
Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.
Data availability
The trained generative model used in some of our work is already published by Patruno and colleagues, and is available at https://github.com/m-mokaya/RIOP/blob/main/models/random.prior.new. The raw data needed to reproduce the experiments in this work are provided at https://github.com/m-mokaya/RIOP/blob/main/data. Our training data (also available in above links) are from ChEMBL: https://www.ebi.ac.uk/chembl/.

Code availability
The code used in this study is available at https://github.com/m-mokaya/RIOP. Example notebooks for each experiment are available at https://github.com/m-mokaya/RIOP/tree/main/notebooks and https://doi.org/10.5281/zenodo.7406695.

References
1. Schneider, P. & Schneider, G. De novo design at the edge of chaos. J. Med. Chem. 59, 4077–4086 (2016).
2. Waring, M. J. et al. An analysis of the attrition of drug candidates from four major pharmaceutical companies. Nat. Rev. Drug Discov. 14, 475–486 (2015).
3. Hay, M., Thomas, D. W., Craighead, J. L., Economides, C. & Rosenthal, J. Clinical development success rates for investigational drugs. Nat. Biotechnol. 32, 40–51 (2014).
4. Bunnavge, M. E. Getting pharmaceutical R&D back on target. Nat. Chem. Biol. 7, 335–339 (2011).
5. Hughes, J., Rees, S., Kalindjian, S. & Philpott, K. Principles of early drug discovery: principles of early drug discovery. Br. J. Pharmacol. 162, 1239–1249 (2011).
6. Bohacek, R. S., McMartin, C. & Guida, W. C. The art and practice of structure-based drug design: a molecular modeling perspective. Med. Res. Rev. 16, 3–50 (1996).
7. Kim, S. et al. PubMed chemistry substance and compound databases. Nucleic Acids Res. 44, D1202–D1213 (2016).
8. Romano, J. D. & Tatonetti, N. P. Informatics and computational methods in natural product drug discovery: a review and perspectives. Front. Genet. 10, 368 (2019).
9. Lin, X., Li, X. & Lin, X. A review on applications of computational methods in drug screening and design. Molecules 25, 1375 (2020).
10. Besnard, J. et al. Automated design of ligands to polypharmacological profiles. Nature 492, 215–220 (2012).
11. Gómez-Bombarelli, R. et al. Automatic chemical design using a data-driven continuous representation of molecules. ACS Cent. Sci. 4, 268–276 (2018).
12. Stumpfe, D. & Bajorath, J. Similarity searching. WIREs Comput. Mol. Sci. 1, 260–282 (2011).
13. Horvath, D. A virtual screening approach applied to the search for trypanothione reductase inhibitors. J. Med. Chem. 40, 2412–2423 (1997).
14. Surabhi, S. & Singh, B. K. Computer-aided drug design: an overview. J. Drug Deliv. Ther. 8, 504–509 (2018).
15. Segler, M. H. S., Kogej, T., Tyczka, C. & Waller, M. P. Generating focused molecule libraries for drug discovery with recurrent neural networks. ACS Cent. Sci. 4, 120–131 (2018).
16. Mauser, H. & Stahl, M. Chemical fragment spaces for de novo design. J. Chem. Inf. Model. 47, 318–324 (2007).
17. Hartenfeller, M., Proschak, E., Schüller, A. & Schneider, G. Concept of combinatorial de novo design of drug-like molecules by particle swarm optimization. Chem. Biol. Drug Des. 72, 16–26 (2008).
18. Dey, F. & Cafisch, A. Fragment-based de novo ligand design by multiojective evolutionary optimization. J. Chem. Inf. Model. 48, 679–690 (2008).
19. Elton, D. C., Boukouvalas, Z., Fuge, M. D. & Chung, P. W. Deep learning for molecular design—a review of the state of the art. Mol. Syst. Des. Eng. 4, 828–849 (2019).
20. Baldi, P. Autoencoders, unsupervised learning, and deep architectures. In Proc. ICML Workshop on Unsupervised and Transfer Learning (eds. Guyon, I. et al.) Vol. 27, 37–49 (PMLR, 2012).
21. Jin, W., Barzilay, R. & Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. Preprint at arXiv https://doi.org/10.48550/arXiv.1802.04364 (2018).
22. Weining, D. SMILES: a chemical language and information system: 1. Introduction to methodology and encoding rules. J. Chem. Inf. Comput. Sci. 28, 31–56 (1988).
23. Lim, J., Ryu, S., Kim, J. W. & Kim, W. Y. Molecular generative model based on conditional variational autoencoder for de novo molecular design. J. Cheminform. 10, 31 (2018).
24. Goodfellow, I. J. et al. Generative adversarial networks. Preprint at arXiv https://doi.org/10.48550/arXiv.1406.2664 (2014).
25. Putin, E. et al. Reinforced adversarial neural computer for de novo molecular design. J. Chem. Inf. Model. 58, 1194–1204 (2018).
26. Guimaraes, G. L., Sanchez-Lengeling, B., Outeiral, C., Farias, P. L. C. & Aspuru-Guzik, A. Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. Preprint at arXiv https://doi.org/10.48550/arXiv.1705.10843 (2017).
27. Vaswani, A. et al. Attention is all you need. Preprint at arXiv https://doi.org/10.48550/arXiv.1706.03762 (2017).
28. Grechishnikova, D. Transformer neural network for protein-specific de novo drug generation as a machine translation problem. Sci. Rep. 11, 321 (2021).
29. Bagal, V., Aggarwal, R., Vinod, P. K. & Priyakumar, U. D. MolGPT: molecular generation using a transformer-decoder model. J. Chem. Inf. Model. 62, 2064–2076 (2022).
30. Zheng, S. et al. Deep scaffold hopping with multimodal transformer neural networks. J. Cheminform. 13, 87 (2021).
31. He, J. et al. Transformer neural network for structure constrained molecular optimization. Preprint at ChemRxiv https://doi.org/10.26434/chemrxiv.14416133.v1 (2021).
32. Goldberg, Y. A Primer on neural network models for natural language Processing. J. Artif. Intell. Res. 57, 345–420 (2016).
33. Kotsias, P.-C. et al. Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. Nat. Mach. Intell. 2, 254–265 (2020).
34. Bjerrum, E. J. & Threlfall, R. Molecular generation with recurrent neural networks (RNNs). Preprint at https://arxiv.org/abs/1705.04612 (2017).
35. Arús-Pous, J. et al. Exploring the GDB-13 chemical space using deep generative models. J. Cheminform. 11, 20 (2019).
36. Arús-Pous, J. et al. Randomized SMILES strings improve the quality of molecular generative models. J. Cheminform. 11, 71 (2019).
37. Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. 8, 229–256 (1992).
38. Popova, M., Isayev, O. & Tropsha, A. Deep reinforcement learning for de novo drug design. Sci. Adv. 4, eaaq7885 (2018).
39. Guo, J. et al. Improving de novo molecular design with curriculum learning. Nat. Mach. Intell. 4, 555–563 (2022).
40. Olivecrona, M., Blaschke, T., Engkvist, O. & Chen, H. Molecular de-novo design through deep reinforcement learning. J. Cheminform. 9, 48 (2017).
41. Soviany, P., Ionescu, R. T., Rota, P. & Sebe, N. Curriculum learning: a survey. Preprint at arXiv https://doi.org/10.48550/arXiv.2101.10382 (2021).
42. Krenn, M., Häse, F., Nigam, A., Friederich, P. & Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): a 100% robust molecular string representation. Mach. Learn. Sci. Technol. 1, 045024 (2020).
43. O’Boyle, N. & Dalke, A. DeepSMILES: an adaptation of SMILES for use in machine-learning of chemical structures. Preprint at ChemRxiv https://doi.org/10.26434/chemrxiv.7097960.v1 (2018).
44. Gaulton, A. et al. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **40**, D1100–D1107 (2012).

45. Landrum, G. RDKit: Open-Source Cheminformatics (RDKit, 2006).

46. Ertl, P. & Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminform.* **1**, 8 (2009).

47. Bemis, G. W. & Murcko, M. A. The properties of known drugs. 1. Molecular frameworks. *J. Med. Chem.* **39**, 2887–2893 (1996).

48. Polykovskiy, D. et al. Molecular Sets (MOSES): a benchmarking platform for molecular generation models. *Front. Pharmacol.* **11**, 565644 (2020).

49. Elman, J. L. Learning and development in neural networks: the importance of starting small. *Cognition* **48**, 71–99 (1993).

50. Blaschke, T. et al. REINVENT 2.0: an AI tool for de novo drug design. *J. Chem. Inf. Model.* **60**, 5918–5922 (2020).

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research council (grant no. EP/S024093/1) and Exscientia. For the purpose of Open access, we have applied a CC BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

Author contributions

M.M. developed the code. M.M., F.I., A.R.B. and C.M.D. designed the experiments. M.M. performed the experiments and analyses. M.M. wrote the manuscript and all of the other authors revised it. A.R.B. and C.M.D. supervised the work. All of the authors read and approved of the final version of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Extended data is available for this paper at https://doi.org/10.1038/s42256-023-00636-2.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-023-00636-2.

Correspondence and requests for materials should be addressed to Charlotte M. Deane.

Peer review information Nature Machine Intelligence thanks the anonymous reviewers for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

© The Author(s), under exclusive licence to Springer Nature Limited 2023
Extended Data Fig. 1 | Simple and complex target substructures. Diagrams of (a) simple and (b) complex target structures used for Tanimoto similarity and substructure generation experiments.
Extended Data Fig. 2 | Similarity distributions of generated sets in simple and complex optimization experiments. Generating molecules similar to (a) a simple target molecule, (b) a complex molecule without diversity filters and (c) a complex molecule with diversity filters. SrIOP (blue), ReCL (orange), training (dotted) and Prior (green) distributions for each optimization task. Both methods can produce entire datasets that match the simple target structure, however, only SrIOP is able to generate molecules similar to the complex target structure.
Extended Data Table 1 | Mean property values for molecules generated from four property optimization tasks using increasing training data percentage representations

| Property (Reward min) | Percentage representation in training |
|-----------------------|--------------------------------------|
|                       | 0     | 2     | 5     | 7     | 10    |
| HBA (8)               | 9     | 8     | 13    | 16    | 19    |
| MW (527)              | 624   | 715   | 748   | 735   | 787   |
| QED (0.83)            | 0.901 | 0.908 | 0.905 | 0.902 | 0.879 |
| TPSA (122)            | 175   | 190   | 206   | 201   | 207   |

Mean property values for molecules generated from four property optimization tasks using increasing training data percentage representations.
### Extended Data Table 2 | Comparison of generated set composition of each agent trained during SrIOP and REINVENT for TPSA optimization

| Run      | # In reward Range | # Scaffolds | Internal Diversity | % In Training data |
|----------|--------------------|-------------|--------------------|--------------------|
| SrIOP 1  | 437                | 352         | 0.762              | 37.5               |
| SrIOP 2  | 468                | 198         | 0.609              | 28.0               |
| SrIOP 3  | 499                | 20          | 0.395              | 0                  |
| SrIOP 4  | 494                | 55          | 0.450              | 0                  |
| REINVENT | 476                | 3           | -                  | 0                  |

Comparison of generated set composition of each agent trained during SrIOP and REINVENT for TPSA optimization.
| Method | Number of attempts | Number of failed attempts | Percentage of failed attempts / % |
|--------|--------------------|---------------------------|----------------------------------|
| ReCL   | 40                 | 8                         | 20                               |
| SrIOP  | 40                 | 12                        | 30                               |
| Total  | 80                 | 20                        | 25                               |

Proportion of failed substructure generation attempts using different SMILES across all molecules tested using ReCL and SrIOP.
Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

- n/a Confirmed
- X The exact sample size (n) for each experimental group/condition, given as a discrete number and unit of measurement
- X A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
- X The statistical test(s) used AND whether they are one- or two-sided
- Only common tests should be described solely by name; describe more complex techniques in the Methods section.
- X A description of all covariates tested
- X A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
- X A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
- X For null hypothesis testing, the test statistic (e.g. F, t, r) with confidence intervals, effect sizes, degrees of freedom and P value noted
- Give P values as exact values whenever possible.
- X For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
- X For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
- X Estimates of effect sizes (e.g. Cohen’s d, Pearson’s r), indicating how they were calculated.

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about: availability of computer code

| Data collection | Our work is built on open source, already published software tools [https://www.nature.com/articles/s42256-022-00494-4]. All our code and results are available at: https://doi.org/10.5281/zenodo.7406695. |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data analysis   | Our software and analysis is written in Python. Our work is built on open source, already published software tools [https://www.nature.com/articles/s42256-022-00494-4]. We have used these tools to make comparisons to our own novel methods. Our novel methods are open-source and available at https://github.com/m-mokaya/RIOP.git or https://doi.org/10.5281/zenodo.7406695. |

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about: availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy.

The trained generative model used in some of our work is already published by Guo et al [Guo, J. et al. Improving de novo molecular design with curriculum...].
Human research participants

Policy information about: studies involving human research participants and Sex and Gender in Research.

| Reporting on sex and gender | n/a |
|----------------------------|-----|
| Population characteristics | n/a |
| Recruitment                | n/a |
| Ethics oversight           | n/a |

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

Life sciences  Behavioural & social sciences  Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

| Sample size | Once trained, we sampled 500 molecules from our models. No sample calculation was done. We chose to sample 500 molecules as this provided a large enough sample from which we could draw clear conclusions about the properties of generated sets of molecules with reasonable computational expense. |
|-------------|---------------------------------------------------------------------------------------------------------------|
| Data exclusions | No data was excluded. |
| Replication | To ensure reproducibility, we repeated specific experiments (Effects of Percentage representation - section 3.2) three times and reported the averaged results with deviation reported in the supplementary information. For other experiments (Curriculum Learning for Generated Library Control (3.3), Comparison to other curriculum learning methods (3.4) and Limitations of SMILES-based Molecular Generators (3.5)), the agent models were sampled multiple times and consistently produced molecules with the same properties. The experiments can be replicated using the datasets and code provided (see data availability statement). |
| Randomization | Randomization is not relevant to our study because there was no subjective allocation of samples to experimental groups. |
| Blinding | Blinding was not relevant to the study. Our work investigated the performance of already published and novel Deep Reinforcement Learning molecular generation tools. Investigator bias could not have any effect on the outcome of the study. |

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

### Materials & experimental systems

| n/a | Involved in the study |
|-----|-----------------------|
| ☑ | Antibodies |
| ☑ | Eukaryotic cell lines |
| ☑ | Palaeontology and archaeology |
| ☑ | Animals and other organisms |
| ☑ | Clinical data |
| ☑ | Dual use research of concern |

### Methods

| n/a | Involved in the study |
|-----|-----------------------|
| ☑ | ChiP-seq |
| ☑ | Flow cytometry |
| ☑ | MRI-based neuroimaging |