ARTICLES
https://doi.org/10.1038/s42256-020-0174-5
nature machine intelligence

Check for updates

Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks

Panagiotis-Christos Kotsias[1], Josep Arús-Pous[1,2], Hongming Chen[1,3], Ola Engkvist[1], Christian Tyrchan[4] and Esben Jannik Bjerrum[1]✉

Deep learning has acquired considerable momentum over the past couple of years in the domain of de novo drug design. Here, we propose a simple approach to the task of focused molecular generation for drug design purposes by constructing a conditional recurrent neural network (cRNN). We aggregate selected molecular descriptors and transform them into the initial memory state of the network before starting the generation of alphanumeric strings that describe molecules. We thus tackle the inverse design problem directly, as the cRNNs may generate molecules near the specified conditions. Moreover, we exemplify a novel way of assessing the focus of the conditional output of such a model using negative log-likelihood plots. The output is more focused than traditional unbiased RNNs, yet less focused than autoencoders, thus representing a novel method with intermediate output specificity between well-established methods. Conceptually, our architecture shows promise for the generalized problem of steering of sequential data generation with recurrent neural networks.

The disruptive impact deep generative models have delivered over the past couple of years has found applications in various aspects of content creation. Neural networks have proven potent in generating news headlines[1], synthesizing music[2] or poems[3] and drawing realistic paintings[4]. In the life sciences, they have driven innovation in tasks such as bioactivity and synthesis prediction[5], segmentation of biological images[6] and de novo design of molecules[7]. In the latter domain, in particular, deep learning serves as a prominent stakeholder by offering the capability to direct the generative process towards chemical regions of interest[8–10]. A challenging task deep networks are trying to address is inverse molecular design[11]—the generation of molecular structures that meet desired conditions, such as specific physicochemical properties or properties predicted by quantitative structure–activity relationship (QSAR) models.

The 'simplified molecular-input line-entry system' (SMILES)[12] is a popular choice[13] to represent molecules when using recurrent neural networks (RNNs). The alphanumeric nature of SMILES strings makes them compatible with state-of-the-art natural language-processing algorithms, such as RNNs, performing sequence modelling and generation. In particular, RNNs are a widely accepted approach to the task of sequence modelling because of their ability to memorize previously predicted characters of a partially finished SMILES string and incorporate them into their inference while building up the complete sequence[14].

Unbiased RNN generative models trained on a relatively small number of SMILES strings have been shown to be able to cover a much larger chemical space[15]. Moreover, augmentation of a dataset using SMILES with randomized atom order has demonstrated state-of-the-art performance with respect to the uniformity and completeness of the coverage of chemical regions, compared to simply using their canonical variants[16]. After learning the general rules of the chemical space, for example atom type, bond type and size of molecules, the prior network can be further specialized using smaller datasets in a transfer learning fashion[17] or using reinforcement learning[18–20].

More complicated architectures, such as autoencoders[21], which include two jointly trained neural networks responsible for converting the input to and back from a latent representation, have been extensively benchmarked[22,23]. The quality of the latent space of an autoencoder was also proven to benefit from the usage of randomized SMILES strings[24–26]. Moreover, the latent space representation of a molecule can be used in optimizing QSAR endpoints using generative adversarial networks (GANs)[27], Bayesian optimization[21] or particle swarm optimization[28]. The combination of a heteroencoder[25], trained on pairs of randomized SMILES strings of the same molecule, with a GAN[29] has further demonstrated automatic navigation towards properties of interest.

Alternatively, learning to precondition structure generation eliminates the need for optimization loops. One approach demonstrated this capability by concatenating SMILES strings with the properties of interest as input to a variational autoencoder[30]. Molecular graphs[31] have also been used in pairs along with the desired change in properties as conditions on which to train a variational autoencoder. Latent representations that are generated by a GAN architecture may also be exploited as input conditions for decoding neural networks[29].

In this work, we demonstrate that molecule-side information, such as molecular descriptors, can be incorporated into the RNN-based generative process. We construct conditional recurrent neural networks (cRNNs) by setting the internal states of long short-term memory cells (LSTMs[32]) after some input conditions.

[1]Hit Discovery, Discovery Sciences, BioPharmaceuticals R&D, AstraZeneca, Gothenburg, Sweden. [2]Department of Chemistry and Biochemistry, University of Bern, Bern, Switzerland. [3]Centre of Chemistry and Chemical Biology, Guangzhou Regenerative Medicine and Health Guangdong Laboratory, Guangzhou, China. [4]Medicinal Chemistry, Research and Early Development, Respiratory & Immunology, BioPharmaceuticals R&D, AstraZeneca, Gothenburg, Sweden. ✉e-mail: esben.bjerrum@astrazeneca.com

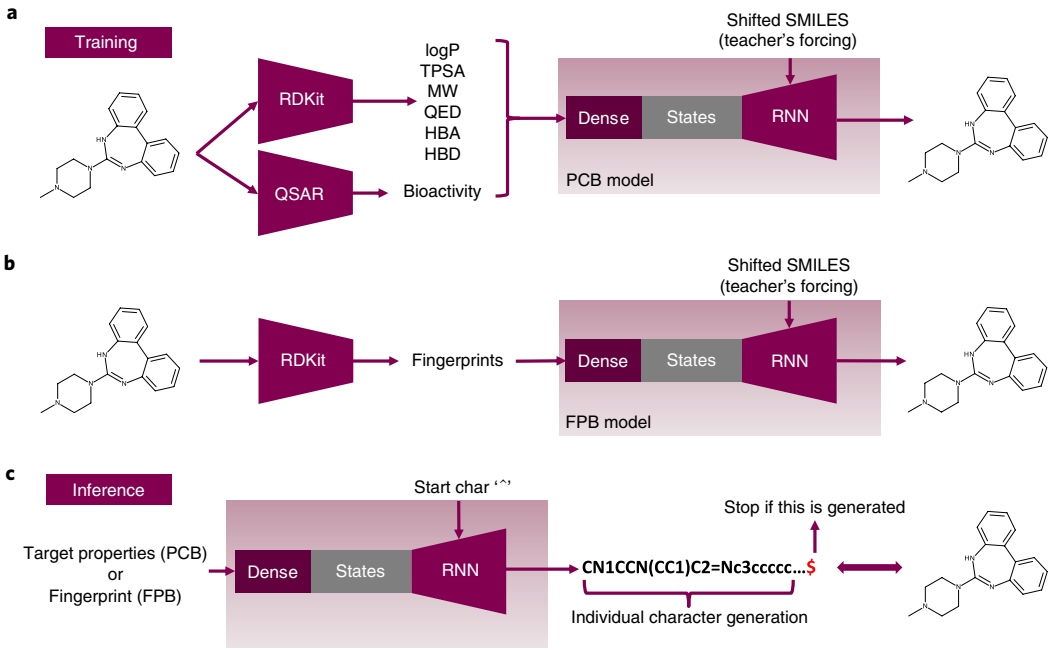


Fig. 1 | cRNN models based on different conditions. a, The physchem-based (PCB) model accepts six scalar properties: the Wildman–Crippen partition coefficient (logP), topological polar surface area (TPSA), molecular weight (MW), drug-likeness (QED), number of hydrogen-bond acceptors (HBA) and donors (HBD) as calculated by the RDKit Python library, concatenated with the probabilistic bioactivity prediction of the QSAR model. **b**, The fingerprint-based (FPB) model accepts a 2,048 bit Morgan fingerprint vector calculated by RDKit. Both models are trained on randomized SMILES strings as targets. **c**, Model inference is biased by the conditional seed and triggered by the starting character '^'. Inference stops when '$' is generated.

The architecture is related to, but conceptually simpler than, a conditional autoencoder, as we only utilize an RNN-based decoder part. The generation is conditioned with properties calculated directly from the molecular structure or QSAR models, so the encoder part is no longer needed. The conditional seed successfully steers the focus of the RNN towards a particular subset of the chemical domain, such as bioactive compounds, with respect to a specific protein target. Our approach complements existing state-of-the-art conditional generative models such as conditional variational autoencoders, reinforcement learning and so on, and may be used for populating of specialized molecular libraries. We also demonstrate a novel way of assessing the focus of a probabilistic sequence generator using negative log-likelihood (NLL) plots. Owing to the nature of the problem for which our method is showcased, it may also be generalized to other applications where conditioning of sequential data is needed, such as natural language generation or time-series forecasting.

Results and discussion

Resulting datasets. The filtering process described in the Methods resulted in the sizes of datasets shown in Table 1. The QSAR support vector classification model with parameters $C=5.53$ and $\gamma=0.022$ was selected as the one with the highest F1-score (0.92) towards the DRD2 validation set. This model was used to label all compounds in the ChEMBL dataset, leading to 2.3% of the ChEMBL compounds being classified with a probability greater than 50% of being active against the DRD2 receptor (Table 1). As shown in Extended Data Fig. 1, the property distributions of the two datasets largely follow each other, except that the QED score of the DRD2 dataset is shifted towards higher values because those molecules are expected to be a priori drug-like.

NLL distributions of datasets. The NLL of sampling molecules from the ChEMBL25 dataset and from the known active compounds of the DRD2 dataset was calculated for all different models.

Table 1 | Size and percentage of active compounds per dataset

Dataset	Total samples	Active %
DRD2_TRAIN	71,512	6.7[a]
DRD2_VALID	17,800	6.3[a]
DRD2_TEST	17,817	6.4[a]
ChEMBL25_TRAIN	1,347,173	2.3[b]
ChEMBL25_TEST	149,679	2.3[b]

[a]Known active compounds. [b]Predicted active compounds ($P \geq 0.5$) by the QSAR model.

In total, 1,000 molecules were randomly selected from each of the ChEMBL25 train and ChEMBL25 test datasets and the active compounds of the DRD2 train and DRD2 test datasets. After performing 10,000 randomization operations per molecule, all unique random strings per molecule were collected. Then, by using equation (5), the NLL of sampling a molecule was approximated as the cumulative likelihood over all its uniquely derived random representations, given each model.

Figure 2 shows all different NLL distributions using the smoothened estimate of the density function of the underlying histograms. As showcased in Supplementary section 'Likelihood of sampling of canonical SMILES', selecting the canonical form to calculate the NLL plots of Fig. 2 is not expected to alter the quantitative conclusions that can be drawn from them.

The FPB model results in the sharpest distribution of NLL values with the lowest mean and variance compared to the other three models with respect to all datasets. Similarly, the PCB model shows the second lowest NLL mean value per dataset. The arrangement of the plots is as expected, because the amount of chemical information in the 2,048 bits of a Morgan fingerprint exceeds the information that is contained within the seven scalar descriptors used in the PCB model, especially from a structural point

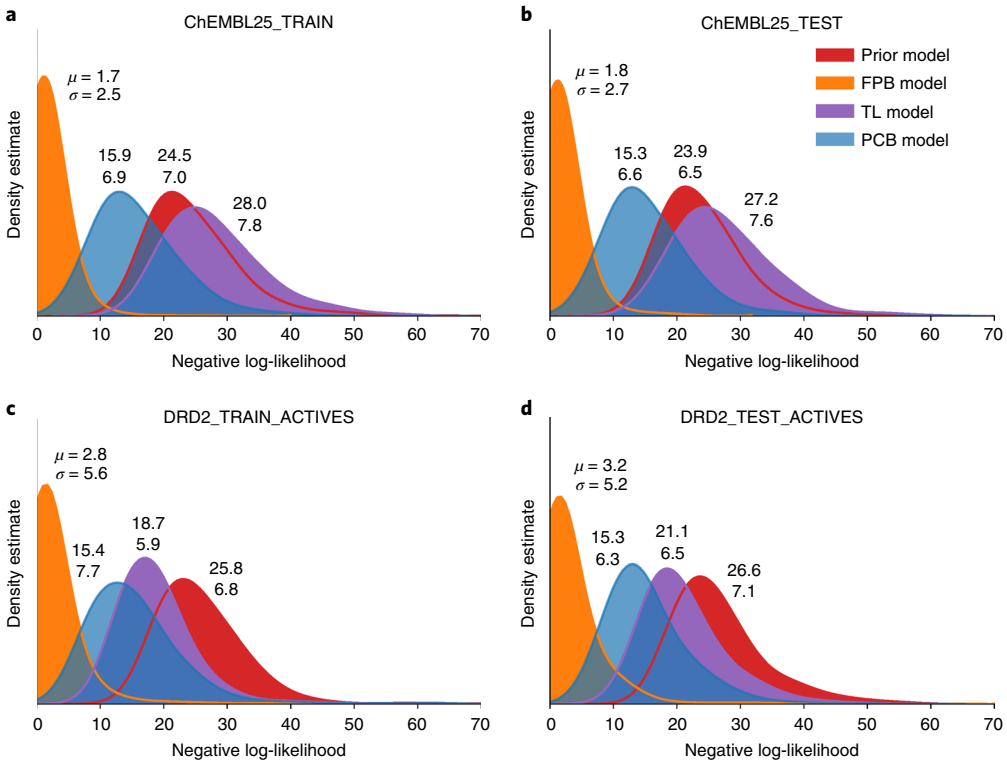


Fig. 2 | NLL of sampling known molecules. a–d, The NLL values of 1,000 molecules randomly drawn from each of four datasets (ChEMBL25_TRAIN (**a**), ChEMBL25_TEST (**b**), DRD2_TRAIN_ACTIVES (**c**) and DRD2_TEST_ACTIVES (**d**)) were calculated using the PCB, FPB, transfer learning (TL) and prior models. For each molecule, 10,000 randomizations were performed, and its NLL was approximated as the cumulative NLL of sampling all unique random strings obtained for that molecule out of the 10,000 randomization attempts. The plots show the estimate of the density of the underlying NLL histograms using the kdeplot function of the seaborn Python library with a bandwidth value of 3.0. The mean and standard deviation of the sampled data distributions are annotated. The ChEMBL25 sets consist of both predicted active and inactive compounds, whereas only the known actives were selected from the DRD2 sets for this test. The graphs are truncated at a maximum NLL value of 70.

of view. The graphs of the conditional models show a slight shift towards higher values for the DRD2 datasets due to the uncertainty that is inherent to unseen data. Nevertheless, both conditional models have a lower mean NLL—and thus a higher probability—of sampling a SMILES string representation corresponding to the molecule from which the conditions originated, compared to the prior network both before and after being trained with transfer learning. The transfer learning model curve exchanges its relative position with the prior model curve between the two datasets because the focus of the model trained with transfer learning has been shifted away from the majority of molecules in ChEMBL and thus it is more difficult to sample their respective SMILES strings.

Ideally, all models should be able to sample the intended chemical space uniformly and this would be expressed by zero variance of the NLL distribution, which should approximate a Dirac distribution. Under such ideal conditions, it would be possible to estimate the size of the output space by simply inverting the (constant) probability of sampling any molecule; for example, a probability of 0.01 would mean that, in total, 100 molecules could be sampled. As an example, a sharp NLL distribution around a value of 10 would imply a uniform probability distribution at a value of 4.54×10^{-5} or an equiprobable output space of 22,000 unique molecules. Similarly, NLL values of 20 and 30 would point to output domains of ~10^8 and 10^{13} molecules, respectively. Even though the distributions of Fig. 2 are far from Dirac distributions, a comparison of the distributions may serve as a qualitative insight into the relative change in the order of magnitude of their output space.

Additionally, the position of the distributions can be interpreted in two ways. First, the closer to zero the NLL distribution moves, the more deterministic the output of the model gets. This can be due to either limited generalizability of the model or a more detailed description of the target, such as in the case of a conditional network. Second, differences in NLL distributions between train and test sets can be a sign of overfitting or mode collapse[15]. This seems to be the case with the transfer learning model, which exhibits a distribution with a lower mean NLL towards the active compounds in the DRD2 train dataset compared to the unseen active ones in the DRD2 test set. In contrast, the NLL distributions of sampling all four datasets with either of the conditional networks, regardless of the dataset, are on par, which makes overfitting a less likely cause. Here, the similar distributions, regardless of the dataset, demonstrate that the conditional models can generate valid SMILES that correspond to both active and inactive compounds with equal ease, given that the states are set accordingly.

Sampling of active molecules. The structures shown in Fig. 3 were suggested by the two conditional networks using known active compounds from the DRD2 test set as conditional seeds, which were selected randomly (shown in the centre). The SMILES strings corresponding to the exemplified molecules in the dashed circle were generated by the FPB model, whereas those outside it were generated by the PCB model. A batch of 256 SMILES strings was sampled per model per seed, and all molecules displayed were filtered to have a QED score greater than 0.8 and were predicted to be active by the QSAR model with a probability greater than 0.8, given that

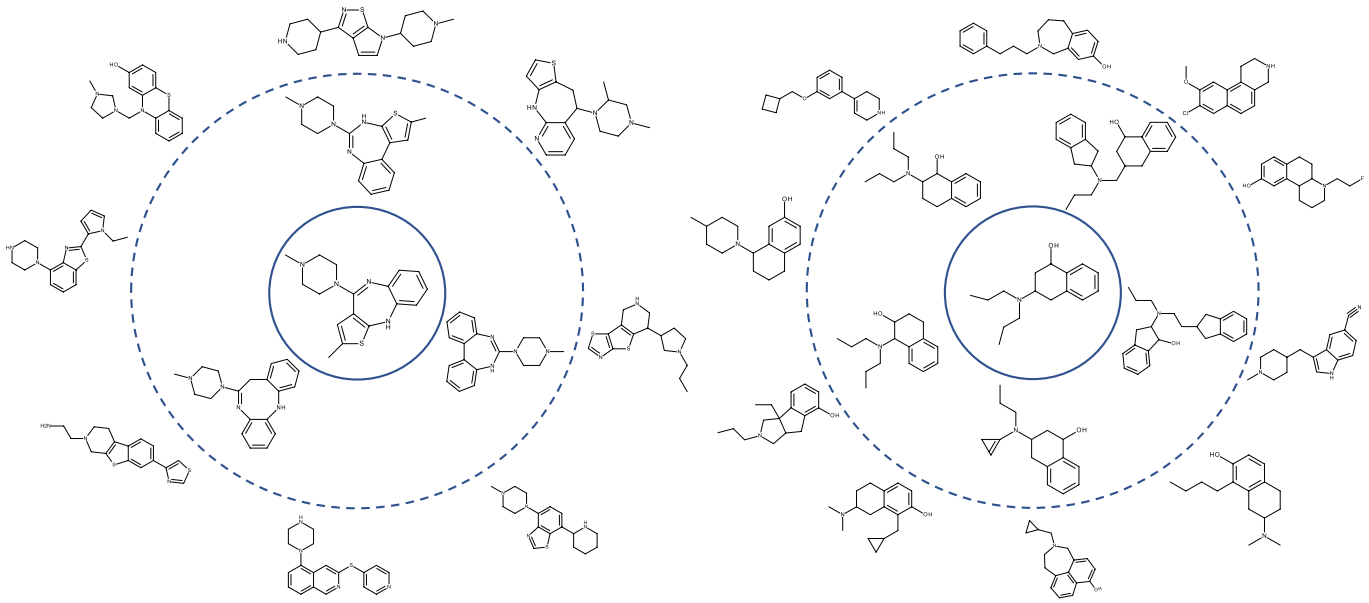


Fig. 3 | Unique structures corresponding to generated SMILES strings from two different known active seeds. The seeds were randomly selected from the DRD2 test set and are shown in the centre. The selected FPB (within the dashed circle) and PCB (outside the dashed circle) generations shown have QED values of ≥0.8 and a predicted active probability of ≥0.8. The FPB-generated molecules mostly maintain the seeding scaffold whereas the PCB-generated ones change scaffolds. A quantitative investigation of more structures is reported in Table 2 and the rest of the unique generated molecules for those two given seeds are shown in Supplementary Figs. 4–11.

both seeds met those values. Of all 256 generated SMILES strings in each batch sampled by the FPB model, ~5% referred to unique molecules that jointly satisfy these challenging constraints, because of high repeatability in the output (discussed in the next section). Similarly, ~5% and 22% of the PCB-generated SMILES strings corresponded to unique molecules that meet the specifications of the first and second seed, respectively. The rest of the unique molecules corresponding to valid generated SMILES strings for those two given seeds are shown in Supplementary Figs. 4–11.

The FPB-based generations demonstrate almost identical structure to the seed, at least at a scaffold level. On the other hand, the PCB-generated molecules have clearly different scaffolds from the seed, which can be attributed to the fact that the selected physicochemical descriptors do not encode structural information directly.

The correlation between the seed and the output of the models was further investigated by calculating the Tanimoto similarity of multiple batches of generated SMILES strings. For that purpose, 100 seeds were randomly selected from the unseen active compounds of the DRD2 test set and, for each one, 256 SMILES strings were generated in a batch by each of the conditional models, yielding a total of 25,600 SMILES strings. For each batch, the pairwise Tanimoto similarities were calculated between the Murcko scaffolds of the associated seed and of all unique molecules behind the generated SMILES strings. Given that fingerprints are not a complete molecular representation, they may be decoded to different molecules than the ones from which they originated, yet frequently with the same scaffold. To account for fingerprints that are not naturally decoded to the exact seeding molecule, the similarity of scaffolds is shown because it enjoys higher values even when scaffold decoration is slightly different. By doing this, we emphasize even more the fact that the FPB model maintains the structural characteristics of the input whereas the output of the PCB is structurally dissimilar to the molecule (and the scaffold of the molecule) from which the seeding conditions originated. The exact reconstructability of the two models on a molecular level is reported in Table 2.

The resulting Tanimoto similarity histograms are plotted in Extended Data Fig. 2a, while histograms of the predicted probability of them being active towards DRD2 are plotted in Extended Data Fig. 2b. The PCB-generated scaffolds tend to be dissimilar to their seeds, in contrast to the FPB-generated ones, the similarity of which to the seeding scaffold follows a bimodal distribution that is shifted

Table 2 | Comparison of cRNN and transfer learning models with respect to custom metrics and the MOSES suite

Metrics		Models		
		PCB	FPB	TL
MOSES				
Valid	↑	0.881	0.951	**0.968**
Unique@1k	↑	0.996	0.276	**1.000**
Unique@10k	↑	**0.996**	0.304	**0.996**
FCD	↓	7.981	**5.590**	8.438
SNN	↑	0.341	**0.774**	0.375
Frag	↑	0.920	**0.966**	0.938
Scaf	↑	0.094	**0.491**	0.193
IntDiv	↑	0.845	0.834	**0.846**
Custom				
Novelty[a]	↑	0.878	0.299	**0.953**
Predicted active fraction[b]	↑	**0.536**	0.194	0.474
Reconstructability[c]	-	≤0.001	0.630	-

The DRD2 test set was used as a reference set for the MOSES framework and the seed conditions were drawn from it to be used by the PCB and the FPB cRNN models. Upwards-pointing arrows show that higher scores are considered better. Downwards-pointing arrows show that lower scores are considered better. Numbers in bold show the best score for that metric. Molecules with a predicted probability greater than 0.5 by the QSAR model were considered active. Uniqueness, novelty, predicted active fraction and reconstructability were assessed on a molecular level using the underlying canonical SMILES behind all generated strings. [a]Calculated with respect to the molecules behind the generated SMILES strings and the merged active molecules of the DRD2_TRAIN and ChEMBL25_TRAIN datasets only. [b]Fraction of 25,600 generated SMILES strings that are valid and refer to unique and predicted active molecules. [c]Calculated based on the most frequently sampled SMILES string out of 256 strings per conditional seed.

to the right, showing that similar or identical scaffolds are generated. However, in both cases, the distribution of active probabilities is comparable (Extended Data Fig. 2b), proving that both models can generate SMILES strings that refer to predicted active compounds given the appropriate conditions. Evaluation of the performance of the QSAR model on all 'unseen' known active molecules of the DRD2 test set shows that it misclassifies 10% of all cases as inactive (with a predicted active probability less than 0.5), whereas in 4% of all cases it mis-assigns a probability of being active in the range of 0.0–0.1. This implies that the inherent imperfections in the QSAR model may be responsible for the respective mode collapse observed in Extended Data Fig. 2b. The reason why the peak around an active probability of 0 of the FPB curve is higher than for PCB is because the FPB model offers a substantially lower number of unique molecules behind its generated SMILES strings and, thus, misclassification errors are enhanced when normalizing the plots with the histogram density instead of bin count.

This supports the previous observation that the PCB model can generate different scaffolds from the same seed. Additionally, it identifies the sampling domain of each model. The main advantage of using fingerprints is that structural restrictions are directly encoded, a fact that is of use when scaffolds that are similar or identical to the seed need to be generated. On the other hand, using physicochemical properties as conditions offers a more versatile sampling, so this model could thus be applicable to explorations outside of a known scaffold, yet within the boundaries of the desired property setpoints.

Benchmarking. To quantitatively assess the performance of the proposed cRNN architecture against the selected baseline as well as other published work in the field, it was run through relevant and compatible metrics from the two main benchmarking suites in the field of de novo molecular generation: MOSES[22] and GuacaMol[23].

MOSES benchmark. The two cRNN models, along with the baseline trained with transfer learning, were tested with respect to the metrics provided by the MOSES framework[22]. For that purpose, 25,600 SMILES were additionally sampled by the model trained with transfer learning, similar to the sampling done for the other models as described in the previous sections. The metrics were calculated with respect to the active compounds of the DRD2 test set that was used as a reference dataset.

The PCB model performs the worst with respect to most metrics, except for the predicted active fraction and uniqueness of underlying canonical SMILES among 10,000 sampled SMILES strings. However, the metrics need to be interpreted carefully. The seed conditions used for the generation were extracted from active compounds of the DRD2 test set, which were not included in the training set of both conditional models. The active class is heavily under-represented in the datasets on which they were trained (only 2.3% of predicted actives in ChEMBL; Table 1) and thus the set of conditional seeds corresponds to a demanding task, which becomes even harder for the PCB model to fulfil because much less information is included in the physicochemical descriptors than in the fingerprints. On the contrary, the transfer learning model was trained directly on known actives and it is independent of any input during generation, while trying to replicate what has been seen during training. Lacking input conditions offers an implicit advantage over the conditional models in terms of valid generated SMILES strings, because specific input combinations may cause a consistent drop in generated validity. However, within a sample of 10,000 generated SMILES strings, the PCB and the transfer learning models are on par regarding uniqueness of the underlying canonical SMILES behind their sampled strings. This metric is a performance indicator, yet it does not fully expose the differences between the models, simply because the output space is too large to generate enough strings with duplicate canonical forms within only 10,000 sampled SMILES strings. On the other hand, the FPB model has low uniqueness, but this is expected as the more deterministic nature and the lower number of possible SMILES to sample from a single fingerprint naturally leads to duplicated outputs and penalized uniqueness.

The Fréchet ChemNet distance[33] (FCD) underlines the chemical distance between the reference and the generated distributions. As such, it is heavily in favour of the conditional models, because the seeds drawn from the test set purposely force the generated distributions towards it and consequently towards lower FCD values. Moreover, because the DRD2 train and test sets had been clustered and the fact that the transfer learning model was further trained on one of them explains the deviation from the other with respect to the FCD metric. Internal diversity also exhibits expected behaviour for a similar reason; the seeds narrow the output down compared to an ideally random sampler in the DRD2 active domain, such as the transfer learning model.

Among all 25,600 SMILES generations, a higher novelty of underlying canonical SMILES was achieved by the transfer learning model. This was due to the lower validity and uniqueness of the conditional models because the upper novelty boundary is defined by the product of validity and uniqueness. For the PCB model that boundary is $\sim 0.881 \times 0.996 = 0.877$, which is reached as seen in Table 2. This is an indicator that the PCB model, even though it suffers from lower validity compared to the transfer learning model, does not copy the training dataset. Similarly, the upper novelty boundary for the FPB is around $0.951 \times 0.304 = 0.289$, which was slightly exceeded because uniqueness@10k calculated by MOSES is probably lower than the complete uniqueness of all 25,600 generated SMILES strings. For the transfer learning model, the upper boundary $0.968 \times 0.996 = 0.964$ was almost reached as well. As observed, even though the absolute number of novel compounds was higher for the transfer learning model, none of the models replicated the training datasets.

It is noteworthy that a higher fraction of predicted active molecules on the basis of the underlying unique canonical SMILES strings was sampled by the PCB model, whereas the least of them were generated by the FPB model. The FPB model was punished because of its high reconstructability, which negatively affects its uniqueness score.

More specifically, the molecular reconstructability of the input descriptors was assessed by trying to retrieve the molecule that was represented by them at each batch. By identifying the most frequently sampled canonical SMILES string in batches of 256 generated strings given a single conditional seed, almost 65% of the FPB-generated batches primarily consisted of strings with the same canonical form as the molecule behind the seeding fingerprint. Those were considered successful reconstructions. Further experimentation with a deeper FPB model with four decoding layers and 512 LSTM units each made it possible to increase the reconstructability to 72%. Fingerprints are commonly thought of as being non-invertible (Table 1[9]) due to information loss in the embedding and hashing operations. However, as Morgan fingerprints can be understood as graph convolutional embeddings[34], they are in fact partly invertible when considered in the scope of the training set. Nonetheless, reconstructions were very scarce when using the physicochemical descriptors in the PCB model, because 256 sampled SMILES strings were not enough to identify a specific molecule in the diverse chemical space behind a set of given input conditions.

To investigate whether novelty of the conditional models is influenced by the training or seeding dataset, 100 new conditions were drawn from each one of the training and test subsets of ChEMBL. Then, the novelty of the unique canonical forms behind all valid SMILES strings out of 256 generated strings (one batch) per set of conditions was assessed with respect to both datasets. The results are shown in Extended Data Fig. 3. As hypothesized, both models may use the conditions stemming from unseen molecules and

Table 3 | Comparison of the cRNN models with the generative models benchmarked in the GuacaMol suite

Benchmark	Best of dataset	SMILES LSTM	SMILES GA	Graph GA	Graph MCTS	cRNN (ours)
Celecoxib red.	0.505	**1.000**	0.607	**1.000**	0.378	**1.000**[a]
Troglitazone red.	0.419	**1.000**	0.558	**1.000**	0.312	**1.000**[a]
Thiothixene red.	0.456	**1.000**	0.495	**1.000**	0.308	**1.000**[a]
logP (−1.0)	**1.000**	**1.000**	**1.000**	**1.000**	0.980	**1.000**[b]
logP (8.0)	**1.000**	**1.000**	**1.000**	**1.000**	0.979	**1.000**[b]
TPSA (150.0)	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**[b]
CNS MPO	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**[b]
QED	**0.948**	**0.948**	**0.948**	**0.948**	0.944	**0.948**[b]

The cRNN architecture (rightmost) achieves a maximum score in all relevant test cases. Best scores per metric are highlighted in bold. GA, genetic algorithm; MCTS, Monte Carlo tree search. [a]Considering the FPB cRNN. [b]Considering the PCB cRNN.

generate strings that describe structures that are not present in either dataset. For any of the models, the difference between datasets is insignificant, reflecting a consistent generation of SMILES strings that point to novel molecules, regardless of the origin of the seeding conditions.

GuacaMol benchmark. To compare the cRNN architecture against more generative models other than the chosen baseline, we chose to test it on all applicable goal-directed tasks defined in the GuacaMol[23] scoring suite considering the selected input descriptors for the PCB and FPB models.

For the FPB model, the tasks corresponding to rediscovery of celecoxib, troglitazone and thiothixene were employed. Rediscovery is defined as a maximization of the similarity score (1.0) between the calculated ECFC4 fingerprints of the generated structures and the target molecule. It is important to underline that, even though the FPB model has been trained on Morgan fingerprints, rediscovery is awarded a score of 1.0, regardless of the fingerprint representation, if the correct structure is generated. Moreover, the models tested with GuacaMol used prior knowledge, where applicable, in the form of 100–300 known highest-scoring molecules from the ChEMBL dataset as initial points for the optimization. They were also exposed to the target of interest via the scoring function in a feedback loop. To account for the benefit of steering the output using the feedback loop and to ensure a fair comparison, the FPB cRNN was preconditioned with the Morgan fingerprint of the target. According to GuacaMol, a maximum of approximately 10,000 generations (SMILES or graphs) took place using each model, and an early stop was allowed if the maximum score was achieved. Similarly, the FPB model was asked to generate 39 batches of 256 SMILES strings using the same conditional seed, for each of the three targets. It is noteworthy that it rediscovered all three targets even from the very first batch of 256 generated SMILES strings.

Regarding property satisfaction benchmarks, the ones that were applicable to the input descriptors with which the PCB cRNN was trained were the two logP targets and the TPSA, QED and the central nervous system multi-parameter optimization (CNS MPO) tasks. Similarly, the 100 top-scoring molecules from the ChEMBL25 test dataset were selected as conditional seeds for the PCB model to generate a single batch of 256 SMILES strings from. It is noteworthy that all molecules in ChEMBL and all virtual compounds generated by the models of the GuacaMol suite could not exceed a QED score of 0.948; this was also observed with the results of our method. Additionally, no molecule with a QED score higher than this value is reported in ref. [35], meaning that there might be an inherent natural upper bound of QED around 0.948, regardless of its definition. Given that high QED scores are scarce, only 54 known molecules in the ChEMBL25 test dataset had a value of ~0.948, thus fewer seeds were used. All the results are summarized in Table 3.

Overall, the results demonstrate that the cRNN architecture performs on par with the best scoring literature algorithms featured in the GuacaMol benchmarking suite, achieving a maximum score for all eight given tasks.

Control of generated properties. The primary advantage of the PCB model is the ability to generate SMILES strings of molecules that follow the desirable properties. This was tested by using 10 conditional seeds derived from randomly selected active compounds from the DRD2 test set whose QED scores are all greater than 0.5. For each conditional seed, a batch of 256 SMILES were generated and the physicochemical properties defined in the input conditions were calculated for all valid SMILES using RDKit. As shown in Fig. 4, most of the properties of the generated valid SMILES exhibit only a small deviation from the defined conditional setpoint, with the QED property having relatively large variance around the reference level.

All 10 property combinations that were used as a reference for this experiment were drawn from known active molecules from the DRD2 test set to challenge the PCB model with truly rare combinations, because the active molecules of the DRD2 test set have been excluded from ChEMBL and were also clustered to be dissimilar from the training and validation sets of the QSAR model. The DRD2 activity setpoint for this experiment, instead of being set to active for all 10 seeding conditions due to prior knowledge, was set according to the QSAR model's prediction for each of the 10 known active seeds because the generated SMILES strings would be evaluated by the QSAR model anyway. Due to the QSAR model's imperfection, two of the known active seeds (5 and 8) were falsely predicted as inactive, and the setpoint for those two was set as inactive. Therefore, the upper limit of the percentage of predicted active SMILES strings is expected to be 80%. After evaluating the probability of all generated SMILES strings that corresponded to valid molecules, ~40% of them were predicted as active with a probability not less than 0.5.

To further investigate the capability of a cRNN to control the properties of the molecules corresponding to its generated SMILES strings, more experiments were conducted where single properties were varied in both directions while keeping the rest of them fixed. A molecule from within the first and third quartiles with respect to all properties of the DRD2 test dataset was selected from which to obtain the initial conditions. Then, for each of the descriptors apart from the active probability, five values were tried out in a stepwise ascending fashion, spanning the value range between the first and third quartiles of each property, while keeping the rest of the conditions at the initial level. The tested conditions correspond to

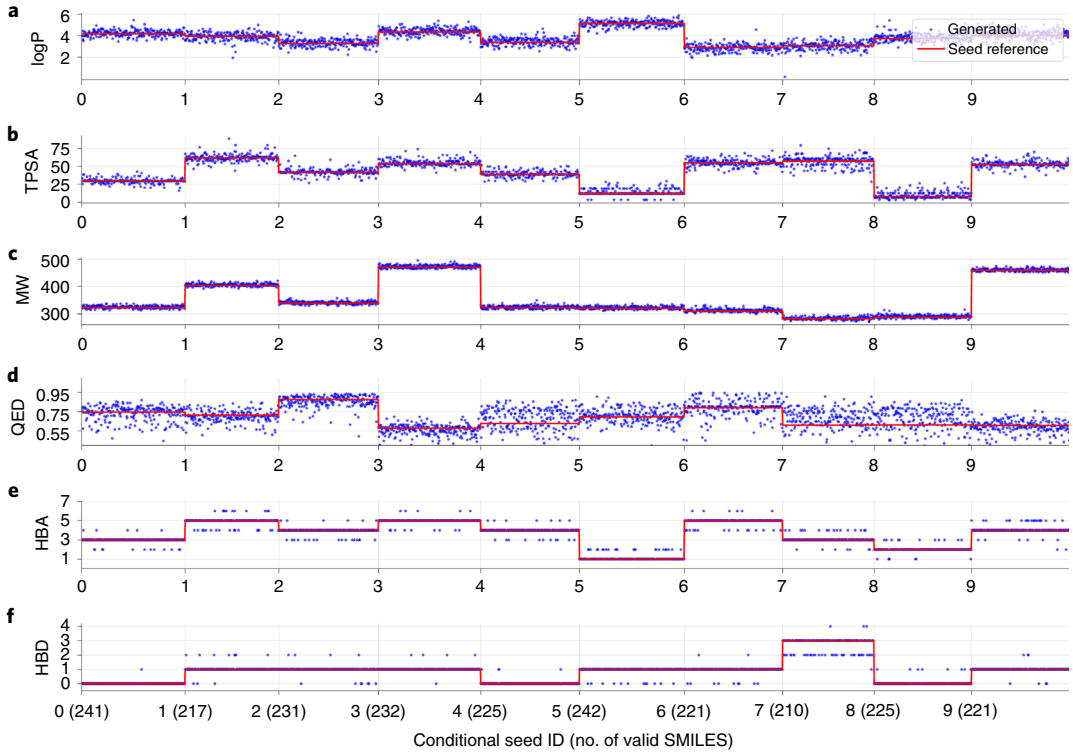


Fig. 4 | Property satisfaction with the PCB model. Reference properties (red) are plotted against the properties of generated valid SMILES (blue) for 10 random conditional seeds from the DRD2 test set. The properties considered are the logP, TPSA, MW, QED, HBA and HBD. The numbers in parentheses denote the number of unique valid molecules out of a batch of 256 generated SMILES strings. The patterns of all generated properties follow the reference. The QED constraint is the hardest to satisfy.

arbitrary property setpoints, unlike the ones shown in Fig. 4. The reference (red line) and generated (blue dots) properties of all valid SMILES strings are shown in Extended Data Fig. 4. Each column of cells per plot corresponds to the tuning of a single property while keeping the other five conditions fixed at the initial values. Overall, logP, TPSA, molecular weight and HBD setpoints were adequately matched in the generated molecular properties, followed by HBA, which seems to be unstable for low values of logP and high values of MW. The QED formula contains the weighted sum[35] of all the other five properties; consequently, the requested conditions along with the QED setpoint may render it impossible for that equation to be satisfied. Therefore, the QED property was hard to keep at the reference value as shown by the large spread around the target value (Extended Data Fig. 4).

This is particularly evident in the region around low values of MW or requested high values of QED for the given seed. In the first case, logP and QED decreased under the influence of the value of MW, which was controlled by the cRNN. From the short length of the step, it is observed that this batch of SMILES suffered from high invalidity and, as far as the valid examples are concerned, their logP and QED values were much lower than the setpoint. Similarly, requested high values of QED, given the values of the other five properties, were impossible to achieve, which affected the values of all properties and eventually led to none of their setpoints being respected, as annotated with the arrow markings in Extended Data Fig. 4.

These are the cases for which input combinations were ill-defined and resulted in either unattractive or invalid structures, something that has also been observed in the latent space vectors of autoencoders[27]. In the cRNN context, such combinations may refer to under-represented regions in the training dataset due either to a lack of relevant samples in the source or conflicts between the requested descriptor ranges. The conditions are entangled because they depend on each other, as observed from the behaviour of the QED score. In most cases, the user is probably interested in tuning only one of the properties rather than restraining many of them; nonetheless, all property conditions ought to be set at reasonable values to avoid the entanglement problem. More sophisticated sampling approaches, such as the LatentGAN architecture[29], could potentially address the entanglement problem. In particular, the generator component of LatentGAN may be used to autonomously propose a valid combination of input properties that lead to active generations towards bioactivity targets.

Exclusivity of sampling. Sampling the cRNN model with the seed conditions derived from a query structure should theoretically make it more likely to generate structures similar to the seed (Fig. 2) and less likely to sample dissimilar molecules. To investigate this hypothesis, 100,000 molecules were randomly selected from the ChEMBL test set and clustered using the DBSCAN algorithm[36], based on the Euclidean distance of their five scaled physicochemical properties (logP, TPSA, MW, HBA and HBD). A value of the maximum neighbour distance, $\varepsilon = 0.1$ and 10 minimum samples for associating core points were selected as parameters of the DBSCAN algorithm.

Next, two clusters of molecules (with sizes of 53 and 57, respectively) were manually selected to keep the variance of their descriptors within a range as narrow as possible, with preferably small overlap. The distributions of logP, TPSA and MW of the selected clusters are shown in Fig. 5a–c. All the molecules of the first cluster resulted in an HBA count of four and an HBD count of zero, while all molecules of the second cluster resulted in counts of four and one, respectively. The selected clusters show minimal or no overlapping with respect to logP, TPSA and HBD count, whereas they share

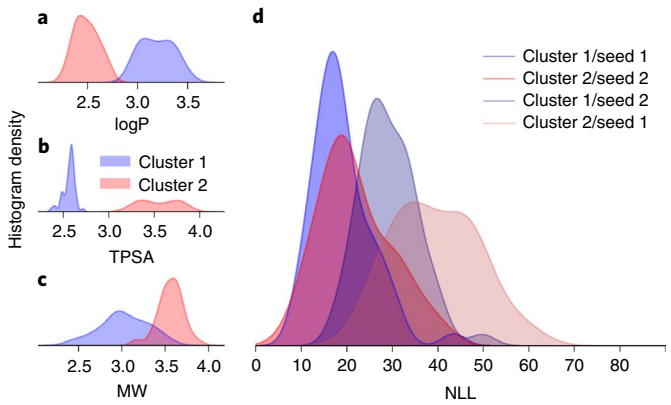


Fig. 5 | Exclusivity of sampling. a–c, Distribution of logP (**a**), TPSA (**b**) and MW (**c**) of each cluster. **d**, Distribution of calculated NLL of sampling each cluster using the two cluster centres as seeds, interchangeably. All distribution curves were fit using the kdeplot method of the seaborn Python library and default settings. A total of 10,000 randomization operations were performed on each molecule and all resulting unique randomized strings per molecule were considered when calculating the NLL using equation (5). It is shown that using a relevant seed makes it more probable to sample SMILES strings that correspond to chemically neighbouring molecules than molecules from another cluster.

the same count of HBA and similar values of MW. The values of QED and predicted probability of being active were not considered during clustering.

A total of 10,000 randomization attempts were performed on each molecule and all resulting unique randomized strings per molecule were considered when calculating the NLL using equation (5). The seed conditions were selected as the coordinates of the geometric centre of each cluster. The conditional NLL of sampling the molecules of each cluster under different seeds was calculated according to equation (5) (Fig. 5d). The cross-conditional NLL was calculated for each cluster by swapping the conditional seeds of both clusters. Theoretically, the generation of molecules from these two clusters during conditional sampling should be mutually exclusive using their own cluster centre as seed. In other words, using each cluster centre as the seed should give a higher probability to sample molecules within the relevant cluster.

This hypothesis is supported by Fig. 5d. When the conditional vector is derived from the seed of cluster 1, it is more likely to sample SMILES from the same molecular cluster (dark blue curve) than the ones from the second cluster (light red curve, Fig. 5d). The same applies when conditioning the generation with the molecules of the second cluster (dark red and light blue curves, Fig. 5d). Even though there is an overlap between the self-conditional and cross-conditional NLL curves, in both cases the former ones describe lower NLL values, thus showing that relevant molecules are more likely to be sampled. Overall, both clusters are comparably probable to be sampled when their own seeds are used as conditions (dark blue and dark red curves, Fig. 5d). Finally, by comparing the self-conditional NLL curves of Fig. 5d to the curves of Fig. 2, it is observed that the NLL curves of Fig. 5d are all shifted towards higher NLL values. This is expected though, because the conditions considered for each molecule were not derived from its own properties but instead from the mean properties of the cluster.

On the use of NLL for conditional model assessment. Given the size of the chemical subspace that generative models represent, sampling 1,000 or 10,000 molecules, as the main benchmarking suites[22,23] propose, may not sufficiently exemplify a model's capacity. For uniformly trained models[16] with a potential target space of zillions of equiprobable molecular strings, the expectation of rediscovering a particular molecule of interest from a left-out testing dataset as a proof of concept may be unrealistic. One may argue that the ability of a model (or the lack of it) to find a specific molecule in a finite number of attempts is indicative of the probability of distinguishing it from all other molecules and, thus, of the model's focus on that target. However, in such an experimental set-up, we can either reject a model's inability or fail to reject it, because accepting this hypothesis would require a significantly larger number of attempts.

In this work, we have attempted to gain insight into different aspects of the proposed cRNN architecture using distributions of CNLL values, as described by equation (1). In Fig. 2a,b, we employ the CNLL values to assess the training quality of the cRNN models, effectively bridging the gap between unbiased and conditional models. Instead of attempting to rediscover some known active molecules from the DRD2 training and testing datasets, we prove that they would be almost as easy to sample as molecules from ChEMBL using the cRNN models as shown in Fig. 2c,d. By retrospectively calculating the NLL values for all SMILES sequences in the complete DRD2 datasets, we demonstrate the correct focus of the output of the model in a transparent fashion while exhaustively using all the available ground truth.

Consequently, we see two main benefits. First, evaluation of the output is independent of any custom scoring (QSAR) model that induces additional noise to the results, which otherwise reflect the multiplicative performance of the generative and QSAR models. This also deals with potential mode-collapsing problems of the QSAR model when applied on unseen or novel data by eliminating them from the evaluation process. Second, the CNLL formula will yield an interpretable score for any arbitrary sequence, regardless of the size of the output space of a model. By using a meaningful threshold, this score would directly reflect the model's ability to reach the sequence of interest.

Moreover, by exchanging conditional seeds between clusters as shown in Fig. 5d, the cross-conditional NLL values can indicate the exclusivity of the focus of a model. In the field of de novo design, this could be translated to probabilistic avoidance of structural alerts, such as toxic substructures, and could serve as a meaningful test case during benchmarking of different models.

The definition of the CNLL as proposed in equation (1) allows for the evaluation of the probability of generating a sequence given a fixed set of descriptors. As evinced by the experiments in this work, the more information that is induced into the network by the choice of descriptors, the more deterministic its output becomes. Through this operationalization, one may appreciate the stochasticity in the learnt decision process of an RNN. Thus, the combination of manually selecting the input features to train on, along with post-processing of the results using CNLL plots, could be an alternative approach to assess the collective feature importance between discrete sets of inputs of an RNN for sequence generation—a field that is under research for black box models[37,38].

All in all, we believe that such an analysis based on NLL statistics is essential for a fair comparison between models and can be adopted by all probabilistic sequence generators, regardless of domain. Yet, to extend existing benchmarks to the proposed novel direction, except for predicting a sequence, all relevant models should be modified to deliver the joint probability of sampling that sequence, or its equivalent NLL.

Applications to drug discovery and beyond. Among the core contributions of this manuscript lies the controlled specificity of the output space of the proposed cRNN architecture. The FPB cRNN, as seen from Fig. 2, has the least stochastic output of all the models, a behaviour that is close to what would be expected from autoencoding networks. The PCB NLL values for all datasets lie between the NLL values of the FPB and the unbiased models. This is significant,

because it implies that the PCB model has a more diverse output space than the FPB (or similar autoencoding networks) while maintaining its focus on the property setpoints, as shown in Fig. 4 and Extended Data Fig. 4. This offers an incomparable advantage over the baseline model trained with transfer learning, because the baseline does not provide any degrees of freedom to the user to shape the output after it has been trained on a dataset. This means that the PCB model is able to find multiple near solutions to the multi-objective optimization problem at hand, whereas most current work in the literature focuses on optimization of just a single property at a time, that is, maximization of logP.

Moreover, we have shown that a cRNN can learn either physicochemical or structural characteristics, depending on the set of descriptors that are chosen by the user to expose the network to during training. Here, the selected properties consist of five physicochemical properties, a well-defined weighted average of them (QED) and a data-driven scoring function (QSAR model) based on public data. This diverse selection showcases the method in a transparent and reproducible way, while at the same time underlining the versatility of the algorithm for drug design purposes. Therefore, the cRNN architecture provides a way of addressing the inverse QSAR problem directly as the PCB cRNN is able to generate molecular structures with desired properties.

On the contrary, other available methods suggest the use of optimization algorithms[21,28] or reinforcement learning[18] to close the loop and steer one or more initial candidate molecules towards the aspired region of the chemical domain in an iterative process. Such optimization approaches require looping over a cost or desirability function, whereas in our case a batch of 256 potentially interesting SMILES strings with properties close to predefined target values can be directly generated with a single forward pass of the trained cRNN. The main advantage of the proposed algorithm over this family of methods is that the inference time of a cRNN is not affected by the arbitrary complexity of the input, because it is exposed to it during training and its weights are adjusted to a specific task. It exhibits quasi-constant inference time, because the implication of sampling a sequence of unknown length on runtime performance is a common denominator for both types of algorithm. In contrast, optimization algorithms that are applied on universally pre-trained models require ad hoc exploration of the chemical space during runtime. This characteristic allows for interactive applications to be built, where a constant feedback cycle permits smooth experimentation, such as allowing the user to dynamically select the target properties and visualize the results within a laboratory set-up.

Nevertheless, a combination of the two distinct approaches could potentially yield even greater benefits. The pre-trained cRNN could complement a reinforcement learning loop[18] by proposing abundant meaningful starting points that would speed up or enhance its convergence in a lead-optimization fashion. Other optimization techniques, such as particle swarm[28] or Bayesian optimization[21], could be used to finetune the conditional seed on top of a cRNN instead of a simple decoder, which in the case of the PCB model would be chemically interpretable, whereas the FBP model could suggest a series of similar compounds based on the optimized seed.

Most importantly, however, our proposed method addresses the general inverse design problem, where a recommender system proposes solutions on a multidimensional manifold that conform to desired specifications. One may extrapolate from our case study to a more generalized approach to the task of customizing the specificity of—and not just the context of—sequential data generation. Such examples could be natural language generation, where the focus of the context is set according to keywords, or autoregression of time-series, where the initial conditions could be set via the cRNN states. Last, but not least, based on the authors' prior experience in the field, maintaining the levels of several process (state) variables at a specified reference using just a single output, regardless of the mode collapse that appears around extreme setpoints as shown in Extended Data Fig. 4, could be exploited in the context of decoupled control of a nonlinear dynamical system. It would be stimulating to see a comparison of cRNNs against existing solutions based on artificial neural networks[39]. In the same context, the cross-conditional NLL could also reveal the probabilistic distance of the output from undesirable states, such as structural resonance or singular configurations of robotic arms.

Conclusions

In this work, the effect of introducing molecular descriptors as inputs to an existing SMILES generator architecture based on RNNs has been investigated. Primarily, it has been shown that known molecules are more likely to be rediscovered when sampling using the descriptor conditions that represent them as inputs to a cRNN, compared to a prior unbiased model that is simply trained on the complete molecular dataset. Our approach also demonstrated the capacity of generating novel compounds that were predicted active against the DRD2 receptor, which were also chemically closer to known active compounds than the ones generated by a baseline model trained with transfer learning. Additionally, a larger fraction of predicted actives was generated by the cRNN than the baseline model. After evaluating our model against literature benchmarks, the cRNN architecture was proven at least as good as the top-scoring models of a benchmarking suite, achieving the maximum score in eight relevant goal-directed tasks. Using molecular fingerprints as conditions focuses the molecular generation even more than physicochemical properties, by acting as structural restrictions that impose a scaffold on the output that is similar, if not identical, to the reference. This also demonstrated the capability of the proposed architecture to function as a fingerprint inverter, by being able to resample the original molecule even up to 72% of the time by using a more complex network. On the other hand, physicochemical properties are more versatile and lead to molecules with more diverse structures and different scaffolds than the molecule from which the conditions were derived. The cRNN architecture tackles the inverse QSAR problem by directly shaping the properties of the generated molecules while avoiding online optimization loops. Nonetheless, even though we have been able to optimize the conditions independently of each other, not all input combinations led to valid structures due to the conditions being correlated. As an example, this was observed when conditioning with a high QED setpoint while keeping the other conditions, which are constituents of the QED score calculation, fixed. Most notably, the cRNN has thus been demonstrated as a potentially useful architecture with an arbitrarily intermediate output space between unbiased character-based RNNs and fully steered autoencoders with a 1:1 relation between latent space vectors and molecules. Additionally, our experiments exemplify a novel way of assessing the focus of the conditional output of a model using NLL plots. Due to the nature of the problem this approach targets, it is expected that the proposed architecture can also be of importance in applications in sequential content creation other than drug design.

Methods

Datasets. The datasets used in this work originate from two publicly available sources: ChEMBL[40] and ExCAPE-DB[41]. Data from ChEMBL were used to train the generative neural network, while data regarding the dopamine receptor D2 (DRD2) target from ExCAPE-DB were used to train a QSAR model using a support vector classifier to estimate the likelihood of a generated compound being potent towards DRD2.

ChEMBL. The neural network was trained with a subset of ChEMBL version 25. Initially, the complete dataset was standardized with the MolVS Python module[42] using the super parent setting, which standardizes fragment, charge, isotope, stereochemistry and tautomeric states. Molecules were filtered to only contain the atoms [H, C, N, O, F, S, Cl, Br] and for a total of fewer than 50 heavy atoms. Next, the known active molecules found in the DRD2 dataset (see section 'ExCAPE-DB')

were removed from the dataset. The dataset was split into training and test subsets in a 9:1 ratio. During training, 10% of the training subset was used as a fixed validation set.

ExCAPE-DB. All data regarding the DRD2 entry in ExCAPE-DB were downloaded[43] and preprocessed as follows. First, duplicate compounds as well as SMILES strings[12] that were not sanitizable by RDKit v2018.09.1[44] were removed from the DRD2 dataset. In total 7,129 compounds had a pXC50 value greater than five and were selected as known actives along with 100,000 random DRD2 measured-inactive compounds from ExCAPE-DB. Stereochemical information was removed by converting all molecules to non-isomeric SMILES strings. The dataset was further reduced to exclude SMILES strings that were longer than the ones in ChEMBL or contained characters not found in ChEMBL. This led to removing strings with iodine and phosphorus. All active molecules were clustered based on the pairwise Tanimoto distance of their Morgan fingerprints with a radius of two using the implementation of the Butina algorithm[45] found in RDKit. The maximum distance threshold for the algorithm to associate neighbours was fixed to 0.4, with a value above this dictating different clusters. All clusters were sorted based on their size and were assigned to the train, validation and test subsets iteratively using a '4-1-1' scheme; that is, for every four clusters assigned to the train set, one cluster was assigned to the validation set and one to the test set in order of decreasing cluster size.

The curated datasets used to train all models are available from https://github.com/pcko1/Deep-Drug-Coder/tree/master/datasets.

SMILES strings randomization and vectorization. During training, the atom order of all molecules was randomized using RDKit. After converting them back to SMILES, every constituting character was one-hot encoded. Every SMILES string was thus represented by a two-dimensional (2D) array with dimensions corresponding to the length of the vocabulary and the maximum canonical SMILES length found in ChEMBL, with an offset of five extra characters to account for randomized SMILES that were longer than their canonical representation. The delimiting characters '^' and '$' were inserted in the beginning and end of each one-hot-encoded string, respectively. Resulting arrays that corresponded to shorter SMILES strings were padded with the end character '$'. The considered vocabulary consisted of 35 tokens that included all common unique alphanumeric characters found in ChEMBL and DRD2 datasets after filtering, the delimiters '^' and '$', and the token '?' to account for one-hot encoding of unknown characters.

The randomization and vectorization of all SMILES strings was performed dynamically using a modified version of the molvecgen Python package[46] during training.

DRD2 QSAR model. A probabilistic support vector machine classification model was used for bioactivity prediction. The standard implementation of a support vector machine (SVM) from the scikit-learn v0.20.3[47] Python package was used, with the radial basis function as a kernel function. The model was trained to discriminate active compounds from inactive ones based on their 2,048-bit-radius 2 Morgan fingerprint representations. Because a poor choice of the regularization parameter, C, and kernel coefficient, γ, may have a detrimental effect on the performance of an SVM, these were optimized with a randomized search in which 50 different values per parameter were drawn from two exponential distributions with replacement. The choice of fingerprint type and radius was based on published work[48,49] and was not optimized further.

Recurrent neural network. The neural network resembles the decoder architecture described in ref. [25]. It was implemented in Keras v2.2.4[50] with a TensorFlowGPU v1.12.0 backend[51] and is schematically shown in Fig. 1. The network accepts a vector of molecular descriptors as inputs to a set of six Dense feedforward layers of 256 units each, using the ReLU[52] activation function. The output of each individual Dense layer is used to set either the cell state or the hidden state of each of the recurrent layers of the network. There are, in total, three unidirectional recurrent layers in the network, each consisting of 256 LSTM[32] neurons. The output of the final LSTM layer is fed to a feedforward layer with 35 units, which is the length of the character space, using softmax activation. Batch normalization was applied to the outputs of all LSTMs and all but the last Dense layers. Keras CUDA-enabled CuDNNLSTM units were used in the recurrent layers.

The model was trained for 100 epochs with randomized SMILES strings following the 'teacher's forcing' method[53], using the ground truth at each step as prior knowledge instead of the character previously predicted by the network. A batch size of 128 sequences was used along with the Adam optimizer with default parameters[54] and an initial learning rate of 10^{-3}. A custom learning rate schedule was used, where the learning rate was kept constant for the first 50 epochs and then decayed exponentially at each epoch, down to a value of 10^{-6} at the final epoch.

A copy of the trained model was modified for the purpose of predicting single characters to jointly form SMILES strings. While maintaining the trained connection weights, the shape of the output of the last feedforward layer was set to a 1D vector expressing the probability of sampling each of the known characters at every step. Also, the LSTM layers were set to stateful mode. During inference, a single character per iteration is sampled out of this vector of probabilities using multinomial sampling. After setting the initial states according to the descriptors of interest, the biased generation is triggered by feeding the start- character '^' to the network and ends when the end character '$' is sampled.

Two different cRNN models were constructed and trained following this procedure, each based on different input descriptors. The first physchem-based (PCB) model is shown schematically in Fig. 1a. The model uses the Wildman–Crippen partition coefficient[55] (logP), topological polar surface area (TPSA), molecular weight (MW), number of hydrogen bond acceptors (HBA), number of hydrogen-bond donors (HBD) and the drug-likeness score[35] (QED) calculated using their RDKit implementations as well as the soft label predicted by the QSAR support vector classification (SVC) model described above. The calculated values were scaled individually to achieve a distribution with zero mean and unit variance, and they were concatenated into a single input vector.

The second fingerprint-based (FPB, Fig. 1b) model was trained solely on Morgan fingerprints of radius 2 and 2,048 bits, which are similar to extended connectivity fingerprints (ECFPs). The training and inference schemes of the cRNN models are described in Fig. 1a–c, respectively.

Model training and inferencing was performed on an NVIDIA Tesla V100 GPU on a 64 bit CentOS v7.5 server with 128 GB of RAM. The training process of the PCB and FPB models utilized 5 and 25 GB of RAM, respectively.

Transfer learning model. The baseline model consists of the same neural network architecture as described above with the notable difference that the initial states, instead of being set based on known descriptors, are instead being reset to zero in the beginning of the generation of each string. This approach is similar to the prior network described in ref. [18] with the difference that each character is treated independently rather than within multi-character tokens. The network was likewise trained with teacher's forcing, learning the character set and the grammar of the SMILES strings found in ChEMBL. The selected RNN dimensions were identical to the ones in the case of the cRNN.

Next, the prior model was further trained exclusively with the known actives of the DRD2 train dataset for an additional 200 epochs, following a transfer learning strategy[56]. The initial learning rate was set to $10^{-4.5}$ and was decayed exponentially to 10^{-6} by the end of the training.

Likelihood of sequences and molecules. The likelihood of sampling a given SMILES string was estimated using NLL as previously described[15], with a modification that incorporates the knowledge that is induced into the initial states of the generation in the case of a conditional model. The conditional NLL (CNLL) is defined as

$$\mathrm{CNLL}(S|c) = -\left[\ln P(X_1 = T_1|c) + \sum_{i=2}^{N} \ln P(X_i = T_i|X_{i-1} = T_{i-1}, \ldots, X_1 = T_1, c)\right] \quad (1)$$

where T_i are the characters in the known SMILES sequence S, X_i are the predicted model outputs, N is the length of sequence S, and c refers to the seeding conditions. The sign of the log-likelihood is made negative to reflect that higher values correspond to more improbable sequences.

The true probability of sampling a molecule M is given by the cumulative probability over all its U unique random representations S_j:

$$P(M|c) = \sum_{j=1}^{U} P(S_j|c) = \sum_{j=1}^{U} \mathrm{e}^{-\mathrm{CNLL}(S_j|c)} \quad (2)$$

Thus, the CNLL of a molecule is given by the negative of the natural logarithm of equation (2):

$$\mathrm{CNLL}(M|c) = -\ln \sum_{j=1}^{U} \mathrm{e}^{-\mathrm{CNLL}(S_j|c)} \quad (3)$$

Because the true number of all unique representations of a molecule is a priori unknown, the likelihood of a molecule can be approximated by a smaller number of unique representations u such that

$$\mathrm{CNLL}(M|c) = \lim_{u \to U} \left(-\ln \sum_{j=1}^{u} \mathrm{e}^{-\mathrm{CNLL}(S_j|c)}\right) \quad (4)$$

where

$$\mathrm{CNLL}(M|c,u) \triangleq -\ln \sum_{j=1}^{u} \mathrm{e}^{-\mathrm{CNLL}(S_j|c)} \quad (5)$$

is the approximation of the true molecular likelihood given input conditions c and a set of u unique random string representations S_j. Finally, from equations (2) and (4) we can derive that

$$P(M|c) \geq P(M|c,u) \geq P(S_j|c) \Leftrightarrow -\ln(P(M|c)) \leq -\ln(P(M|c,u)) \leq -\ln(P(S_j|c))$$
$$\Leftrightarrow \mathrm{CNLL}(M|c) \leq \mathrm{CNLL}(M|c,u) \leq \mathrm{CNLL}(S_j|c)$$

This means that the molecular NLL cannot be higher than the NLL of any individual random sequence corresponding to the given molecule.

Data availability

The curated datasets used to train all models are available at https://github.com/pcko1/Deep-Drug-Coder/tree/master/datasets.

Code availability

The Python code and the trained neural networks used in this work are available under MIT licence[57] in the Deep Drug Coder (DDC) GitHub repository https://github.com/pcko1/Deep-Drug-Coder and https://doi.org/10.5281/zenodo.3739063, which also includes an optional encoding network to constitute a molecular heteroencoder.

Received: 20 November 2019; Accepted: 14 April 2020;
Published online: 18 May 2020

References

1. Lopyrev, K. Generating news headlines with recurrent neural networks. Preprint at https://arxiv.org/pdf/1512.01712.pdf (2015).
2. Briot, J.-P., Hadjeres, G. & Pachet, F.-D. *Deep Learning Techniques for Music Generation* (Springer, 2020).
3. Wang, Z. et al. Chinese poetry generation with planning based neural network. In *Proceedings of 26th International Conference of Computing and Linguistics* 1051–1060 (COLING 2016 Organizing Committee, 2016).
4. Elgammal, A., Liu, B., Elhoseiny, M. & Mazzone, M. CAN: Creative adversarial networks, generating 'art' by learning about styles and deviating from style norms. Preprint at https://arxiv.org/abs/1706.07068 (2017).
5. Segler, M. H. S., Preuss, M. & Waller, M. P. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature* **555**, 604–610 (2018).
6. Ronneberger, O., Fischer, P. & Brox, T. U-Net: convolutional networks for biomedical image segmentation. In *Proceedings of Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015* (eds Navab, N., Hornegger, J., Wells, W. M. & Frangi, A. F.) 234–241 (Springer, 2015).
7. Chen, H., Engkvist, O., Wang, Y., Olivecrona, M. & Blaschke, T. The rise of deep learning in drug discovery. *Drug Discov. Today* **23**, 1241–1250 (2018).
8. Xu, Y. et al. Deep learning for molecular generation. *Future Med. Chem.* **11**, 567–597 (2019).
9. 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).
10. Vamathevan, J. et al. Applications of machine learning in drug discovery and development. *Nat. Rev. Drug Discov.* **18**, 463–477 (2019).
11. Sanchez-Lengeling, B. & Aspuru-Guzik, A. Inverse molecular design using machine learning: generative models for matter engineering. *Science* **361**, 360–365 (2018).
12. Weininger, D. SMILES, a chemical language and information system. 1: Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **28**, 31–36 (1988).
13. Schwalbe-Koda, D. & Gómez-Bombarelli, R. Generative models for automatic chemical design. Preprint at https://arxiv.org/pdf/1907.01632.pdf (2019).
14. Lipton, Z. C., Berkowitz, J. & Elkan, C. A critical review of recurrent neural networks for sequence learning. Preprint at https://arxiv.org/pdf/1506.00019.pdf (2015).
15. Arús-Pous, J. et al. Exploring the GDB-13 chemical space using deep generative models. *J. Cheminform.* **11**, 20 (2019).
16. Arús-Pous, J. et al. Randomized SMILES strings improve the quality of molecular generative models. *J. Cheminform.* **11**, 71 (2019).
17. Segler, M. H. S., Kogej, T., Tyrchan, C. & Waller, M. P. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.* **4**, 120–131 (2018).
18. Olivecrona, M., Blaschke, T., Engkvist, O. & Chen, H. Molecular de-novo design through deep reinforcement learning. *J. Cheminform* **9**, 48 (2017).
19. Zhou, Z., Kearnes, S., Li, L., Zare, R. N. & Riley, P. Optimization of molecules via deep reinforcement learning. *Sci. Rep.* **9**, 10752 (2019).
20. Popova, M., Isayev, O. & Tropsha, A. Deep reinforcement learning for de novo drug design. *Sci. Adv* **4**, 7 (2018).
21. Gómez-Bombarelli, R. et al. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **4**, 268–276 (2018).
22. Polykovskiy, D., Artamonov, A., Veselov, M., Kadurin, A. & Nikolenko, S. Molecular Sets (MOSES): a benchmarking platform for molecular generation models. Preprint at https://arxiv.org/pdf/1811.12823.pdf (2019).
23. Brown, N., Fiscato, M., Segler, M. H. S. & Vaucher, A. C. GuacaMol: benchmarking models for de novo molecular design. *J. Chem. Inf. Model.* **59**, 1096–1108 (2019).
24. Bjerrum, E. J. SMILES enumeration as data augmentation for neural network modeling of molecules. Preprint at https://arxiv.org/pdf/1703.07076.pdf (2017).
25. Bjerrum, E. J. & Sattarov, B. Improving chemical autoencoder latent space and molecular de novo generation diversity with heteroencoders. *Biomolecules* **8**, 131 (2018).
26. Winter, R., Montanari, F., Noé, F. & Clevert, D. A. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem. Sci.* **10**, 1692–1701 (2019).
27. Blaschke, T., Olivecrona, M., Engkvist, O., Bajorath, J. & Chen, H. Application of generative autoencoder in de novo molecular design. *Mol. Inform.* **37**, 1–11 (2018).
28. Winter, R. et al. Efficient multi-objective molecular optimization in a continuous latent space. *Chem. Sci* **10**, 8016–8024 (2019).
29. Prykhodko, O. et al. A de novo molecular generation method using latent vector based generative adversarial network. *J. Cheminform* **11**, 74 (2019).
30. 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).
31. Jin, W., Barzilay, R. & Jaakkola, T. S. Multi-resolution autoregressive graph-to-graph translation for molecules. Preprint at https://chemrxiv.org/articles/Multi-Resolution_Autoregressive_Graph-to-Graph_Translation_for_Molecules/8266745/1 (2019).
32. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1–32 (1997).
33. Preuer, K., Renz, P., Unterthiner, T., Hochreiter, S. & Klambauer, G. Fréchet ChemNet distance: a metric for generative models for molecules in drug discovery. *J. Chem. Inf. Model.* **58**, 1736–1741 (2018).
34. Duvenaud, D. K. et al. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems* Vol. 28 (eds Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M. & Garnett, R.) 2224–2232 (Curran Associates, 2015).
35. Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S. & Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nat. Chem.* **4**, 90–98 (2012).
36. Ester, M., Kriegel, H., Xu, X. & Miinchen, D. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge*, *Discovery and Data Mining* 226–231 (AAAI Press, 1996).
37. Škrlj, B., Džeroski, S., Lavrač, N. & Petkovič, M. Feature importance estimation with self-attention networks. Preprint at https://arxiv.org/pdf/2002.04464.pdf (2020).
38. Olden, J. D., Joy, M. K. & Death, R. G. An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecol. Model.* **178**, 389–397 (2004).
39. Hung, L. & Chung, H. Decoupled control using neural network-based sliding-mode controller for nonlinear systems. *Expert Syst. Appl.* **32**, 1168–1182 (2007).
40. Gaulton, A. et al. The ChEMBL database in 2017. *Nucleic Acids Res.* **45**, D945–D954 (2017).
41. Sun, J. et al. ExCAPE-DB: an integrated large scale dataset facilitating big data analysis in chemogenomics. *J. Cheminform.* **9**, 41 (2017).
42. Swain, M. MolVS: Molecule Validation and Standardization v0.1.1 (2019); https://molvs.readthedocs.io/en/latest/
43. Sun, J. et al. ExCAPEDB (2019); https://solr.ideaconsult.net/search/excape/
44. Landrum, G. et al. RDKit: Open-Source Cheminformatics Software (2019); https://www.rdkit.org/
45. Butina, D. Unsupervised data base clustering based on daylight's fingerprint and Tanimoto similarity: a fast and automated way to cluster small and large data sets. *J. Chem. Inf. Comput. Sci.* **39**, 747–750 (1999).
46. Bjerrum, E. J. Molvecgen: Molecular Vectorization and Batch Generation (2019); https://github.com/EBjerrum/molvecgen
47. Pedregosa, F. et al. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
48. O'Boyle, N. M. & Sayle, R. A. Comparing structural fingerprints using a literature-based similarity benchmark. *J. Cheminform.* **8**, 36 (2016).
49. Probst, D. & Reymond, J. L. A probabilistic molecular fingerprint for big data settings. *J. Cheminform.* **10**, 66 (2018).
50. Chollet, F. Keras (2019); https://keras.io/
51. Abadi, M. et al. TensorFlow: large-scale machine learning on heterogeneous distributed systems. Preprint at https://arxiv.org/pdf/1603.04467.pdf (2016).
52. Arora, R., Basu, A., Mianjy, P. & Mukherjee, A. Understanding deep neural networks with rectified linear units. Preprint at https://arxiv.org/pdf/1611.01491.pdf (2016).
53. Williams, R. J. & Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.* **1**, 270–280 (2008).
54. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *3rd International Conference for Learning Representations, (ICLR) 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (eds Bengio, Y. & LeCun, Y.) (2015).
55. Wildman, S. A. & Crippen, G. M. Prediction of physicochemical parameters by atomic contributions. *J. Chem. Inf. Comput. Sci.* **39**, 868–873 (1999).

56. Tan, C. et al. A survey on deep transfer learning. In *Artificial Neural Networks and Machine Learning — ICANN 2018* (eds Krurková, V., Manolopoulos, Y., Hammer, B., Iliadis, L. & Maglogiannis, I.) 270–279 (Springer, 2018).
57. MIT Licence; https://opensource.org/licenses/MIT

Acknowledgements
We thank the entire MolecularAI team at AstraZeneca for their invaluable input and the fruitful discussions held during development of the present work. J.A.-P. is supported financially by the European Union's Horizon 2020 research and innovation programme under a Marie Skłodowska-Curie grant (agreement no. 676434, 'Big Data in Chemistry', 'BIGCHEM'; http://bigchem.eu).

Author contributions
P.-C.K. and E.J.B. planned the project and jointly performed analysis of the results. P.-C.K. developed the necessary code. E.J.B. supervised the overall project. J.A.-P. assisted with the preprocessing of the datasets. J.A.-P., H.C., O.E. and C.T. provided valuable feedback on the methods used, the experimental set-up and the results at every stage. P.-C.K. wrote the manuscript and all authors reviewed it.

Competing interests
The authors declare no competing interests.

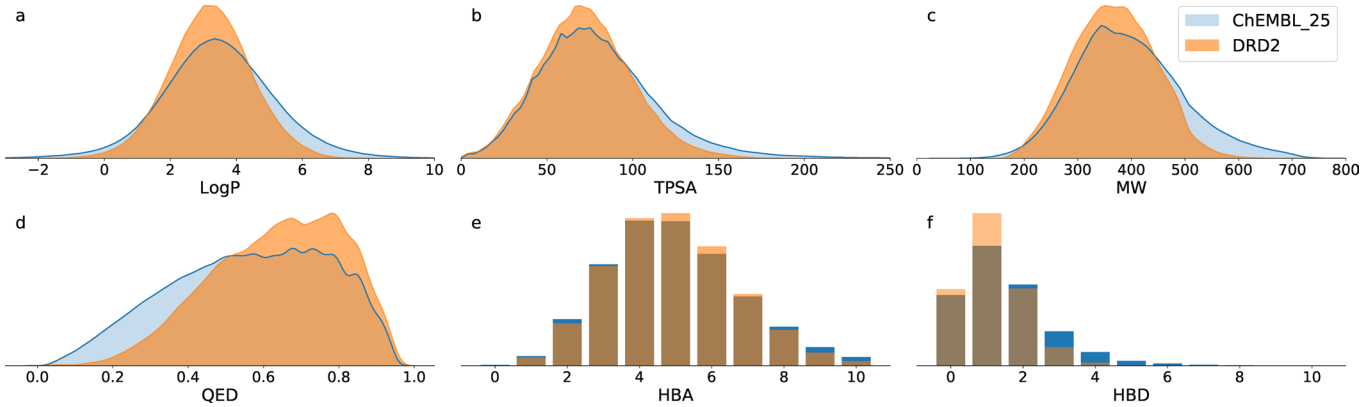
Additional information
Extended data is available for this paper at https://doi.org/10.1038/s42256-020-0174-5.

Supplementary information is available for this paper at https://doi.org/10.1038/s42256-020-0174-5.

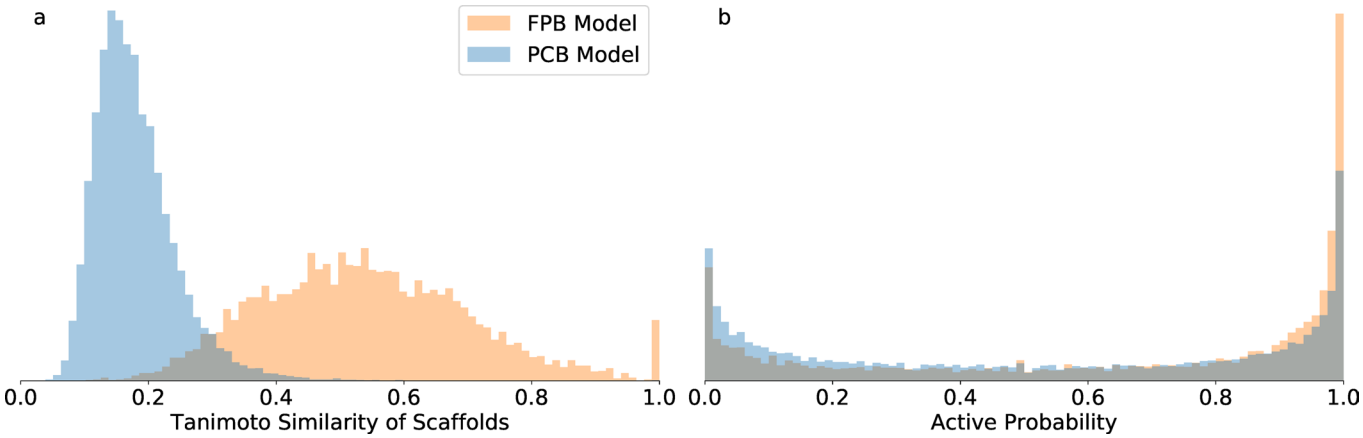
Correspondence and requests for materials should be addressed to E.J.B.

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

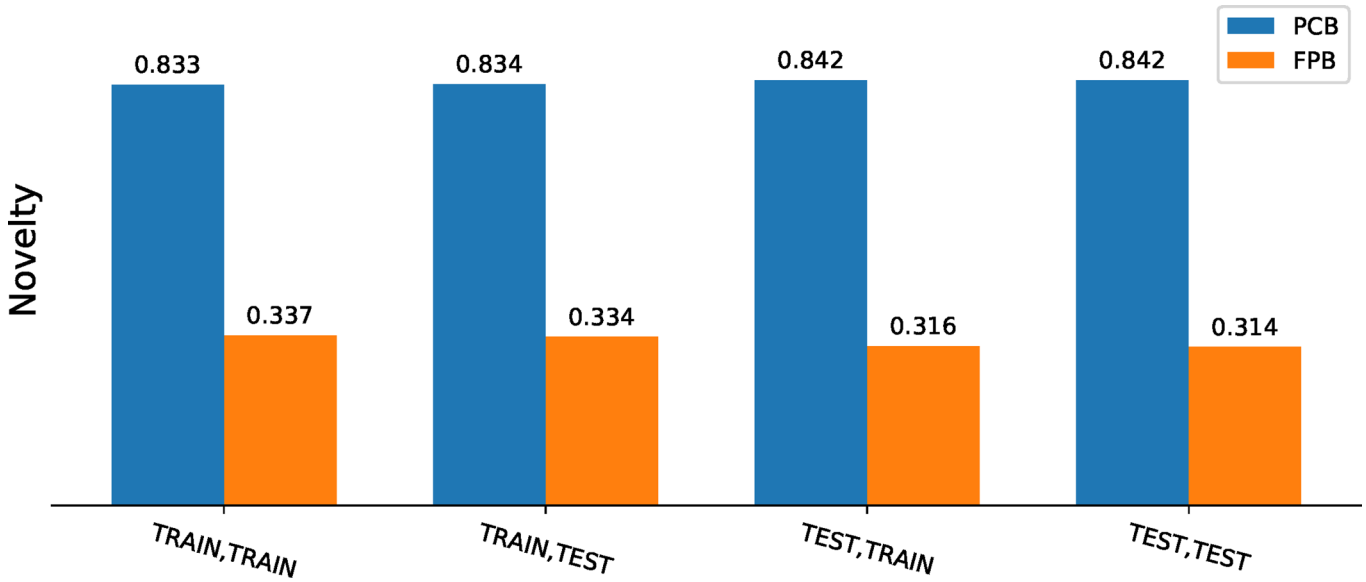
Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



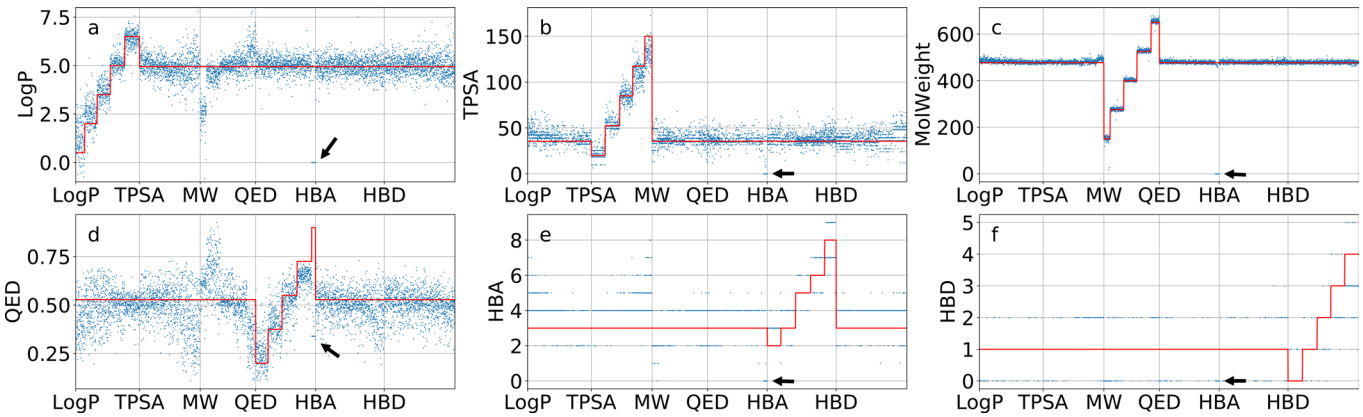
Extended Data Fig. 1 | Distribution of physicochemical properties of datasets. a, Wildman-Crippen coefficient (logP), **b**, topological polar surface area (TPSA), **c**, molecular weight (MW), **d**, drug-likeness score (QED), **e**, number of hydrogen bond acceptors (HBA) and **f**, hydrogen bond donors (HBD) with respect to the complete CHEMBL25 and DRD2 datasets before splitting. Subfigures a-d show the continuous histogram density as estimated by the kdeplot method of the seaborn Python library using default parameters.



Extended Data Fig. 2 | Tanimoto similarity and predicted activity of generated structures. a, Distribution of pairwise Tanimoto similarity of uniquely generated Murcko scaffolds to the seeding Murcko scaffold. The physchem-based (PCB) model generates SMILES that correspond to new scaffolds whereas the fingerprint-based (FPB) model generates scaffolds that are more similar or even identical to the seeding scaffold. **b**, Predicted active probability of all unique structures behind all generated SMILES strings per model. Both models generate SMILES that are predicted to be active with similar probability distributions.



Extended Data Fig. 3 | Novelty of uniquely generated underlying molecules with respect to different datasets. Novelty is assessed with respect to the train and test ChEMBL datasets using the physchem-based (PCB) and fingerprint-based (FPB) models. The first element of every pair on the x-axis corresponds to the dataset the conditions were drawn from. The second element represents the dataset with respect to which novelty was calculated. For any model the difference between datasets is insignificant, reflecting a consistent generation of novel compounds regardless of the seeding conditions. The numbers correspond to the fraction of valid unique novel molecules out of 25,600 generated SMILES strings.



Extended Data Fig. 4 | Optimization of properties individually in every direction with the physchem-based model. The pattern of the molecular properties of the generated valid SMILES (blue dots) seems to follow the set conditions (red lines). The length of a step represents the number of valid SMILES for that setpoint out of 256 sampled SMILES strings. Low molecular weight or high QED setpoints lead to unstable generation of valid SMILES for the given condition. QED displays the largest deviations from the seed conditions and is the hardest property to control as the formula contains a weighted sum of the other five properties. The area annotated by arrows refers to an input combination with a high QED target that caused the output to collapse with respect to the rate of valid SMILES and the fulfillment of the specified conditions. The exact percentage of unique molecules stemming from all valid SMILES sampled at each step is shown in Supplementary Fig. 12.