ARTICLES
https://doi.org/10.1038/s42256-021-00403-1
nature machine intelligence

Multi-constraint molecular generation based on conditional transformer, knowledge distillation and reinforcement learning

Jike Wang[1,2,3,7], Chang-Yu Hsieh[4,7], Mingyang Wang[1,7], Xiaorui Wang[5], Zhenxing Wu[1], Dejun Jiang[1], Benben Liao[4], Xujun Zhang[1], Bo Yang[1], Qiaojun He[1], Dongsheng Cao[6] ✉, Xi Chen[2,3] ✉ and Tingjun Hou[1,3] ✉

Machine learning-based generative models can generate novel molecules with desirable physiochemical and pharmacological properties from scratch. Many excellent generative models have been proposed, but multi-objective optimizations in molecular generative tasks are still quite challenging for most existing models. Here we proposed the multi-constraint molecular generation (MCMG) approach that can satisfy multiple constraints by combining conditional transformer and reinforcement learning algorithms through knowledge distillation. A conditional transformer was used to train a molecular generative model by efficiently learning and incorporating the structure–property relations into a biased generative process. A knowledge distillation model was then employed to reduce the model's complexity so that it can be efficiently fine-tuned by reinforcement learning and enhance the structural diversity of the generated molecules. As demonstrated by a set of comprehensive benchmarks, MCMG is a highly effective approach to traverse large and complex chemical space in search of novel compounds that satisfy multiple property constraints.

The development of bioactive compounds aided by machine learning predictions has emerged as a key component in many modern drug discovery campaigns. Machine learning-based de novo drug design approaches (including molecular generation) are quite attractive if they can accelerate the discovery and/or optimization of novel ligands with desirable therapeutic effects. This lofty goal of automatic design is still a number of years away from being a mainstream practice. The main obstacle is that drug design is an inherently multi-constrained optimization process[1–12]; for instance, a lead compound must manifest strong and specific binding towards one or multiple intended targets, high drug-likeness, low toxicity and so on. On top of these expectations, we further require a machine learning model to maximize the diversity and novelty of its output (that is, more diverse and novel molecular structures). These stringent and sometimes conflicting demands entail continual developments of new techniques to further improve the success rates of drug discovery programs.

Since the first application of an autoencoder in 2016, officially published in 2018[10,13], various machine learning-based generative models have been developed and customized for de novo design in the past few years. Many encouraging results have convincingly demonstrated the potential of these approaches[14–16]. Several machine learning frameworks and models—such as sequence-based recurrent neural networks (RNNs)[17–26], variational autoencoders (VAEs)[13–15,27–34], reinforcement learning (RL) and generative adversarial networks[16,35–39]—were found to be particularly effective at delivering promising lead compounds; however, the multi-objective nature of drug discovery demands more than what could be provided by these earlier models. To cope with the challenging multi-constrained optimization tasks, a new generation of models have been proposed in which conditional tokens were introduced into model training to build conditional generative models[27,33,40,41], applying Bayesian optimization[42] (and similar search strategies incorporating the mechanism of exploitation-exploration tradeoff) after training to find the desired regions in the latent space from the trained generative models such as VAE[13,28,32], or using RL and generative adversarial networks[26,39,43–45] to fine-tune a model to induce a bias into the output distribution[14,25,26,36–39].

The fundamental difficulty in directly generating bioactive molecules can essentially be understood by the analogy of finding a needle in a haystack. These bioactive molecules not only represent an almost negligible portion of the accessible chemical space, but also form small clusters that are sparsely scattered around different corners in the space. As alluded earlier, the challenge is further exacerbated by additional requirements such as structural diversity and novelty. These conditions collectively imply that an ideal model should efficiently traverse across distant regions in the chemical space. Reinforcement learning can be used to fine-tune the parameters of generative models to guide the free parameter space towards a set with an optimal objective function value such as bioactivity.

Although RL (guided by the feedback from a QSAR model or other predictive methods) has been confirmed capable of finding a highly diverse set of bioactive molecules, it still takes a substantial amount of optimization steps to learn these patterns via training with rewards. To alleviate the low efficiency of RL, Blaschke et al.[46] used transfer learning to quickly focus on certain regions in the chemical

[1]Innovation Institute for Artificial Intelligence in Medicine of Zhejiang University, College of Pharmaceutical Sciences, Zhejiang University, Zhejiang, P. R. China. [2]School of Computer Science, Wuhan University, Wuhan, P. R. China. [3]State Key Lab of CAD&CG, Zhejiang University, Zhejiang, P. R. China. [4]Tencent Quantum Laboratory, Tencent, Shenzhen, P. R. China. [5]College of Chemistry and Chemical Engineering, Lanzhou University, Lanzhou, P. R. China. [6]Xiangya School of Pharmaceutical Sciences, Central South University, Changsha, P. R. China. [7]These authors contributed equally: Jike Wang, Chang-Yu Hsieh, Mingyang Wang. ✉e-mail: oriental-cds@163.com; robertcx@whu.edu.cn; tingjunhou@zju.edu.cn

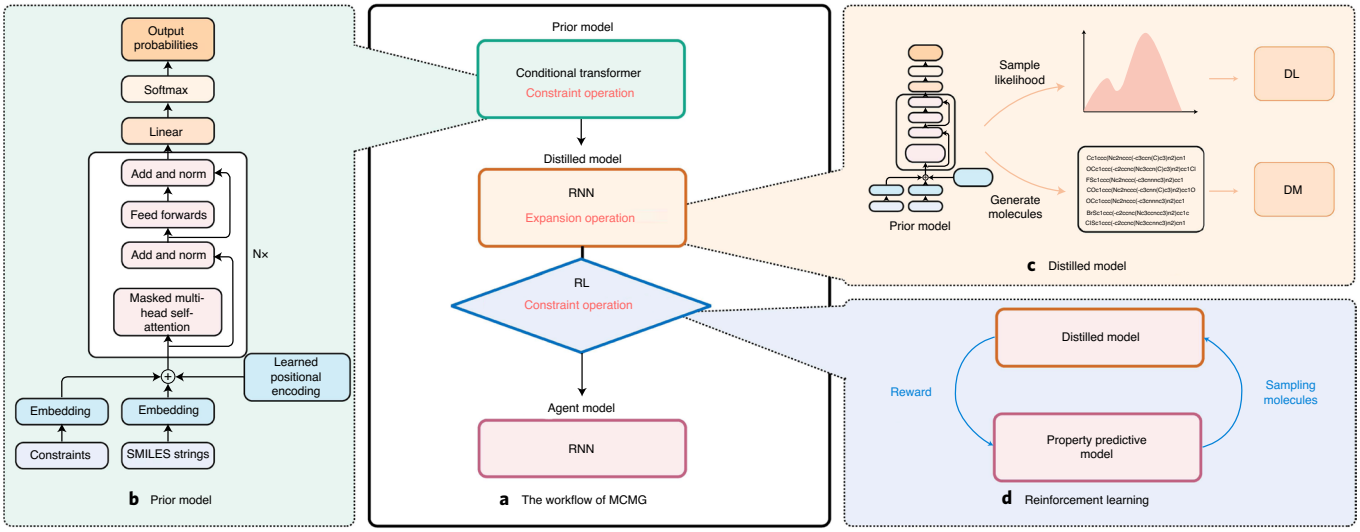

Fig. 1 | The architecture of MCMG. a, The workflow of the MCMG approach. **b**, The architecture of the prior model. **c**, The training process of two different distilled methods. **d**, The optimization process of RL.

space if one has access to an example set of bioactive compounds. Before the RL optimization process, they used the RNN model as a pretrained model and then retrained it with known active molecules. Under this framework, the output distribution of the model has already been biased towards the given set of desired molecules before RL handles the job of fine-tuning. In this case, as expected, RL usually identifies suitable spots with specific requirement of molecular properties in the chemical space under less training; however, certain tradeoffs can take place depending on how transfer learning is conducted. A potential side effect is a significant contraction of the accessible chemical space, makes the model settle into a local optimum, implying that the generated molecules tend to be more similar to the compounds in the transfer learning training set.

We set out to address the following challenge: enhance the efficiency of a molecular generative model to output desirable molecules by preconditioning the generative model without compromising its output diversity in a multi-constrained task. To this end, we proposed a novel molecular generative method named multi-constraint molecular generation (MCMG). First, a conditional transformer[47] was employed to build the generative model due to its superior performance in natural language processing. Molecules can be represented by the simplified molecular-input line-entry system (SMILES), which is regarded as a chemical language; a conditional transformer can therefore be used to generate SMILES naturally. A knowledge distillation model was then employed to reduce the model's complexity so it could be efficiently fine-tuned by RL, which is originally proposed to transfer the knowledge learned from a large model or an ensemble of multiple models to another lightweight model for fast deployment. This distillation method can also greatly improve the structural diversity of the generated molecules. We thoroughly investigated and benchmarked MCMG under two evaluation schemes. MCMG illustrates superior capability in multi-task constrained generation tasks. Compared with existing baselines, MCMG achieved excellent performances in all experiments, and under the same conditions it could generate more molecules with desirable properties. Furthermore, the generated molecules have higher structural diversity and possess more types of scaffolds.

Results and discussion

MCMG approach. As shown in Fig. 1a, MCMG consists of three essential submodels. The first model is an initial-stage model called the prior model (Fig. 1b). The second model is a distilled model based on RNN (Fig. 1c), with which we tried two different knowledge distillation methods: distilled likelihood (DL) and distilled molecules (DM). The last is an agent model fine-tuned by RL (Fig. 1d). We imposed the first constraint in the prior model through introducing a conditional token. For the conditional model, the sampling distribution over the reconstructed chemical space often highly peaks around a few regions. To mitigate this problem, we employed a customized model-distillation protocol that can smear out the distribution and yield more diversity and novelty. Finally, we used RL to readjust the model again in the hope to discover more suitable regions in the chemical space that can be either overlooked (more like cold spots overshadowed by hot spots if we metaphorically view conditioned chemical space as a heat map) or not fully captured by the conditional transformer.

The advantages of a conditional transformer over conditional RNNs. In view of the superior performance of transformers in the field of natural language processing, it was adopted as the prior model. Indeed, we verified that conditional transformers are able to better capture and utilize the structure–property relations for the generative tasks than conditional RNNs. To validate the higher quality of the generated molecules, we built a conditional RNN (c-RNN) by adding conditional tokens into training just like our conditional transformer (c-transformer). Our first task is to compare the RNN, c-RNN and c-transformer to determine which model is more suitable to be used as the focused prior model.

Let us illustrate our findings using task 2 as an example. We first trained and sampled 5,000 molecules from each model, and then applied the MOSES evaluation metrics for a comprehensive benchmark as given in the first three rows of Table 1. We found that the unconditioned RNN cannot generate molecules that possess all of the four constraints simultaneously. If an RL algorithm was directly used to fine-tune this RNN-based molecular generative model, the RL agent will encounter difficulty to bias the probability distribution towards the set of desired molecules. On the other hand, the RL agent should have an easier time to fine-tune preconditioned prior models.

As clearly shown in Table 1, compared with the c-RNN, the c-transformer has a huge lead in validity and success, whereas the c-transformer and c-RNN are essentially tied for the other four evaluation metrics (that is, SNN, Frag, Novelty and IntDiv). The

Table 1 | The MOSES evaluation metrics of the molecules generated by c-transformer, c-RNN, prior model and distilled models

Model	Validity↑[a]	SNN↓	Frag↑	Novelty↑	IntDiv↑	Success↑
RNN	**0.969**	0.489	**0.998**	0.965	**0.876**	0%
c-RNN	0.809	**0.403**	0.823	**0.995**	0.839	9.80%
c-Transformer (Prior)	**0.904**	**0.427**	0.774	0.993	0.830	**25.4%**
DL	0.813	**0.432**	0.605	**0.993**	0.718	**23.6%**
DM	**0.886**	0.434	**0.832**	0.983	**0.835**	4.90%

[a]↑Represents the higher the better, whereas ↓ represents the lower the better. Please see the Methods for definitions. Bold text indicates the best result.

c-transformer performs better overall for conditional molecular generations. Transformers do not rely on the past hidden states to capture dependencies on previous words but instead process a sentence as a whole to allow parallel computing, decrease training time and reduce performance degradation due to long-term dependencies; the c-transformer can therefore capture more information than the c-RNN. The generated molecular distributions for the four property constraints are separately presented in the four different panels in Supplementary Fig. 1.

The performances of the distilled models. We next analysed the two distinct approaches (DL and DM) to distill the focused prior model. Similarly, 5,000 molecules were sampled by each model for assessments. The last three rows of Table 1 list the MOSES evaluation metrics for the molecules generated by the two distilled models and the prior model. It is clear that DL performs almost identically to the prior model in every aspect. This is not surprising as the DL model is supposed to strictly learn the likelihood function defined in equation (3).

The DM model, however, behaves much differently. It suffers a steep drop in the success rate for the generated molecules to satisfy all four constraints, but this does not imply that the DM model fails to pick up the preferred molecules (just not that many to satisfy the four constraints simultaneously). Although the success rate of DM is lower, two other important properties (that is, Frag and IntDiv) are considerably improved. The distributions of the four properties for the molecules generated by the prior model and two distilled models are displayed in Supplementary Fig. 2.

The distribution of the distilled models. We found that the distribution of the molecules generated by the DM model is distinct from those generated by the other aforementioned models. In fact, we confirmed that the DM model reconstructs a much larger chemical space than the DL model. We then followed the studies reported by Blaschke et al.[46] and Tripp et al.[48] to draw similar figures to visualize the chemical space. As shown in Fig. 2a, we assume that the entire ellipse is an unconditioned chemical space, the blue space is the reconstructed chemical space for DL, the green space is the reconstructed chemical space for DM, and the red dots represent the regions containing the desired molecules. It can be seen that the blue space is much more compact than the green space. As the density of the red points in the blue space is high, it would be easy for RL to optimize the DL model and reach a local optimum in few steps. The green space is relatively large, whereas the presence of the red dots is not so dense (compared with the blue space), but the overall density of the red dots is still relatively higher than the red dots distributed over the entire unconditional space. Hence, RL can also efficiently fine-tune the DM model to identify promising regions in the larger green space. The much broader chemical space supports higher molecular diversity.

To corroborate the above conjecture, we sampled 5,000 molecules and evaluated the average negative log-likelihood (NLL) for each of the following models: RNN, DM, DL and semi-DM. A smaller averaged NLL implies less randomness in the generated sequence of SMILES, which implies less variability and more restricted chemical space on the reconstruction by the generative model. As shown in Fig. 2d, the NLL of the DL distribution peaks at around the lower end of the spectrum, so its reconstructed chemical space is the smallest, and the trend is followed by DM, semi-DM and RNN. We will further elaborate on the benefits brought by the DM and semi-DM models with their larger reconstructed chemical spaces in the following experiments.

Evaluation setting 1. We first reported the benchmark studies for two experimental tasks under the evaluation setting 1 (experiment 1), which is to build and save an optimal molecular generative model. This process is illustrated by Fig. 2b. The goal of this evaluation setting is to use RL to find the best area in the chemical space; the size and position of the area can be adjusted by RL.

We presented the performance of the best optimized instance (in terms of the highest success rate during the training) for each model. More precisely, Table 2 and Supplementary Table 1 report the evaluations on the 5,000 molecules generated by each model for tasks 1 and 2, respectively. For task 1, we found that all models perform extremely well for the success rate (above 90%) except REINVENT; however, the relatively underperforming REINVENT achieves a solid result with a success rate of 79.46%. Although the best-performing MCMG-likelihood (MCMGL) reaches a success rate of nearly 100%, the model suffers a problem that is revealed by the novelty metrics in the upper section of Table 2. We found that REINVENT 2.0 also suffers from a similar problem.

It is explained in the "Benchmark" section in the Methods that novelty (in the conditional metrics) is given by the comparison of the similarity between the successful molecules and the positive molecules (satisfying all the four constraints) in the training set. As the DRD2 dataset contains a large number of active molecules (2,665), all models encounter difficulty proposing molecules that are sufficiently distinct from the existing actives; however, the problem is particularly severe for REINVENT2.0 and MCMGL. Due to the use of transfer learning, REINVENT2.0 generates molecules that are highly similar to the molecules in the training set and result in low novelty. In terms of Div, REINVENT2.0 and MCMGL perform similarly while underperforming the other three benchmarked models. Although the success rates for both REINVENT2.0 and MCMGL are quite high, their poor performance against novelty implies that very limited practical advantages can be expected from these models.

Due to this observation, real success—defined as the percentage of the generated unique successful molecules that meet these constraints—seems to be a more convincing metric to assess the usefulness of a molecular generative model. The difference between success and real success provides another view on the problem of MCMGL and REINVENT2.0.

In task 2 we included a few more baselines for comparison. Task 2 is much more challenging than task 1. As shown in Supplementary Table 1, all models used in task 1 experience a decline in their success

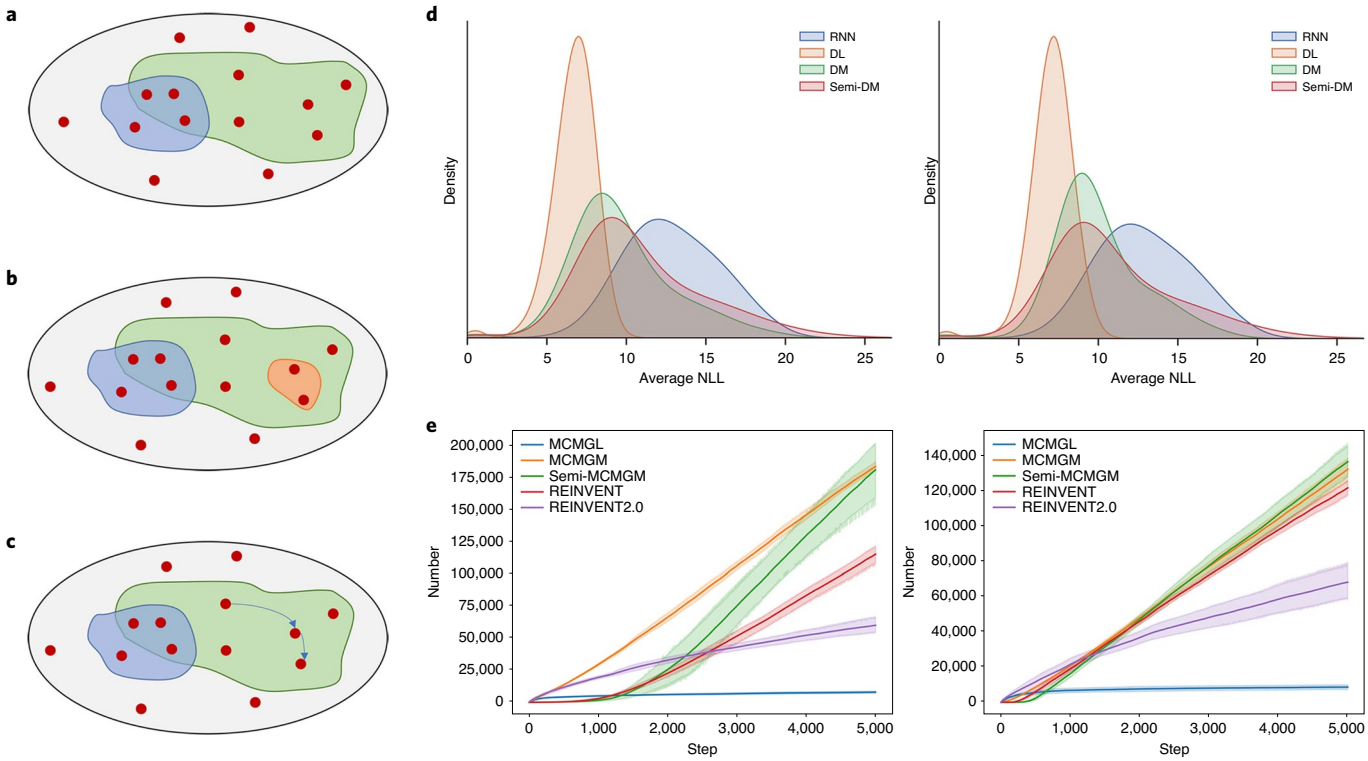

Fig. 2 | Illustration of chemical space and results of evaluation setting 1. a, Chemical space of DL (blue) and DM (green). The red area is the desirable area containing the molecules with desirable properties. **b**, An illustration of evaluation setting 1, which aims to find the area (orange) to maximize the density of the desirable area (red). **c**, An illustration of evaluation setting 2, which aims to collect the desirable molecules during the optimization process. **d**, The distribution of the average NLL of sampling from models. Left: the distribution of task 1. Right: the distribution of task 2. **e**, The relationship between the number of generated unique successful molecules and the number of optimized steps in tasks 1 (left) and 2 (right). The *x*-axis represents the number of the RL optimization steps, whereas the *y*-axis represents the number of the unique successful molecules.

Table 2 | The conditional and MOSES evaluation metrics for the successful molecules generated by the models for task 1 of experiment 1

	Models	Reinvent	Reinvent2.0	MCMGL	MCMGM	Semi-MCMGM
Conditional metrics	Success↑	79.46%	91.10%	**99.62%**	91.83%	94.18%
	Novelty↑	23.9%	0.2%	17.1%	25.3%	**39.6%**
	Div↑	0.718	0.559	0.587	0.751	**0.714**
	Real success↑	72.8%	72.6%	15.2%	**89.26%**	82.69%
MOSES metrics	Unique↑	0.916	0.797	0.153	**0.972**	0.878
	Frag↑	**0.448**	0.394	0.046	0.334	0.214
	SNN↓	0.559	0.570	**0.505**	0.541	0.525
	IntDiv↑	0.620	0.517	0.479	**0.668**	0.624
	Novelty↑	**0.993**	0.987	0.975	0.992	0.992

rates. REINVENT undergoes the steepest drop in the success rate and we hypothesized that the active regions (with respect to both targets in task 2) are much smaller and/or sparsely scattered; hence, it is much more challenging for an RL agent to identify these spots without any initial bias instilled in the prior probability distribution over the chemical space. Compared with task 1, the success rates of REINVENT and semi-MCMGM therefore drop by 26.6% and 15.6%, respectively. For RationaleRL, Div is the highest, which is closely related to the way that generates molecules. RationaleRL generates molecules by splicing on the active fragments, which often produces unreasonable structures that are very hard to be synthesized (Supplementary Table 2). The distributions of the properties of the successful molecules generated by the models for tasks 1 and 2 in experiment 1 can be found in Supplementary Figs. 3 and 4.

The results summarized in Table 2 and Supplementary Table 1 clearly demonstrate that the proposed MCMG methods have made some solid progress in the development of multiconditional generative models for de novo drug design. We would like to draw attention in particular to the performance of semi-MCMGM model that is ranked second. The fact that semi-MCMGM yields highly competitive performance is non-trivial. Semi-MCMGM allows us to work with target-independent preconditioning (for instance, QED (quantitative estimate of drug-likeness) and SA (synthetic accessibility score) in this study, but other properties such as ADMET

Table 3 | The conditional and MOSES evaluation metrics for the successful molecules generated by the models in experiment 2

		Models	Reinvent	Reinvent2.0	MCMGL	MCMGM	Semi-MCMGM
Task 1	Conditional metrics	Novelty↑[a]	5.0%	0.1%	10.8%	20.9%	**28.7%**
		Div↑	0.702	0.570	0.768	0.785	**0.792**
	MOSES metrics	Unique↑	0.903	0.468	0.105	0.876	**0.934**
		Frag↑	**0.415**	0.443	0.183	0.316	0.328
		SNN↓	0.541	0.589	0..622	0.573	**0.537**
		IntDiv↑	0.653	0.517	0.622	0.708	**0.722**
		Novelty↑	**0.994**	0.983	0.889	0.978	0.992
Task 2	Conditional metrics	Novelty↑[a]	64.5%	26.8%	36.2%	**71.4%**	70.3%
		Div↑	0.671	0.620	0.630	**0.686**	0.679
	MOSES metrics	Unique↑	0.841	0.533	0.085	0.852	**0.856**
		Frag↑	0.552	0.507	0.514	**0.577**	0.551
		SNN↓	**0.376**	0.393	0.397	0.382	**0.376**
		IntDiv↑	0.612	0.563	0.343	**0.626**	0.619
		Novelty↑	**1.000**	**1.000**	0.999	**1.000**	**1.000**

properties need to also be considered in the future) and still get outstanding performances against prior arts.

Evaluation setting 2. The evaluation setting 2 (experiment 2) is not to focus on getting an optimal model but rather collecting useful molecules during the RL-assisted fine-tuning stage (Fig. 2c). The training of each model terminated at 5,000th step, regardless of model convergence. All of the successful molecules generated in these 5,000 steps were collected for each model, and 10,000 molecules were randomly selected from the pool as the representatives. This representative set of 10,000 molecules was subsequently evaluated to infer the model's performance.

Table 3 shows how models perform (according to the evaluation setting 2) for the two identical tasks in the previous section. In task 1, we observed a similar pattern of performances under the evaluation setting 1. REINVENT2.0 and MCMGL especially score extremely low on novelty in the conditional metrics. As for Div, semi-MCMGM performs extremely well, close to 0.8, while REINVENT2.0 scores less than 0.6. According to the MOSES metrics, the performance for both REINVENT2.0 and MCMGL scores lowly on the uniqueness indicator, implying a large number of duplicated molecules are generated. Overall, in task 1, semi-MCMGM achieves the most impressive performance, exceling in five out of the seven metrics. In task 2, the novelty values for all models are higher than those for the models in task 1, which is consistent with the observation for task 1. In task 2, MCMGM achieves the best performance on five metrics. On the whole, the MCMG models perform better than the baselines.

We next analysed the cumulative number of the unique successful molecules with respect to the number of the RL optimization steps. As shown in Fig. 2e, at the beginning, MCMGM, MCMGL and REINVENT2.0 quickly accumulate a large number of successful molecules, whereas REINVENT and semi-MCMGM only start to pick up the pace at around the 1,000th step. The reason is that REINVENT and semi-MCMGM are sampled from the molecules from a much larger chemical space (without being subjected to a focused subspace either due to transfer learning or pre-conditioning), and RL needs much more iterations to learn traits of desired molecules. For MCMGL, after it outputs a certain number of unique and successful molecules, no matter how many more steps it is further trained, it can hardly generate more unique and successful molecules, and REINVENT2.0 also encounters the same problem. Hence, the cumulative curves reach a plateau for both models, as shown in Fig. 2e. This is because the chemical spaces reconstructed by these two models are small, and the number of the legitimate (that is, successful) molecules is low, and RL will easily settle into a local optimum, so that many successful molecules end up being repeatedly generated. As for the other three models bearing much larger chemical spaces, their cumulative numbers continue to increase with the training steps. Especially, for semi-MCMGM, the cumulative number of successful molecules continue to rise steadily after the 1,000th step. REINVENT also gives a similar curve in Fig. 2e, but the growth rate is not as high as that for semi-MCMGM. Among all of the models, MCMGM offers the most maximum number of unique and successful molecules, maintaining a very high growth rate over the entire training course. The right panel of Fig. 2e describes the cumulative trend for task 2. The curves for most models manifest similar trends as task 1. The distributions of the properties of the successful molecules generated by the models for tasks 1 and 2 in experiment 2 are displayed in Supplementary Figs. 5 and 6.

Scaffold analysis. In this part, we performed scaffold analysis on 10,000 molecules used for the assessments under the evaluation setting 2. The Murcko scaffolds[49] of all of the molecules were extracted and compared with those of the known active compounds, and the similarity between them was calculated. Supplementary Table 3 summarizes the number of the scaffolds whose similarity is less than or equal to a threshold value, and the average of the scaffold similarity was recorded in the last row. For task 1, semi-MCMGM holds an absolute advantage over the other methods for the scaffold novelty. For semi-MCMGM, we found more than 1,000 distinct scaffolds with a maximum similarity less than or equal to 0.4, and an average similarity of only 0.446. For task 2, the results are similar, whereas MCMGM and semi-MCMGM show an advantage in the novelty of the scaffolds of the generated molecules. To better demonstrate that MGMGM and semi-MCMGM have better capability to generate molecules with unique scaffolds, an example for the scaffolds with the similarity $\leq$0.2 (we call it top differential scaffolds) in task 1 is visualized in Fig. 3. The visualization results illustrate that our model can generate molecules that are significantly different from the real active molecules.

RL exploration. Our study is dedicated to the initial state of the generative model. In fact, the use of different RL objectives may have significant impact on the resulting data. It can be seen that both of MCMGL

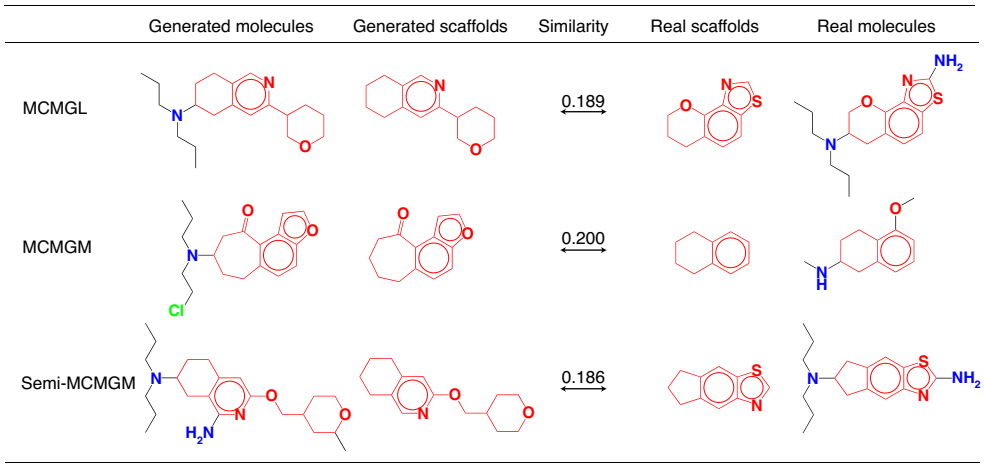

Fig. 3 | The top differential scaffold results generated by the three model configurations in task 1. The first column is the molecules generated by MGMG, the second column is the scaffolds of the above molecule, the third column is the similarity between the scaffold of the generated molecules and the scaffold of the real molecules, the fourth column is the scaffolds of the molecule in the real activity dataset, and the fifth column is the real activity molecules.

and REINVENT2.0 fall into local optima, which is caused by the too small chemical space restricted by the model before RL. Different RL strategies can be used according to different tasks to make the model avoid falling into local optima. Blaschke et al.[50] proposed a very good method named diversity filter to solve this problem, and we call this strategy that can avoid local optima as RL-exploration mode. To verify the effectiveness of MCMG, we performed the same experiments for RL-exploration mode (please see Supplementary Tables 4 and 5 for detailed results). Overall, MCMG-series achieved better performance in different evaluation settings or different modes.

Conclusion

We proposed the MCMG approach by encompassing a conditional transformer, a knowledge-distilled RNN and an RL training. We made solid progress on the challenging problem of balancing the convergence speed and output diversity for a molecular generative model. We benchmarked several versions of MCMG under two evaluation settings and many common tasks to ensure that our models offer practical advantages under various scenarios. Under the evaluation setting 1, compared with other models, MCMG achieves the highest success rate on two different tasks (DRD2, QED and SA; JNK3, GSK3β, QED and SA) with 94.18% and 80.2% versus 91.10% (REINVENT2.0) and 78.0% (RationleRL). The real success rate for MCMG is 89.26% and 70.9% for these two tasks, and MCMG enjoys a wide leading margin compared with the other models (72.8% for REINVENT and 51.7% for RationaleRL), suggesting that MCMG can generate more unique and successful molecules. Under the evaluation setting 2, the molecules generated by MCMG show the lowest similarity with the known active molecules, suggesting that more novel active molecules are generated for the two tasks. In conclusion, MCMG achieves promising performance in multi-objective molecular generative tasks and offers a highly effective way to traverse large and complex chemical space in search of potential drug candidates.

Methods

Datasets. The training dataset was obtained from the work of Olivecrona et al.[26], which was selected from the ChEMBL[51] dataset. The bioactivity dataset includes the experimental bioactivity data for three different protein targets, namely dopamine type 2 receptor (DRD2), c-Jun N-terminal kinase-3 (JNK3)[52] and glycogen synthase kinase-3 beta (GSK3β)[53,54]. The DRD2 dataset was provided by Olivecrona et al.[26], which contains 100 K negative and 7,219 positive compounds. The JNK3 dataset[52] contains the inhibition data for 50,000 negative and 2,665 positive compounds, whereas the GSK3β dataset[53,54] contains the inhibition data for 50,000 negative and 740 positive compounds. The JNK3 and GSK3β datasets are available from the study by Li and colleagues[41].

Model architecture of MCMG. A brief overview of the entire workflow is illustrated in Fig. 1a. A conditional transformer[47] model was first trained to conditionally generate the desired molecules with a moderate success rate, and then this prior model was distilled to a RNN to facilitate the subsequent integration with RL. A distilled RNN can not only alleviate the training burden on RL but also give an overall best-performing model in this study.

Prior model. The architecture of the prior model is provided in Fig. 1b. This model is expected to learn to generate molecules with desirable properties encoded by a set of conditional tokens. The standard transformer model was modified for molecular generation by simplifying the decoder and removing the associated multihead attention layers across the encoder–decoder interface. This modification is necessary; otherwise, we will face the risk of data leakage during the sequential generation of SMILES and compromise the learning process.

The prior model (a conditional transformer) was taught to output the SMILES characters in a sequential and autoregressive fashion. The next symbol to be generated was determined by the previously generated partial sequence, and the masked multihead self-attention layer was adopted to prevent information leakage from the tokens in the undecoded part of the sequence. The multihead self-attention layer is the core of the prior model, which is composed of several scaled dot-product (multiplicative) attention functions and facilitates the model capture key information in a sequence. The attention mechanism formula can be succinctly described as follows:

$$\text{Attention}\,(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

where Q, K, V represent query, key and value matrix, respectively; d_k is the dimension of K.

As the attention layers attend to the entire sequence at once, an additional positional encoding should be attached to each token to restore the notion of a linear sequence. For better performance in this study, we did not use the standard sinusoidal positional encoding when choosing the learned positional embedding scheme introduced by Vaswani et al.[47].

As this prior model is expected to output molecules with desirable properties, we added a set of extra tokens (as a constraint code c) to distinguish the appropriate molecules from the rest in the training dataset. The constraint code c is an n-bit string indicating whether a molecule satisfies each of n constraints or not (that is, 1 or 0 for each property). Hence, the prior model learns a joint embedding of the constraints and SMILES during the training process. Aside from the processing of the constraint codes, the rest of the training process is similar to a standard seq2seq training. In short, one defines a beginning token and an end token, GO and EOS. GO + SMILES are taken as the input and SMILES + EOS as the target for training the model to encode and decode proper molecules with the SMILES syntax.

More precisely, the prior model was trained to maximize the following negative log-likelihood with a given c:

$$L(D) = -\sum_{i=1}^{n} \log p\,(x_i | x_{<i}, c) \quad (2)$$

During the inference stage, the prior model was expected to generate molecules according to a learnt conditional probability distribution $p(x|c)$:

$$p(x|c) = \prod_{i=1}^{n} p(x_i|x_{<i}, c) \quad (3)$$

Distilled model. As mentioned in the introduction, it is difficult to optimize a transformer via an RL algorithm, and the reconstructed chemical space of such a conditional transformer is too concentrated, which will create the potential problem of trapping into a local optimum. In response to this challenge, we proposed to distill the prior model into a more compact architecture[55], which retained only the knowledge related to the conditioned chemical space (that is, potential regions with desired molecules). In this study, we investigated two approaches of knowledge distillations.

The first option is to build an RNN with three-layer gated recurrent units[56] to learn the likelihood of a subset of the molecules sampled from the prior model. We randomly sampled 128 SMILES from the prior model in each step to train the RNN. The training process is as follows. First, the molecules from the prior model conditioned on a given constraint token were sampled. Second, they were input together with token GO and the constraint code c (for example, as "high QED") as the starting tokens into the RNN sequentially until the end token EOS was reached or the maximum allowed length was reached. Finally, the generated SMILES strings were taken as the training set, and the likelihood of the prior generated in this process was used as the target to train the distilled model. The distilled model can thus learn the likelihood distribution of desired molecules as given by the prior model. The distilled model and the final model (RL fine-tuned) trained in this manner are named DL and MCMGL, respectively.

The second option is to directly use the conditional transformer model to generate a dataset of one million desired molecules (given a set of appropriate conditional tokens), which was then used to train an RNN with the same structure described above. Under this training, the distill model cannot learn the exact likelihood function from the prior model and the set of generated molecules cannot be used to infer the exact likelihood function due to the randomness induced by the temperature effect at the decoding part of the generative process. Hence, the distilled model affords the RL algorithm a less-focused chemical space to explore, and the benefit of this training procedure will be explained in a later section. We name the distilled model and the final model trained in this manner as DM and MCMG-molecules (MCMGM), respectively. A recap of these two distinct approaches is summarized in Fig. 1c.

Finally, we also proposed two different models, semi-DM and semi-MCMGM. In this case, the prior model was trained to generate molecules that satisfy only a subset of the desired properties. For instance, in our benchmark studies below, we attempted to only introduce computationally cheap and reliable labels such as QED and SA scores, and refrained from labeling molecules with predicted bioactivity. The rest of the training procedure remains the same as that for MCMGM.

Agent model. It is difficult for RL to train a neural network to perform a specific task from scratch; instead, it is more feasible to use RL to fine-tune a pretrained neural network. For instance, RNNs have been fine-tuned by RL to generate specific contents[57]. In this case, we adopted the RL algorithm in REINVENT to fine-tune the distilled model introduced earlier, and built a customized reward function for multiple objectives commonly required for molecular generations in drug design.

We summarize the training process for the agent model, which assumes exactly the same RNN architecture as the distilled model introduced above. First, the agent model still generated SMILES in a character-by-character fashion. Along the course, the generation of each token was regarded as an action under the RL framework. The agent model should go through the episodes of molecular generation to eventually learn a conditional probability $p(A|S)$ of possible actions to be taken when the decoded partial sequence is found in the state S. The RL framework can only train an agent model if proper rewards for an action, A, are specified by the environment.

Finally, the loss function for the RL training was defined as follows: we sampled the SMILES strings from the agent model and recorded the likelihood as log $p(A)_{\text{agent}}$. The generated SMILES strings were then input into the distilled model to estimate the likelihood log $p(A)_{\text{middle}}$. The score $S(A)$ for each generated molecule was weighted by a coefficient and then added to the likelihood log $p(A)_{\text{middle}}$ to get the augmented likelihood:

$$\log p(A)_{\text{aug}} = \log p(A)_{\text{distilled}} + \sigma S(A) \quad (4)$$

Hence the loss function is:

$$\text{Loss} = [\log p(A)_{\text{aug}} - \log p(A)_{\text{agent}}]^2 \quad (5)$$

DRD2, JNK3 and GSK3β prediction models. The inhibition strengths for JNK3 and GSK3β were predicted by the predictive models reported by of Jin and co-workers[58], which were established by random forest[59] using the Morgan fingerprints (ECFP4)[60] based on a dataset from Li and colleagues[41]; we achieved AUROC scores of 0.86 and 0.86 for JNK3 and GSK3β, respectively. The DRD2 prediction model was obtained from the work reported by Olivecrona and colleagues[26], which is a support vector machine classifier with a Gaussian kernel built by scikit-learn[61].

Evaluation settings. Two evaluation settings were used to assess the quality of multi-conditional generative models. The first is adopted by Jin et al.[58], and the goal is to build and save an optimal molecular generative model, which can be reused in later stages. Removal of repeated optimizations from scratch will greatly benefit scientific researchers. The second setting is not to focus on getting an optimal model but rather collecting useful molecules during the RL-assisted fine-tuning stage as done in the REINVENT[26] and REINVENT2.0[46] frameworks. The rationale is that the ultimate objective is to get a set of qualified generated molecules, and the focus should be placed on the appropriate detail.

The above two evaluation settings require different experimental schemes. The focus of the first is to assess the ability of the best optimized model (ability herein refers to the proportion of the desirable molecules generated, and the quality of the molecules generated by the best optimized model according to some typical indicators such as novelty and diversity). The second focuses on a model's efficiency to generate a fixed number of desirable molecules under the same conditions and the quality of the generated molecules.

We conducted two different multi-objective optimization experiments under both settings using the scoring function $S(m)$ reported by Jin and co-workers[58], The QED and SA scores for each molecule were calculated by RDKit[62], and the inhibition strengths for DRD2, JNK3 and GSK3β were predicted by the prediction models. The specific details are outlined below.

Task 1: DRD2, QED and SA.

In this task, our goal is to generate molecules with DRD2 activity higher than or equal to 0.5, QED higher than or equal to 0.6, and SA less than or equal to 4. The SA score is calculated by the equation (6), and the scoring function $S(m)$ of task 1 is shown in the equation (7):

$$\text{SAscore} = \text{FragmentScore} - \text{ComplexityPenalty} \quad (6)$$

$$S(m) = \text{DRD2}(m) + \text{QED}(m) + \text{SA}(m) \quad (7)$$

Task 2: JNK3, GSK3β, QED and SA.

In this task, our goal is to generate molecules with DRD2 activity higher than or equal to 0.5, QED higher than or equal to 0.6, and SA score less than or equal to 4. The $S(m)$ of task 2 is calculated by the following equation:

$$S(m) = \text{JNK3}(m) + \text{GSK3}\beta(m) + \text{QED}(m) + \text{SA}(m) \quad (8)$$

Baselines and training setting. Experiment 1. We compared our method with JT-VAE, GCPN, RationaleRL, REINVENT and REINVENT2.0 in task 2. It should be pointed out that REINVENT2.0 in this article specifically refers to REINVENT2.0, which uses transfer learning and RL mode. The RationaleRL and REINVENT models were trained according to the process described in Jin and colleagues's study[58]. It should be noted that because the qualities of the molecules generated by JT-VAE and GCPN are very low and the training sets used in Jin's study and our study are the same, we did not repeat the experiment but directly used the results reported by Jin et al.[58]. As RationaleRL, which was retrained on the new dataset, requires a lot of computing resources and also human intervention, we did not test it in task 1 and the subsequent experiments. The performance of JT-VAE and GCPN are very poor, so we did not test them in the subsequent experiments.

In the process of the RL optimization in experiment 1, MCMG, REINVENT and REINVENT2.0 were fine-tuned by 5,000 steps at a learning rate of 0.0001, and RationaleRL was fine-tuned by 100 steps at the default learning rate of 0.0005. As for MCMG, REINVENT and REINVENT2.0, we saved the models every 500 steps during the RL optimization process, run the RL optimization process five times and tested the model with the highest Success evaluation metric.

Experiment 2. In the two tasks in experiment 2, we used REINVENT and REINVENT2.0 as the baselines. MCMG, REINVENT and REINVENT2.0 were fine-tuned by 5,000 steps at a learning rate of 0.0001, and all of the successful molecules generated during the RL optimization process were saved. The RL optimization processes were run a total of five times.

Benchmark. Two distinct sets of evaluation metrics were used in our study. One is the standard metrics proposed by Jin et al.[58], and the other is the well-established MOSES[63] evaluation metrics which is the main benchmark in the field of de novo molecular generation.

Conditional evaluation metrics. Success: In this study, a proposed molecule is regarded to be successful (or useful) when it satisfies these properties: QED ≥ 0.6, SA ≤ 4.0, and the predicted inhibition strengths of JNK3 and GSK3β ≥ 0.5. The percentage of the generated molecules that meet these constraints is the success rate. This item is only used in experiment 1.

Real success: the percentage of the generated unique molecules that meet these constraints. This item is only used in experiment 1.

Diversity: calculated based on the Tanimoto distance sim(X, Y) with respect to the Morgan fingerprints of a pair of successful molecules, as per the following equation:

$$\text{Diversity} = 1 - \frac{2}{n(n-1)} \sum_{X,Y} \text{sim}\,(X, Y) \tag{9}$$

Novelty: the comparison of the similarity between the successful molecules and the positive molecules (satisfying all four constraints) in the training set. For each successful molecule G, the nearest neighbor molecule G_{SNN} is selected from the positive molecules in the training set. In short, novelty is defined as follows:

$$\text{Novelty} = \frac{1}{n} \sum_{G} 1\left[\text{sim}\,(G, G_{\text{SNN}} < 0.4)\right]. \tag{10}$$

MOSES evaluation metrics. The MOSES evaluation metrics assess the quality of the generated molecule dataset G based on training set T:

Validity: the proportion of the valid molecules in the generated molecules.

Uniqueness: the proportion of the unique structures generated.

Novelty: the proportion of molecules in G but not in T:

$$\text{Novel}\,(G) = 1 - \frac{|\text{set}(G \cap T)|}{G} \tag{11}$$

Internal diversity (IntDiv): the measure of the diversity for G by calculating the average Tanimoto coefficient (TC) in the generated molecular set:

$$\text{IntDiv}\,(G) = 1 - \frac{1}{|\text{set}(G)|^2} \sum_{(a,b) \in \text{set}(G)} TC\,(m_a, m_b) \tag{12}$$

Fragment similarity (Frag): used to measure how frequently various molecular fragments appear in the training and test sets. The frequency of fragments is calculated using the BRICS function in RDKit:

$$\text{Frag}\,(G, T) = 1 - \cos(f_G, f_T) \tag{13}$$

Nearest neighbor similarity (SNN): SNN is defined as the average Tanimoto similarity between a molecule m_G from G and its nearest neighbor molecule m_T in T:

$$\text{SNN}\,(G, T) = \frac{1}{|G|} \max T\,(m_G, m_T) \tag{14}$$

Data availability

The training dataset was obtained from the study reported by Olivecrona et al.[26], which selected the data from the ChEMBL[51] dataset. The bioactivity dataset includes the experimental bioactivity data for three different protein targets, namely DRD2, JNK3 and GSK3β. The DRD2 dataset was provided by Olivecrona et al.[26], which contains 100,000 negative and 7,219 positive compounds. The JNK3 dataset[52] contains the inhibition data for 50,000 negative and 2,665 positive compounds, whereas the GSK3β dataset[53,54] contains the inhibition data for 50,000 negative and 740 positive compounds. The JNK3 and GSK3β datasets are available from the study of Li and colleagues[41].

Code availability

The code used in the study is publicly available from the GitHub repository: https://github.com/jkwang93/MCMG (ref. [64]).

Received: 9 March 2021; Accepted: 9 September 2021;
Published online: 18 October 2021

References

1. 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. Design Eng.* **4**, 828–849 (2019).
2. 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).
3. Chen, H. & Engkvist, O. Has drug design augmented by artificial intelligence become a reality? *Trends Pharmacol. Sci.* **40**, 806–809 (2019).
4. Ekins, S. et al. Exploiting machine learning for end-to-end drug discovery and development. *Nat. Mater.* **18**, 435–441 (2019).
5. Mater, A. C. & Coote, M. L. Deep learning in chemistry. *J. Chem. Inf. Model.* **59**, 2545–2559 (2019).
6. Jørgensen, P. B., Schmidt, M. N. & Winther, O. Deep generative models for molecular science. *Mol. Inf.* **37**, 1700133 (2018).
7. Yang, X., Wang, Y., Byrne, R., Schneider, G. & Yang, S. Concepts of artificial intelligence for computer-assisted drug discovery. *Chem. Rev.* **119**, 10520–10594 (2019).
8. Hessler, G. & Baringhaus, K.-H. Artificial intelligence in drug design. *Molecules* **23**, 2520 (2018).
9. Batool, M., Ahmad, B. & Choi, S. A structure-based drug discovery paradigm. *Int. J. Mol. Sci.* **20**, 2783 (2019).
10. Xu, Y. et al. Deep learning for molecular generation. *Future Med. Chem.* **11**, 567–597 (2019).
11. Button, A., Merk, D., Hiss, J. A. & Schneider, G. Automated de novo molecular design by hybrid machine intelligence and rule-driven chemical synthesis. *Nat. Mach. Intell.* **1**, 307–315 (2019).
12. Moret, M., Friedrich, L., Grisoni, F., Merk, D. & Schneider, G. Generative molecular design in low data regimes. *Nat. Mach. Intell.* **2**, 171–180 (2020).
13. Gómez-Bombarelli, R. et al. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Sci.* **4**, 268–276 (2018).
14. Zhavoronkov, A. et al. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nat. Biotechnol.* **37**, 1038–1040 (2019).
15. Polykovskiy, D. et al. Entangled conditional adversarial autoencoder for de novo drug discovery. *Mol. Pharmaceutics* **15**, 4398–4405 (2018).
16. Putin, E. et al. Adversarial threshold neural computer for molecular de novo design. *Mol. Pharm.* **15**, 4386–4397 (2018).
17. Bjerrum, E. J. & Threlfall, R. Molecular generation with recurrent neural networks (RNNs). Preprint at https://arxiv.org/abs/1705.04612 (2017).
18. Gupta, A. et al. Generative recurrent networks for de novo drug design. *Mol. Inf.* **37**, 1700111 (2018).
19. Pogány, P., Arad, N., Genway, S. & Pickett, S. D. De novo molecule design by translating from reduced graphs to SMILES. *J. Chem. Inf. Model.* **59**, 1136–1146 (2019).
20. Liu, X., Ye, K., van Vlijmen, H. W. T., Ijzerman, A. P. & van Westen, G. J. P. An exploration strategy improves the diversity of de novo ligands using deep reinforcement learning: a case for the adenosine A2A receptor. *J. Cheminf.* **11**, 35 (2019).
21. Segler, M. H. S., Kogej, T., Tyrchan, C. & Waller, M. P. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Sci.* **4**, 120–131 (2018).
22. Yang, X., Zhang, J., Yoshizoe, K., Terayama, K. & Tsuda, K. ChemTS: an efficient python library for de novo molecular generation. *Sci. Technol. Adv. Mater.* **18**, 972–976 (2017).
23. Grisoni, F., Moret, M., Lingwood, R. & Schneider, G. Bidirectional molecule generation with recurrent neural networks. *J. Chem. Inf. Model.* **60**, 1175–1183 (2020).
24. Merk, D., Friedrich, L., Grisoni, F. & Schneider, G. De novo design of bioactive small molecules by artificial intelligence. *Mol. Inf.* **37**, 1700153 (2018).
25. Popova, M., Isayev, O. & Tropsha, A. Deep reinforcement learning for de novo drug design. *Sci. Adv.* **4**, eaap7885 (2018).
26. Olivecrona, M., Blaschke, T., Engkvist, O. & Chen, H. Molecular de-novo design through deep reinforcement learning. *J. Cheminf.* **9**, 48 (2017).
27. Lim, J., Ryu, S., Kim, J. W. & Kim, W. Y. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *J. Cheminf.* **10**, 31 (2018).
28. Kusner, M. J., Paige, B. & Hernández-Lobato, J. M. in *Proc. 34th International Conference on Machine Learning* Vol. 70. (eds. Doina, P. & Yee Whye, T.) 1945–1954 (PMLR, 2017).
29. Liu, Q., Allamanis, M., Brockschmidt, M. & Gaunt, A. L. in *Proc. 32nd International Conference on Neural Information Processing Systems* 7806–7815 (Curran Associates Inc., 2018).
30. Simonovsky, M. & Komodakis, N. in *International Conference on Artificial Neural Networks* 412–422 (Springer, 2018).
31. Bjerrum, E. J. & Sattarov, B. Improving chemical autoencoder latent space and molecular de novo generation diversity with heteroencoders. *Biomolecules* **8**, 131 (2018).
32. Jin, W., Barzilay, R. & Jaakkola, T. in *Proc. 35th International Conference on Machine Learning* Vol. 80. (eds. Jennifer, D. & Andreas, K.) 2323–2332 (PMLR, 2018).
33. Kang, S. & Cho, K. Conditional molecular design with deep generative models. *J. Chem. Inf. Model.* **59**, 43–52 (2019).
34. Kingma, D. P. & Welling, M. Auto-encoding variational Bayes. Preprint at https://arxiv.org/abs/1312.6114 (2014).
35. Kadurin, A., Nikolenko, S., Khrabrov, K., Aliper, A. & Zhavoronkov, A. druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Mol. Pharmaceutics* **14**, 3098–3104 (2017).
36. Sanchez-Lengeling, B., Outeiral, C., Guimaraes, G. L. & Aspuru-Guzik, A. Optimizing distributions over molecular space. An objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC). Preprint at *ChemRxiv* https://doi.org/10.26434/chemrxiv.5309668.v3 (2017).
37. Guimaraes, G. L., Sanchez-Lengeling, B., Farias, P. L. C. & Aspuru-Guzik, A. Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. Preprint at https://arxiv.org/abs/1705.10843 (2017).

38. Putin, E. et al. Reinforced adversarial neural computer for de novo molecular design. *J. Chem. Inf. Model.* **58**, 1194–1204 (2018).
39. Yu, L., Zhang, W., Wang, J. & Yu, Y. in *Proc. 31st AAAI Conference on Artificial Intelligence* 2852–2858 (AAAI Press, 2017).
40. Sohn, K., Yan, X. & Lee, H. in *Proc. 28th International Conference on Neural Information Processing Systems* Vol. 2, 3483–3491 (MIT Press, 2015).
41. You, J., Liu, B., Ying, Z., Pande, V. & Leskovec, J. in *Advances in Neural Information Processing Systems* 6410–6421 (2018).
42. Brochu, E., Cora, V. M. & Freitas, N. d. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Preprint at https://arxiv.org/abs//1012.2599 (2010).
43. Cao, N. D. & Kipf, T. MolGAN: an implicit generative model for small molecular graphs. Preprint at https://arxiv.org/abs/1805.11973 (2018).
44. Jaques, N. et al. in *Proc. 34th International Conference on Machine Learning* Vol. 70, 1645–1654 (JMLR.org, 2017).
45. Sutton, R. S. & Barto, A. G. *Introduction to Reinforcement Learning* (MIT Press, 1998).
46. Blaschke, T. et al. REINVENT 2.0: an AI tool for de novo drug design. *J. Chem. Inf. Model.* **60**, 5918–5922 (2020).
47. Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Proc. Syst.* **30**, 5998–6008 (2017).
48. Tripp, A., Daxberger, E. & Hernández-Lobato, J. M. in *Advances in Neural Information Processing Systems* 11259–11272 (2020).
49. Bemis, G. W. & Murcko, M. A. The properties of known drugs. 1. Molecular frameworks. *J. Med. Chem.* **39**, 2887–2893 (1996).
50. Blaschke, T., Engkvist, O., Bajorath, J. & Chen, H. Memory-assisted reinforcement learning for diverse molecular de novo design. *J. Cheminf.* **12**, 68 (2020).
51. Anna, G. et al. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **40**, 1100–1107 (2012).
52. Ip, Y. T. & Davis, R. J. Signal transduction by the c-Jun N-terminal kinase (JNK)-from inflammation to development. *Curr. Opin. Cell Biol.* **10**, 205–219 (1998).
53. Shang, L. et al. RAGE modulates hypoxia/reoxygenation injury in adult murine cardiomyocytes via JNK and GSK-3 beta signaling pathways. *PLoS ONE* **5**, e10092 (2010).
54. Tanabe, K. et al. Glucose and fatty acids synergize to promote B-cell apoptosis through activation of glycogen synthase kinase 3 beta independent of JNK activation. *PLoS ONE* **6**, e18146 (2011).
55. Hinton, G., Vinyals, O. & Dean, J. Distilling the knowledge in a neural network. *Computer Sci.* **14**, 38–39 (2015).
56. Cho, K. et al. Learning phrase representations using RNN Encoder decoder for statistical machine translation. Preprint at https://arxiv.org/abs/1406.1078 (2014).
57. Jaques, N., Gu, S., Turner, R. E. & Eck, D. Tuning recurrent neural networks with reinforcement learning. Preprint at https://arxiv.org/abs/1611.02796v1 (2017).
58. Jin, W., Barzilay, R. & Jaakkola, T. Composing molecules with multiple property constraints. Preprint at https://arxiv.org/abs/2002.03244v1 (2020).
59. Breiman, L. Random forests. *Mach. Learn.* **45**, 5–32 (2001).
60. David, R. & Mathew, H. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **50**, 742–754 (2010).
61. Pedregosa, F. et al. Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
62. Freeze, J. G., Kelly, H. R. & Batista, V. S. Search for catalysts by inverse design: artificial intelligence, mountain climbers, and alchemists. *Chem. Rev.* **119**, 6595–6612 (2019).
63. Polykovskiy, D. et al. Molecular Sets (MOSES): a benchmarking platform for molecular generation models. *Front. Pharmacol.* **11**, 565644 (2020).
64. Wang J. et al. *Code Repository jkwang93/MCMG: v1.1.0* (Zenodo, 2021); https://doi.org/10.5281/zenodo.5205570

Acknowledgements

We want to thank Z. Liu for insightful discussion on this study. This work was financially supported by National Key R&D Program of China (grant no. 2016YFA0501701), National Natural Science Foundation of China (grant no. 81773632), Natural Science Foundation of Zhejiang Province (grant no. LZ19H300001), Key R&D Program of Zhejiang Province (grant no. 2020C03010), and Fundamental Research Funds for the Central Universities (grant no. 2020QNA7003).

Author contributions

T.J.H., C.Y.H., D.S.C. and X.C. designed the research study. J.K.W. developed the method and wrote the code. J.K.W., M.Y.W., X.R.W., D.J.J., B.B.L., X.J.Z. B.Y. and Q.J.H. performed the analysis. J.K.W., M.Y.W., T.J.H. and C.Y.H. wrote the paper. All authors read and approved the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-021-00403-1.

Correspondence and requests for materials should be addressed to Dongsheng Cao, Xi Chen or Tingjun Hou.

Peer review information *Nature Machine Intelligence* thanks J.B. Brown, Jose Jimenez-Luna, and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.