DEEP GENERATIVE MODELS FOR CONDITIONED MOLECULE GENERATION

Chia Yong Hui[1], Esther Toh Jia Ling[2], Goh Enxi Xanthe[3], Shen Bingquan[4], Lim Jing[4]

[1]NUS High School of Mathematics and Science, 20 Clementi Avenue 1, Singapore 129957
[2]Nanyang Girls' High School, 2 Linden Dr, Singapore 288683
[3]Raffles Girls' School, 2 Braddell Rise, Singapore 318871
[4]DSO National Laboratories, 12 Science Park Drive, Singapore 118225

ABSTRACT

Deep Generative Models (DGMs) have shown significant promise in drug discovery by enabling efficient exploration of the vast chemical space. Traditional drug design approaches face limitations due to the complexity and size of this space. In this study, we propose a multi-objective optimization approach using a transformer-based architecture to generate drug-like molecules with desired properties, such as quantitative drug-likeness (QED), logarithm of partition coefficient (LogP), and synthetic accessibility score (SAS). We employ reinforcement learning (RL) with Proximal Policy Optimization (PPO) and Genetic Algorithm to improve performance. Our results demonstrate the model's ability to produce drug-like molecules with the desired properties, showcasing its potential to accelerate the drug discovery process.

BACKGROUND AND PURPOSE OF RESEARCH

The process of drug discovery has been a major challenge and has not only increased in duration, taking generally 12-15 years, but also in risk and cost. [1] Despite advancements in various aspects of the drug development and discovery process, the success rate remains low, at only 10%–15%. [2]

In recent years, efforts have been made to apply deep learning, especially generative models, to the drug design field. Nevertheless, generating valid and desired molecular structures remains a difficult task, primarily due to the vast chemical space and the limited availability of labelled data. Methods like those by Gomez et al. still struggled to achieve high success rates because of the complex grammatical rules of SMILES (Simplified molecular- input line-entry system) strings. [3]

Despite these challenges, advancements in the field of machine learning have made deep generative models (DGMs) a promising solution to improve efficiency in the discovery process. They have the potential to save time and cost of at least 25 to 50% in the discovery and pre-clinical stages. [2,4] Furthermore, DGMs can utilize cheminformatics tools to eliminate invalid molecules, thereby focusing efforts on high-potential compounds. This capability streamlines early drug discovery and enables more effective use of resources.

Recently, there has also been increasing focus on reinforcement learning approaches. Nevertheless, these methods have their limitations, in scalability, data efficiency, and robustness. [6,7] The typical Q-learning algorithm applies only to discrete action and state

space, leading to inefficient learning. On the other hand, recently developed, PPO demonstrates relatively high potential in overcoming these challenges. It has the stability and reliability of trust-region methods and is simpler to implement. [6]

Additionally, while machine learning methods present impressive results, traditional approaches like GAs remain effective for constrained optimization. GAs, which mimic natural selection and mutation, can outperform ML techniques in constrained tasks [8,9].

HYPOTHESIS

In this paper, we propose a multi-objective optimization approach that combines a transformer-based architecture with reinforcement learning, followed by constrained optimization. The model is designed to optimize QED, LogP, and SAS values, thereby generating molecules with favourable chemical properties. Using SMILES string representations, the transformer learns the underlying chemical rules that define valid molecules. Reinforcement learning, specifically PPO, is then applied to refine the generation process, ensuring that the molecules align better with the desired property objectives. We also experimented with using GA to refine the outputs by maintaining the desired properties while exploring different molecular modifications via mutations and crossovers.

METHODOLOGY AND RESULTS

Pre-Training with Transformer Model

Our model utilizes a transformer-based architecture, which is commonly used for autoregressive language modelling, to generate SMILES strings with desired molecular properties, such as QED, SAS and LogP. The properties are encoded into a 3D vector via linear transformation with each dimension representing one of the properties, and is used as the input of the model. The core of the model consists of 6 stacked transformer encoder-decoder blocks. Each block is composed of a multi-head self-attention mechanism and a feed-forward network. The final output is then passed through a linear layer to project it into the vocabulary space. This generates a distribution over possible tokens, from which the next token is sampled.

To train the model, we used a subset of the Zinc dataset that consists of 250k molecules. During training, the model receives the molecule properties as condition vectors as the input and the SMILES strings as target output. The model uses teacher forcing where the target SMILES is initially masked, and the model is required to predict the next token in the SMILES. The next token of the actual SMILES will then be revealed, and the model will have to predict the next token again. This repeats until the full SMILES is generated. Generation is done in an autoregressive manner, one token at a time, conditioning each new token on the previously generated token and condition vectors. Top-p sampling is used, where tokens are selected based on their cumulative probability:

$$\sum_{x \in T_p} P(x_i | x_{1:i-1}, c) \geq p \tag{1}$$

where $P(x_i|x_{1:i-1}, c)$ is the probability of token x_i conditioned on the sequence generated so far $x_{1:i-1}$ and the condition vector c. T_p is the subset of tokens from vocabulary where the cumulative probability exceeds p, which we have set to be 0.9.

We compared our transformer model with an LSTM model, which is also a highly effective model for handling sequential data. We utilized the following metrics to evaluate the effectiveness of the models:

1. Validity: the percentage of valid molecules the model is able to generate, which is evaluated by RDKit
2. Diversity: the percentage of unique molecules that the model can generate
3. Novelty*: the percentage of molecules generated that are not in the training dataset
4. $rmse_{QED}$*: the root mean squared error (RMSE) of QED values (quantifying the drug-likeness of molecules based on eight common molecular properties: molecular weight, AlogP, hydrogen bond donors, hydrogen bond acceptors, polar surface area, rotatable bonds, aromatic rings, and structural alerts) of the generated molecules relative to the target QED
5. $rmse_{SAS}$*: the RMSE of SAS values (an estimation of molecular synthesis feasibility based on fragment scores and complexity) of the generated molecules relative to the target SAS.
6. $rmse_{LogP}$*: the RMSE of LogP values (logarithm of the partition coefficient, measuring hydrophobicity) of the generated molecules relative to the target LogP.

*Novelty, $rmse_{QED}$, $rmse_{SAS}$ and $rmse_{LogP}$ metrics were calculated with the exclusion of invalid molecules.

We randomly sampled specific property value combinations from the dataset to ensure that the values represent plausible and meaningful targets and generated 1000 molecules to evaluate.

	Validity	Diversity	Novelty	$rmse_{QED}$	$rmse_{SAS}$	$rmse_{LogP}$
LSTM	96.5%	**97.8%**	**99.8%**	0.133	0.205	0.541
Transformer	**99.6%**	81.8%	99.6%	**0.070**	**0.146**	**0.289**

Table 1.1. Comparison of results for main metrics between LSTM and transformer, for target QED: 0.7, SAS: 2.0 and LogP: 5.0

	Validity	Diversity	Novelty	$rmse_{QED}$	$rmse_{SAS}$	$rmse_{LogP}$
LSTM	97.3%	**92.4%**	**100.0%**	0.141	0.524	1.339
Transformer	**97.8%**	85.1%	99.9%	**0.079**	**0.383**	**0.708**

Table 1.2. Comparison of results for main metrics between LSTM and transformer, for target QED: 0.55, SAS: 5.0 and LogP: -1.6

The results from Tables 1.1 – 1.3 highlight that both models are capable of generating molecules of high validity and novelty. Additionally, while the LSTM model performs better in generating unique molecules, the RMSE scores for the transformer was lower.

Due to the low diversity, we decided to incorporate Byte Pair Encoding(BPE). Instead of tokenising at the character level, BPE identifies and encodes common structures in the dataset as tokens. Each token would then represent a more meaningful group of atoms and bonds, allowing for enhanced exploration of the chemical space and thereby generating more unique molecules.

	Validity	Diversity	Novelty	$rmse_{QED}$	$rmse_{SAS}$	$rmse_{LogP}$
Single Char	**99.6%**	81.8%	**99.6%**	**0.070**	0.146	**0.289**
BPE	97.9%	**98.4%**	99.1%	0.160	**0.139**	0.453

Table 2.1. Comparison of results for main metrics for different tokenizer approaches, for target QED: 0.7, SAS: 2.0 and LogP: 5.0

	Validity	Diversity	Novelty	$rmse_{QED}$	$rmse_{SAS}$	$rmse_{LogP}$
Single Char	**97.8%**	85.1%	**99.9%**	**0.079**	0.383	**0.708**
BPE	93.7%	**91.9%**	99.5%	0.109	**0.346**	0.737

Table 2.2. Comparison of results for main metrics for different tokenizer approaches, for target QED: 0.55, SAS: 5.0 and LogP: -1.6

The metrics presented in Tables 2.1 to 2.2 demonstrate that the implementation of BPE improves diversity, as well as $rmse_{SAS}$ score slightly.

Reinforcement Learning

After training the base model, we implemented PPO to fine-tune the model to be able to generate molecules with properties closer to the specified targets and achieve lower RMSE scores.

For each episode, we use the pre-trained model to generate SMILES strings and for each generated molecule, the reward function evaluates how closely its properties match the targets, with the properties assigned different weights:

$$reward_t = 0.45 \cdot reward_{QED} + 0.15 \cdot reward_{SAS} + 0.40 \cdot reward_{LogP} \quad (2)$$

where,

$reward_{QED} = 1 - |QED_{gen} - QED_{target}|$

$reward_{SAS} = 1 - |SAS_{gen} - SAS_{target}|$

$reward_{LogP} = 1 - |LogP_{gen} - LogP_{target}|$

The rewards are used in the calculation of return R_t is calculated for each episode and is then used to calculate advantage A_t. The PPO updates the policy $\pi_\theta(a|s)$, parameterized by θ, using the clipped surrogate objective:

$$L^{CLIP}(\theta) = \bar{E}_t[\min(r_t(\theta) \cdot A_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon) \cdot A_t] \quad (4)$$

where,

$\bar{E}_t$ denotes the expectation over timesteps
ϵ represents the clipping range that is set to 0.2
$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{old}(a_t|s_t)}$ denotes the probability ratio comparing the new policy to the old policy,
where a_t denotes the action taken at time t and s_t denotes the state at time t

In equation 7, the clipped term ensures that the policy update remains within the bound of ϵ , and the prevents excessively large policy updates, improving training stability.

	Validity	Diversity	Novelty	$rmse_{QED}$	$rmse_{SAS}$	$rmse_{LogP}$
Pre-PPO	**97.9%**	98.4%	99.1%	**0.160**	0.139	0.453
Post-PPO	97.6%	**99.1%**	**99.5%**	0.161	**0.136**	**0.403**

Table 3.1. Comparison of results for main metrics before and after RL, for target QED: 0.7, SAS: 2.0 and LogP: 5.0

	Validity	Diversity	Novelty	$rmse_{QED}$	$rmse_{SAS}$	$rmse_{LogP}$
Pre-PPO	**93.7%**	**91.9%**	99.5%	0.109	0.346	**0.737**
Post-PPO	91.7%	90.4%	99.5%	**0.080**	**0.308**	0.740

Table 3.2. Comparison of results for main metrics before and after RL, for target QED: 0.55, SAS: 5.0 and LogP: -1.6

The results in Tables 3.1 – 3.2 show that RL can enhance the performance of the model, with slight improvements in most of the metric scores.

Genetic Algorithm

We applied GA after generating molecules. A combination of mutations, fragmentation and crossover was applied on the generated molecules, such that there is a balance between conservative edits and disruptive edits. The mutations include the following:

1. Add atom: a random atom is appended and bonded with an existing atom in the molecule
2. Insert atom: inserts a new atom between two randomly chosen atoms
3. Change atom: replaces an existing atom with a different atom
4. Remove atom: a terminal atom (bonded to only one other atom within a molecule) is removed
5. Modify bond: changes the bond type between two randomly selected atoms (eg. single bond to double bond)

6. Add ring: adds a new ring structure and bonds the new ring with a random existing atom in the molecule
7. Delete cyclic bond: removes a randomly selected bond that is part of a ring in the molecule

To implement crossovers, fragmentation was first implemented:

1. Ring Fragmentation cuts all rings bonds, producing the ring fragments
2. Non-Ring Fragmentation: Cuts all non-ring bonds, producing acyclic fragments

Then, the fragments are combines from two parent molecules to create new offsprings, where ring-based and non-ring based crossovers are randomly implemented. Each molecules undergoes 100 iterations of mutation and 30 iterations of crossover, and the properties are calculated for each new molecule, and only those that are closer to the targets are retained.

	Validity	Diversity	Novelty	$rmse_{QED}$	$rmse_{SAS}$	$rmse_{LogP}$
Pre-GA	97.6%	99.1%	99.5%	0.161	0.136	0.403
Post-GA	**100.0%**	**99.6%**	**100.0%**	**0.135**	**0.070**	**0.257**

Table 4.1. Comparison of results for main metrics before and after constrained optimization, for target QED: 0.7, SAS: 2.0 and LogP: 5.0

	Validity	Diversity	Novelty	$rmse_{QED}$	$rmse_{SAS}$	$rmse_{LogP}$
Pre-GA	91.7%	90.4%	99.5%	0.080	0.308	0.740
Post-GA	**100.0%**	**99.8%**	**100.0%**	**0.066**	**0.192**	**0.397**

Table 4.2. Comparison of results for main metrics before and after constrained optimization, for target QED: 0.55, SAS: 5.0 and LogP: -1.6

From Table 4.1, we can observe how GA can reduce the RMSE scores for the 3 properties. Therefore, GA is a suitable method in ensuring that the molecules generated are optimized for the 3 target properties.

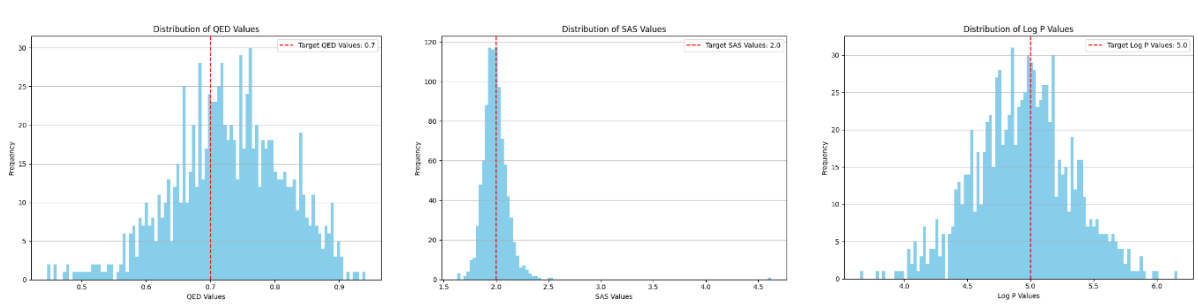


Fig 1.1. Distribution of QED Fig 1.2. Distribution of SAS Fig 1.3. Distribution of LogP

Fig 1.1 to 1.3 show the final outputs, with the graphs showing that the values of the properties are closely centred around the specified targets, especially for the SAS.

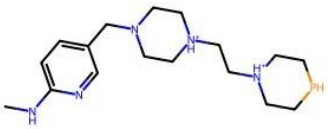
Target:	Target:	Target:
QED: 0.9, SAS: 1.5, LogP: 3	QED: 0.7, SAS: 2, LogP: 5	QED: 0.55, SAS: 5, LogP: -1.6
		
QED: 0.905, SAS: 1.495, LogP: 2.997	QED: 0.698, SAS: 1.992, LogP: 5.010	QED: 0.548, SAS: 5.027, LogP: -1.599

Fig 2. shows the top molecule generated for each target after GA.

CONCLUSION AND DISCUSSION

In this project, we addressed the challenge of de novo drug design by generating novel drug-like molecules that meet specific molecular properties, such as QED, SAS and LogP. We introduce a hybrid approach that combines a transformer-based generative model and GA. This hybrid approach allows for a balance between producing valid molecules and ensuring diversity. We also demonstrated how RL and BPE can be effective for improving the results for the autoregressive generation task.

Unlike methods that optimize for a single property, our approach evaluates multiple properties simultaneously, allowing for generation of more balanced and practically relevant molecules. However, QED account for any biological interaction or pathway that the drug may utilise and future work could explore incorporation of additional biological properties such as modelling the interactions between receptors and potential drug molecules to improve the practical relevance of the generated molecules.

REFERENCES

[1] S. M. Paul et al., “How to improve R&D productivity: the pharmaceutical industry’s grand challenge,” Nature Reviews Drug Discovery, vol. 9, no. 3, pp. 203–214, Feb. 2010, doi: https://doi.org/10.1038/nrd3078.

[2] D. Sun, W. Gao, H. Hu, and S. Zhou, “Why 90% of clinical drug development fails and how to improve it?,” *Acta Pharmaceutica Sinica B*, vol. 12, no. 7, Feb. 2022, doi: https://doi.org/10.1016/j.apsb.2022.02.002.

[3] R. Gómez-Bombarelli et al., “Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules,” ACS Central Science, vol. 4, no. 2, pp. 268–276, Jan. 2018, doi: https://doi.org/10.1021/acscentsci.7b00572.

[4] “The Drug Discovery Process: What Is It and Its Major Steps,” *blog.biobide.com*. https://blog.biobide.com/the-drug-discovery-process.

[5] “Unlocking the potential of AI in Drug Discovery Current status, barriers and future opportunities.” Available: https://cms.wellcome.org/sites/default/files/2023-06/unlocking-the-potential-of-AI-in-drug-discovery_report.pdf.

[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” *arXiv.org*, 2017. https://arxiv.org/abs/1707.06347.

[7] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, “Trust Region Policy Optimization,” arXiv.org, 2015. https://arxiv.org/abs/1502.05477.

[8] N. Yoshikawa, K. Terayama, M. Sumita, T. Homma, K. Oono, and K. Tsuda, “Population-based De Novo Molecule Generation, Using Grammatical Evolution,” Chemistry Letters, vol. 47, no. 11, pp. 1431–1434, Nov. 2018, doi: https://doi.org/10.1246/cl.180665.

[9] J. H. Jensen, “A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space,” Chemical Science, vol. 10, no. 12, pp. 3567–3572, 2019, doi: https://doi.org/10.1039/c8sc05372c.

APPENDIX

A. Results

	Validity	Diversity	Novelty	$rmse_{QED}$	$rmse_{SAS}$	$rmse_{LogP}$
LSTM	99.9%	**54.9%**	**99.3%**	0.092	0.159	0.407
Transformer	**100.0%**	22.6%	99.0%	**0.034**	**0.100**	**0.334**

Table 1.3. Comparison of results for main metrics between LSTM and transformer, for target QED: 0.9, SAS: 1.5 and LogP: 3.0

	Validity	Diversity	Novelty	$rmse_{QED}$	$rmse_{SAS}$	$rmse_{LogP}$
Single Char	**100.0%**	22.6%	**99.0%**	**0.034**	0.100	0.334
BPE	99.9%	**74.4%**	98.8%	0.041	**0.083**	**0.316**

Table 2.3. Comparison of results for main metrics for different tokenizer approaches, for target QED: 0.9, SAS: 1.5 and LogP: 3.0

	Validity	Diversity	Novelty	$rmse_{QED}$	$rmse_{SAS}$	$rmse_{LogP}$
Pre-PPO	99.9%	74.4%	**98.8%**	**0.041**	**0.083**	**0.316**
Post-PPO	99.9%	**93.5%**	98.5%	0.110	0.113	0.351

Table 3.3. Comparison of results for main metrics before and after RL, for target QED: 0.9, SAS: 1.5 and LogP: 3.0

	Validity	Diversity	Novelty	$rmse_{QED}$	$rmse_{SAS}$	$rmse_{LogP}$
Pre-GA	99.9%	93.5%	98.5%	0.110	0.113	0.351
Post-GA	**100.0%**	**97.5%**	**100.0%**	**0.090**	**0.091**	**0.206**

Table 4.3. Comparison of results for main metrics before and after constrained optimization, for target QED: 0.9, SAS: 1.5 and LogP: 3.0