Deep Reinforcement Learning Exploration in Continuous Latent Space for Molecular Design

Hsin-Jung Yang, [1] Chih-Hsuan Yang, [1] Parker Sornberger, [2] Rebekah Duke, [2] Chad Risko, [2] Baskar Ganapathysubramanian, [1] Soumik Sarkar [1]

[1]Department of Mechanical Engineering, Iowa State University
[2]Department of Chemistry and Center for Applied Energy Research, University of Kentucky

Abstract

Recently, deep reinforcement learning (DRL) has begun to emerge as a competitive alternative to generative models in automated molecular design. In this work, we proposed a DRL approach for designing molecules with desirable properties. Notably, the DRL agent traverses and explores a continuous latent space created by a variational autoencoder (VAE) and learns to exploit the space through experience. Self-Referencing Embedded Strings (SELFIES) is utilized for molecular representation to ensure the validity of the generated molecules and the adoption of the continuous latent space allows easier molecular string perturbation and navigation without manually crafting the perturbation rules. Our initial results show that the DRL agent is able to generate valid molecules with given property target.

Introduction

Molecule discovery is expensive and time-consuming owing to the vast, discrete, and unstructured nature of the chemical space. (Kirkpatrick and Ellis 2004) Still, automating molecular design with deep learning remains an appealing challenge as advances in such a field would accelerate the discovery of new materials. In the past few years, generative models (Li, Zhang, and Liu 2018; Gupta et al. 2018; Gómez-Bombarelli et al. 2018; Zhavoronkov et al. 2019) demonstrated promising results on generating new molecules with desired properties. Nevertheless, one notable limitation of these approaches is the lack of a feedback loop to improve the generated molecules beyond the original database. To address this problem, deep reinforcement learning (DRL), an algorithmic paradigm typically used for sequential decision-making, has begun to emerge as a competitive alternative (Putin et al. 2018; Zhou et al. 2019; Schreck, Coley, and Bishop 2019). DRL is often formalized as a discrete-time stochastic process in which an agent learns an optimal policy by continuously interacting with the environment, where the policy is parameterized by a deep neural network, a powerful function approximator that allows the non-linear mapping of a state to optimal decisions. This way, instead of entirely relying on existing databases, DRL expanded the capacity to learn from out of database cases such as the molecules it generates and can still be able to learn valid structural motifs based on the feedback of the environment. In this work, we propose a DRL-based method to generate molecules with desired properties. The DRL agent traverses and explores a continuous latent space created by a variational autoenccoder (VAE) (Kingma and Welling 2013) by perturbing the latent vectors and learns a policy through experience. To ensure validity, we adopted Self-Referencing Embedded Strings (SELFIES) (Krenn et al. 2019), a 100% robust string-based representation to represent molecules.

Related Works

String-based molecular representation is one of the earliest ways to represent molecules and remains very popular nowadays. In particular, the Simplified molecular-input line-entry system (SMILES) (Weininger 1988) is widely adopted among the community. Many works (Olivecrona et al. 2017; Guimaraes et al. 2017; Neil et al. 2018; Popova, Isayev, and Tropsha 2018; Putin et al. 2018) adopt the SMILES representation and use recurrent neural networks (RNN) to encode the strings of SMILES representations to properties and are optimized via different approaches such as RL, Monte-Carlo Search, or Generative Adversarial Networks (GANs). These works successfully generated molecules with given desirable properties, yet, like other literature that work with the SMILES representation, struggle massively with chemical validity. To resolve this problem, researches came up with a wide variety of approaches, with one being the Self-Referencing Embedded Strings (SELFIES) (Krenn et al. 2019). In their work, the SELFIES representation demonstrated 100% validity and is shown that it can represent all molecules, which opens the door for developing more robust de novo molecular design machine learning methods. In addition to SELFIES, researchers also turned to other forms of representations, such as graph-based molecule representations, where the atoms and bonds are represented by nodes and edges (You et al. 2018; Zhou et al. 2019; SV et al. 2022), and 3D molecule representations (Simm, Pinsler, and Hernández-Lobato 2020; Flam-Shepherd, Zhigalin, and Aspuru-Guzik 2022), which typically provide more structural information than other methods and allows geometrical constraints. Combined with RL, these methods (You et al. 2018; Zhou et al. 2019; SV et al. 2022; Simm, Pinsler, and Hernández-Lobato 2020; Flam-Shepherd, Zhigalin, and Aspuru-Guzik 2022) showcased close to or even 100% validity.

However, despite with the much improved validity situation, researchers still face another big challenge: the large, discrete, and unstructured chemical space. The vast nature of the chemical space makes it impractical for exhaustive search, and the discrete and unstructured nature often requires manually crafted perturbation rules for exploration, as (Gómez-Bombarelli et al. 2018) pointed out in their work. In the same work, they also also demonstrated that smoothed continuous latent space exhibits good predictive power as well as the ability to perform gradient-based optimization methods in the latent space. Recently, (Thiede et al. 2022) explores using the concept of curiosity to train better RL agents to generate molecules based on SELFIES representation (Krenn et al. 2019) , which has a different focus compared to this work.

Methodology

In this section, we provide a description of the DRL setup. Our objective is to use DRL to explore a continuous latent space encoded by a variational autoencoder (VAE) (Kingma and Welling 2013) and learn a policy that would generate molecules with desired properties. On a high level, the agent explores the latent space by perturbing the latent representation and develops a policy as experience accumulates. Figure 1 shows the main components at each step of exploration, which are the state s_t, action a_{t+1}, reward r_{t+1}, and policy π_t: At each step, the agent takes takes an action a_t according to its current policy π_t, and receives an updated sate of the environment s_{t+1} and reward r_{t+1}. A custom OpenAI Gym environment (Brockman et al. 2016) using RDKit (Landrum et al. 2006) integrating a VAE was created to train against the agent.

Variational Autoencoder (VAE) The VAE is designed to provide a continuous and entirely valid latent space for the RL agent, i.e., it converts a discrete molecule representation into a real-valued continuous vector, which is guaranteed to be decoded back to a molecule string that represents a valid molecule. The continuity of the space is ensured by the design of the decoder, which is a feed-forward neural network that acts as a classifier which elects the most likely symbols for each character in the molecule string. This way, every latent vector is ensured to have a corresponding SELFIES representation string. The encoder of the VAE is RNN-based and takes in a converted one-hot encoding from SELFIES and outputs the latent vector. In addition, the validity of the generated molecule strings is guaranteed with the adoption of Self Referencing Embedded Strings (SELFIES), which is a 100% robust string-based representation (Krenn et al. 2019). We trained the VAE on the Quantum Machines 9 (QM9) dataset (133k molecules) (Ruddigkeit et al. 2012; Ramakrishnan et al. 2014) and the Organic Crystals in Electronic and Light-Oriented Technologies (OCELOT) dataset (Ai et al. 2021), which has around 30k molecules. Note that owing to the design of the decoder, adjacent latent vectors might have the same SELFIES representation.

State Space The state space S of the environment is the set of all possible states s_t, which is a 500-dimensional vector defined as the immediate latent representation vector created by the agent at time t after action a_{t-1}. Since states s_t are outputs encoded by the encoder in the VAE, the upper and lower bounds of each dimension differs. Therefore, s_t is normalized such that each dimension is bounded within $[-1, 1]$.

Action Space The action space A of the environment consists of the set of all possible actions a_t, which is a 2-dimensional vector (p_t, m_t) with the p_t indicating the location of which the perturbation happens, and m_t the amount of perturbation need to be made to the selected dimension of representation vector, i.e.,

$$s_{t+1}[p_t] = s_t[p_t] + m_t$$

Since the distribution over actions is modeled by Gaussian distribution, the action space is technically not bounded, yet still, there are some actual limits by which the agent needs to abide, such as the length of the vector as well as the upper and lower bounds of each vector dimension. Therefore, similar to other Open AI Gym implementations, clipping happens if the agent generates actions that exceed these limits. The resultant clipping state vector would be the closest possible state in the latent space.

Reward Design and Zindo In order to guide the behaviour of the RL agent, we designed both intermediate and terminal rewards. These rewards are domain specific and are given to the agent based on evaluations from a evaluator based on Zindo (J.Ridley and Zerner 1973), a semi-empirical quantum chemistry method: At each step when action a_t is executed, the resultant state s_{t+1} is immediately evaluated by the evaluator, the agent then receives intermediate rewards based on the evaluation, and will be given an extra terminal reward if episode-ending criterion is met, e.g., the agent generates a molecule with desired properties. The domain specific rewards include molecule properties such as the molecular weight (MW), the highest occupied molecular orbital (HOMO), and the lowest occupied molecular orbital (LUMO). Depending on the objective of the training, the reward may be defined purely based on one of the three values or a weighted sum of any combination of the trio. In this work, we focus on optimizing the HOMO of molecules; a staged reward is given to the agent. Let the absolute value of the difference between the current HOMO value and the target HOMO value be δ, then the reward function can be written as:

$$r_t = \begin{cases} 0, \ 2.5 < \delta \\ 1, \ 1.5 < \delta < 2.5 \\ 5, \ 0.3 < \delta < 1.5 \\ 10, \ \delta < 0.3. \end{cases}$$

RL Agent Given the need for continuous action outputs as well as considering efficiency and stability, we chose the Proximal Policy Optimization (PPO) (Schulman et al. 2017) as the DRL algorithm, and adopted the PPO-Clip implementation from PFRL (Fujita et al. 2021). The PPO-Clip maintains two policy networks, $\pi_\theta(a_t|s_t)$ and $\pi_{\theta_k}(a_t|s_t)$. The $\pi_\theta(a_t|s_t)$ is the current policy in question that needs refinement, whereas $\pi_\theta(a_t|s_t)$ is the previous policy that was used to collect samples. The policy is optimized by the following equation:

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}} \, E_{s,a\sim\pi_{\theta_k}}[L(s,a,\theta_k,\theta)]$$

where

$$L(s,a,\theta_k,\theta) = min(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a), g(\epsilon, A^{\pi_{\theta_k}}))$$

$$g(\epsilon, A) = \begin{cases} (1+\epsilon)A, & A \geq 0 \\ (1-\epsilon)A, & A < 0 \end{cases}$$

and A the advantage equation, a measure of the relative advantage of an action, usually denoted as $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$. The tunable hyperparmaeter ϵ defines how far away the gradient ascent can go from the previous policy. The clipping mechanism works like a regulizer to prevent the policy to change drastically between steps.

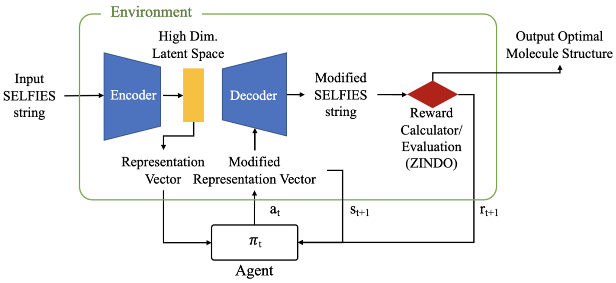

Figure 1: Overview of the framework. The environment consists of a variational autoencoder and the ZINDO property calculator. The arrows represent the flow of data.

Experiments

Experiment Setup

To test out the effectiveness of our method, we conducted experiments in the following tasks:

Validity Since many methods adopting string-based molecular representation struggle with the validity of the generated molecules, we deem that validity checks are essential as a justification of the usage of SELFIES. The validity of the molecules are performed with the sanitization check included in the RDKit (Landrum et al. 2006) as in the work of (You et al. 2018). A molecule is considered valid once it passes the sanitization check.

Molecular Design with Desired Properties In this task, we train the agent to generate molecules that has a specific molecular property targeted to a given range. Such task would be beneficial for creating a library of molecule candidates that has properties suitable for certain applications. In our case, we've chosen the highest occupied molecular orbital (HOMO) as the targeted property and given a range of -7±0.3 eV.

All experiments were performed on a machine using NVIDIA GTX 1060 GPU and Intel(R) Xeon(R) CPU E5-2620 v2 CPU. The DRL agent was trained against an environment that resets to the same initial state (a molecule with a CC(O)C(N)(CO)C#N SMILES string) at each episode. The terminal criteria of each episode include: (1) The agent comes up with a molecule with desired HOMO value, or (2) the agent exceeded 500 steps. The total number of steps across episodes during a training run is limited to 500,000.

Results and Discussion

Since as mentioned earlier, due to the nature of the latent space, some (adjacent) latent vectors may have identical SELFIES strings (repeated states), we further conducted a filtering process to the logged data to focus on the unique molecules. Of all the visited states, 325,118 of them are unique molecules.

Validity Sanitizing checks in RDKit are then performed on the 325,118 unique molecules. 325,108 of them passed, yielding a close to 100% validity. Though further investigation is still required to understand the cause that leads to the 10 molecules that failed, the high validity rate, to a large extent, justifies the use of SELFIES representation (Krenn et al. 2019).

Property Optimization Our initial results show that the agent is capable of generating molecules that meets the given criterion (-7 ± 0.3eV HOMO). Among all the visited states, around a third to a quarter of them are successful terminal states that generates a molecule with a -7 ± 0.3 eV HOMO. However, many of them are duplicated/repeated states, after some filtering, we end up with around 40 unique molecules. Figure 2 presents one of the roll outs of the DRL agent, showing the path that the agent took to optimize HOMO. It is also interesting to see that how the molecule gradually evolves through the episode to become the final molecule, as the intermediate molecules might be helpful when it comes to analyzing synthesizability. In addition, the trained DRL agent also demonstrated capacity to generate diversified molecules with similar properties, as shown in Figure 3: Starting from the initial molecule, the agent generated very different looking molecular structures that still has similar HOMO values. Note that the the HOMO values are yet to be verified by more accurate calculation methods such as DFT.

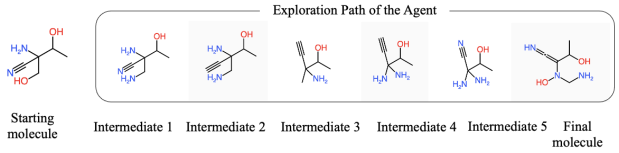

Figure 2: Schematic of a roll out: The agent begins with the initial state (leftmost), visits the intermediate states sequentially from left to right, and eventually ending at the terminal state (rightmost), which has a HOMO value within the targeted -7 ± 0.3 eV range.

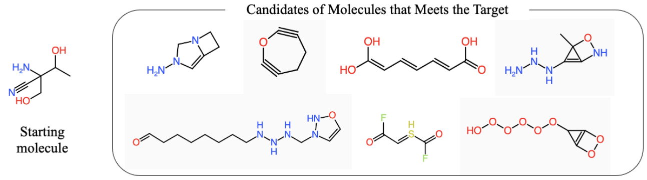

Figure 3: Demonstration of diversity. Different molecule structures that meets the same criteria from different roll outs.

Acknowledgements

This work was partly supported by the National Science Foundation under grant OIA-2019574.

References

Ai, Q.; Bhat, V.; Ryno, S. M.; Jarolimek, K.; Sornberger, P.; Smith, A.; Haley, M. M.; Anthony, J. E.; and Risko, C. 2021. OCELOT: An infrastructure for data-driven research to discover and design crystalline organic semiconductors. *The Journal of Chemical Physics*, 154(17): 174705.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. Openai gym. *arXiv preprint arXiv:1606.01540*.

Flam-Shepherd, D.; Zhigalin, A.; and Aspuru-Guzik, A. 2022. Scalable Fragment-Based 3D Molecular Design with Reinforcement Learning. *arXiv preprint arXiv:2202.00658*.

Fujita, Y.; Nagarajan, P.; Kataoka, T.; and Ishikawa, T. 2021. ChainerRL: A Deep Reinforcement Learning Library. *Journal of Machine Learning Research*, 22(77): 1–14.

Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; and Aspuru-Guzik, A. 2018. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2): 268–276.

Guimaraes, G. L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P. L. C.; and Aspuru-Guzik, A. 2017. Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models.

Gupta, A.; Müller, A. T.; Huisman, B. J.; Fuchs, J. A.; Schneider, P.; and Schneider, G. 2018. Generative recurrent networks for de novo drug design. *Molecular informatics*, 37(1-2): 1700111.

J.Ridley; and Zerner, M. 1973. An intermediate neglect of differential overlap technique for spectroscopy: Pyrrole and the azines.

Kingma, D. P.; and Welling, M. 2013. Auto-Encoding Variational Bayes.

Kirkpatrick, P.; and Ellis, C. 2004. Chemical space. *Nature*, 432: 823.

Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; and Aspuru-Guzik, A. 2019. SELFIES: a robust representation of semantically constrained graphs with an example application in chemistry. *CoRR*, abs/1905.13741.

Landrum, G.; Serizawa, T.; Nuzillard, J.-M.; and Dalke, A. 2006. RDKit: Open-source cheminformatics. http://www.rdkit.org.

Li, Y.; Zhang, L.; and Liu, Z. 2018. Multi-objective de novo drug design with conditional graph generative model. *Journal of cheminformatics*, 10(1): 1–24.

Neil, D.; Segler, M.; Guasch, L.; Ahmed, M.; Plumbley, D.; Sellwood, M.; and Brown, N. 2018. Exploring Deep Recurrent Models with Reinforcement Learning for Molecule Design.

Olivecrona, M.; Blaschke, T.; Engkvist, O.; and Chen, H. 2017. Molecular De Novo Design through Deep Reinforcement Learning.

Popova, M.; Isayev, O.; and Tropsha, A. 2018. Deep reinforcement learning for de novo drug design. *Science Advances*, 4(7): eaap7885.

Putin, E.; Asadulaev, A.; Ivanenkov, Y.; Aladinskiy, V.; Sanchez-Lengeling, B.; Aspuru-Guzik, A.; and Zhavoronkov, A. 2018. Reinforced adversarial neural computer for de novo molecular design. *Journal of chemical information and modeling*, 58(6): 1194–1204.

Ramakrishnan, R.; Dral, P. O.; Rupp, M.; and von Lilienfeld, O. A. 2014. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1.

Ruddigkeit, L.; Van Deursen, R.; Blum, L. C.; and Reymond, J.-L. 2012. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *Journal of chemical information and modeling*, 52(11): 2864–2875.

Schreck, J. S.; Coley, C. W.; and Bishop, K. J. 2019. Learning retrosynthetic planning through simulated experience. *ACS central science*, 5(6): 970–981.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Simm, G. N. C.; Pinsler, R.; and Hernández-Lobato, J. M. 2020. Reinforcement Learning for Molecular Design Guided by Quantum Mechanics.

SV, S. S.; Law, J. N.; Tripp, C. E.; Duplyakin, D.; Skordilis, E.; Biagioni, D.; Paton, R. S.; and St John, P. C. 2022. Multi-objective goal-directed optimization of de novo stable organic radicals for aqueous redox flow batteries. *Nature Machine Intelligence*, 1–11.

Thiede, L. A.; Krenn, M.; Nigam, A.; and Aspuru-Guzik, A. 2022. Curiosity in exploring chemical spaces: intrinsic rewards for molecular reinforcement learning. *Machine Learning: Science and Technology*, 3(3): 035008.

Weininger, D. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1): 31–36.

You, J.; Liu, B.; Ying, R.; Pande, V.; and Leskovec, J. 2018. Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation.

Zhavoronkov, A.; Ivanenkov, Y. A.; Aliper, A.; Veselov, M. S.; Aladinskiy, V. A.; Aladinskaya, A. V.; Terentiev,

V. A.; Polykovskiy, D. A.; Kuznetsov, M. D.; Asadulaev, A.; et al. 2019. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nature biotechnology*, 37(9): 1038–1040.

Zhou, Z.; Kearnes, S.; Li, L.; Zare, R. N.; and Riley, P. 2019. Optimization of molecules via deep reinforcement learning. *Scientific reports*, 9(1): 1–10.