Reward Conditioned Policy for Molecular Optimization

Minsu Kim **Hyungu Kang**
Korea Advanced Institute of Science and Technology (KAIST)
{min-su, gusrn0505}@kaist.ac.kr

Abstract

This paper suggests a novel deep reinforcement learning method for *de novo* molecular generation, which is a representative computational chemistry task to automatically discover a new drug with a special property. The *de novo* generation can be cast as reinforcement learning (RL), where policy explores finding high-quality molecules in the massive chemical space. The chemical space is composed of *graph* structures, where the atom is a node and chemical bonding is an edge. Therefore, combining RL with graph representation learning is a crucial technique, getting considerable attention. However, those techniques have a significant limitation where the reward function for molecule generation is very extensive and has high uncertainty. To tackle these limitations, we suggest utilizing offline pre-collected data (which is highly reliable) for *de novo* molecular generation. To this end, we propose reward conditioned policy that can generate high-quality molecule graphs leveraging offline pre-collected data, with two novel auto-encoder structures: *generator-AE*, and *plugin-CVAE*. The generator-AE is designed to reconstruct molecular graphs. The plugin-CVAE, which is a reward-conditioned variational auto-encoder is designed to reconstruct the latent vector of *generator-AE* under the condition of reward. In the test phase, our method has three following steps: (a) we input high rewards and normal distributed latent into the decoder of plugin-CVAE, (b) the latent vector (for *generator-AE*) generated from plugin-CVAE becomes input of *generator-AE* decoder, (c) finally, *generator-AE* decoder generate high-quality molecular graph. Our method has two benefits: (a) it can be used as an offline-RL method, which only utilizes pre-collected offline datasets without an online oracle, and (b) it can be used as a sample-efficient online-RL where it extends offline datasets using online exploration. Experiment results show that the proposed method makes the state-of-the-art result of penalized octanol-water partition coefficient optimization task.

1 Task Description

1. **Dataset.** We have chemical dataset **ZINC250K** [1].
2. **Oracles.** We have chemical oracles (i.e. reward metric) as panelized logp (**PlogP**:water-solubility), **QED** (toxic screening) and **GuacaMol** [2] benchmarks.
3. **Offline Setting.** We only provide oracle values for **ZINC250K** for offline training. The major purpose is to design new molecules having maximum oracle value *without online oracle calls*.

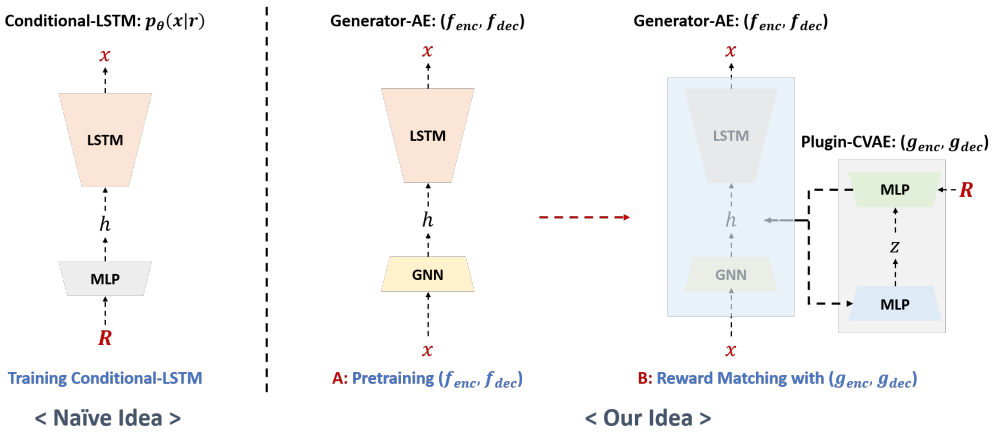


Figure 1: An overview of our idea

4. **Online Sample Efficient Setting.** We also validate the online adaptability of the offline model with *limited* oracle call (inspired by [3]). The major purpose is to design new molecules having maximum oracle value.

2 Idea Sketch

2.1 Notations

1. Molecule Graph: $\boldsymbol{x} = \{\mathcal{V}, \mathcal{E}\}$. The $\mathcal{V}$ contains atoms and $\mathcal{E}$ contains bonds between atoms.
2. Reward: $\boldsymbol{R}$.
3. Dataset for online oracle round t: $\mathcal{D}_t$. The $\mathcal{D}_0$ indicates an offline dataset.

Note that molecule graph $\boldsymbol{x}$ can be represented with a Simplified molecular-input line-entry system (SMILES) [4], which is sequential data. Therefore, we handle SMILES-LSTM [5] as generating neural networks for $\boldsymbol{x}$.

2.2 Naiive Reward Conditioned Policy with SMILES LSTM

The most representative reward conditioning method directly maps reward value into latent space such as the *decision-transformer* [6]. Therefore we suggest *conditional-LSTM* as our baseline which is reward conditioned policy with SMILES LSTM. As shown in Fig. 1, we iteratively train *conditional-LSTM*, $p_\theta(\boldsymbol{x}|\boldsymbol{R})$ as follows:

1. Samples data from offline dataset: $(\boldsymbol{x}, \boldsymbol{R}). \sim \mathcal{D}_0$.
2. Maximize likelihood to produce $\boldsymbol{x}$ for $p_\theta(\boldsymbol{x}|\boldsymbol{R})$ using cross-entropy loss.

2.3 Our Idea

Our major idea is utilizing two auto-encoding scheme: *Generator-AE* and *Plugin-CVAE*. The *Generator-AE* (f_{enc}, f_{dec}) is trained to reconstruct $\boldsymbol{x}$. The *Plugin-CVAE*, which is R conditioned variational auto-encoder (g_{enc},g_{dec}), is trained to reconstruct $\boldsymbol{h}$. As shown in Fig. 1, we train our scheme as:

1. **Phase A.**: Pretrain *Generator-AE*: $f_{dec}(f_{enc}(\boldsymbol{x})) \approx \boldsymbol{x}$ using reconstruction loss.
2. **Phase B.**: Finetunes *Plugin-CVAE*: $g_{dec}(g_{enc}(\boldsymbol{h}, \boldsymbol{R})) \approx \boldsymbol{h} = f_{enc}(\boldsymbol{x})$ using VAE loss. In this process, the *Generator-AE* is not updated.

After training, our generative model p is composited as: $p(\boldsymbol{x}|\boldsymbol{R}) = f_{dec}(g_{dec}(\boldsymbol{z}|\boldsymbol{R}))$, where $\boldsymbol{z} \sim \mathcal{N}(0, I)$.

Benefit of our model. Our model is beneficial with generalization capability with limited reward data because we designed a compact reward matching system using *Plugin-CVAE* (excluding SMILES LSTM in the reward matching process which is expensive to train).

3 Preliminary Experimental Results

3.1 Training Results of *Generator-AE*

This section gives experimental results of the **Phase A** of Section 2.3: training of *Generator-AE*. Specifically, this task aims to reconstruct a molecular graph, using a sequential decoder (i.e. SMILES LSTM). To evaluation reconstruction performances, we report two metrics: sequence accuracy and element accuracy. Let N be the number of the input molecular graphs, and $\{K_i\}_{i=1}^{N}$ be the length of each SMILES string of the corresponding molecular graph.

Sequence Accuracy. The sequence accuracy acc_{seq} is evaluated as:

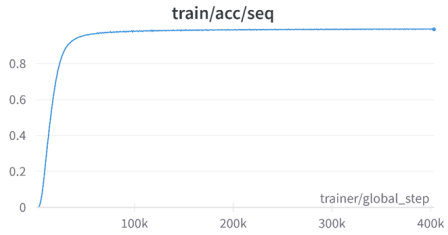
$$acc_{seq} = \frac{1}{N} \sum_{i=1}^{N} 1_{\{f_{dec}(f_{enc}(x))=x\}} \tag{1}$$

Sequence accuracy is metric to measure how many SMILES sequences were perfectly reconstructed.

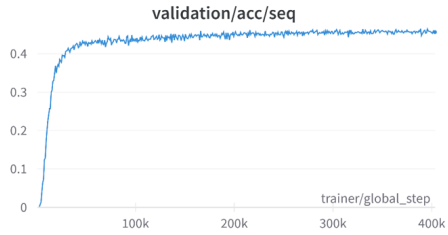
Element Accuracy. The element accuracy acc_{elem} is evaluated as:

$$acc_{elem} = \frac{1}{N} \frac{1}{\sum_i K_i} \sum_{i=1}^{N} \sum_{j=1}^{K_i} 1_{\{f_{dec}(f_{enc}(x))_j=x_j\}} \tag{2}$$

The element accuracy is a more generous metric than the sequence accuracy, where it measures the number of *element* which reconstructed successfully.

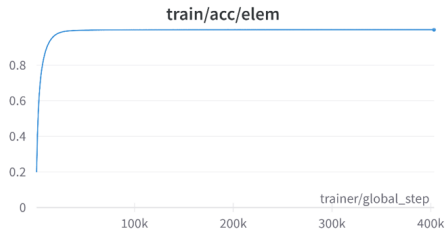


(a) Sequence accuracy on training.



(b) Sequence accuracy on validation.

Figure 2: Training and validation graph of sequence accuracy.



(a) Element accuracy on training.

(b) Element accuracy on validation.

Figure 3: Training and validation graph of element accuracy.

Reconstruction Results. As shown in Fig. 2 and Fig. 3, both sequential accuracy and element accuracy give high reconstruction quality, except the fact that the validation reconstruction only

Table 1: Performance evaluation of **PlogP**. Types of methods are indicated by deep reinforcement learning (DRL), deep embedding optimization (DEO), genetic algorithm (GA), and offline reinforcement learning (Off-RL).

Methods	Type	Objective	Num Online shot
GVAE+BO [9]	DEO	2.87	∞
SD-VAE [10]	DEO	3.50	∞
ORGAN [11]	DRL	3.52	∞
VAE+CBO [12]	DEO	4.01	∞
ChemGE [13]	GA	4.53	∞
CVAE+BO [14]	DEO	4.85	∞
JT-VAE [15]	DEO	4.90	∞
ChemTS [16]	DRL	5.60	∞
GCPN [17]	DRL	7.86	∞
MRNN [18]	DRL	8.63	∞
MolDQN [19]	DRL	11.84	∞
GraphAF [20]	DRL	12.23	∞
GB-GA [21]	GA	15.76	∞
DA-GA [22]	GA	20.72	∞
MSO [23]	DEO	26.1	∞
PGFS [24]	DRL	27.22	∞
GEGL [5]	DRL	31.40	∞
R-cond-policy-offline (ours)	Off-RL	20.10*	0
R-cond-policy-online (ours)	DRL	**77.12**	4,000

reaches about 0.5. Increasing the reconstruction accuracy is a crucial challenge for the molecule generation community, not only for the molecule optimization community; improving reconstruction accuracy may improve optimization quality also. We leave it for further research; next section, we show our framework can compete with the state-of-the-art, even though we use a poor auto-encoder model.

3.2 Performance Evaluation on *Water-solubility*

This section measures *panalized octanol-water partition coefficient* (**PlogP**) as an evaluation metric. The **PlogP** is the most widely used metric for *de novo* molecular optimization. However, several works [7, 8] have pointed out that the **PlogP** metric is ill-defined as a molecule scoring function; this metric assign a high score for **unrealistic** molecules which is unstable.

In this work, we show that the proposed method can make state-of-the-art results on **PlogP** task, only using few shot online adaptation. We compare with several online baselines where the scores are measured without considering sample efficiency (see Table 1).

Furthermore, our framework can avoid generating **unrealistic** molecules as it can control generation quality using input query of reward-conditioned policy $p(\boldsymbol{x}|\boldsymbol{R})$. Specifically, we can adjust $\boldsymbol{R}$ to get enough score **realistic** molecules to contrast with the existing online search method that just aims to maximize the score (We will experiment with this until final reports).

4 Future Direction

1. Ablation Studies.
2. Evaluating performance on various chemical metrics.
3. Evaluating sample efficiency of the offline dataset.

References

[1] John J Irwin and Brian K Shoichet. Zinc- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, 45(1):177–182, 2005.

[2] Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3):1096–1108, 2019.

[3] Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor W Coley. Sample efficiency matters: A benchmark for practical molecular optimization. *arXiv preprint arXiv:2206.12411*, 2022.

[4] David Weininger. 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, 1988.

[5] Sungsoo Ahn, Junsu Kim, Hankook Lee, and Jinwoo Shin. Guiding deep molecular optimization with genetic exploration. *Advances in neural information processing systems*, 33:12008–12021, 2020.

[6] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

[7] Wenhao Gao and Connor W Coley. The synthesizability of molecules proposed by generative models. *Journal of chemical information and modeling*, 60(12):5714–5723, 2020.

[8] Connor W Coley. Defining and exploring chemical spaces. *Trends in Chemistry*, 3(2):133–145, 2021.

[9] MJ Kusner, B Paige, and JM Hernández-Lobato. Grammar variational autoencoder. in international conference on machine learning. 2017.

[10] Hanjun Dai, Yingtao Tian, Bo Dai, Steven Skiena, and Le Song. Syntax-directed variational autoencoder for structured data. In *International Conference on Learning Representations*, 2018.

[11] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *arXiv preprint arXiv:1705.10843*, 2017.

[12] Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chemical science*, 11(2):577–586, 2020.

[13] Naruki Yoshikawa, Kei Terayama, Masato Sumita, Teruki Homma, Kenta Oono, and Koji Tsuda. Population-based de novo molecule generation, using grammatical evolution. *Chemistry Letters*, 47(11):1431–1434, 2018.

[14] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[15] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pages 2323–2332. PMLR, 2018.

[16] Xiufeng Yang, Jinzhe Zhang, Kazuki Yoshizoe, Kei Terayama, and Koji Tsuda. Chemts: an efficient python library for de novo molecular generation. *Science and technology of advanced materials*, 18(1):972–976, 2017.

[17] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in neural information processing systems*, 31, 2018.

[18] Mariya Popova, Mykhailo Shvets, Junier Oliva, and Olexandr Isayev. Molecularrnn: Generating realistic molecular graphs with optimized properties. *arXiv preprint arXiv:1905.13372*, 2019.

[19] Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific reports*, 9(1):1–10, 2019.

[20] Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382*, 2020.

[21] Jan H Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical science*, 10(12):3567–3572, 2019.

[22] AkshatKumar Nigam, Pascal Friederich, Mario Krenn, and Alán Aspuru-Guzik. Augmenting genetic algorithms with deep neural networks for exploring the chemical space. *arXiv preprint arXiv:1909.11655*, 2019.

[23] Robin Winter, Floriane Montanari, Andreas Steffen, Hans Briem, Frank Noé, and Djork-Arné Clevert. Efficient multi-objective molecular optimization in a continuous latent space. *Chemical science*, 10(34):8016–8024, 2019.

[24] Sai Krishna Gottipati, Boris Sattarov, Sufeng Niu, Yashaswi Pathak, Haoran Wei, Shengchao Liu, Simon Blackburn, Karam Thomas, Connor Coley, Jian Tang, et al. Learning to navigate the synthetically accessible chemical space using reinforcement learning. In *International Conference on Machine Learning*, pages 3668–3679. PMLR, 2020.