L-MolGAN: An improved implicit generative model for large molecular graphs

Yutaka Tsujimoto[1], Satoru Hiwa[2], Yushi Nakamura[2], Yohei Oe[2]
and Tomoyuki Hiroyasu[2 *]

[1] Graduate School of Life and Medical Sciences, Doshisha University, Kyoto, Japan

[2] Department of Biomedical Sciences and Informatics, Doshisha University, Kyoto, Japan

* Correspondence: Tomoyuki Hiroyasu (tomo@is.doshisha.ac.jp)

Abstract

Deep generative models are used to generate arbitrary molecular structures with the desired chemical properties. MolGAN is a renowned molecular generation models that uses generative adversarial networks (GANs) and reinforcement learning to generate molecular graphs in one shot. MolGAN can effectively generate a small molecular graph with nine or fewer heavy atoms. However, the graphs tend to become disconnected as the molecular size increase. This poses a challenge to drug discovery and material design, where large molecules are potentially inclusive. This study develops an improved MolGAN for large molecule generation (L-MolGAN). In this model, the connectivity of molecular graphs is evaluated by a depth-first search during the model training process. When a disconnected molecular graph is generated, L-MolGAN rewards the graph a zero score. This procedure decreases the number of disconnected graphs, and consequently increases the number of connected molecular graphs.

The effectiveness of L-MolGAN is experimentally evaluated. The size and connectivity of the molecular graphs generated with data from the ZINC-250k molecular dataset are confirmed using MolGAN as the baseline model. The model is then optimized for a quantitative estimate of drug-likeness (QED) to generate drug-like molecules. The experimental results indicate that the connectivity measure of generated molecular graphs improved by 1.96 compared with the baseline model at a larger maximum molecular size of 20 atoms. The molecules generated by L-MolGAN are evaluated in terms of multiple chemical properties, QED, synthetic accessibility, and log octanol–water partition coefficient, which are important in drug design. This result confirms that L-MolGAN can generate various drug-like molecules despite being optimized for a single property, i.e., QED. This method will contribute to the efficient discovery of new molecules of larger sizes than those being generated with the existing method.

Keywords: deep learning, generative adversarial network, graph convolutional network, molecular graph

1. Introduction

Machine learning-based molecular design of drugs is used to efficiently determine the desired molecular structure in drug discovery. It also aids the automated search for unknown molecular structures of the desired properties and predict their physical properties without requiring the domain knowledge of organic chemistry. A renowned classical molecular design model is inverse quantitative structure-activity relationship (inverse-QSAR) [1]. Based on the QSAR model—an analytical model of the relationship between molecular structure and bioactivity, formulated using molecular descriptors quantifying the features of the molecular structure—inverse-QSAR performs a backward prediction of the molecular structure from the desired bioactivity. Therefore, to obtain a molecular structure with the desired bioactivity, it is necessary to select the appropriate molecular descriptors that are equivalent to the raw data of feature engineering in machine learning. However, it is difficult to identify the descriptors correlated with the desired bioactivity from the numerous available molecular descriptors, which is a core problem in inverse-QSAR analysis.

Several molecular-structure search methods based on deep generative models, which generate new data with similar features as the original without the availability of predetermined feature vectors for the dataset, have been proposed and developed. Most adopt a graph-based approach in which the molecular structure is represented as a graph and are classified into two approaches in terms of the molecular generation process: sequential iterative process and one-shot generation [2].

In the sequential iterative process, molecules are assembled stepwise by adding atoms and bonds to a predefined scaffold. The advantage of the generative model [3–6] when combined with the sequential iterative process is the assurance of chemical validity of the generated molecules. Thus, it is possible to obtain functional molecules by reliably generating larger molecules. However, the disadvantage of the sequential iterative process is the increased computational cost of verifying the valence, topological prediction of molecular structure, and graph isomorphism to calculate the reconstruction error when iteratively assembling molecules.

In one-shot generation, a molecule is generated by determining the combination of atoms and bonds in a single step. The advantage of the generative model combined with one-shot generation [2,7–10] is the simplicity of its architecture and algorithm. Its computational cost is smaller than the sequential iterative process. Consequently, the generative model can be optimized in a short time. However, the one-shot generation method can only generate small molecular graphs because the number of possible connections between atoms in larger molecules increases quadratically, increasing the likelihood of the generation of chemically invalid molecules [8].

One of the most successful generative models using the one-shot generation scheme is the molecular generative adversarial network (MolGAN) [7]. MolGAN generates small molecular graphs with the desired chemical properties by combining GANs [11] and reinforcement learning. It can generate chemically valid molecules if the number of heavy atoms used for molecular representation is nine or fewer. However, when this number exceeds nine, many disconnected molecular graphs are generated.

To overcome this issue, we propose a large MolGAN (L-MolGAN), an improved version of the MolGAN model, for generating larger, more connected molecular graphs. Increasing the number of connected molecular graphs in MolGAN will lead to the rapid generation of large molecular graphs. We integrated into L-MolGAN a mechanism that enhances the generation of connected graphs in the generative process of MolGAN. The first stage of the model judges if the generated molecular graph is connected or disconnected. If the graph is disconnected, it will be penalized during model training. Consequently, the generation of disconnected molecular graphs is suppressed in the model optimization process.

The contributions of this study are:

1. An improved MolGAN that produces large (up to 20 atoms), novel molecules without disconnections.
2. A molecular graph expansion mechanism that penalizes, and consequently suppresses, the production of disconnected graphs.

The remainder of this paper is organized as follows. Section 2 presents an overview of the proposed L-MolGAN and a method to represent the molecular graph and the framework of the original MolGAN. A method to improve the connectivity of molecular graphs generated by L-

MolGAN is also described in this section. Then, in Sections 3 and 4, the effectiveness of the proposed model is validated by comparing its performance in generating new molecules with that of the original MolGAN using a publicly available dataset of drug-like molecules. Finally, the paper is concluded in Section 5.

2. Method

2.1 Model Architecture

MolGAN, which is the baseline model, consists of GANs (generator and discriminator) and a reward network. In this model, the molecular structure is represented by a graph. The L-MolGAN adds a mechanism called molecular graph expansion to the baseline model that increases the number of generated connected molecular graph. The model architecture of L-MolGAN is illustrated in Fig 1. The L-MolGAN differs from the original MolGAN only in terms of the molecular graph expansion mechanism, highlighted by the colored box in the figure.

Fig 1 Model architecture of L-MolGAN for generating large molecular graph. It consists of a generator, a discriminator, a reward network, and a molecular graph expansion mechanism. Molecular graphs are generated by inputting into the generator vectors sampled from a prior distribution. The discriminator classifies the input molecular graph into generator-produced or

dataset. The reward network predicts the chemical properties of the input molecular graph.

GANs were used to learn the molecular features of the training dataset, and the reward network was trained to predict the chemical properties of the given molecular graph. A multi-layer perceptron (MLP) was adopted for all three components, the generator, the discriminator, and the reward network, similar to the baseline study by De Cao et al. [7]. In the following subsections, we shall explain the molecular representation and each network model, as well as the proposed modifications to the baseline model.

2.2 MolGAN

2.2.1 Molecular representation as a graph

Studies related to the artificial generation of molecules using deep generative models [12–14] represented molecules as strings using the simplified molecular-input line-entry system (SMILES) [15]. The linear SMILES is in turn generated string using a recurrent neural network and long short-term memory. Thus, the molecule of interest was artificially produced.

However, the inherent syntax of SMILES is complex, and the chemical structure and properties of a molecule can vary drastically with the order of the string and changes in a single character. In addition, the same molecule has multiple string representations, making it impossible to determine a unique SMILES [16, 17]. To avoid these problems, researchers have developed molecular graphs that represent molecules based on the graph theory.

Molecular graphs are an intuitive, more robust representation of molecules compared with intermediate representations such as SMILES. In this study, the molecules were treated as labeled undirected graphs. A molecular graph was defined as $G = (V, E)$, where E and V denote a set of edges and nodes, respectively. Each atom and each bond that make up a molecule correspond to a node $v_i \in V$ and an edge $(v_i, v_j) \in E$, respectively. The molecular graph consists of two types of matrix: the node feature matrix and the adjacency matrix. The node $v_i \in V$ in the molecular graph G was defined by the one-hot vector x_i in T dimensions, where T represents the number of types of atoms. From this vector, the type of atom, which is an attribute of node v_i, can be determined. The node feature matrix is represented by aggregating all node feature vectors. The edge $(v_i, v_j) \in E$ in the molecular graph G indicates that nodes v_i and v_j are connected. In addition to the connections between nodes, the type of bond $y \in \{1, \dots, Y\}$ is considered in the molecular graph, where Y is the number of bond types. In this study, the node feature matrix $\mathrm{X} = [\mathrm{x}_1, \dots, \mathrm{x}_\mathrm{N}]^\mathrm{T} \in \mathbb{R}^{N \times T}$ and the adjacency matrix $\mathrm{A} \in \mathbb{R}^{N \times N \times Y}$ were used to identify the types of atoms in all node sets of the molecular graph G and the adjacency matrix.

2.2.2 Generative adversarial networks

GANs are deep generative models that aim to generate samples similar to a training set by approximating the model distribution to an empirical distribution. In computational molecular design, adversarial generation is an important strategy for producing molecular species similar to a given molecular dataset.

GANs can be interpreted as an implicit generative model as it does not need assume a specific probability distribution for the model distribution when approximating the empirical distribution. This eliminates the need for an explicit likelihood function for approximating the probability distribution. On the one hand, the variational autoencoder (VAE) [18], a likelihood-based model, adopts a method to approximate the empirical and model distributions by assuming in advance the latter to be Gaussian and maximizing the evidence lower bound instead. On the other hand, GANs adopt a method to approximate the model distribution to the empirical distribution by parameterizing the distribution with a deep neural network and estimating its density ratio. GANs mainly consist of two deep neural networks to approximate the distribution by density ratio estimation: generator G_{θ}, generates a new sample $G(z;\theta)$ similar to the training sample $x \sim p_{data}$ by inputting a random number $z \sim p_{z}$ obtained from a prior distribution p_{z}; discriminator D_{φ}, which accurately identifies the input data as a training sample $x \sim p_{data}$ or a sample $G(z;\theta)$ generated by the generator. Training generator G_{θ} to generate samples similar to the empirical distribution means will yield worse identification results for the samples produced by the generator. In other words, the density ratio estimation problem is replaced by a classification problem, which can be effectively solved by deep neural networks are good. Therefore, these deep neural networks can be considered players in the minimax game of Equation 1, which shows the expected value of the cross-entropy error.

$$\min_{\theta} \max_{\varphi} E_{x \sim p_{data}(x)}[\log D_{\varphi}(x)] + E_{z \sim p_z(z)}[\log(1 - D_{\varphi}(G_{\theta}(z)))] \quad (1)$$

In adversarial learning, the generator is trained to generate samples similar to the training set and misidentify them to the discriminator. In contrast, the discriminator is trained to correctly discriminate between the samples generated by the generator and those from the training set. With this process, the two models coevolve in adversary, with the generator minimizing the second term in Equation 1 and the discriminator maximizing the linear sum of the first and second terms. The alternate optimization the two neural networks through back-propagation, a sample is eventually generated such that the discriminator cannot distinguish between real and fake samples.

2.2.3 Generator

The generator can generate molecular graphs with the desired chemical properties. In this study, its architecture was a simple MLP with four layers. The number of units in each layer was 256, 512, 1024, and 2200, respectively. By inputting a random number z sampled from the standard normal distribution $N(0, I)$ into the generator, we output the adjacency matrix $\tilde{A}$ and the node feature matrix $\tilde{X}$ representing the molecular graph. The output graph $\tilde{G} = (\tilde{A}, \tilde{X})$ is a probabilistic complete graph, which is interpreted as a categorical distribution for the types of atoms and bonds. Here, $\tilde{A}$ contains the existence probabilities of the nodes and edges for each bond type, and $\tilde{X}$ the class probabilities of the nodes. To enable its transformation into a chemically valid molecular graph, the discrete graph $G = (A, X)$ was obtained using the argmax function on the output probabilistic complete graph $\tilde{G} = (\tilde{A}, \tilde{X})$. The adjacency matrix was defined as $\mathrm{A} \in \mathbb{R}^{N \times N \times Y}$, and the node feature matrix as $\mathrm{X} = [\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{N}]^\mathrm{T} \in \mathbb{R}^{N \times T}$. The maximum number of nodes in the molecular graph was set to $N = 20$, and the number of bond types to $Y = 5$. The five types of bonds are single bond, double bond, triple bond, aromatic bond, and no bond. The number of types of atoms was set to $T = 10$: carbon, nitrogen, oxygen, fluorine, phosphorus, sulfur, chlorine, bromine, and iodine, and one-padding symbol. Thus, the maximum number and types of atoms and bond types were restricted. These constraints shall be used to determine the generator architecture. The dimensions of the output adjacency and output node feature matrices were represented by $N \times N \times Y$ (i.e., $20 \times 20 \times 5$) and $N \times T$ (i.e., 20×10), respectively.

To output the two types of matrices simultaneously, an output layer is required to output the 2200-dimensional vector, which is the sum of the number of elements of the adjacency and node feature matrices. The number of units in the output layer depends on these constraints. The random number inputs to the generator had 256 dimensions. Based on the results of existing research, the number of units in each hidden layer was set as a multiple of the number of dimensions of the input random numbers. The 2200-dimensional vector output from the generator was split into two vectors—2000- and 200-dimensional vectors—to create the adjacency and node feature matrices. These divided vectors were then transformed into the dimensionality of each defined matrix. Consequently, the output molecular graph is a complete probabilistic graph.

The final output molecular graph is a chemically valid molecular graph. Therefore, the argmax function was used to break the weak bonds in the complete graph. The output of this operation on the adjacency matrix $\tilde{A}$ is the adjacency matrix A binarized at [0,1]. The node feature matrix $\tilde{X}$ was also binarized using the same process. Finally, a new molecular graph with the correct valence was generated through the optimized molecular generation process.

However, this adversarial generation process only generates molecular species similar to the training set. Moreover, it is necessary to introduce methods to improve the properties of the generated molecules such as reinforcement learning, which uses a deterministic policy in the process of molecule generation. We incorporated the deep deterministic policy gradient method [19] into a generative model to optimize the non-differentiable chemical indices based on the literature. The stochastic policy is expressed as $\pi(a|s;\theta)$. This denotes the policy π_θ that probabilistically selects action a for state s. In this case, θ is a parameter used when the policy is being modeled. The deterministic policy μ_θ is the policy $a = \mu_\theta(s)$, where action a is uniquely determined for a certain state s. This policy is optimized by updating θ to maximize the behavioral value function for this behavior. In this study, the policy was G, and state s was represented as a random number z. Thus, for a random number z, the molecular graph is uniquely generated according to the deterministic policy. In the deep deterministic policy gradient method, the deterministic policy and action value functions were approximated using a deep neural network. Therefore, a property prediction neural network, which can be trained using gradients, was introduced into the action value function for calculating rewards. The rewards can then be used to generate molecules with indistinguishable chemical properties. These properties can be maximized by varying the policy parameters in the direction of the approximated action value gradient.

By formulating these series of processes, we trained the generator such that the objective function $L(\theta)$ in Equation 2 was minimized. A molecular graph with the desired chemical properties similar to the training data was generated by minimizing the linear combination of the GAN loss, L_{GAN}, and the reinforcement learning loss, L_{RL}:

$$L(\theta) = \lambda \cdot L_{RL} + (1 - \lambda) \cdot L_{GAN}, \qquad (2)$$

where λ is a hyperparameter that balances between adversarial learning and property optimization. This tunable parameter takes values in the range of $\lambda \in [0, 1]$.

2.2.4 Discriminator and Reward network

The architecture of these two neural networks, discriminator and reward network, were implemented by a simple MLP with three layers. The three hidden layers of both the discriminator and the reward network had 512, 256, and 2 units, respectively. The input molecules were discriminated by the discriminator as the training set or molecules sampled by the generator. The chemical properties of the input molecules were predicted using the reward network. In the generation process, the discriminator outputs the discrimination rate of authenticity based on the feature vector of the entire molecular graph, and the reward network outputs the predicted score of the chemical property. The generator can be optimized by feeding back the outputs.

However, a simple MLP cannot directly handle the graph structure data. Therefore, it would be necessary to develop a graph convolution operation specific to the graph structure data before inputting molecular graphs into the two models. The type of bond between atoms must be considered when convoluting the molecular graph. Therefore, based on the literature, we used a relational graph convolution operation that considers the attributes of the edges on a graph [20, 21]. This operation uses the adjacency matrix to convolute the node information for each edge attribute as follows:

$$h_i^{(l+1)} = \sigma(\sum_{r \in R} \sum_{j \in N_i^r} \frac{1}{|N_i|} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)}) \qquad (3)$$

where h_i^l is the feature representation of node v_i in the lth layer, $\underline{R}$ is the set of relations, and N_i^r is the set of nodes connected by the relation r in node v_i. Thus, a linear transformation was performed by extracting the neighboring node information for each relation. The self-loop was convolved similarly.

Finally, a nonlinear transformation was performed over the input signal by the activation function σ, and the feature representation $h_i^{(l+1)}$ of the $l+1$st layer was output. The convolution of a node uses its own information as well as information from its neighboring nodes. The output of the hidden layer was recursively used same as in a neural network by accumulating the convolutions. Finally, each convolved node information was aggregated into a single feature representation. Each time the convolution operation was repeated, the neighboring node information was convolved; thus, a global feature representation revealing the entire graph was obtained from the local features.

The generator and discriminator were used to facilitate the adversarial learning of the molecular generation model. The discriminator was trained to maximize Equation 1. The parameters of the generator was updated via backpropagation through the discriminator to the generator. The generator and the reward network were used to optimize the chemical properties of molecules. The generator used the reward network output as a reward, and the parameters of the two models were updated using the deep deterministic policy gradient method. In addition, the reward network was trained by back-propagating the error between the output of the reward network and the estimated property using RDKit [22], a chemoinformatics tool. Reinforcement learning for chemical properties optimization was performed once for every three iterations of adversarial learning. The parameters of the reward network were fixed in adversarial learning, whereas the parameters of the discriminator are fixed in the chemical properties optimization.

2.2.5 Molecular graph generation using the trained MolGAN

The optimized generator was extracted from the trained MolGAN model and used to generate new molecules by inputting random numbers sampled from the standard normal distribution into the generator. Changes in these numbers resulted in different molecular graphs. This allowed the generator to generate not only known molecules but also unknown ones included in the training dataset.

2.3 L-MolGAN and Molecular Graph Expansion Mechanism

According to the literature [7, 8], the number of nodes in the generated molecular graph is small, which is the problem we aim to solve. Earlier studies evaluated the MolGAN under the condition that only nine heavy atoms can be used to produce a molecular graph without disconnection. However, this limit is not practical in drug discovery, especially for larger molecules because the more the atoms, the more the disconnected graphs. To solve this problem, we propose modifications to the MolGAN.

We suppress the generation of disconnected graphs by penalizing them during the model training process. The detailed algorithm is as follows: 1) for each generated graph, its connectivity is checked by depth-first search (DFS) and 2) if the graph is disconnected, its chemical property score is set to zero as a penalty; otherwise, its score is predicted by the reward network. This is similar to the general training process. DFS is a recursive and exhaustive algorithm used to search all nodes of a graph or a tree. With DFS, the entire graph is traversed by starting at a certain node in the molecular graph and following the edges. If all the nodes in the graph can be reached, the graph is considered connected. Repeated penalizations to a disconnected graph will suppress its generation and increase the number of connected graphs generated. We refer to these modifications as the “molecular graph expansion mechanism,” and rename the resulting improved MolGAN as L-MolGAN.

3. Experiment

We shall investigate the effectiveness of the L-MolGAN by comparing it with the baseline MolGAN. In all experiments, we set the QED as a singular objective to derive new drug candidates and trained two modes (i.e., the model training was performed to optimize QED score with an RL objective). Its effectiveness was evaluated in terms of 1) how well it works for large molecular graph generation and 2) how many novel drug-like molecules it generates. The general settings of the model training and its evaluation metrics shall be described in Subsections 3.1 to 3.3. Then, three different numerical experiments shall be described in the Subsections 3.4 to 3.6.

3.1 Dataset

In this study, ZINC-250k [23], a renowned molecular datasets, was used in the experiments to generate molecular graphs. ZINC-250k is made up of 250 000 commercial drug-like molecules randomly selected from the ZINC database. The maximum number of constituent heavy atoms of a molecule in ZINC-250k is 38. Particularly, a subset of ZINC-250k was sampled by randomly choosing 15 000 molecules from ZINC-250k, with the maximum number of constituent heavy atoms limited to 20, which is approximately twice the molecular size of that used in the baseline study by De Cao et al. [7].

3.2 Evaluation metrics

We employed the generally used indices of validity, novelty, and uniqueness to evaluate the molecular generation model. Validity is the percentage of chemically valid molecules among the generated molecules. Note that validity is not a measure of the connectivity of molecules but only the valence of atoms. Novelty is the percentage of valid molecules among the generated molecules not included in the training data. In this study, these molecules were defined as novel molecules. Uniqueness is the percentage of generated molecules that are valid as well as unique. This measure indicates the degree of diversity among the molecules generated. Furthermore, species, the number of unique and connected molecular graphs, was introduced to clearly represent the number of unique molecules that were derived. The ideal molecular generation model should generate novel, valid, and connected molecules. Additionally, connectivity, which is the percentage of connected graphs, is one of the most important metrics introduced in this study. It indicates the percentage of valid and connected molecular graphs among the ones generated.

Furthermore, three chemical indicators were used to evaluate the chemical properties of the generated molecules, QED [24], solubility, and synthetic accessibility (SA) score. In QED, drug-like properties were calculated using a weighted geometric mean based on the distribution of multiple drug-properties data. Solubility indicates the degree of hydrophilicity of a molecule, which was quantified by the logP coefficient. This coefficient is defined as the logarithm of the concentration ratio of different solvents [25]. The SA score indicates the ease of synthesis of a molecule [26]. In this experiment, all chemical-property scores were manipulated to take values in the range of [0,1]. Note that the property scores of molecules with disconnected graphs in the L-MolGAN were set to zero as a penalty.

3.3 Model training

MolGAN and L-MolGAN were trained using the Adam optimizer [27] with a learning rate of 0.0001 to optimize the QED for all the experiments. Mini-batch training was conducted to stabilize the learning. The batch size was set to 100. With an early stopping strategy, the model training was terminated when the average change in loss during 10 epochs was less than 1.0% or when the maximum number of epochs (300) had been reached. Mode collapse [28], a situation where similar data are generated regardless of the arbitrariness of numbers input to the generator, is one of the crucial issues in GANs. To circumvent this issue, we used mini-batch training and the early stopping strategy mentioned above.

In another study [7], researchers terminated model training when the uniqueness score fell below 2.0%. However, this cause the generated molecules to become more homogenous because several epochs would be solely dedicated to satisfying the termination criterion. Therefore, we focused on the average loss change in the training process to determine the termination criterion, rather than thresholding for each property score.

3.4 Experiment I: Parameter study of learning balance

We investigated the extent to which the value of parameter λ, which balances the chemical properties optimization and adversarial learning, affects the characteristics of the generated molecules. The optimal choice of λ for the molecular generation model was determined through this experimental task. The value of λ was varied from 0.0 to 1.0 in increments of 0.2, and the model was trained in five trials for each value.

As explained earlier, we optimized the QED to generate drug-like molecules. The trained model that maximized the sum of validity, novelty, uniqueness, and QED scores was selected as the reference to evaluate the performance of the molecular generation model.

3.5 Experiment II: Performance comparison of proposed method with existing method

The proposed method and the baseline model (MolGAN) were compared in terms of performance using the evaluation metrics described in Subsection 3.2. The representative model for each method was chosen through a parametric study of λ.

3.6 Experiment III: Generation and evaluation of novel

molecules

The proposed method was evaluated in terms of the number of novel drug-like molecules that can be derived. Here, new molecules were generated by inputting into the pretrained generator random numbers sampled 5000 times from the standard normal distribution. The chemical properties of the generated molecules were evaluated using RDKit. Novel molecules with the desired chemical properties were identified from the generated molecules. The 20 molecules with the highest QED scores were chosen, and their chemical properties were examined in terms of novelty and ease of synthesis.

In addition, in drug discovery and materials design, the generated molecules should not only satisfy a single property (such as the QED) but also possess other properties such as synthesizability. Therefore, SA and logP were chosen in addition to the QED to evaluate the molecules generated in this experiment. However, there is a tradeoff between QED and logP [29]. There is no single best molecule but several ‘good’ molecules that exist within the envelope of all the generated molecules. Here, we refer to them as ‘dominant molecules’ and chose them in terms of the three chemical properties, QED, SA, and logP, for each combination of two of the three properties. Furthermore, we classified them into hydrophiles and lipophiles based on the logP score and verified if the dominant molecules possessed both the properties. Both hydrophilicity and lipophilicity are important properties considered in drug design.

4. Results and Discussion

4.1 Experiment I: Parametric study of learning balance λ

The effect of λ on the molecules generated by the L-MolGAN are summarized in Table 1. The table indicates that the mean value of validity increases with an increase in λ, while its standard deviation decreases. The same tendency was observed for the connectivity and QED scores. In contrast, the uniqueness decreased as λ increased. We believe this could have been caused by the occurrence of mode collapse in the model training, as reported by a conventional study [7]. Consequently, as λ increases, the generative model becomes more susceptible to mode collapse. This would also affect the diversity of the generated molecules, as evidenced by the decrease in the standard deviation of every molecular property index with an increase in λ.

Table 1 Comparison of properties of molecules generated at different λ by the proposed method. Each value indicates the mean and the standard deviation of each metric for five trials.

λ	**Validity [%]**	**Uniqueness [%]**	**Novelty [%]**	**Connectivity [%]**	**QED**	**SA**	**logP**	**Species**
0.0	28.62 ± 6.43	19.66 ± 6.53	100.00 ± 0.00	59.82 ± 15.5	0.62 ± 0.03	0.29 ± 0.05	0.54 ± 0.05	72.60 ± 41.67
0.2	80.72 ± 2.71	8.46 ± 2.40	100.00 ± 0.00	85.57 ± 9.10	0.77 ± 0.04	0.21 ± 0.11	0.59 ± 0.06	138.00 ± 41.74

0.4	94.70 ± 3.34	4.10 ± 4.92	100.00 ± 0.00	95.26 ± 4.92	0.82 ± 0.02	0.12 ± 0.05	0.60 ± 0.04	72.60 ± 41.67
0.6	**95.00 ± 3.31**	**3.21 ± 1.09**	**100.00 ± 0.00**	**94.85 ± 5.58**	**0.85 ± 0.03**	**0.16 ± 0.07**	**0.62 ± 0.03**	**76.80 ± 20.98**
0.8	98.79 ± 1.88	0.30 ± 0.11	100.00 ± 0.00	99.81 ± 0.24	0.82 ± 0.05	0.10 ± 0.06	0.55 ± 0.08	10.20 ± 5.12
1.0	96.79 ± 0.21	0.04 ± 0.02	100.00 ± 0.00	100.00 ± 0.0	0.86 ± 0.05	0.16 ± 0.09	0.55 ± 0.11	1.60 ± 0.89

The average QED score from the training data was 0.76 ± 0.12. The QED score of the newly generated molecules for λ = 0.0 was smaller than the average. The scores with other λ values were greater than the average. Validity, connectivity, and QED for λ = 0.0 were remarkably smaller than those of the other settings. The generator should be trained not only to improve the chemical property score, but also to suppress the generation of invalid molecules during the optimization. However, at λ = 0.0, the model training was completely dedicated to the adversarial learning of the generative model, rather than the chemical properties optimization. It is plausible that the overall performance at λ = 0.0 was the weakest because of the generation of several invalid graphs.

From these results, we chose λ = 0.6 as the optimal value, which maximizes the total values of all the considered evaluation metrics. Furthermore, a single representative model, which had the largest total value among the five trials for $\lambda = 0.6$, was chosen. This setting shall be used for all subsequent experiments with MolGAN as well as L-MolGAN.

Fig 2 Distribution of QED scores of molecules generated by the proposed method at the different λ. The black solid line labeled as 'ZINC subset' indicates the distribution of molecules included in the training dataset. Kernel density estimation has been used to depict the QED distribution. Only the connected molecular graphs are used for the density estimation.

Fig 2 shows the distribution of QED scores of molecules generated by the proposed method, estimated using the best-performing generative model for each λ value. Additionally, 'ZINC subset' indicates the distribution of molecules included in the training dataset.

We shall focus on the peak of each distribution to determine the effect of λ. Because each distribution has multiple peaks, we shall focus only on the highest one. As λ increases, the peak shifts to a higher QED score. Specifically, when λ was larger than 0.4, the peak shifted to a higher QED score than that of the training data. From the fact that the model was trained to maximize the QED score, we confirmed that the model was well-optimized. Note that a narrower distribution was obtained owing to the mode collapse with a larger λ.

4.2 Experiment II: Performance comparison of the proposed

method with existing methods

Table 2 lists the results of the performance comparison between MolGAN and L-MolGAN in generating molecules. The results confirm the validity, connectivity, and QED scores of the L-MolGAN were better than those of the MolGAN.

Table 2 Comparison of molecules generated by MolGAN and L-MolGAN

Model	Validity [%]	Uniqueness [%]	Novelty [%]	Connectivity [%]	QED	SA	logP	Species
MolGAN	94.53	**5.97**	**100.00**	48.12	0.85	**0.47**	0.60	44.00
L-MolGAN	**98.91**	4.88	**100.00**	**94.32**	**0.88**	0.23	**0.66**	**88.00**

Particularly, connectivity of graphs saw an improvement of 1.96 times in L-MolGAN over the existing model. However, the uniqueness and SA of L-MolGAN were worse than those of MolGAN. This indicates that the molecules generated by MolGAN were more diverse than L-MolGAN and are relatively easy to synthesize. However, the lower connectivity score of MolGAN indicates the presence of several disconnected molecules in the generated molecules. In this regard, the L-MolGAN is more effective in generating valid as well as connected molecules than the MolGAN.

In addition, the lower SA score in the L-MolGAN suggests that the model generates molecular graphs with more complex molecular structures because they have mostly connected nodes. Both models generated entirely novel molecules that did not exist in the training data; however, MolGAN had a connectivity score of 48.12%. In contrast, the L-MolGAN achieved a higher connectivity score of 94.32%. These results suggest that the proposed model generates larger, more novel drug-like molecules, and has more practical implications for drug discovery compared with the existing method.

Fig 3 Distributions of the QED score in the molecules generated by the representative models of MolGAN and L-MolGAN. Kernel density estimation has been used to depict the QED distribution. Only the connected molecular graphs were used for the density estimation.

Fig 3 illustrates the distribution of QED scores estimated from the molecules generated by the L-MolGAN and MolGAN. These distributions only represent the connected molecular graphs. The average QED values of the training data, MolGAN, and L-MolGAN were 0.76, 0.81, and 0.88, respectively. In Fig 3, the training data and MolGAN have a single peak, whereas L-MolGAN has two. In addition, the peak positions of the training data and MolGAN were similar. However, the distribution of MolGAN is narrower than that of the training data and has higher QED scores than the training data. These results indicate that MolGAN has been successful at chemical properties optimization.

A closer look at the distribution in L-MolGAN in Fig 3 reveals two large peaks; one is close to the peak position of the MolGAN and the training data, while the other is located where the QED score is higher. We hypothesize that the peak in MolGAN and one of the peaks in L-MolGAN were close to those of the training data because the optimization of chemical properties was strongly affected by the properties of mode of the QED distribution in the training data.

Another peak in the L-MolGAN was located where the QED score was greater than 0.9. Notably, in this range of QED, the distributions of MolGAN and training data contained few molecules. This indicates that the L-MolGAN can exploit molecular graphs with better QED scores than the MolGAN. We assumed this was so because the L-MolGAN generated many connected molecular graphs. Improvement in connectivity would contribute to the generation of substituents and molecular skeletons with higher QED scores. Moreover, because the QED score is based on the physical properties of a molecular graph, it can be even calculated for disconnected graphs. For this reason, contrary to our presupposition, the chemical properties would be optimized for disconnected graphs as well. These results suggest that the L-MolGAN can overcome this issue in MolGAN.

4.3 Experiment III: Generation and evaluation of novel molecules

Fig 4 illustrates a two-dimensional description of the best 20 molecules with the highest QED scores generated by the optimized generator of L-MolGAN.

Fig 4 Two-dimensional representation of 20 molecules with the best QED scores. The numbers at the bottom of each molecule represent the corresponding QED, logP, and SA scores.

Most molecules contained one or two sulfur atoms in their structures. In addition, many had QED scores of 0.9 or higher, indicating that their chemical properties were superior to those of the ZINC dataset. However, because their SA scores were significantly small, the molecules generated may be unrealistic. Therefore, we focused on the relationship between the structure of the generated molecules and their synthesizability.

1,3-Thiazole was included as the common substructure of the top 20 molecules. Thiazole is a nitrogen-containing five-membered heterocyclic compound, which is a common skeleton in molecules used in pharmaceuticals and agrochemicals. The bicyclic ring skeleton was also found to be a common substructure within thiazole-containing molecules with an SA of 0.0. This skeleton is composed of five carbon or sulfur atoms bridging the carbon atoms at the 2 and 5 positions of the thiazole ring. Nine out of the top twenty molecules had these characteristics. The bicyclic ring skeleton is difficult to synthesize because of the high steric strain of the molecule.

Fig 5 Three-dimensional representation of the thiazole and bicyclic skeletons.

Fig 5 represents the three-dimensional model of the common thiazole and bicyclic ring skeletons included in the generated molecules. Because the thiazole ring skeleton has a planar structure, the atoms and substituents (i.e., the two methyl groups) in the ring lie on the same plane because the thiazole ring is aromatic. However, the planarity of the thiazole ring and its surroundings in the bicyclic ring skeleton is broken. The planarity of the thiazole ring was not maintained at the 2 and 5 carbon positions. This steric strain is affected by the number of atoms to be bridged. Therefore, we concluded that they were unrealistic due to the steric strain caused by the bicyclic framework. However, the L-MolGAN could generate these molecules, which have not yet been discovered. Therefore, it was worth an attempt to synthesize them as drug candidates.

Fig 6 highlights the dominant molecules chosen in the QED–SA space. The dotted line indicates the envelope of the generated molecules. We confirmed that a tradeoff between the QED and SA scores, and the existence of eight dominant molecules. The dominant molecules [A] to [F] with a high QED score had a common cyclic substructure, whereas molecules [G] and [H] with low QED scores were chain-like. This suggested the potential of heterocyclic compounds as oral drugs. It is plausible that the synthesis became difficult when the ring structure contained two or more heteroatoms. Consequently, molecules [A] to [C] exhibited an

SA of less than 0.2, and that of molecules [D] to [F] approximately 0.5.

Fig 6 Dominant molecules identified through QED and SA scores ([A] to [H]). The points indicate all the molecules generated by the L-MolGAN. The dotted line indicates the envelope of the generated molecules.

Subsequently, the dominant molecules were selected in SA–logP space, as shown in Fig 7. The molecule [H] was also chosen in the QED–SA space. We only reported molecules with higher lipophilicity based on the logP score. The molecule [H] was also chosen in the QED–SA space. The dominant molecules with higher lipophilicity in the QED–logP space are also shown in Fig 8. The molecule [B] was also chosen in the QED–SA space. Several dominant molecules were sulfur-containing compounds.

Fig 7 Dominant molecules with higher lipophilicity identified through SA and logP scores ([H], [I] and [J]). The points indicate all the molecules generated by the generator of L-MolGAN. The dotted line indicates the envelope of the generated molecules.

Fig 8. Dominant molecules with higher lipophilicity identified through QED and logP scores ([B], [K] to [Q]). The points indicate all the molecules generated by the L-MolGAN.

The dotted line indicates the envelope of the generated molecules.

The dominant molecules with higher hydrophilicity in the QED–logP space are shown in Fig 9. Molecules [A] and [C] are also chosen in the QED–SA space. Fig 10 indicates molecules in the SA–logP space. Molecules [E], [G], and [H] were already chosen in the QED–SA space. In addition, the molecule [R] was also chosen in the QED–logP space.

Fig 8 Dominant molecules with higher hydrophilicity identified through QED and logP scores ([A], [C], [R] to [U]). The points indicate all the molecules generated by the L-MolGAN. The dotted line indicates the envelope of the generated molecules.

Fig 9 Dominant molecules with higher hydrophilicity identified through SA and logP scores ([E], [G], [H], [R] and [V]). The points indicate all the molecules generated by the L-MolGAN. The dotted line indicates the envelope of the generated molecules.

These results revealed the generation of a variety of dominant molecules by the L-MolGAN. Although the model was trained to optimize only the QED for drug discovery, a variety of molecules were identified among several combinations of the three chemical properties. Additionally, there was no best single molecular graph that simultaneously optimized the three chemical properties or their combinations. This motivated us to search for a variety of molecular graph among conflicting optimization goals of plural chemical properties. Future studies should apply a multi-objective optimization framework to the proposed method to search for dominant molecules with higher chemical property scores and more diverse chemical structures and properties.

5. Conclusions

The performance of the MolGAN deteriorates when generating a molecular graph with a molecular size larger than nine atoms, owing to the increase of disconnected graphs. This introduces challenges in the design of drugs with large molecules. We addressed this challenge by adding to the MolGAN a molecular graph expansion mechanism that penalizes disconnected graphs and referred to it as L-MolGAN. The L-MolGAN improved the number of connected graphs generation on the ZINC-250k molecular dataset by a factor of 1.96, compared with the MolGAN. We also confirmed the generation of a variety of drug-like molecules by the L-MolGAN, even though it was optimized for a single property, i.e., QED. The L-MolGAN shall contribute to the efficient discovery of new molecules larger than those generated by the MolGAN.

Acknowledgements

We would like to thank Editage [http://www.editage.com] for editing and reviewing this manuscript for English language. No funding to declare.

References

[1] Churchwell CJ, Rintoul MD, Martin S, Visco Jr DP, Kotu A, Larson RS, et al. The signature molecular descriptor: 3. inverse-quantitative structure–activity relationship of ICAM-1 inhibitory peptides. J Mol Graph Model. 2004;22(4): 263-73.

[2] Madhawa K, Ishiguro K, Nakago K, Abe M. Graphnvp: An invertible flow model for generating molecular graphs. arXiv:1905.11600 [Preprint]. 2019 May 28. https://arxiv.org/abs/1905.11600.

[3] You J, Liu B, Ying Z, Pande V, Leskovec J. Graph convolutional policy network for goal-directed molecular graph generation. In: Advances in Neural Information Processing Systems 31. 2018. pp. 6412-22. arXiv:1806.02473 [Preprint]. 2019 [cited 2019 Feb 25]. Available from: https://arxiv.org/abs/1806.02473.

[4] Jin W, Barzilay R, Jaakkola T. Junction tree variational autoencoder for molecular graph generation. In: International Conference on Machine Learning PMLR. 2018. pp. 2323-32.

[5] Liu Q, Allamanis M, Brockschmidt M, Gaunt AL. Constrained graph variational autoencoders for molecule design. arXiv:1805.09076 [Preprint]. 2018. https://arxiv.org/abs/1805.09076.

[6] Li Y, Vinyals O, Dyer C, Pascanu R, Battaglia P. Learning deep generative models of Graphs. arXiv:1803.03324 [Preprint]. 2018. https://arxiv.org/abs/1803.03324.

[7] De Cao N, Kipf T. MolGAN: An implicit generative model for small molecular graphs. arXiv:1805.11973 [Preprint]. 2018. https://arxiv.org/abs/1805.11973.

[8] Simonovsky M and Komodakis N. Graphvae: Towards generation of small graphs using variational autoencoders. In: The International Conference on Artificial Neural Networks, Springer. 2018. pp. 412-22.

[9] Hong SH, Ryu S, Lim J, Kim WY. Molecular generative model based on an adversarially regularized autoencoder. J Chem Inf Model 2019;60(1): 29-36.

[10] Ma T, Chen J, Xiao C. Constrained generation of semantically valid graphs via regularizing variational autoencoders. arXiv:1809.02630 [Preprint]. 2018. https://arxiv.org/abs/1809.02630

[11] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. arXiv:1406.2661 [Preprint] 2014. https://arxiv.org/abs/1406.2661

[12] Kusner MJ, Paige B, Hernandez-Lobato JM, Grammar variational autoencoder. In: The International Conference on Machine Learning, PMLR. 2017. pp. 1945-54.

[13] Gomez-Bombarelli R, Wei JN, Duvenaud D, Hernandez-Lobato JM, Sanchez-Lengeling B, Sheberla D, et al. Automatic chemical design using data-driven continuous representation

of molecules ACS central science. 2018;4(2): 268-76.

[14] Benhenda M. Chemgan challenge for drug discovery: Can AI reproduce natural chemical diversity? arXiv:1708.08227 [Preprint]. 2017. https://arxiv.org/abs/1708.08227.

[15] Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Comput Sci 1988;28(1): 31-36.

[16] Bjerrum E J. SMILES enumeration as data augmentation for neural network modeling of molecules. arXiv:1703.07076 [Preprint]. 2017. https://arxiv.org/abs/1703.07076.

[17] Alperstein Z, Cherkasov A and Rolfe J T. All smiles variational autoencoder. arXiv:1905.13343. 2019. https://arxiv.org/abs/1905.13343

[18] Kingma DP, Welling M. Auto-encoding variational Bayes. arXiv:1312.6114 [Preprint] 2013. https://arxiv.org/abs/1312.6114.

[19] Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, et al. Continuous control with deep reinforcement learning. arXiv:1509.02971. 2015. https://arxiv.org/abs/1509.02971.

[20] Schlichtkrull M, Kipf TN, Bloem P, Van Den Berg R, Titov I, Welling M. Modeling relational data with graph convolutional networks. In: European Semantic Web Conference Springer. 2018. pp. 593-607.

[21] Li Y, Tarlow D, Brockschmidt M, Zemel R. Gated graph sequence neural networks. arXiv:1511.05493 [print] 2015. https://arxiv.org/abs/1511.05493.

[22] Landrum G. Rdkit: Open-source cheminformatics. [Online] http://www.rdkit.org.

(Accessed) 3(04):2012;2006.

[23] Irwin J, Sterling T, Mysinger MM, Bolstad ES, Coleman RG, ZINC: A free tool to discover chemistry for biology. J Chem Inf Model 2012;52(7): 1757-68.

[24] Bickerton GR, Paolini GV, Besnard J, Muresan S, Hopkins AL. Quantifying the chemical beauty of drugs. Nat Chem 2012;4(2): 90-98.

[25] Comer J, Tam K. Lipophilicity Profiles: Theory and Measurement. In: Bernard Testa, Han van de Waterbeemd, Gerd Folkers, Richard Guy. Pharmacokinetic Optimization in Drug Research: Biological, Physicochemical and Computational Strategies. Zurich: Wiley Online Library; 2001. pp. 275-304.

[26] Ertl P, Schuffenhauer A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. J Cheminformatics 2009;1: 1-11.

[27] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv:1412.6980 [Preprint] 2014. https://arxiv.org/abs/1412.6980.

[28] Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X. Improved techniques for training GANs. arXiv:1606.03498 [Preprint]. 2016. https://arxiv.org/abs/1606.03498.

[29] Damani F, Sresht V, Ra S. Black box recursive translations for molecular optimization. arXiv:1912.10156 [Preprint]. 2019. https://arxiv.org/abs/1912.10156.