Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Structure-aware conditional variational auto-encoder for constrained molecule optimization

Junchi Yu[a,b,c], Tingyang Xu[c], Yu Rong[c], Junzhou Huang[c], Ran He[a,b,d,*]

[a] *University of Chinese Academy of Sciences, China*
[b] *Institute of Automation, Chinese Academy of Sciences, China*
[c] *Tencent AI Lab, China*
[d] *Center for Excellence in Brain Science and Intelligence Technology, CAS, China*

ARTICLE INFO

Article history:
Received 10 January 2021
Revised 28 December 2021
Accepted 7 February 2022
Available online 9 February 2022

2010 MSC:
00-01
99-00

Keywords:
Molecule optimization
Conditional generation
Drug discovery

ABSTRACT

The goal of molecule optimization is to optimize molecular properties by modifying molecule structures. Conditional generative models provide a promising way to transfer the input molecules to the ones with better property. However, molecular properties are highly sensitive to small changes in molecular structures. This leads to an interesting thought that we can improve the property of molecules with limited modification in structure. In this paper, we propose a structure-aware conditional Variational Auto-Encoder, namely SCVAE, which exploits the topology of molecules as structure condition and optimizes the molecular properties with constrained structural modification. SCVAE leverages graph alignment of two-level molecule structures in an unsupervised manner to bind the structure conditions between two molecules. Then, this structure condition facilitates the molecule optimization with limited structural modification, namely, constrained molecule optimization, under a novel variational auto-encoder framework. Extensive experimental evaluations demonstrate that structure-aware CVAE generates new molecules with high similarity to the original ones and better molecular properties.

1. Introduction

Molecule optimization is one fundamental problem in biochemistry since it helps to generate molecular structures with specific desired properties [1,2]. However, it still remains challenging because the space of possible molecules is vast and searching in such space is burdensome owing to its combinatorial nature [3]. Especially, molecular properties are highly sensitive to small changes in molecular structures [4]. For example, one could increase the solubility of ethyl-benzene by simply changing the hydrogen on the methyl to a hydroxyl. Such property sensitivities of molecules actually attract chemists' interests because a small change in a molecule indicates small changes in the synthetic paths. This leads to constrained molecule optimization shown in Fig. 1, which requires limited structural modifications but significant improvements on molecular properties.

Nowadays, it becomes natural to apply deep learning on molecule optimization due to its successes in many other fields. Prior deep learning works mainly focus on multi-step solutions. Traditional VAE based methods resort to representation learning with well-designed variational auto-encoder (VAE) architecture to approach the potential molecular space [5–7]. Then, these methods do multi-step gradient ascent with respect to the certain property in the latent space to search the better molecules. Some reinforcement learning (RL) based methods adjust the hidden molecular space by treating molecular properties as rewards [8]. Beyond that, Jin et al. [9] views molecule optimization as paired graph-to-graph translation to avoid the time and computational cost in multi-step solutions. It learns molecule optimization as an one-step projection from a pair of molecules with significantly improved property and similar structure in the pair. Another way to model molecule optimization is to isolate the property and structure in the latent space [10] via disentanglement. Recent literature focuses on the subgraph-based methods which combine different motifs for the synthesized molecules. Despite fruitful results obtained by those deep learning methods, few methods focus on constrained molecule optimizations. The challenge still remains how to model property sensitivities with structural constraints. In other words, can we leverage the structural similarities as a condition to facilitate the constrained molecule optimization?

Recalling methods in deep graph learning, graph alignment has received increasing attention to capture correlated structural sim-

[*] Corresponding author at: Institute of Automation, Chinese Academy of Sciences, China.
E-mail address: rhe@nlpr.ia.ac.cn (R. He).

https://doi.org/10.1016/j.patcog.2022.108581

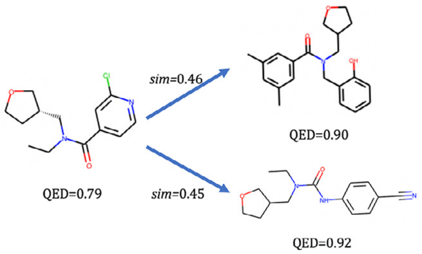

Fig. 1. An instance for constrained molecule optimization. Only limited changes in structure are allowed to improve the drug likeness (QED) of a molecule. The similarity is computed via Tanimoto Similarity over Morgan Fingerprints of two molecules.

ilarities among graphs. The goal of graph alignment is to identify similar nodes across multiple graphs [11,12]. However, it is nontrivial to model the similarity of molecules with the graph alignment. First of all, it is laborious and time-consuming of manually identifying anchor points or anchor edges in molecule pairs for alignment. Besides, the one-to-one mapping setting of current graph alignment methods hinders its implementation to align the similar nodes and substructures in the molecule graphs as different molecules have different numbers and types of nodes (atoms). Most importantly, the substructures in molecules are vital to the molecule property while the graph alignment only seek for correspondence at node level.

To this end, we propose *structure-aware* Conditional Variational Auto-Encoder (SCVAE) for constrained molecule optimization. SCVAE reconstructs one of a pair of two similar molecules by taking the other one in the pair as a *structural condition*. We first employ *two-level soft graph alignment* to exploit structural similarity of molecule pairs. Specifically, we decompose a molecule into small motifs and view it as a tree to model structure similarity at a low-resolution. Then, at graph level, we examine the atom correspondence for structural similarity at a high-resolution. This strategy allows us to effectively model the similarity of molecules at different level. Furthermore, we view the structural similarity as an extra component in the evidence lower bound to train the whole framework. In the training phase, SCVAE treats the one with poorer molecular property in the pair as structure condition and reconstructs the one with better molecular property in the pair so that we can obtain new molecules with better property by feeding another molecule as a condition in the generation phase. We evaluate our model on five constrained molecule optimization tasks on ZINC and QM9 dataset. Experiment results demonstrate that our model can exploit structural information in paired molecules to achieve superior optimizing results.

2. Related work

2.1. Molecule generation and optimization

Molecules are usually represented as SMILES strings and graphs. To ensure the chemical validity of the generated molecules, the former [5,6] mainly resort to syntactic and semantic constraints in the decoding process. However, those methods cannot fully maintain chemical valence and thus result in invalid molecules. To this end, Jin et al. [7,9] view molecules as a combination of cliques and achieve 100% valid molecules by masking the cliques that lead to invalid results. You et al. [8] further incorporates molecule-reinforcement rules and domain-specific rewards. Later, given a starting molecule, Seff et al. [13] explores new ones by locally operating reversible inductive moves.

Although there has been a great breakthrough in molecule generation, molecule optimization remains challenging. Madhawa et al. [1], Kusner et al. [5], Dai et al. [6], Jin et al. [7] resort to Bayesian Optimization and do gradient ascent w.r.t target property in the latent space. Shi et al. [2], You et al. [8] start from a single atom or subgraph and constantly connect with new parts that have the most RL rewards until it reaches the maximum length. Built upon JT-VAE, VJTNN [9] introduces a one-step manner for constrained molecule optimization by selecting paired molecules with similar structure and distinct improvement in properties. Trained with paired data, VJTNN learns a mapping from degenerated molecules to the better ones by modeling the difference between their latent representations. However, it doesn't model the structure similarity between the input pairs, which is essential to constrained molecule optimization. Motivated by the observation that the molecule properties are highly influenced by small motifs [14,15], several works decompose the molecules into small substructures and ensembles the motifs that leads to molecules with better properties [3,16]. Yang et al. [17] performs the molecule optimization by simultaneously augmenting the datasets and training the model via the stochastic expectation maximization method.

2.2. Graph alignment

The purpose of graph alignment is to match the nodes across different graphs. To do that, most works rely on a set of predefined anchor nodes as the queries and search for the candidates which best matches the queries [18]. For example, Yasar and Çatalyürek [12] leverages anchor points to locate nodes in a certain graph and compares the position of nodes across different graphs by a divide-and-conquer strategy. Derr et al. [19] aligns node embedding both on node and graph level with the advantage of adversarial learning. Besides, other works factorize the similarity matrix of the nodes to avoid constructing pair-wise feature comparison [11]. Furthermore, the graph alignment is extended to a more general scenario that involves multiple graphs using intra- and inter- structural embedding [20]. The node correspondence discovered by the graph alignment methods naturally hints the structural similarity between the pairs of graphs, and thus facilitates various tasks. Sun et al. [21] employs the graph alignment to mitigate the non-isomorphic neighborhood structures in knowledge graphs and yields consistent representations for them. Weskamp et al. [22] uses the graph alignment to identity the common features in different proteins to study the functional relationship from their structures. Berg and Lässig [23] develops a searching algorithm to mine the frequent motifs in the interaction network base on graph alignment. Bai et al. [24] enhances the predictive performance of the graph neural networks by aligning the topological structures of the graphs. However, it is nontrivial to employ graph alignment to model the structural similarity across molecules and improve the constraint molecule optimization. The major challenge is that the anchor nodes in molecule graphs are under-defined.

2.3. Conditional generative model

Given samples x with corresponding attributes y, the conditional generative model aims to learn a conditional distribution $p(x|y)$ and generate data by feeding y to the model. Recent works mainly improve the adversarial generative network (GAN) [25] and variational auto-encoder (VAE) [26] by modeling a distribution of hidden space conditioned on the input observation. Two popular models, namely Conditional-GAN [27] and Conditional-VAE [28], have been applied into many fields such as image generation [29–32], text generation [33] and neural language processing [34,35]. These methods usually embed images or text into latent vectors to constrain the hidden space. Besides, other works resort to disentangle the latent representations regarding the attributes in the hidden space Li et al. [36]. Then, the conditional generation is achieved by concatenation of desired latent representations and output the real world data by a decoder. Chen et al. [37] extends GAN with a mutual information cost and thus successfully controls the properties of images by modifying the latent vectors. Higgins et al. [38] learns factorised latent vector in an unsupervised manner via augmenting the Kullback–Leibler divergence penalty of the VAE objective. Although the conditional generation models have greatly facilitated various tasks in many aspects, it remains a challenge to model the structural condition between the molecule graphs and restrict the structural similarity across the input and output molecules, mostly due to the irregular and discrete nature of molecules. Thus, it is imperative to effectively model the similarity of molecules to boost the constrained molecule optimization.

3. Notations

Viewing molecules as graphs, we denote a molecule graph as $G = \{V, E\}$, where V and E are nodes and edges. Similar to Jin et al. [7], we extract a junction tree from the molecule graph and denote it as $T = (V, E)$.

Let f_i be the feature of node i. f_i represents atom type, valence of an atom or type of a subgraph when i represents a graph node or a tree node. We denote the node embedding of molecule x at tree and graph level as x^T and x^G respectively. x^T and x^G are $n_T \times L$ and $n_G \times L$ matrices, where n_T and n_G are the number of nodes at tree and graph level and L is the dimension of node embedding. We adopt Tanimoto Similarity over Morgan Molecule Fingerprints as the similarity metric between molecule x and y, denoted as $sim(x, y)$. For constrained molecule optimization, similar to Jin et al. [9], we first sample such molecule pairs to satisfy $(X, Y) = \{(x, y)|p(y) - p(x) \geq 0, sim(x, y) \geq \delta, x \in (M), y \in (M)\}$, where $p(\cdot)$ denotes the property value of a molecule and (M) denotes a set of molecules. We denote $[\cdot, \dots, \cdot]$ as the operator of concatenation for vectors.

4. Method

In this section, we start with the preliminaries related to the proposed method. Then we elaborate the details of our method.

4.1. Preliminaries

Conditional generative methods The constrained molecule optimization is closely related to the conditional generative methods. Given molecule pairs (X, Y), these methods aim to learn the conditional distribution $p(Y|X)$. Usually it is intractable to directly compute this distribution. Following the framework of conditional variational auto-encoder, VJTNN [9] optimizes a variational lower bound, which takes the form:

$$\log p(Y|X) \geq \mathbb{E}_{z\sim q(z|x)} \log p(y|z, x) - \mathbb{KL}(q(z|x)|p(z)) = \mathcal{L}_{ELBO} \quad (1)$$

where $\mathbb{KL}(\cdot)$ is the Kullback–Leibler divergence. This objective introduces a variational encoder $q(z|x)$ and a probabilistic decoder $p(y|z, x)$ for molecule optimization. The encoder maps x to the latent variable z, while the decoder takes both x and z as input and outputs y with better properties.

However, this objective only models the conditional dependency between x and y. This is insufficient for constrained molecule optimization as the conditional dependency does not necessarily lead to structure similarity. In other words, $Y_{cmo,x} = \{y|p(y) - p(x) \geq 0, sim(x, y) \geq \delta, y \in (M)\}$ is a subset of $Y_{mo,x} = \{y|p(y) - p(x) \geq 0, y \in (M)\}$. Therefore, it is crucial to model the structure similarity between molecule pairs in the optimization.

Junction tree encoder/decoder The junction tree encoder is proposed by Jin et al. [7]. The intuition is that molecules can be represented both at graph level and junction-tree level. Viewing molecules as graphs, it first extracts tree and graph level node embeddings via message passing neural network,

$$e^t_{ji} = g_1\left(f_i, f_j, \sum_{j\in N(i)} \sum_{k} e_{kj}\right)$$

$$x_i = g_2\left(f_i, \sum_{j\in N(i)} e^T_{ji}\right) \quad (2)$$

where e^t_{ji} is the message transmitted from node i to node j in the tth layer, f_i is the feature of node i and g_1, g_2 are neural networks. After T layers of message passing layers, the node embedding is aggregated from the last message passing layer. Note that the parameters in each message passing layer are shared. Therefore, for input pair (G_x, G_y), by processing the message passing on graphs and junction trees, we obtain the node embedding integrated from the inward messages and the node features. The paired tree and graph representations are formulated as $\{x^T_1, x^T_2, \dots, x^T_{n_T}\}, \{y^T_1, y^T_2, \dots, y^T_{m_T}\}, \{x^G_1, x^G_2, \dots, x^G_{n_G}\}, \{y^G_1, y^G_2, \dots, y^G_{m_G}\}$.

Jin et al. [9] further generates a molecule from its embedding by constructing the junction tree node by node in a depth first manner. In tth step, we obtain edge set $E^t = \{(i_1, j_1), (i_2, j_2), \dots, (i_m, j_m)\}$. For node i_t, we update the message h_{i_t,i_t} through a tree GRU proposed by Jin et al. [7]:

$$h_{i_t,j_t} = \mathrm{GRU}(f_{i_t}, \{h_{k,i_t\,(k,i_t)\in E^t, k\neq i_t}\}) \quad (3)$$

Here f_{i_t} is node feature. When the model reaches node i_t, it first decides whether to append a new node or backtrack to the ascendant of i_t. The probability is computed via a network:

$$h_t = \tau\left(W^d_1 f_{i_t} + W^d_2 \sum_{(k,i_t)\in E^t} h_{k,i_t}\right)$$

$$p_t = \sigma(W^d_3 h_t) \quad (4)$$

where $\sigma(\cdot)$ denotes Sigmoid function. If the model decides to extend a new node to the current position i_t, the label is predicted as follows:

$$q_t = Softmax(W^l_1 h_{i_t,j_t}) \quad (5)$$

Here q_t is a distribution over the node vocabulary that masks out any invalid paths of molecular generation. We employ the Junction Tree Encoder into our model but revise the decoder to accept structural conditions.

4.2. Structure conditioned ELBO

We first describe the constrained molecule optimization process of our model. Specifically, given a molecule x, the target molecule y is generated by the following process:

1. Choose a molecule $x \sim X$;
2. Choose a latent vector $z \sim \mathcal{N}(0, I)$
3. Compute the structural similarity c from x and the desired molecule y.

where $\mathcal{N}(0, I)$ is the isotropic Gaussian distribution. According to the generating process above, our model first aims to model the structural similarity c between the molecule pair (x, y) and learns a conditional distribution $p(y|x, c)$. Therefore, by Jensen's inequality, the log-likelihood can be written as:

$$\begin{aligned}\log p(Y|X, C) &= \log \textstyle\int_z p(y, z|x, c)\mathrm{d}z \\ &\geq \mathbb{E}_{z\sim q(z|x,y,c)} \log \tfrac{p(y|z,x,c)p(z|y,c)}{q(z|x,y,c)} \\ &= \mathbb{E}_{z\sim q(z|x,y,c)} \log p(y|z, x, c) \\ &\quad - \mathbb{KL}(q(z|x, y, c)|p(z|x, c)) \\ &= L_{SC}\end{aligned} \quad (6)$$

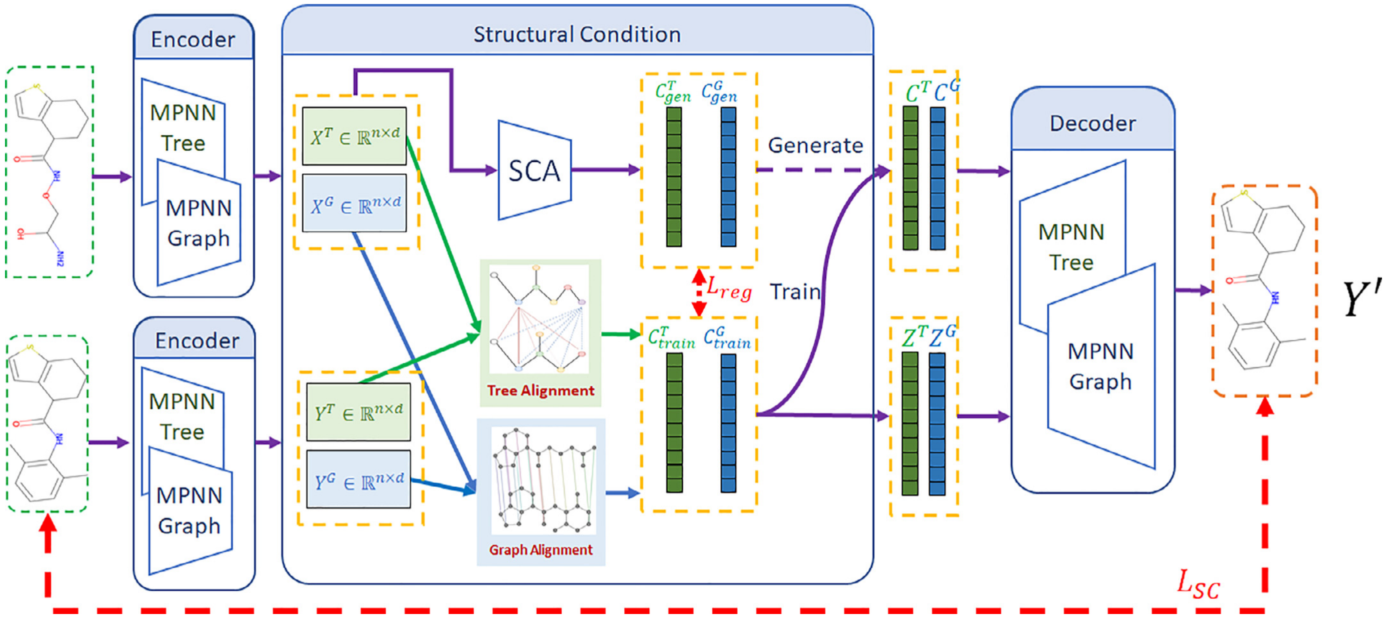

Fig. 2. The framework of SCVAE. It first extracts tree and graph node embeddings of the input pair and performs soft graph alignment at two-level to model structure conditions and outputs target molecule with desired property. We ignore the information flow from the input molecule X to the decoder and the molecule pair (X, Y) to the latent variable Z in this figure for simplicity. c^T_{train}, c^G_{train} are treated as the structure condition in training while c^T_{gen}, c^G_{gen} are extracted with a structure condition approximator in generation/optimization.

where z is the latent variable, $q(z|x, y, c)$ is the variational posterior to approximate the true posterior $p(z|x, y, c)$ and L_{SC} is the structure conditioned evidence lower bound. We assume $p(z|x, c)$ is the standard Gaussian distribution. Similar to VAE, we employ the reparameterization trick to obtain z:

$$(\mu, \log\sigma^2) = g(x, y, c; \phi_g)$$
$$z \sim \mathcal{N}(z; \mu, \sigma^2 I) \tag{7}$$

where ϕ_g is the parameter of the neural network g.

A naive implementation to this objective is to treat c as a molecule similarity metric in the real world (e.g. the Tanimoto Similarity over Morgan Fingerprints), and employs a mapping $f : (\mathcal{X}, \mathcal{Y}) \to \mathcal{R}^+$. However, this will either lead to CPU-bound methods (e.g. RDKIT Toolkit[1]) or require additional training data and a network to precisely approximate f. Therefore, we seek for modeling the similarity of molecules from their topology.

4.3. Two-level alignment representation

We first obtained tree and graph representations of paired molecules (X, Y) from the junction tree encoder from Eq. (2). Then we employ the graph alignment to exploit the structure similarity of molecules from their topology. The intuition behind this is that the node embeddings already contains the information of substructures after the message passing layers. Identifying these similar substructures is essential as it is the key to model the similarity of molecule pairs. We borrow the idea of the graph alignment to match the similar nodes. More specifically, given a node embedding x^G_i in molecule graph G^x as an example, we need to identify the most similar node y^G_j from molecule graph G^y.

As shown in Fig. 2, we employ soft graph alignment on both tree's node embeddings and graph's node embeddings to exploit two-tilevel molecule alignment representation. At tree level, for tree node x^T_i in molecule x, we compute the similarity between tree node y^T_j in molecule y and x^T_i as a probability of alignment from x_i to y_j,

$$w^T_{ji} = \frac{\exp\sigma(x^T_i, y^T_j)}{\sum_j \exp\sigma(x^T_i, y^T_j)}, \tag{8}$$

where $\sigma()$ is a affinity metric between node embeddings across x and y such as distance affinity. w^T_{ji} measures how y^T_j shares the same local pattern of substructure as x^T_i. Meanwhile, we compute the probability of x^T_i being similar to y^T_j as well:

$$w^T_{ij} = \frac{\exp\sigma(x^T_i, y^T_j)}{\sum_i \exp\sigma(x^T_i, y^T_j)} \tag{9}$$

The reason why we compute w^T_{ji} and w^T_{ij} bidirectionally is that these probabilities are normalised over different molecule graphs. Then, we assemble similar node embeddings across two molecule graphs according to the node similarity:

$$m^T_{yx} = g_1\left(\sum_i \left[x^T_i, \sum_j y^T_j w^T_{ji}\right]\right)$$
$$m^T_{xy} = g_1\left(\sum_j \left[y^T_j, \sum_i x^T_i w^T_{ij}\right]\right) \tag{10}$$

where g_1 is a neural network. By this alignment, the node representations of x and y will match to each other in the hidden space. We further use a neural network $f()$ to map m^T_{xy} and m^T_{yx} to a low dimensional alignment representation,

$$c^T_{train} = f([m^T_{xy}, m^T_{yx}]), \tag{11}$$

Similarly, we obtain the alignment representation c^G_{train} at graph level by following the above tree-level paradigm with graph node embeddings $\{x^G_1, x^G_2, \dots, x^G_{n_G}\}$, and $\{y^G_1, y^G_2, \dots, y^G_{m_G}\}$. c^G_{train} and c^T_{train} are treated as the structure similarity c in the training process.

[1] This toolkit is used for evaluating the chemical properties of molecules. https://www.rdkit.org/

4.4. Structure condition approximator

The structure alignment representation c is obtained from molecule pairs (x, y) in training according to Eq. (8). Unfortunately, the target molecule y is absent in the decoding and testing process. However, the structure alignment representation is obtained by comparing the substructure similarity of two molecules, this enlightens us to use a surrogate model to recover c from x. To leverage the structure condition in generation, we therefore introduce a structure condition approximator f_c (SCA in Fig. 2, a multi layer perceptron) to extract c from x.

$$c^T_{gen} = f_c(x^T), \quad c^G_{gen} = f_c(x^G) \tag{12}$$

the loss function is:

$$\mathcal{L}_{reg} = |c^T_{gen} - c^T_{train}|_1 + |c^G_{gen} - c^G_{train}|_1 \tag{13}$$

The new molecule is decoded from the representation of the target molecule via a probabilistic decoder $p(y|x, z, c)$. According to Eq. (3), we model the approximate posterior as isomorphic Gaussian and sample the latent vector at tree and graph level z^T and z^G. The structure alignment representation c is extracted by the structure condition approximator. Therefore, the tree and graph representation of the new molecule is $y^T = [x^T, c^T, z^T]$ and $y^G = [x^G, c^G, z^G]$. We revise the latent space of junction tree decoder to accept the structure alignment representation. The probabilistic decoder takes the two-level representation as input and outputs the target molecule. The overall loss function is,

$$L = -L_{SC} + \lambda L_{reg} \tag{14}$$

where λ is a hyper-parameter.

5. Experiments

In this section, we qualitatively and quantitatively analyse the performance of the proposed method on various constrained molecule optimization tasks.

5.1. Dataset

The experimental design of the dataset is the same as Jin et al. [9]. For constrained molecule optimization, we constrain the output molecule Y and the input X by $sim(X, Y) \geq \delta$, where $sim(X, Y)$ denotes Tanimoto Similarity over Morgan Fingerprints of two molecules. We conduct experiments on ZINC [39] and QM9 [40] datasets. The training set consists of constrained molecule pairs (X, Y) sampling from each dataset with a significant property improvement. We focus on four intrinsic properties of molecules and evaluate the proposed method on following tasks:

Penalized-logP (P-logP) is a logP value which take the inaccessibility and number of rings into account. For P-logP optimization, we first train our model with two similarity constraints $\delta = 0.4/0.6$. The training set includes 99 K/79 K molecule pairs respectively for ZINC dataset, and 33 K/21 K for QM9 dataset. The testing set contains 800 molecules for both datasets.

Drug likeness (QED) measures the drug likeness of a molecule, which is bounded within the range (0,1.0). Our task is to optimize the molecule with QED within (0.7,0.8) into a higher range (0.9,1.0). The training set includes 88K and 31 K molecule pairs with similarity constraints $\delta = 0.4$ for ZINC and QM9 dataset respectively. The testing set contains 800 molecules for both datasets.

Human β-secretase 1(BACE) measures the likelihood of a molecule to be an inhibitor of human β-secretase, which is bounded within the range (0,1.0). Our task is to optimize the molecule with BACE within (0.0,0.01) into a higher range (0.4,1.0). The training set includes 23 K and 16 K molecule pairs with similarity constraints $\delta = 0.4$ for ZINC and QM9 dataset respectively. The test set has 800 molecules for both datasets.

Table 1
P-logP optimization results on ZINC dataset. We report property improvement and diversity over the successfully optimized candidates.

Method	$\delta \geq 0.6$		$\delta \geq 0.4$	
	Improvement	Diversity	Improvement	Diversity
MMPA	1.65 ± 1.44	0.329	3.29 ± 1.12	0.496
JT-VAE	0.28 ± 0.79	–	1.03 ± 1.39	–
GCPN	0.79 ± 0.63	–	2.49 ± 1.30	–
VSeq2Seq	2.33 ± 1.17	0.331	3.37 ± 1.75	0.471
VJTNN	2.33 ± 1.24	**0.333**	3.55 ± 1.67	0.480
SCVAE(L1)	**2.66** ± **1.25**	0.308	**3.95** ± **1.46**	**0.512**
SCVAE(L2)	2.57 ± 1.17	0.291	3.89 ± 1.57	0.502

Brain-blood barrier penetration(BBBP) measures the likelihood of a molecule to penetrate the brain-blood barrier, which is bounded within the range (0,1.0). Our task is to optimize the molecule with BBBP within (0.0,0.3) into a higher range (0.9,1.0). The training set includes 31 K and 22 K molecule pairs with similarity constraints $\delta = 0.4$ for ZINC and QM9 dataset respectively. The test set has 800 molecules for both datasets.

5.2. Baselines

We compare the proposed method with the baselines as follows, **MMPA** [41,42] aims to discover underlying rules to mostly improve the molecule property. **JT-VAE** [7] is the state-off-the-art molecule generative method as it first achieves 100% chemically valid output molecules. For optimization tasks, it searches the optimized molecules by multi-step gradient ascend with respect to the property in the latent space. **VSeq2Seq** [43] employs molecule SMILES strings and learns a sequence-to-sequence translation model, with latent code added into the architecture of Bahdanau et al. [44]. **VJTNN** [9], built upon JT-VAE, is trained with molecule pairs and optimizes a molecule in a one-step manner. **GCPN** [8] combines reinforcement learning and graph neural network to generate a molecule by adding atoms and bonds iteratively. Moreover, adversarial learning is employed to generate realistic molecules.

5.3. Evaluation metrics

Following the same testing protocol proposed by Jin et al. [9], every input molecule is decoded in 20 times with different latent vector z sampled from $\mathcal{N}(0, I)$. For JT-VAE, we do 80 steps of gradient ascent to improve the target properties in the latent space, and choose the one with the best properties. For the task of P-logP optimization, the reported molecules are the successfully optimized molecules with the highest property improvement and under the similarity constraint. We further calculate pairwise average Tanimoto distance over the valid optimized molecules as $dist(X, Y) = 1 - sim(X, Y)$. This metric measures how many diverse molecules the models generate with an input, which refers to Diversity or Div. in Tables 1 and 2 respectively. This is important to measure the capacity of different models. Although constrained molecule optimization only allows limited modification to molecular structures, good models can still exhibit the diversity of such modification. For the tasks of QED, BACE and BBBP optimization, we define a molecule is successfully translated if there is one output that meets the similarity constraint and the property is in the desired range. We report diversity and measure the rate of molecules being successfully translated, which refers to the translation accuracy. Moreover, we report the novelty of the molecules which are the rate of the successfully optimized molecules unseen in the training set.

Table 2
QED, BACE and BBBP optimization results on ZINC dataset. We rerun the baselines on BACE and BBBP tasks under our settings. The other baseline results are copied from Jin et al. [9].

	QED			BACE			BBBP		
Method	Acc.	Div.	Nov.	Acc.	Div.	Nov.	Acc.	Div.	Nov.
MMPA	32.9%	0.236	99.9%	15.3%	0.276	100.0%	31.6%	0.182	99.9%
JT-VAE	8.8%	–	–	1.7%	–	–	2.4%	–	–
GCPN	9.4%	0.216	**100.0%**	9.6%	0.209	100.0%	22.7	0.193	100.0%
VSeq2Seq	**58.5%**	0.331	99.6%	18.9%	0.254	99.1%	35.1%	0.215	100.0%
VJTNN	57.0%	0.389	98.1%	17.6%	0.288	100.0%	38.3%	0.263	100.0%
SCVAE(L1)	58.2%	0.401	97.9%	**23.3%**	**0.291**	**100.0%**	**41.9%**	**0.284**	**100.0%**
SCVAE(L2)	58.1%	**0.423**	98.6%	21.6%	0.274	100.0%	40.7%	0.262	100.0%

Table 3
P-logP optimization results on QM9 dataset. We report property improvement and diversity over the successfully optimized candidates.

Method	$\delta \geq 0.6$		$\delta \geq 0.4$	
	Improvement	Diversity	Improvement	Diversity
MMPA	1.31 ± 1.05	0.296	2.31 ± 1.29	0.401
JT-VAE	0.34 ± 0.31	–	1.01 ± 0.96	–
GCPN	0.67 ± 0.58	–	1.50 ± 1.24	–
VSeq2Seq	2.18 ± 1.31	**0.312**	2.56 ± 1.67	0.423
VJTNN	2.16 ± 1.16	0.301	2.61 ± 1.79	0.441
SCVAE(L1)	**2.31 ± 1.04**	0.257	**3.12 ± 1.36**	**0.451**
SCVAE(L2)	2.24 ± 1.14	0.242	2.96 ± 1.61	0.445

Table 4
QED, BACE and BBBP optimization results on QM9 dataset.

	QED			BACE			BBBP		
Method	Acc.	Div.	Nov.	Acc.	Div.	Nov.	Acc.	Div.	Nov.
MMPA	29.6%	0.229	99.9%	17.5%	0.236	100.0%	25.9%	0.171	99.9%
JT-VAE	9.3%	–	–	2.3%	–	–	2.9%	–	–
GCPN	10.2%	0.246	**100.0%**	16.4%	0.216	100.0%	15.1%	0.173	100.0%
VSeq2Seq	48.9%	0.343	99.4%	21.2%	0.260	99.8%	36.7%	0.245	100.0%
VJTNN	46.5%	0.379	99.2%	23.7%	0.263	100.0%	39.4%	0.247	100.0%
SCVAE(L1)	**51.9%**	**0.384**	98.7%	**26.3%**	**0.268**	**100.0%**	**42.0%**	**0.257**	**100.0%**
SCVAE(L2)	48.7%	0.381	99.1%	25.8%	0.252	100.0%	41.2%	0.249	100.0%

5.4. Performance

Tables 1 and 3 shows the results of P-logP optimization. Our model outperforms the baseline methods on property improvement by a large margin and diversity. Particularly, our model achieves significant improvement comparing to traditional CVAE-based methods including JT-VAE, VJTNN and VSeq2Seq. It is because the proposed method leverages the topology similarity of molecule pairs as condition and formulates it as the graph alignment. Moreover, our unsupervised soft graph alignment strategy facilitates the model to effectively explore the chemical space for better molecules.

Tables 2 and 4 demonstrates QED, BACE and BBBP optimization results. As MMPA and VSeq2Seq are not initially proposed for constrained molecule optimization, we modify them following the experiment details in Jin et al. [9] for fair comparison. SCVAE performs favorably than the other CVAE-based methods in translation accuracy, diversity and novelty. This indicates our model often generates new and diverse outputs with high translation accuracy. Notice that all methods performs better on the ZINC dataset than the QM9 dataset, since we obtain more training data on the former dataset.

We further discuss the influence of different distance metrics in Eq. (13) on SCVAE. In experiment, we employ L1 and L2 loss in Eq. (13) to train the structure condition approximator f_c respectively. We find L1 loss is more stable than L2 loss, since L2 loss is prone to result in a trivial solution that $c^T_{gen} = c^T_{train} = 0, c^G_{gen} = c^G_{train} = 0$ in practice. This trivial solution minimizes the loss in Eq. (13), but the structure similarities $c^T_{gen}, c^T_{train}, c^G_{gen}, c^G_{train}$ are equal to zero and provide no information for the structure constraint, and thus leads to degenerated optimization results. Meanwhile, we find L1 loss is more stable and less likely to yield such trivial solution. Therefore, it is recommended to employ L1 loss in Eq. (13) to train the structure condition approximator f_c. In the experiment, we replace the L1 loss by the L2 loss in Eq. (13) and train the whole framework for several times until it obtains reasonable results. We denote SCVAE with L1 loss and L2 loss as SCVAE(L1) and SCVAE(L2) respectively. As shown in Tables 1–4, SCVAE(L1) outperforms SCVAE(L2) on all datasets. However, SCVAE(L2) still achieves competitive results compared with the baselines. This shows that the performance of our model is robust to different distance metrics. However, we employ L1 loss instead of L2 loss in Eq. (13) to prevent from numerical instability.

Recently, it is popular to employ the subgraph-based optimization methods for constrained molecule optimization. However, these methods are two-step and extracting property-related subgraphs is non-trivial. Compared with the subgraph-based methods, our method is **end-to-end**, and thus is more efficient to train [45]. The major difficulty of constrained molecule optimization is the large chemical space and discrete nature of molecules. Our method formulates the optimization problem as a graph-to-graph translation, and learns a mapping from the input molecule to those with improved property. The whole framework is trained on paired training data, and thus reduce the searching space. To preserve structural similarity, our model first leverages the graph alignment of two-level molecule structures to model the similarity C of pair molecules (X, Y). Then, a structure-aware VAE models the optimization problem as learning a conditional distribution $p(Y|X, C)$.

Table 5
Sensitivity study of the hyperparameter λ on the P-logp Optimization task on ZINC dataset.

λ	SCVAE(L1)		SCVAE(L2)	
	Improvement	Diversity	Improvement	Diversity
0.01	3.72 ± 1.34	0.496	3.79 ± 1.31	0.500
0.05	3.83 ± 1.47	0.520	3.85 ± 1.49	0.513
0.1	3.95 ± 1.46	0.512	3.82 ± 1.47	0.492
0.5	3.92 ± 1.54	0.517	3.89 ± 1.57	0.502
1	3.85 ± 1.39	0.525	3.80 ± 1.52	0.497
5	3.91 ± 1.41	0.511	3.82 ± 1.45	0.511

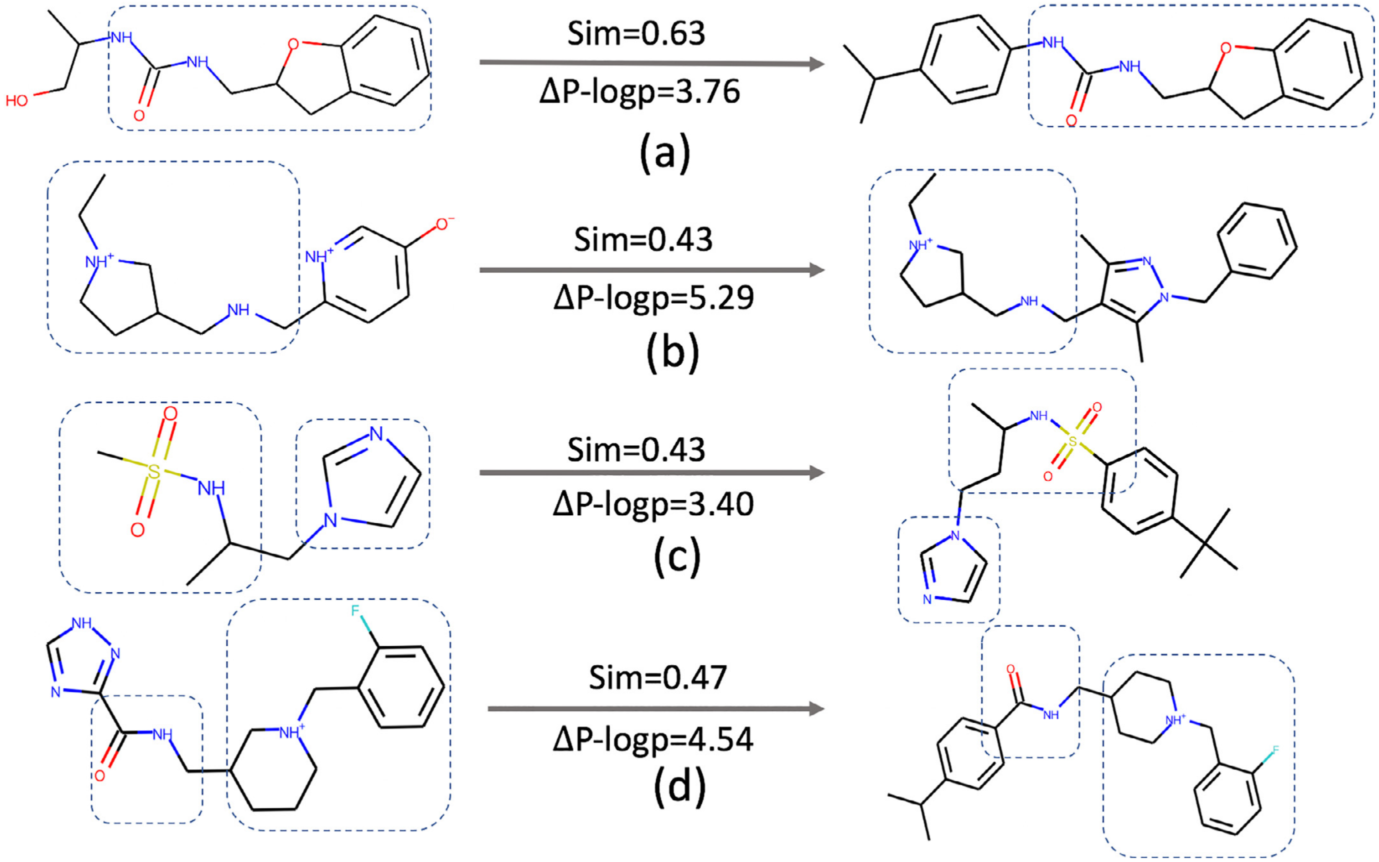

Fig. 3. Optimization results of P-logP task. The input and the output molecules has the same structure in the dashed line. In (a) and (b), SCVAE preserves the scaffold and replaces the hydrophilic-group with lipophilic-group. In (c) and (d), SCVAE mainly changes the connectivity of subgraphs and adds lipophilic-group to improve P-logp.

Once the model is trained, one can efficiently sample the optimized molecules with diverse structures.

5.5. Sensitivity study

We further analyze the influence of hyper-parameter λ, which controls the trade-off between two terms in Eq. (14). We rerun our model with λ varies in $\{0.01, 0.05, 0.1, 0.5, 1, 5\}$ on the P-logp task on ZINC dataset. The result is shown in Table 5, SCVAE(L1) and SCVAE(L2). We notice that a small λ will lead to a slight decline in performance, since a small penalty on Eq. (13) is insufficient for the structure condition approximator f_c to approximate the structure similarity in the decoding process. Meanwhile, a large λ pose a large penalty on Eq. (13) lead to the over-fitting of f_c, and thus yields degenerated results. Moreover, a large λ also lead to a unstable training process and is likely to result in a trivial solution $c^T_{gen} = c^T_{train} = 0, c^G_{gen} = c^G_{train} = 0$. Hence, these similarity embeddings provide no information for the structure condition. In practice, we choose a proper λ to balance the scale of two terms in Eq. (14) for the best result.

5.6. Visualization

We further visualize the structure modification in the constrained optimization process in Fig. 3. It should be noted that these results are not hand-made solutions. The output molecules have large property improvement over the input ones while the structures in the dashed line remains the same. In Fig. 3(a) and (b), our model preserves the scaffolds of the input molecules and simply replaces the hydrophilic-group with the lipophilic-group. This is known as an effective way to improve the P-logP of molecules. In Fig. 3(c) and (d), our model changes the connectivity of subgraphs and adds the lipophilic-group to improve the P-logP value. This indicates that SCVAE can yield reasonable and explainable solutions with limited modification in molecule structure.

To further investigate the validity of soft graph alignment for topology similarity, we visualize the substructure embeddings of a pair of molecules with similar structure via T-SNE [46] for SCVAE and VJTNN. As shown in Fig. 4, the first two columns are molecules with similar structure; the second two columns are the plots of node representations of molecule pairs in the latent space

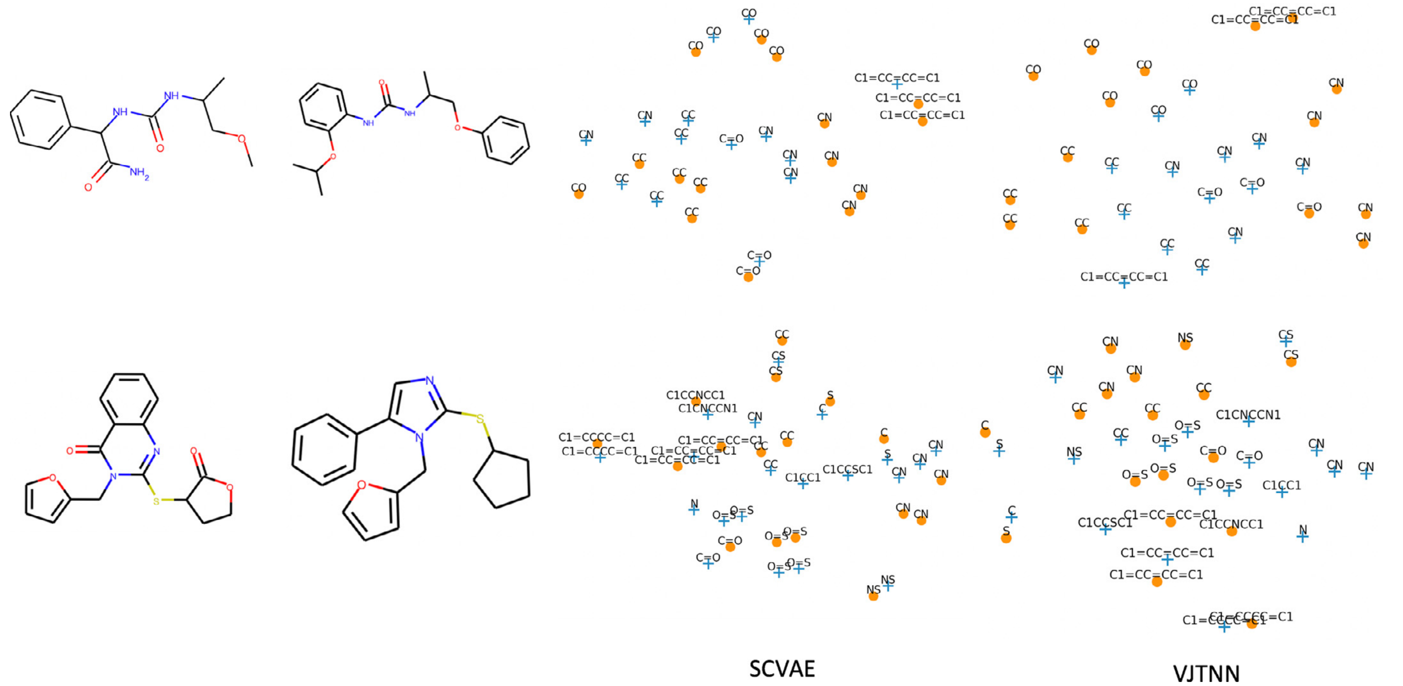

Fig. 4. Visualization of molecule representations. The dots with different shapes represent node embeddings in different molecules.

for our model and VJTNN. Within over 350 different subgraphs in the training pairs, SCVAE still aligns most similar subgraphs to exploit structure similarity across different molecules. Therefore, the proposed method indeed leverages structure similarity as a condition for constrained molecule optimization.

6. Conclusion

In this paper, we have proposed a novel structure-aware conditional VAE framework, namely SCVAE. SCVAE leverages the structure similarities in the molecule pairs as a condition to facilitate the constrained molecule optimization. To do this, viewing molecular graph as a structural condition, SCVAE utilizes graph alignment of tree and graph level molecule structures in an unsupervised manner to bind the structure conditions between two molecules. Then, this structure condition bridges the gap between the input and output molecules via a probabilistic encoder-decoder architecture. By optimizing a structure conditioned evidence lower bound, SCVAE is capable of effectively exploring the chemical space for better molecules under the similarity constraint. Qualitative and quantitative results show SCVAE outputs the molecules with better properties as well as limited structure modification.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is partially funded by National Natural Science Foundation of China (Grant No. U21B2045), National Natural Science Foundation of China (Grant No. U20A20223), and Youth Innovation Promotion Association CAS (Grant No. Y201929).

References

[1] K. Madhawa, K. Ishiguro, K. Nakago, M. Abe, Graphnvp: an invertible flow model for generating molecular graphs, arXiv preprint arXiv:1905.11600 (2019).
[2] C. Shi, M. Xu, Z. Zhu, W. Zhang, M. Zhang, J. Tang, Graphaf: a flow-based autoregressive model for molecular graph generation, in: International Conference on Learning Representations, 2020, pp. 1–12.
[3] B. Chen, T. Wang, C. Li, H. Dai, L. Song, Molecule optimization by explainable evolution, in: International Conference on Learning Representations, 2020, pp. 1–12.
[4] P. Kirkpatrick, C. Ellis, Chemical space, Nature 432 (7019) (2004) 823–824.
[5] M.J. Kusner, B. Paige, J.M. Hernández-Lobato, Grammar variational autoencoder, in: International Conference on Machine Learning, PMLR, 2017, pp. 1945–1954.
[6] H. Dai, Y. Tian, B. Dai, S. Skiena, L. Song, Syntax-directed variational autoencoder for structured data, in: International Conference on Learning Representations, 2018, pp. 1–17.
[7] W. Jin, R. Barzilay, T. Jaakkola, Junction tree variational autoencoder for molecular graph generation, in: International Conference on Machine Learning, PMLR, 2018, pp. 2323–2332.
[8] J. You, B. Liu, R. Ying, V. Pande, J. Leskovec, Graph convolutional policy network for goal-directed molecular graph generation, in: Advances in Neural Information Processing Systems, 2018, pp. 1–12.
[9] W. Jin, K. Yang, R. Barzilay, T. Jaakkola, Learning multimodal graph-to-graph translation for molecule optimization, in: International Conference on Learning Representations, 2018, pp. 1–12.
[10] A. Mollaysa, B. Paige, A. Kalousis, Conditional generation of molecules from disentangled representations, https://openreview.net/forum?id=BkxthxHYvr (2020).
[11] M. Heimann, H. Shen, T. Safavi, D. Koutra, Regal: representation learning-based graph alignment, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 117–126.
[12] A. Yasar, Ü.V. Çatalyürek, An iterative global structure-assisted labeled network aligner, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2614–2623.
[13] A. Seff, W. Zhou, F. Damani, A. Doyle, R.P. Adams, Discrete object generation with reversible inductive construction, in: Advances in Neural Information Processing Systems, 2019, pp. 1–12.
[14] J. Yu, T. Xu, Y. Rong, Y. Bian, J. Huang, R. He, Graph information bottleneck for subgraph recognition, in: International Conference on Learning Representations, 2021, pp. 1–18.
[15] J. Yu, T. Xu, Y. Rong, Y. Bian, J. Huang, R. He, Recognizing predictive substructures with subgraph information bottleneck, IEEE Trans. Pattern Anal. Mach. Intell. (2021) 1–14.

[16] W. Jin, R. Barzilay, T. Jaakkola, Multi-objective molecule generation using interpretable substructures, in: International Conference on Machine Learning, PMLR, 2020, pp. 4849–4859.
[17] K. Yang, W. Jin, K. Swanson, R. Barzilay, T. Jaakkola, Improving molecular design by stochastic iterative target augmentation, in: International Conference on Machine Learning, PMLR, 2020, pp. 10716–10726.
[18] E. Malmi, A. Gionis, E. Terzi, Active network alignment: a matching-based approach, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 1687–1696.
[19] T. Derr, H. Karimi, X. Liu, J. Xu, J. Tang, Deep adversarial network alignment, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 352–361.
[20] X. Chu, X. Fan, D. Yao, Z. Zhu, J. Huang, J. Bi, Cross-network embedding for multi-network alignment, in: The World Wide Web Conference, 2019, pp. 273–284.
[21] Z. Sun, C. Wang, W. Hu, M. Chen, J. Dai, W. Zhang, Y. Qu, Knowledge graph alignment network with gated multi-hop neighborhood aggregation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 222–229.
[22] N. Weskamp, E. Hullermeier, D. Kuhn, G. Klebe, Multiple graph alignment for the structural analysis of protein active sites, IEEE/ACM Trans. Comput. Biol. Bioinf. 4 (2) (2007) 310–320.
[23] J. Berg, M. Lässig, Local graph alignment and motif search in biological networks, Proc. Natl. Acad. Sci. 101 (41) (2004) 14689–14694.
[24] L. Bai, L. Cui, Y. Jiao, L. Rossi, E. Hancock, Learning backtrackless aligned-spatial graph convolutional networks for graph classification, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2020) 783–798.
[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, vol. 27, 2014, pp. 1–12.
[26] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, arXiv preprint arXiv:1312.6114 (2013).
[27] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784 (2014).
[28] K. Sohn, H. Lee, X. Yan, Learning structured output representation using deep conditional generative models, in: Advances in neural Information Processing Systems, vol. 28, 2015, pp. 3483–3491.
[29] J. Yu, J. Cao, Y. Li, X. Jia, R. He, Pose-preserving cross spectral face hallucination, in: International Joint Conference on Artificial Intelligence, 2019, pp. 1018–1024.
[30] L. Gao, D. Chen, Z. Zhao, J. Shao, H.T. Shen, Lightweight dynamic conditional GAN with pyramid attention for text-to-image synthesis, Pattern Recognit. 110 (2021) 107384.
[31] R. He, Y. Li, X. Wu, L. Song, Z. Chai, X. Wei, Coupled adversarial learning for semi-supervised heterogeneous face recognition, Pattern Recognit. 110 (2021) 107618.
[32] J. Zhao, C. Shi, F. Jia, Y. Wang, B. Xiao, Document image binarization with cascaded generators of conditional generative adversarial networks, Pattern Recognit. 96 (2019) 106968.
[33] L. Kang, P. Riba, M. Rusinol, A. Fornes, M. Villegas, Content and style aware generation of text-line images for handwriting recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2021) 1–15.
[34] T. Zhao, R. Zhao, M. Eskenazi, Learning discourse-level diversity for neural dialog models using conditional variational autoencoders, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 654–664.
[35] T. Wang, X. Wan, T-CVAE: transformer-based conditioned variational autoencoder for story completion, in: International Conference on Joint Artificial Intelligence, 2019, pp. 5233–5239.
[36] Y. Li, L. Song, X. Wu, R. He, T. Tan, Learning a bi-level adversarial network with global and local perception for makeup-invariant face verification, Pattern Recognit. 90 (2019) 99–108.
[37] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, Infogan: interpretable representation learning by information maximizing generative adversarial nets, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 2180–2188.
[38] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, beta-vae: learning basic visual concepts with a constrained variational framework, in: International Conference on Learning Representations, 2017, pp. 1–12.
[39] T. Sterling, J.J. Irwin, Zinc 15–ligand discovery for everyone, J. Chem. Inf. Model. 55 (11) (2015) 2324–2337.
[40] B. Ramsundar, P. Eastman, P. Walters, V. Pande, Deep Learning for the Life Sciences: Applying Deep Learning to Genomics, Microscopy, Drug Discovery, and More, O'Reilly Media, Inc., 2019.
[41] E. Griffen, A.G. Leach, G.R. Robb, D.J. Warner, Matched molecular pairs as a medicinal chemistry tool: miniperspective, J. Med. Chem. 54 (22) (2011) 7739–7750.
[42] A. Dalke, J. Hert, C. Kramer, mmpdb: an open-source matched molecular pair platform for large multiproperty data sets, J. Chem. Inf. Model. 58 (5) (2018) 902–910.
[43] R. Gómez-Bombarelli, J.N. Wei, D. Duvenaud, J.M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T.D. Hirzel, R.P. Adams, A. Aspuru-Guzik, Automatic chemical design using a data-driven continuous representation of molecules, ACS. Cent. Sci. 4 (2018) 268–276.
[44] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).
[45] W. Jin, B. Regina, T. Jaakkola, Hierarchical graph-to-graph translation for molecules, arXiv preprint arXiv:1907.11223.pdf (2019).
[46] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008) 2579–2605.

Junchi Yu is a M.S. student at the Institute of Automation, Chinese Academy of Sciences, Beijing, China. He obtained the B.E. degree from Wuhan University, Wuhan, China. His main research interests include generative models, graph neural networks, and graph generations.

Tingyang Xu is a Senior researcher of Machine Learning Center in Tencent AI Lab. He obtained the Ph.D. degree from The University of Connecticut in 2017 and joined Tencent AI Lab in July 2017. In Tencent AI Lab. His main research interests include social network analysis, graph neural networks, and graph generations.

Yu Rong is a Senior researcher of Machine Learning Center in Tencent AI Lab. He received the Ph.D. degree from The Chinese University of Hong Kong in 2016 and joined Tencent AI Lab in June 2017. His main research interests include social network analysis, graph neural networks, and large-scale graph systems.

Junzhou Huang is an Associate Professor in the Computer Science and Engineering department at the University of Texas at Arlington. He also serves as the director of machine learning center in Tencent AI Lab. He received the B.E. degree from Huazhong University of Science and Technology, the M.S. degree from the Institute of Automation, Chinese Academy of Sciences, and the Ph.D. degree in Computer Science at Rutgers, The State University of New Jersey. His major research interests include machine learning, computer vision and imaging informatics.

Ran He is a Professor in Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences. He received the B.E. and M.S. degrees in computer science from Dalian University of Technology, and the Ph.D. degree in Pattern Recognition and Intelligent Systems from the Institute of Automation, Chinese Academy of Sciences. He currently serves as a member of Editor Board of Pattern Recognition (Elsevier), and Entropy. His research interests include information theoretic learning, pattern recognition, and computer vision.