Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation

Li, Ziyao & Ma, Xiaojun; May 14, 2020

Contents

01 Preliminaries & Related Work

02 Markovian Graph Generation

03 Model Details: RL setup in GCPN

04 Experiments & Observations

Preliminaries & Related Work

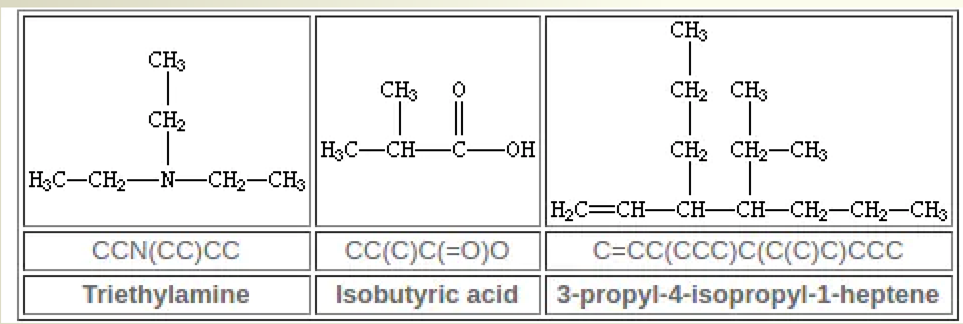
A Brief Introduction of
Molecule Graph Generation

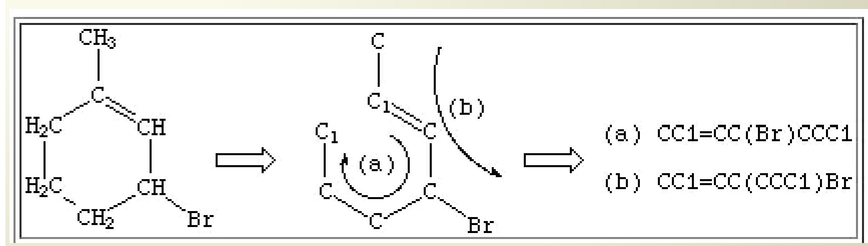
Molecule Graphs (MolGraph)

Representation of Molecules

- Sequential representation: SMILES
- Graph representation: structural formula

(recap high school chemistry)





Graph ConvNets (GCNs)

北京大學 PEKING UNIVERSITY

GCN Convolves over node neighborhoods.

- As Matrix Multiplication:

$$H^{(t+1)} = \sigma\left(AH^{(t)}W^{(t)}\right)$$

- As Neighborhood Aggregation:

$$H_u^{(t+1)} = \sigma\left(\sum_{v \in N(u)} H_v^{(t)} W^{(t)} \right)$$

(two flips of one coin)

MolGraph Generation

Related work in MolGraph Generation (GCN based).

MolGraph + VAE = GraphVAE (2018)

MolGraph + VAE + structural priors = JT-VAE (2018)

MolGraph + GAN = MolGAN (2018)

MolGraph + Flow = GraphNVP (2019); GraphAF (2020)

MolGraph + RL + GAN = GCPN (2018)

(arXiv year)

Markovian Graph Generation (MDP)

Motivations & Basic Schemes Behind GCPN

MolGraph Gen. as MDP

北京大學 PEKING UNIVERSITY

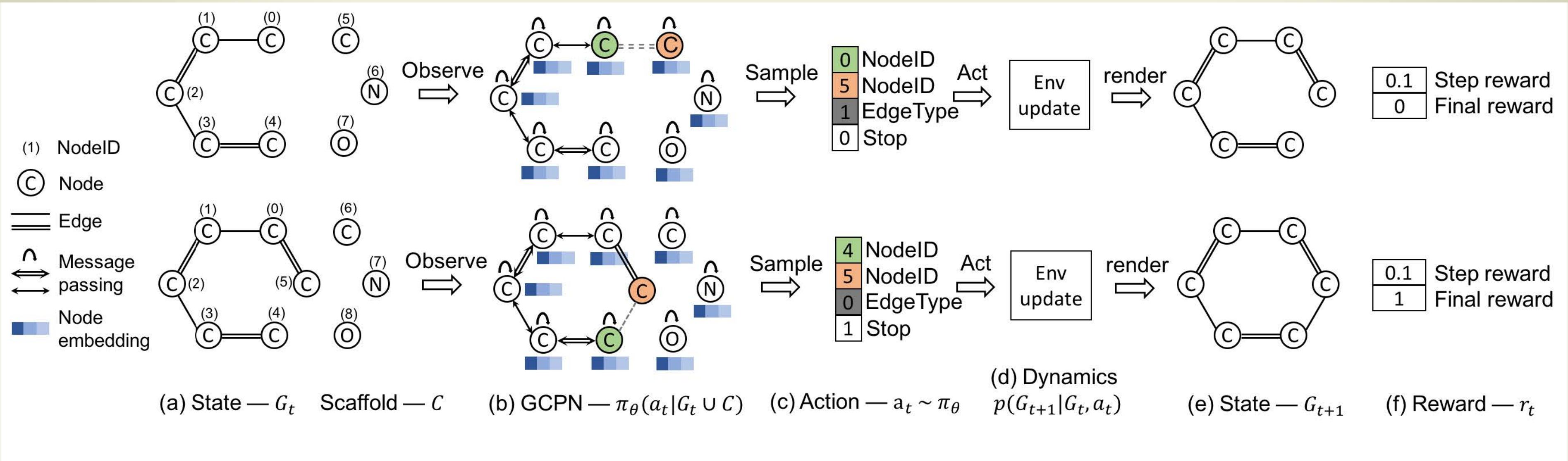
Markovian Decision Process

- **Markovian Condition:**

$$p(s_{t+1}|s_t, s_{t-1}, \cdots) = p(s_{t+1}|s_t)$$

- In GCPN, this condition means in the generation process, each step is determined fully by the <u>intermediately generated molecule</u>.

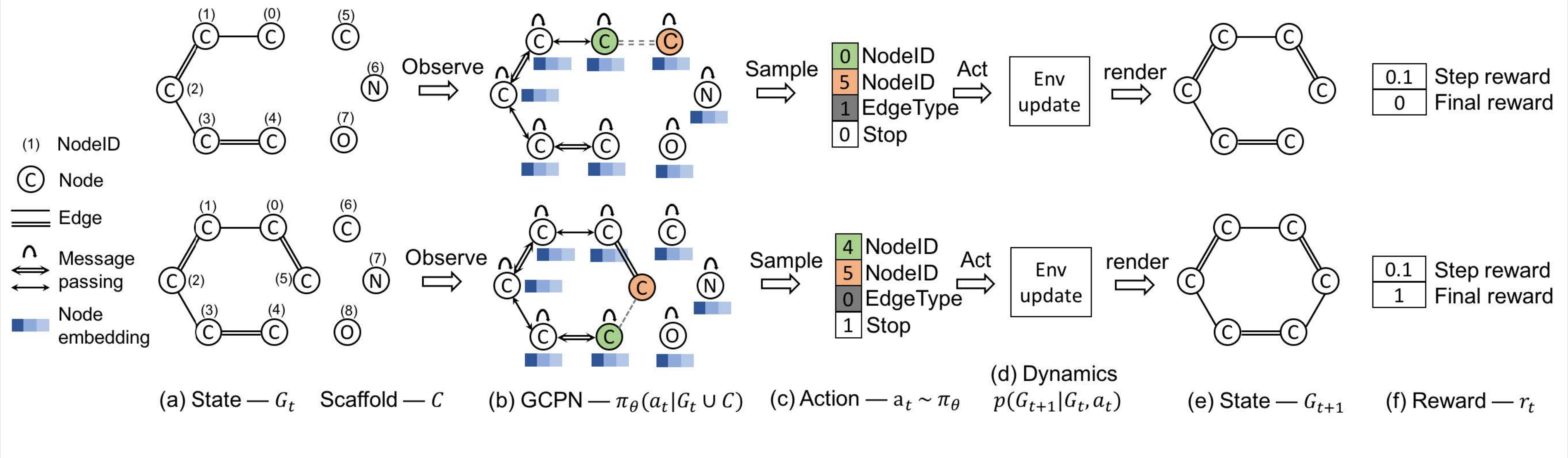
GCPN Overview



Overall techniques: RL + Adversarial Loss

- Sequentially constructing a MolGraph Using MDP
- The agent gain rewards according to chem. rules and GAN validity

GCPN Overview



In each step, an RL agent

- either connect a new atom to an existing one
- or connect two existing atoms.
- (as well as) choose whether to stop.

Model Details: RL setup in GCPN

Details *(brief, actually)*
in the RL Implementation of GCPN

Generation Scheme

state

Using GCN to encode intermediately generated molecule, as the observed state for the agent.

action

4-step action is sampled, based on the GCN embeddings:
(1) candidate atom;
(2) target atom;
(3) bond type;
(4) stop or not.

dynamics

Rendering the graph after each action is sampled – and reject infeasible actions.

rewards

Step rewards and Final rewards

(1) Step reward: (small) reward of mol. including validity reward and GAN reward

(2) Final rewards: molecular properties metrics including – logP; QED; MW; ZINC filters…

stop

Output generated molecule.

State

(Heterogenous) GCN state encoder.

- $$H^{(l+1)} = \text{AGG}(\text{ReLU}(\{\tilde{D}_i^{-\frac{1}{2}}\tilde{E}_i\tilde{D}_i^{-\frac{1}{2}}H^{(l)}W_i^{(l)}\}, \forall i \in (1, ..., b)))$$

- Note: the AGG(·) is conducted for different bond types.

  (distinguish with graph pooling / neighborhood aggregation)

Action

Four-step action generated from states.

Generated & sampled sequentially.

$$
\begin{aligned}
a_{t+1} &= \text{CONCAT}(a_{\text{first}}, a_{\text{second}}, a_{\text{edge}}, a_{\text{stop}}) \\
f_{\text{first}}(s_t) &= \text{SOFTMAX}(m_f(X)), \\
f_{\text{second}}(s_t) &= \text{SOFTMAX}(m_s(X_{a_{\text{first}}}, X)), \\
f_{\text{edge}}(s_t) &= \text{SOFTMAX}(m_e(X_{a_{\text{first}}}, X_{a_{\text{second}}})), \\
f_{\text{stop}}(s_t) &= \text{SOFTMAX}(m_t(\text{AGG}(X))),
\end{aligned}
$$

$$
\begin{aligned}
a_{\text{first}} &\sim f_{\text{first}}(s_t) \in \{0, 1\}^n \\
a_{\text{second}} &\sim f_{\text{second}}(s_t) \in \{0, 1\}^{n+c} \\
a_{\text{edge}} &\sim f_{\text{edge}}(s_t) \in \{0, 1\}^b \\
a_{\text{stop}} &\sim f_{\text{stop}}(s_t) \in \{0, 1\}
\end{aligned}
$$

Dynamics

PEKING UNIVERSITY

Produce ("render") the graph according to the sampled action. Reject infeasible actions. (Deterministic state trans. dynamics.)

Reward

Step (small) **& Final** (large) **Reward**

- Step reward 1: validity. Penalize infeasible actions.
- Step reward 2: -1 * GAN Loss. Penalize “weird” samples.
- Final reward: molecular properties –
  - logP: octanol-water partition coefficient (脂水分配系数)
  - QED: druglikeness （类药性）
  - MW: molecular weights （分子量）
  - Penalty over unrealistic molecules （经验标准）
  - Zinc fn. group filters: (生医标准，来自于商用药物分子库 ZINC)

Training

Policy Gradient + Pretraining.

- **Proximal Policy Optimization (PPO)**

  We will not unpack it here.
  See ref. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. CoRR, abs/1707.06347, 2017.

- Pretraining using generation trajectory of true molecules.

  (could be very important !)

Experiments & Observations

Some Brief Results of GCPN

Experiments

- **Property optimization**

  Gen. mols with some property optimized.

- **Property Targeting**

  Gen. mols with some property closed to a given target.

- **Constraint Property Optimization**

  Conduct property optimization with the molecule contains a given substructure.

Comp. v.s. SOTA (JT-VAE / ORGAN)

北京大學 PEKING UNIVERSITY

Table 1: Comparison of the top 3 property scores of generated molecules found by each model.

Method	Penalized logP				QED			
	1st	2nd	3rd	Validity	1st	2nd	3rd	Validity
ZINC	4.52	4.30	4.23	100.0%	0.948	0.948	0.948	100.0%
ORGAN	3.63	3.49	3.44	0.4%	0.896	0.824	0.820	2.2%
JT-VAE	5.30	4.93	4.49	100.0%	0.925	0.911	0.910	100.0%
GCPN	**7.98**	**7.85**	**7.80**	**100.0%**	**0.948**	**0.947**	**0.946**	**100.0%**

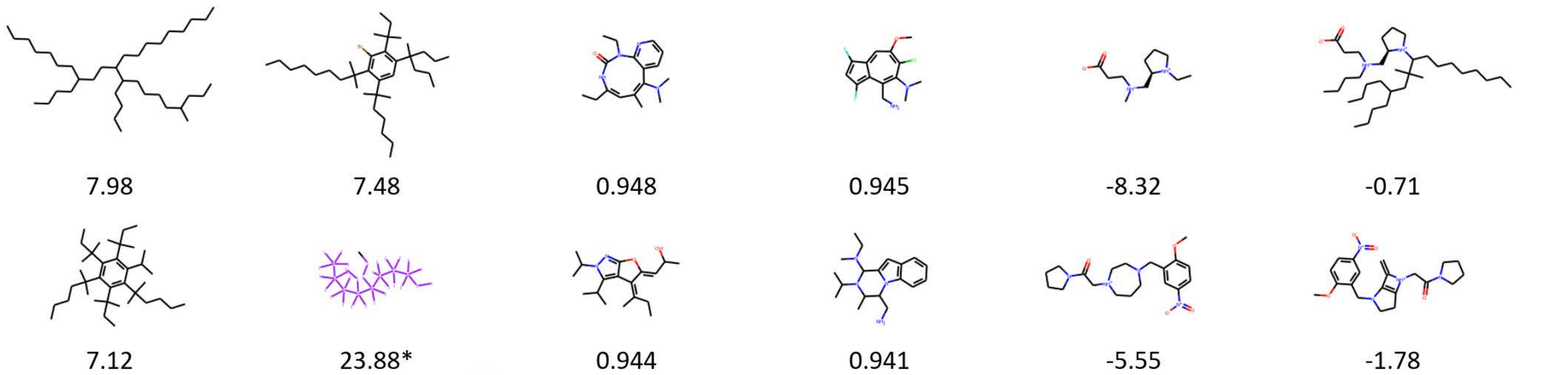
Table 2: Comparison of the effectiveness of property targeting task.

Method	$-2.5 \le \text{logP} \le -2$		$5 \le \text{logP} \le 5.5$		$150 \le \text{MW} \le 200$		$500 \le \text{MW} \le 550$	
	Success	Diversity	Success	Diversity	Success	Diversity	Success	Diversity
ZINC	0.3%	0.919	1.3%	0.909	1.7%	0.938	0	–
JT-VAE	11.3%	**0.846**	7.6%	0.907	0.7%	0.824	16.0%	0.898
ORGAN	0	–	0.2%	**0.909**	15.1%	0.759	0.1%	0.907
GCPN	**85.5%**	0.392	**54.7%**	0.855	**76.1%**	**0.921**	**74.1%**	**0.920**

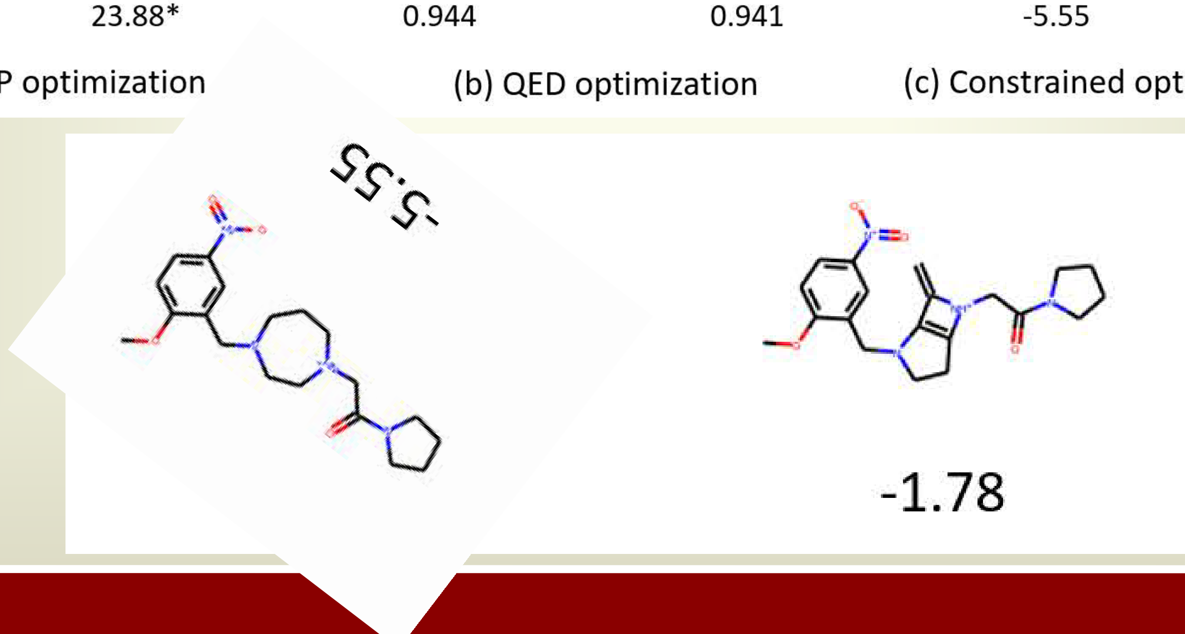
Table 3: Comparison of the performance in the constrained optimization task.

δ	JT-VAE			GCPN		
	Improvement	Similarity	Success	Improvement	Similarity	Success
0.0	1.91 ± 2.04	0.28 ± 0.15	97.5%	$\mathbf{4.20 \pm 1.28}$	$\mathbf{0.32 \pm 0.12}$	**100.0%**
0.2	1.68 ± 1.85	0.33 ± 0.13	97.1%	$\mathbf{4.12 \pm 1.19}$	$\mathbf{0.34 \pm 0.11}$	**100.0%**
0.4	0.84 ± 1.45	$\mathbf{0.51 \pm 0.10}$	83.6%	$\mathbf{2.49 \pm 1.30}$	0.47 ± 0.08	**100.0%**
0.6	0.21 ± 0.71	0.69 ± 0.06	46.4%	$\mathbf{0.79 \pm 0.63}$	$\mathbf{0.68 \pm 0.08}$	**100.0%**

Some demo.



(a) Penalized logP optimization (b) QED optimization (c) Constrained optimization of penalized logP



Summary of Observation

- **Indeed, GCPN produced very “valid” results.**
- **This is probably the results of its elaborated process of generation, including the trajectory pretraining & GAN loss, while the diversity is somehow harmed.**
- **No chem. or bio. motivation is adequately combined in the generation process. Therefore, some actions of the agent seems bizarre.**

Thanks

References

(All available in arXiv so only arXiv ID provided.)

1707.06347 – PPO

1802.03480 – GraphVAE

1802.04364 – JT-VAE

1805.11973 – MolGAN

1806.02473 – GCPN

1905.11600 – GraphNVP

2001.09382 – GraphAF