Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation

Jiaxuan You Bowen Liu

Rex Ying Vijay Pande Jure Leskovec

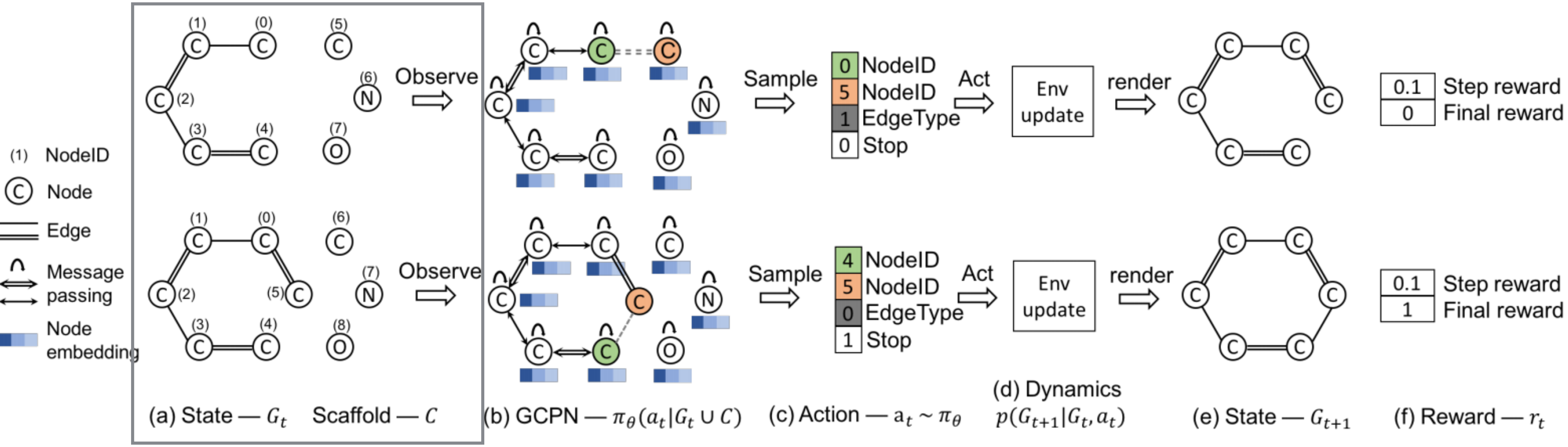
Presenter: Yevgeny Tkach

2019 Spring @
https://qdata.github.io/deep2Read/

Executive Summary

- Molecule generation using GAN approach
- edge-by-edge generation based on reinforcement learning
- Loss function gives is a weighted sum of adversarial loss, domain specific target metrics, validity constraints
- Target metrics are optimized during the learning and not as a post learning optimization process which offers state of the art results in the domain
- Graph representation is performed with GCN
- Training of the generator performed with policy gradient

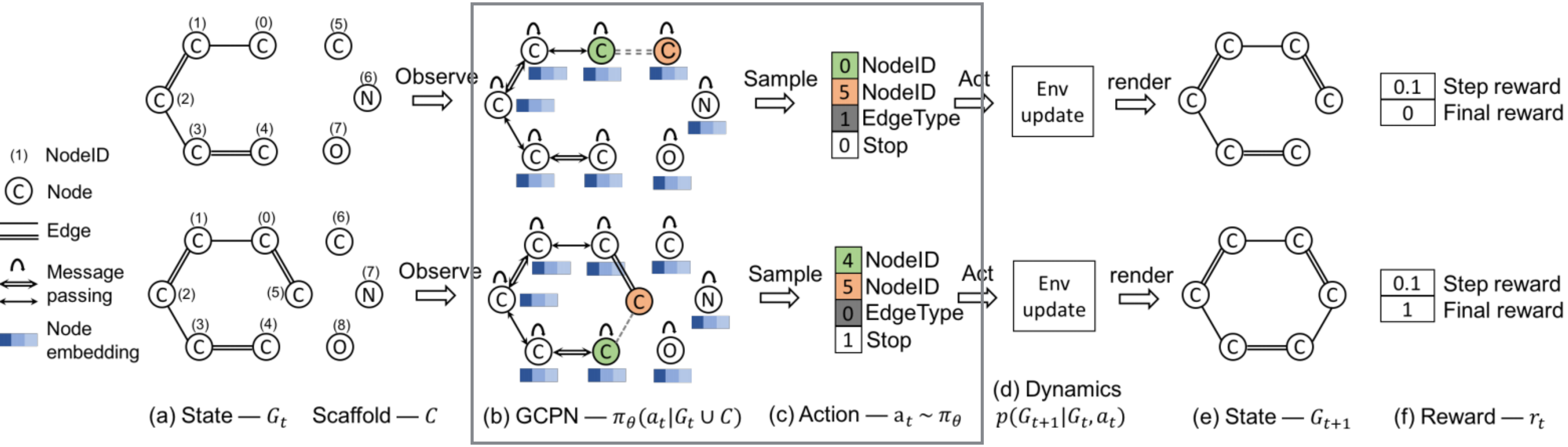
GCPN



(a) State — G_t Scaffold — C (b) GCPN — $\pi_\theta(a_t|G_t \cup C)$ (c) Action — $a_t \sim \pi_\theta$ (d) Dynamics $p(G_{t+1}|G_t, a_t)$ (e) State — G_{t+1} (f) Reward — r_t

- Graph *G = (A,E,F)* ,$A \in \{0,1\}^{n \times n}$,$F \in \mathbb{R}^{n \times d}$,$E \in \{0,1\}^{b \times n \times n}$, $A = \sum_{i=1}^{b} E_i$
- Run GCN $H^{(l+1)} = \text{AGG}(\text{ReLU}(\{\tilde{D}_i^{-\frac{1}{2}} \tilde{E}_i \tilde{D}_i^{-\frac{1}{2}} H^{(l)} W_i^{(l)}\}, \forall i \in (1, ..., b)))$

GCPN



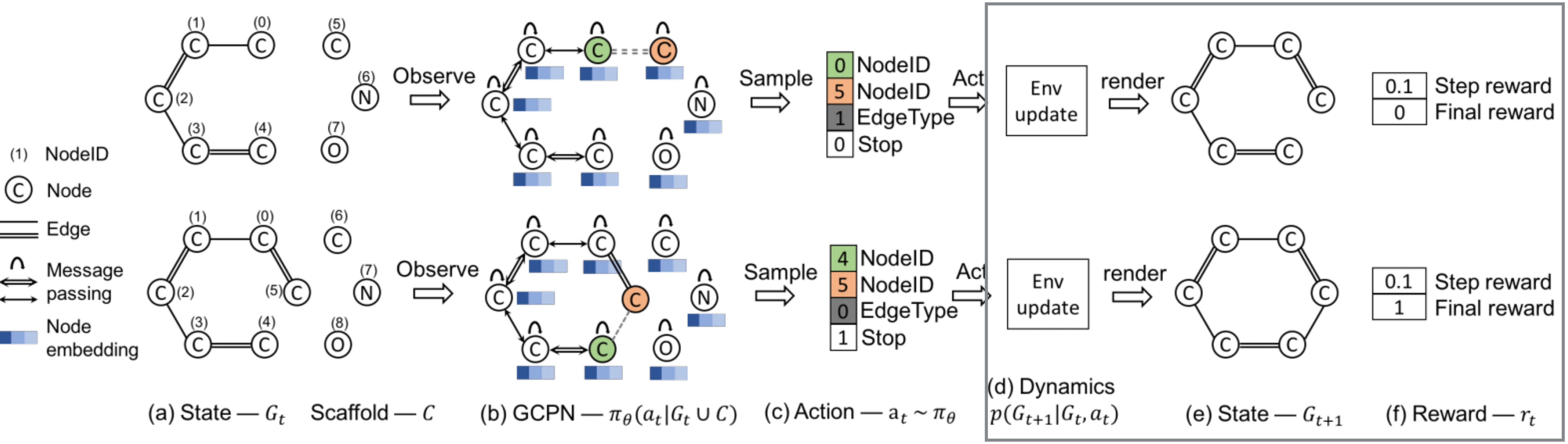
(a) State — G_t Scaffold — C (b) GCPN — $\pi_\theta(a_t|G_t \cup C)$ (c) Action — $a_t \sim \pi_\theta$ (d) Dynamics $p(G_{t+1}|G_t, a_t)$ (e) State — G_{t+1} (f) Reward — r_t

- Run 4 MLP networks to build a 4 part action:

$$a_t = \text{CONCAT}(a_{\text{first}}, a_{\text{second}}, a_{\text{edge}}, a_{\text{stop}})$$

$$\begin{aligned}
f_{\text{first}}(s_t) &= \text{SOFTMAX}(m_f(X)), & a_{\text{first}} &\sim f_{\text{first}}(s_t) \in \{0,1\}^n \\
f_{\text{second}}(s_t) &= \text{SOFTMAX}(m_s(X_{a_{\text{first}}}, X)), & a_{\text{second}} &\sim f_{\text{second}}(s_t) \in \{0,1\}^{n+c} \\
f_{\text{edge}}(s_t) &= \text{SOFTMAX}(m_e(X_{a_{\text{first}}}, X_{a_{\text{second}}})), & a_{\text{edge}} &\sim f_{\text{edge}}(s_t) \in \{0,1\}^b \\
f_{\text{stop}}(s_t) &= \text{SOFTMAX}(m_t(\text{AGG}(X))), & a_{\text{stop}} &\sim f_{\text{stop}}(s_t) \in \{0,1\}
\end{aligned}$$

GCPN



(a) State — G_t Scaffold — C (b) GCPN — $\pi_\theta(a_t|G_t \cup C)$ (c) Action — $a_t \sim \pi_\theta$ (d) Dynamics $p(G_{t+1}|G_t, a_t)$ (e) State — G_{t+1} (f) Reward — r_t

- Check weather the suggested action is legal, based on a 3rd party molecule checker, specifically RDKit.
- If it is legal update the state if it is not resemble from the policy and repeat until a legal action is proposed.
- Keep track of the “step rewards” which adds and removes punishments for suggesting legal or illegal actions.

Reward function

- The reward function is built from intermediate/step reward and final reward.
- Final reward is a weighted sum over domain specific rewards and an adversarial reward.
- Domain specific rewards include scores like octanol-water partition coefficient (logP), druglikeness (QED) and molecular weight (MW).
- Adversarial reward is computed by training a discriminator using a GAN loss to distinguish between generated and true molecules. The discriminator is also a GCN.

$$\min_{\theta}\max_{\phi} V(\pi_\theta, D_\phi) = \mathbb{E}_{x\sim p_{data}}[\log D_\phi(x)] + \mathbb{E}_{x\sim\pi_\theta}[\log D_\phi(1-x)]$$

Experiments

- **Property Optimization -** The task is to generate novel molecules whose specified molecular properties are optimized.
- **Property Targeting -** The task is to generate novel molecules whose specified molecular properties are as close to the target scores as possible.
- **Constrained Property Optimization -** The task is to generate novel molecules whose specified molecular properties are optimized, while also containing a specified molecular substructure.

Experiment results

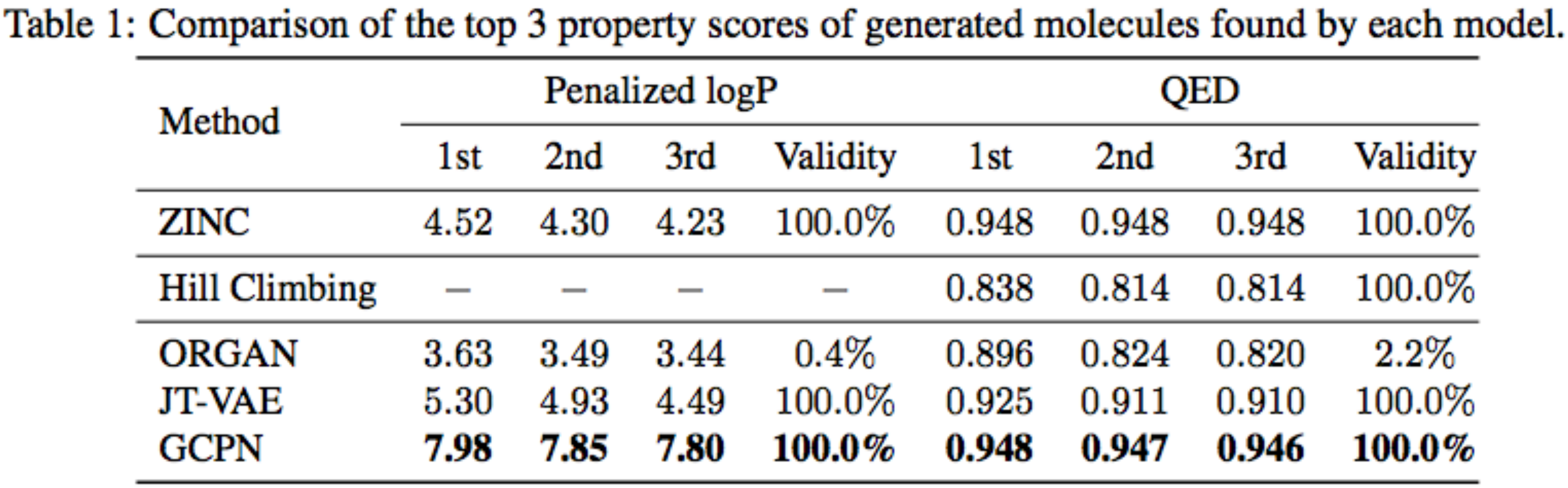


Table 1: Comparison of the top 3 property scores of generated molecules found by each model.

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%
Hill Climbing	–	–	–	–	0.838	0.814	0.814	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%**

Experiment results

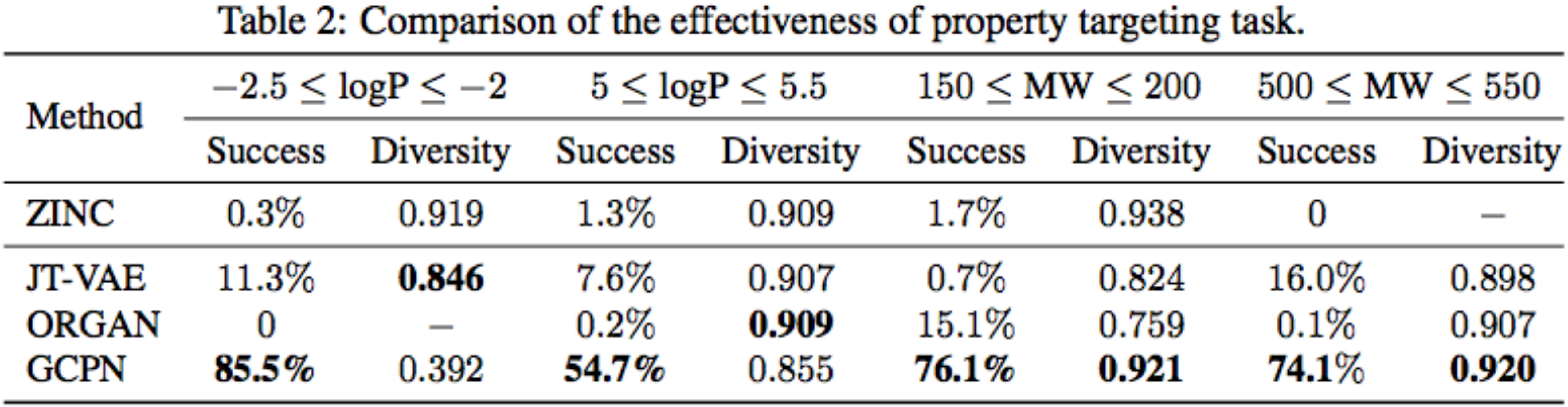


Table 2: Comparison of the effectiveness of property targeting task.

Method	$-2.5 \leq \text{logP} \leq -2$		$5 \leq \text{logP} \leq 5.5$		$150 \leq \text{MW} \leq 200$		$500 \leq \text{MW} \leq 550$	
	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**

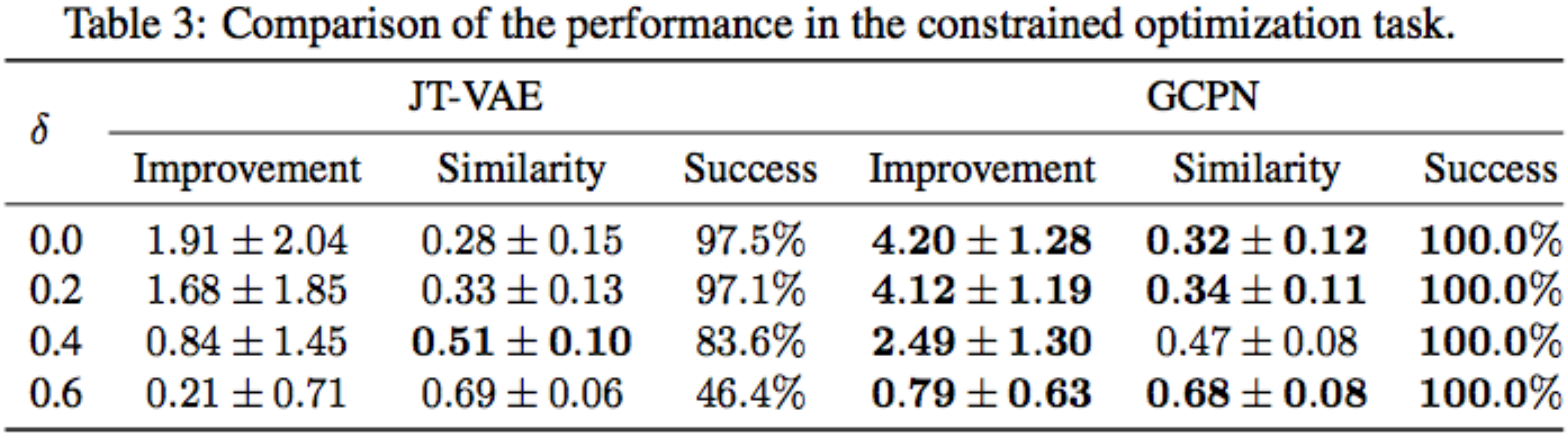


Table 3: Comparison of the performance in the constrained optimization task.

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%

Discussion

- The authors present a very clean policy training approach without sophisticated choice of vucaulary
- Junctional tree approach was not necessarily useful in molecule generation - It would be interesting to check the scaffold approach with RL training framework
- Legality check of actions during training and testing is a powerful technic
- Where applicable RL approach to optimize a target metric seems to be more powerful then a gradient based optimization of an approximate function