De Novo Structure-Based Drug Design Using Deep Learning

Authors: Sowmya Ramaswamy Krishnan, Navneet Bung, Sarveswara Rao Vangala, Rajgopal Srinivasan, Gopalakrishnan Bulusu, and Arijit Roy

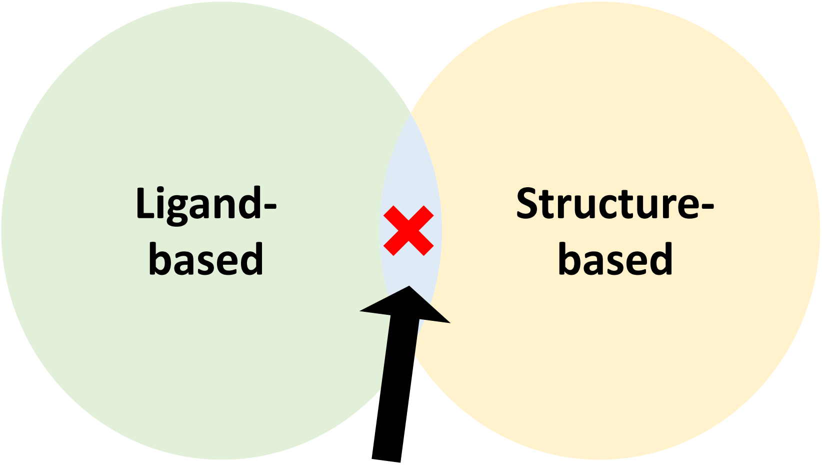
Outline

- Introduction of Drug Design
  - Background
  - Contributions of this paper
- Proposed Methodology
  - Model design
  - Training details
- Experiment Results
  - Generation results
  - Interpretation
- Rethinking and Discussion
  - Existed Problem on Generalization
- Conclusions

Introduction of Drug Design

Ligand-based design:

- General Idea: Based on existed target-molecules data to train a molecule generator. Then apply it to new target.
- Pros:
  - Can provide reliable results
- Cons:
  - Cannot generalize to design drug for novel target



This paper lies in this field

Structure-based design:

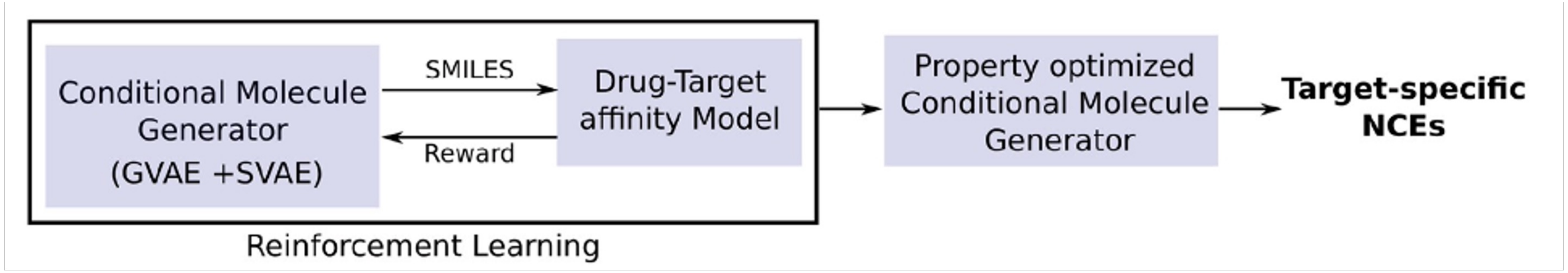
- General Idea: Directly capture the structure information to design
- Pros:
  - Structure is easier to be generalize to novel target
- Cons:
  - Should devise useful feature extraction method for structure information

- Contributions:
  - The first method which proposes to use binding site representation of target protein as structure information
  - Get similar and identical design results on two well-studied target proteins: JAK2 and DRD2
  - Give some interpretations on useful active sites from the trained deep learning model

Methodology

Methodology: Overview

- Combine the structure information of target protein to design molecules
  - Construction of active site graph for target protein
  - Use of ECIF fingerprints to capture active site features
  - Adopt Graphical neural network to capture the interaction between difference active sites
- Conditioned on the target structure information, use pretrained molecule language model to generate appropriate drugs
  - Pretrain SMILES language model on the ChEMBL dataset in an unsupervised way
  - Fuse target information to finish conditional generation
- Boost design performance with reinforcement learning
  - Use a pretrained affinity model as a reward function
  - Train the language model to achieve a high expected reward



Conditional Molecule Generator (GVAE +SVAE)
SMILES
Reward
Drug-Target affinity Model
Reinforcement Learning
Property optimized Conditional Molecule Generator
Target-specific NCEs

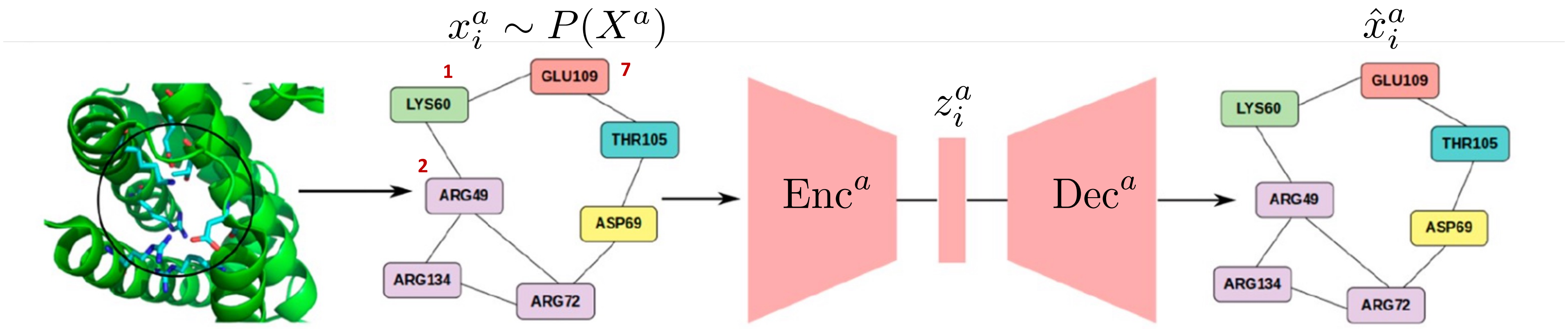
Methodology: Consider Target Structure

- Construct active site based on the target protein $x_i^a \sim P(X^a)$
  - How to construct? PDBbind and scPDB datasets
  - To consider the structure information, they use a graph-based neural network to model the interaction between different sites
- A Graph VAE is trained to reconstruct the adjacency matrix

$$z_i^a = \mathrm{Enc}^a(x_i^a) \quad \hat{x}_i^a = \mathrm{Dec}^a(z_i^a)$$

$$\mathcal{L} = \sum_i \mathrm{CrossEntropy}(x_i^a, \hat{x}_i^a) + KL(q(z^a|x_i^a)||P(z))$$

$$\text{Adjacency Matrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ & & \cdots & & & & \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$



Methodology: Small Molecular Generation

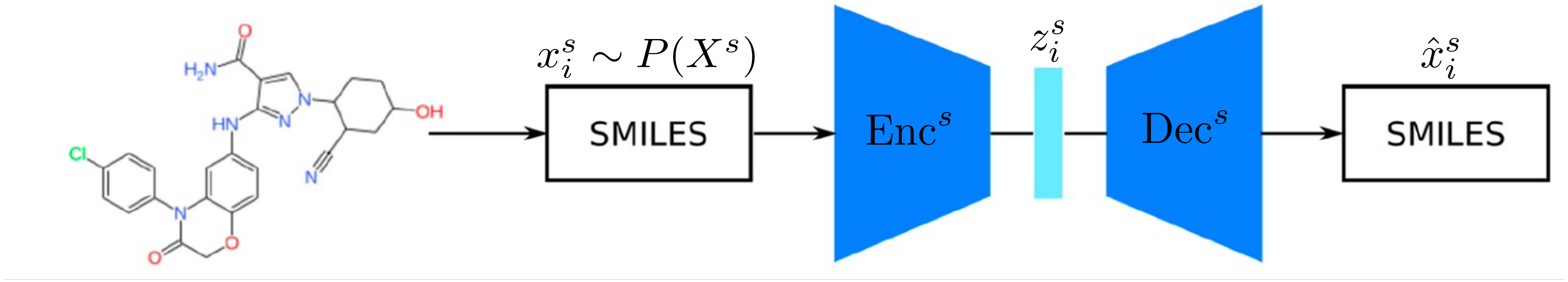
- Generating molecular is abstracted as a SMILES language generation problem $x_i^s = [C, H, O, \ldots, N] \sim P(X^s)$ $P(x_i^s) = \prod_{j=1}^{L} P(x_{i,j}^s | x_{i,1:j-1}^s) P(x_{i,1}^s)$

- Use the ChEMBL dataset, train a sequence generation model in unsupervised learning to predict the SMILES.

$$z_i^s = \mathrm{Enc}^s(x_i^s) \quad \hat{x}_i^s = \mathrm{Dec}^s(z_i^s)$$

$$\mathcal{L} = -\sum_i \log P(x_i^s) + KL(q(z^s|x_i^s)||P(z))$$

Decoding Use special token "!" as input, generate new token at time step t by samping from the distribution $P(\hat{x}_{i,t}^s | \hat{x}_{i,1:t-1}^s)$. Samplting methods include top-k or top-p.



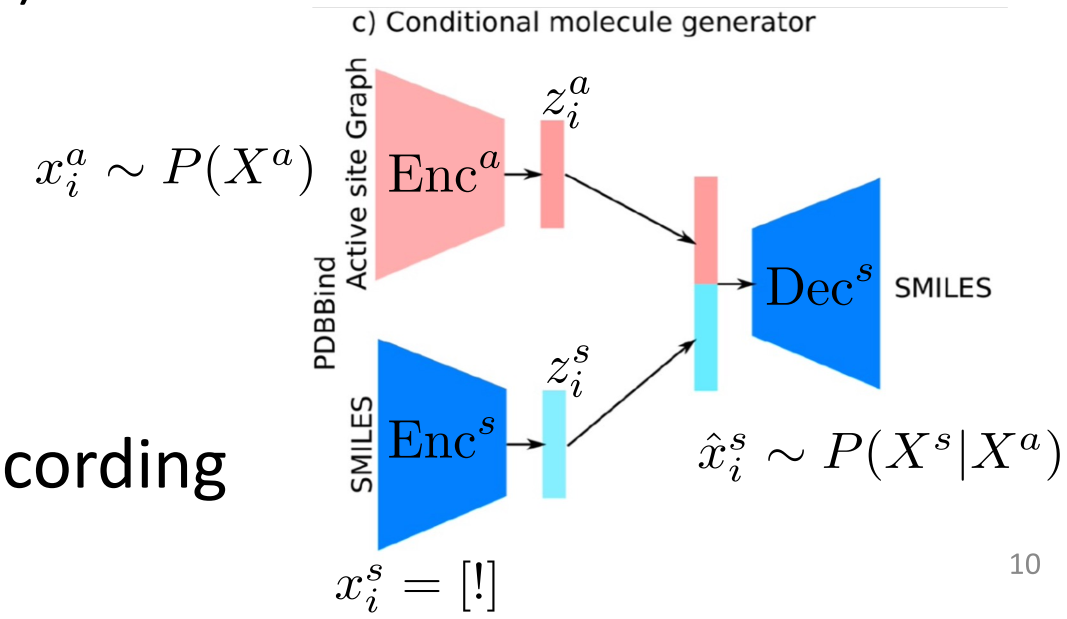
Methodology: Conditional Molecule Generation

- Generate appropriate molecule based on the target activate sites

- Given pairs of active site – molecule $(x_i^a, x_i^s)\ \forall i \in [1, P]$
  - PDBbind dataset
  - Pad the input smiles as special token (e.g. !)
  - Maximize the conditional log-likelihood

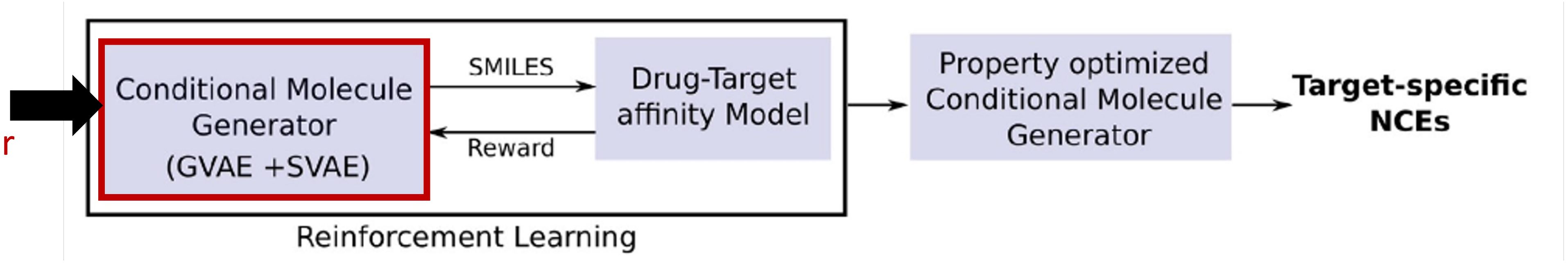
$$z_i^a = \mathrm{Enc}^a(x_i^a) \quad z_i^s = \mathrm{Enc}^s(x_i^s)$$
$$\hat{x}_i^s = \mathrm{Dec}^s([z_i^a, z_i^s])$$
$$\mathcal{L} = -\sum_i \log P(x_i^s | x_i^a)$$

- This trained model is able to generate according molecules based on input activate sites



Methodology: Review the generator part

Pretrain Generator


Conditional Molecule Generator (GVAE +SVAE)
SMILES
Reward
Drug-Target affinity Model
Reinforcement Learning
Property optimized Conditional Molecule Generator
Target-specific NCEs

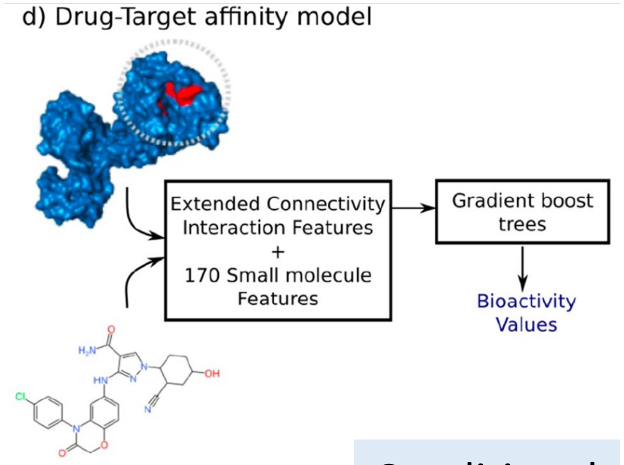
- Why we need pretraining?
  - Pretraining is essential to capture prior graph interaction semantics
  - Pretraining is essential to generate valid molecule sequence
- Next step
  - Further boost the drug design performance to achieve good affinity
  - Use reinforcement learning (RL) to judge the SMILES sequence produced by the conditional generator

Methodology: Drug-Target Affinity Score Model + RL

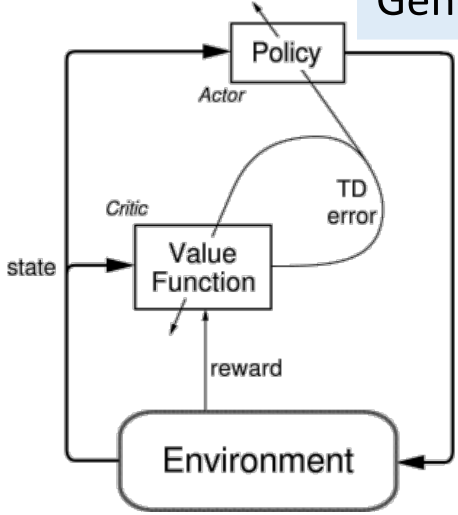
- Test the bioactivity (Affinity) provided by the generator
  - The bioactivity can be determined through experiment but time consuming and indifferentiable
    - PDBbind general set and refined set to experimentally determined $\mathrm{IC}_{50}, \mathrm{K}_i, \mathrm{K}_d$
  - Use estimated model to predict the bioactivity of input
    - Adopt previous EDIF model (A kind of decision tree)

$$s = \mathrm{EDIF}(x^s, t)$$
$$r(s) = \exp(s/3)$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{s_0, a_0, \ldots, s_t, a_t}[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)] Q_w(s_t, a_t)$$
$$= \mathbb{E}_\tau[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) Q_w(s_t, a_t)]$$


d) Drug-Target affinity model
Extended Connectivity Interaction Features + 170 Small molecule Features
Gradient boost trees
Bioactivity Values
H₂N
O
N
HN
OH
Cl
O

Conditional Generator



SMILES

DTA Model

Methodology: Summary

Pretraining:

- Pretrain active site VAE and SMILES VAE

$$\mathrm{Enc}^a, \mathrm{Dec}^a, \mathrm{Enc}^s, \mathrm{Dec}^s$$

- Use pretrain model to maximize conditional generation model

$$(x_i^a, x_i^s)\ \forall i \in [1, P]$$

$$\hat{x}_i^s \sim P(X^s|X^a)$$

Reinforcement Learning:

- Construct affinity model as environment

$$s = \mathrm{EDIF}(x^s, t)$$

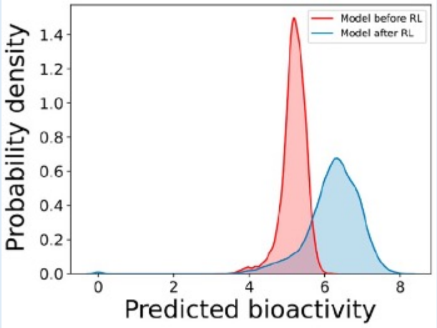
$$r(s) = \exp(s/3)$$

- Use pretrain model to maximize conditional generation model

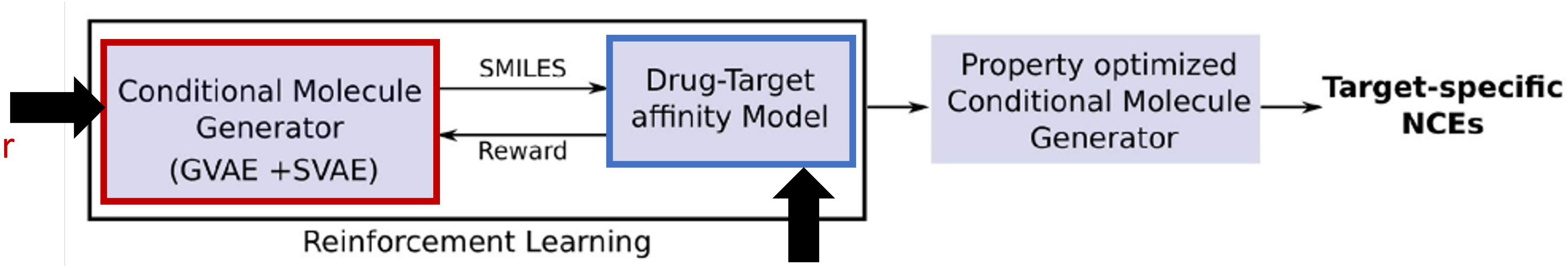
$$\nabla_\theta J(\theta) \quad \text{policy gradient}$$

Inference:

- Only *consider one target protein*, use RL to search until observed distribution shift (sampling)



Pretrain Generator

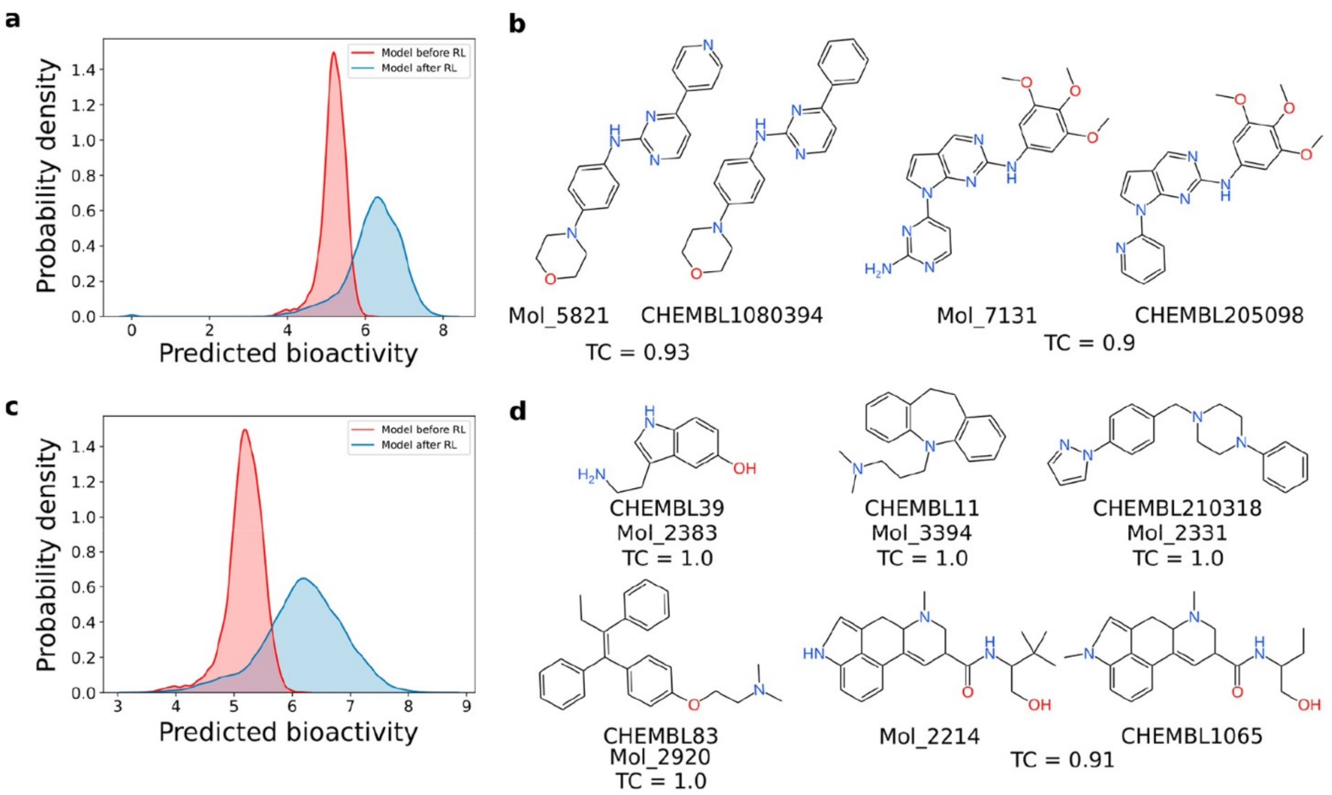

Conditional Molecule Generator (GVAE +SVAE)
SMILES
Reward
Drug-Target affinity Model
Reinforcement Learning
Property optimized Conditional Molecule Generator
Target-specific NCEs

Learning from reward

Experiment Results

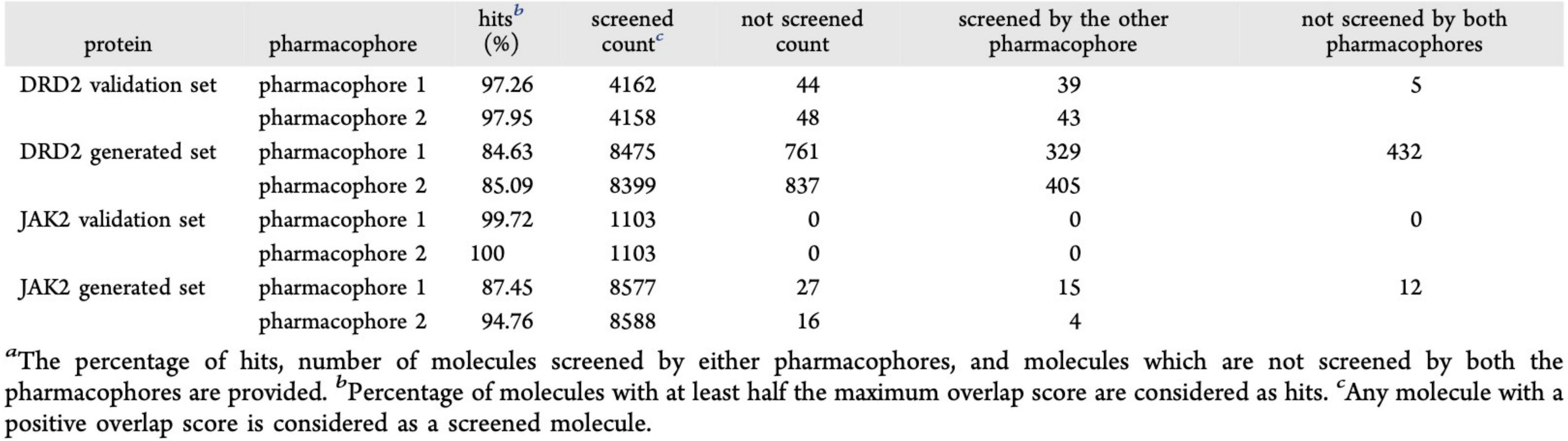
Validation: Generated Results

- Validation studies on two different target proteins: DRD2 and JAK2
  - For each target protein, use RL to do searching
- Generation of small molecules with high similarity to existing inhibitors



Validation: Generated Results

- Preservation of key pharmacophoric features required for efficient binding

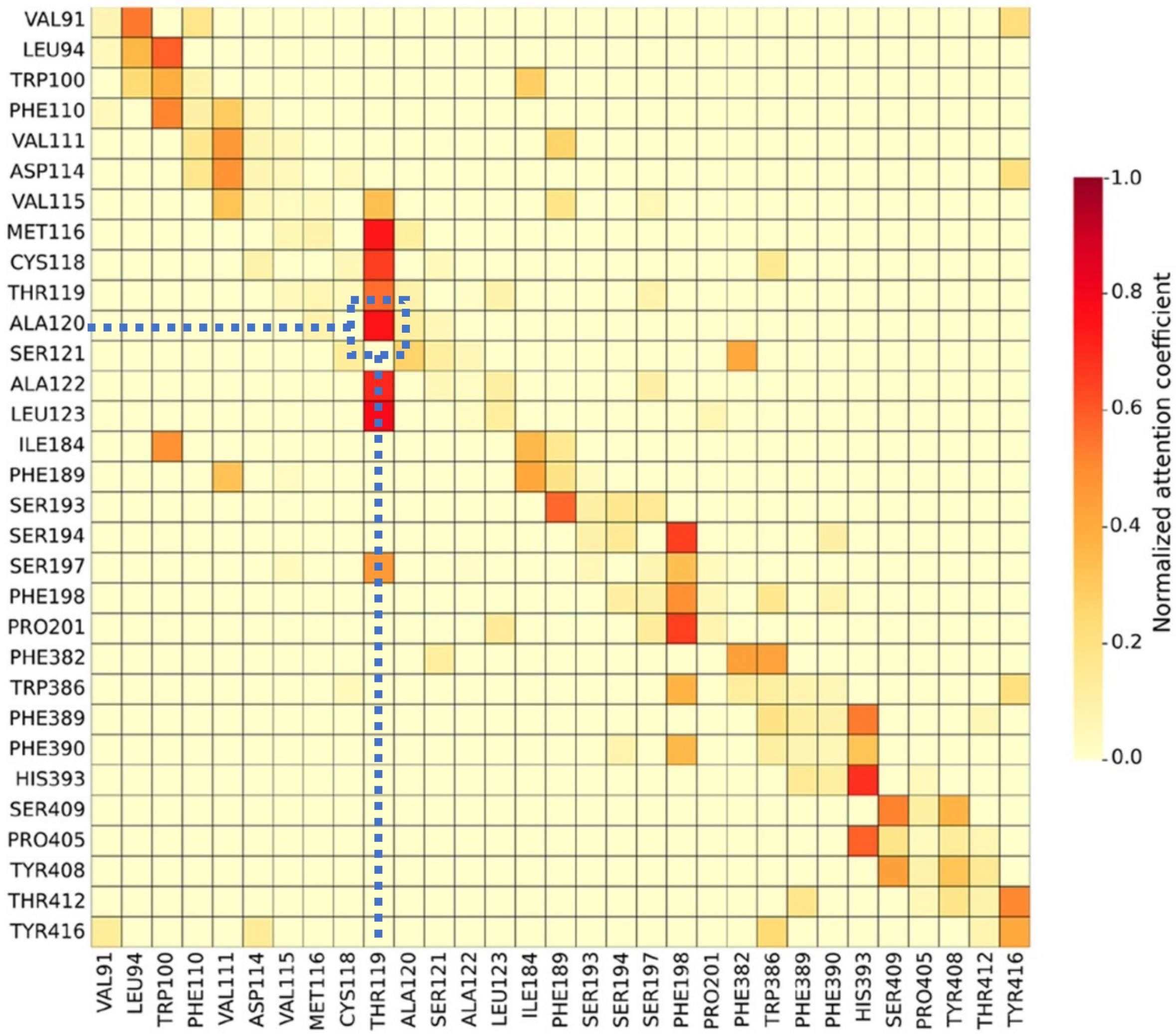


protein	pharmacophore	hits[b] (%)	screened count[c]	not screened count	screened by the other pharmacophore	not screened by both pharmacophores
DRD2 validation set	pharmacophore 1	97.26	4162	44	39	5
	pharmacophore 2	97.95	4158	48	43	
DRD2 generated set	pharmacophore 1	84.63	8475	761	329	432
	pharmacophore 2	85.09	8399	837	405	
JAK2 validation set	pharmacophore 1	99.72	1103	0	0	0
	pharmacophore 2	100	1103	0	0	
JAK2 generated set	pharmacophore 1	87.45	8577	27	15	12
	pharmacophore 2	94.76	8588	16	4	

[a] The percentage of hits, number of molecules screened by either pharmacophores, and molecules which are not screened by both the pharmacophores are provided. [b] Percentage of molecules with at least half the maximum overlap score are considered as hits. [c] Any molecule with a positive overlap score is considered as a screened molecule.

Understanding Graph Interaction

- Visualization methods: entropy histograms and attention coefficient heatmap
  - Interpretability of the model's learning process
- Explanation of attention coefficients
  - Role of specific residues and interactions in the active site
  - *<u>17 of the 149 interactions</u>* with attention weight greater than 0.5 (important)
  - Leu94, Trp100, Asp114, Thr119, Ile184, Phe198, His393, and Tyr416 are verified in the experiments



DRD2

Understanding Target-molecule Interaction

- Test interaction between generated molecules with target protein DRD2
  - These residues (Leu94, Trp100, Ile184, Phe110) form hydrophobic inter- actions with the generated molecules.
- Deep learning models are often criticized as black boxes, but the method proposed in this work could explain the importance of active site residues.

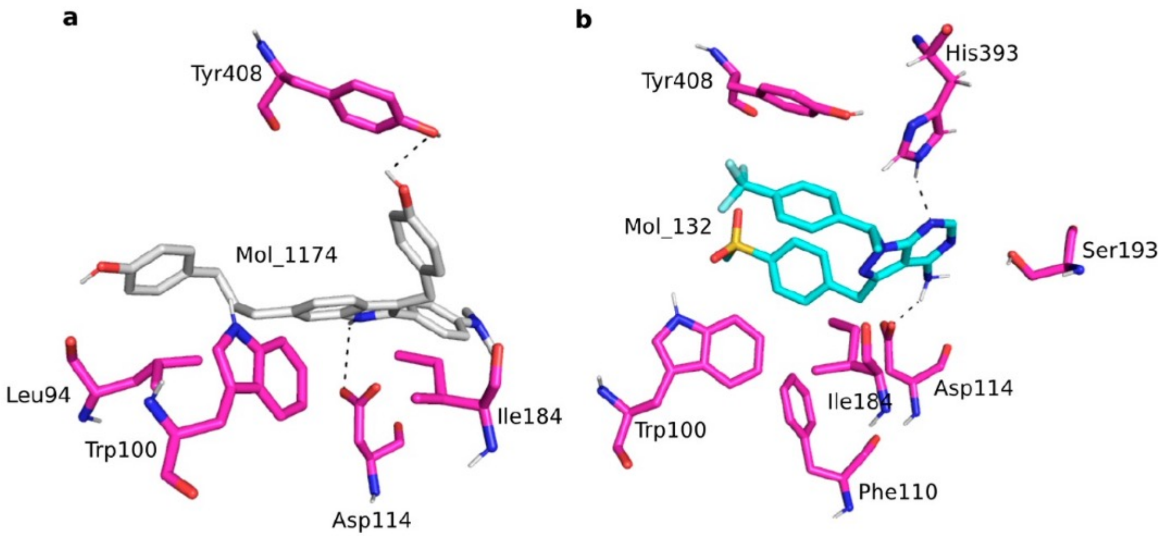


Figure 4. Interactions between key active site residues identified from attention coefficients and selected DRD2-specific generated small molecules: (a) Mol_1174 (white sticks) and (b) Mol_132 (cyan sticks). The residues forming hydrogen bond are shown as dotted lines.

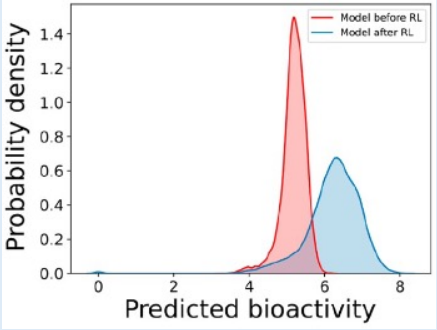
Rethinking and Discussion

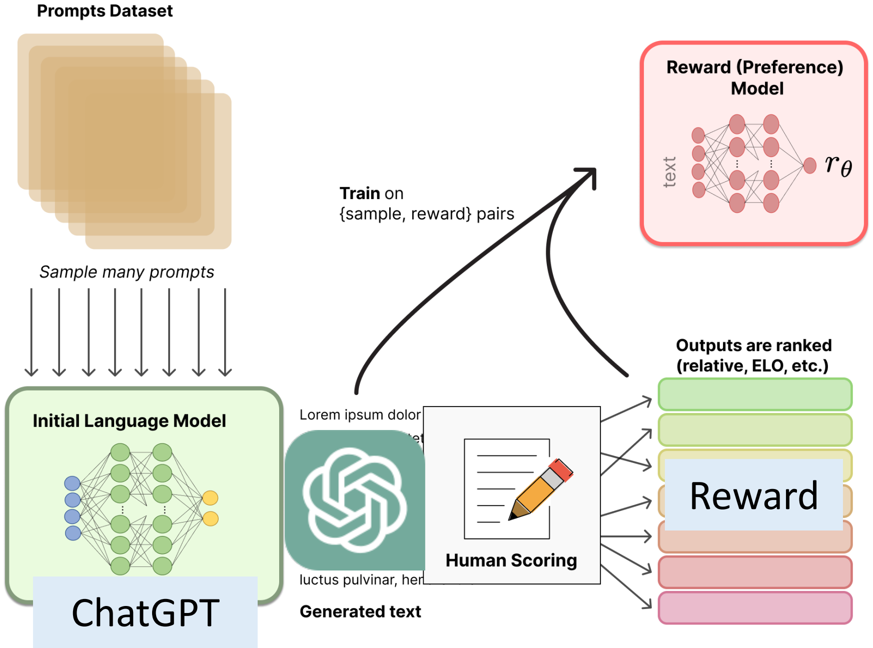
Existed Problem on Generalization

- In inference stage, an overfit problem is introduced in their method.
  - Individually consider one target protein leads the model overfit to this target on the affinity dataset and cannot be generalize to other target proteins
  - It ignores the conditional generation process $\hat{x}_i^s \sim P(X^s|X^a)$ ❌
- Take an essential training component (Reinforcement Learning Human Feedback, RLHF) in ChatGPT as an example. It is bad for the GPT can only produce good answer on one question.

Inference:

- Only *consider one target protein*, use RL to search until observed distribution shift (sampling)





Ref: https://huggingface.co/blog/rlhf

Conclusion

- Recap of the study and the novel structure-based method for small molecule generation
- Importance of distinguishing key residues and interactions in the active site
- Implications for more efficient, cost-effective drug discovery
- Problematic RL method leads to bad generalization ability

Thanks for Your Attention

Appendix

Variational Auto Encoder

Consider observed N samples x_i under i.i.d. assumption from an unknown distribution $P(X)$, which means $x_i \sim P(X)$. To estimate the parameter of unknown $P(X)$ from observed samples, the log-likelihood is considered as

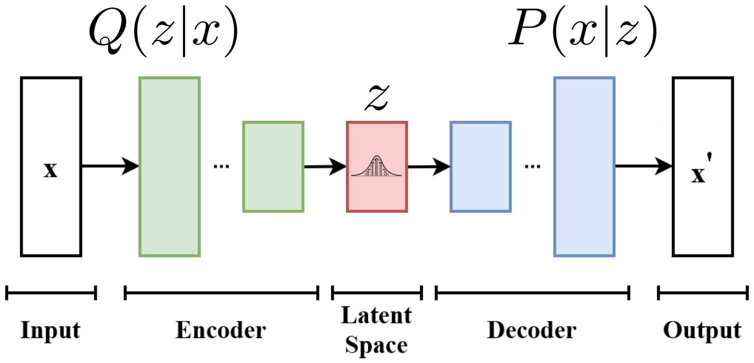
$$\begin{aligned} \log P(\{x_1,\dots,x_N\}) &= \sum_{i=1}^{N} \log P(x_i) \quad \text{(i.i.d. assumption)} \\ \log P(x_i) &= KL(Q(z|x_i)||P(z|x_i)) + \mathbf{E}_{Q(z|x_i)}[\log \frac{P(x,z)}{Q(z|x)}]. \end{aligned} \tag{1}$$

where we introduce the latent distribution of $Q(z|x)$.

Since the KL distance is always postive, therefore, the lowerbound of log-likelihood is

$$\begin{aligned} \log P(x_i) &\geq \mathbf{E}_{Q(z|x_i)}[\log \frac{P(x,z)}{Q(z|x)}] \quad \text{(ELBO)} \\ \mathbf{E}_{Q(z|x_i)}[\log \frac{P(x,z)}{Q(z|x)}] &= \mathbf{E}_{Q(z|x_i)}[\log P(x_i|z)] - KL(Q(z|x_i)||P(z)). \end{aligned} \tag{2}$$

which means to maximize log-likelihood which equals to maximize the lower bound (ELBO).



E.g., SMILES VAE Training

$$z_i^s = \mathrm{Enc}^s(x_i^s) \quad \hat{x}_i^s = \mathrm{Dec}^s(z_i^s)$$

$$\mathcal{L} = -\sum_i \log P(x_i^s) + KL(q(z^s|x_i^s)||P(z))$$

Ref: Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

VAE Pretraining for Active Site

An active site graph can be represented as vertices $\mathcal{V} = \{v_1, \ldots, v_N\}$ and edges $e_{i,j} \in \mathcal{E}$

GAT Network Compute attention score $s_{i,j}$ between adjanced node embedding h_{v_i} and h_{v_j}:

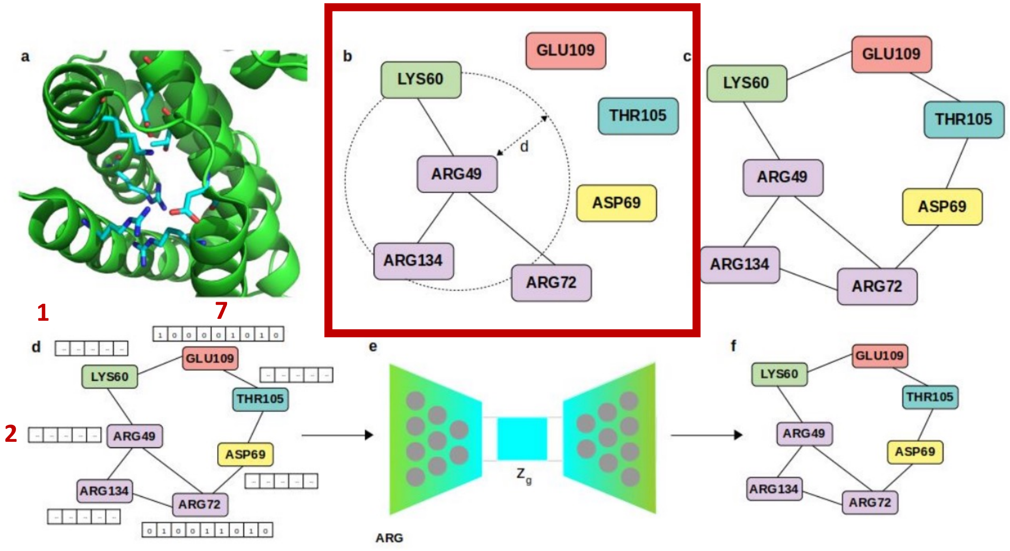
$$\begin{aligned} s_{i,j} &= \frac{\exp(Wh_{v_i} \cdot Wh_{v_j})}{\sum_{(i,k)\in\mathcal{E}} \exp(Wh_{v_i} \cdot Wh_{v_k})} \\ \hat{h}_{v_i} &= \sum_{(i,k)\in\mathcal{E}} s_{i,k} h_{v_k} \end{aligned} \tag{1}$$

Edge Reconstruction After encoded the graph feature, the decoder is asked to reconstruct adjacency matrix of $\mathcal{E}$ (connected edges means 1 otherwise 0) for each vertice v

$$\begin{aligned} z_i^a &= \text{Enc}^a(x_i^a) \quad \hat{x}_i^a = \text{Dec}^a(z_i^a) \\ \mathcal{L} &= \sum_i \text{CrossEntropy}(x_i^a, \hat{x}_i^a) + KL(q(z^a|x_i^a)||P(z)) \quad \text{(ELBO)} \end{aligned} \tag{2}$$

$$\text{CrossEntropy} = -\sum_i x_i^a \log(\hat{x}_i^a)$$

Construct active site graph $d < 4\mathring{A}$



$$\text{Adjacency Matrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ & & & \cdots & & & \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

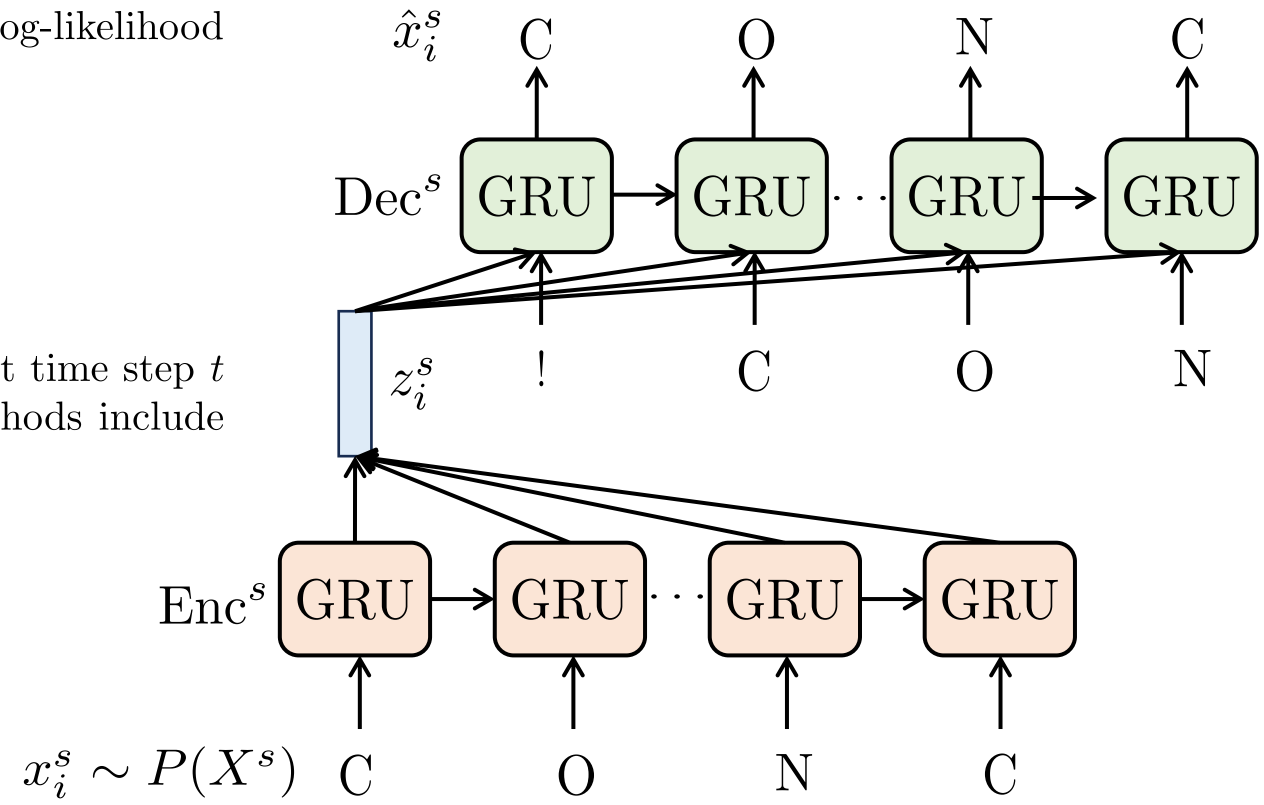
VAE Pretraining for SMILES Language Model

The SMILES sequence $x^s \sim P(X^s)$, therefore, the maximize log-likelihood can be defined as:

$$z_i^s = \mathrm{Enc}^s(x_i^s) \quad \hat{x}_i^s = \mathrm{Dec}^s(z_i^s)$$

$$\mathcal{L} = -\sum_i \log P(x_i^s) + KL(q(z^s|x_i^s)||P(z)) \quad \text{(ELBO)}.$$

Decoding Use special token "!" as input, generate new token at time step t by samping from the distribution $P(\hat{x}_{i,t}^s|\hat{x}_{i,1:t-1}^s)$. Samplting methods include top-k or top-p.



Ref: Joulin, Armand, and Tomas Mikolov. "Inferring algorithmic patterns with stack-augmented recurrent nets." Advances in neural information processing systems 28 (2015).