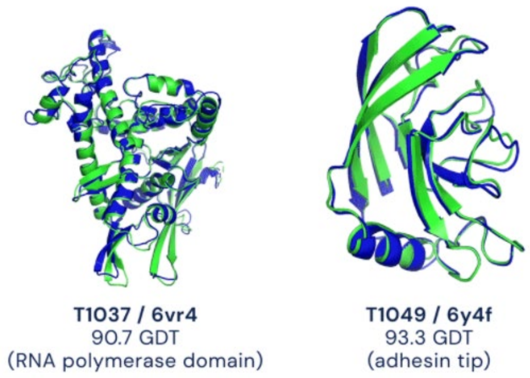
Molecule Optimization by Explainable Evolution

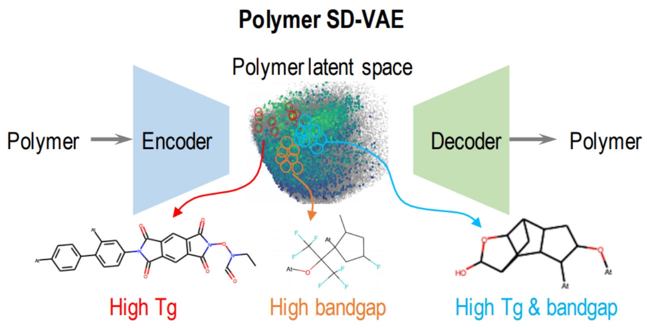
Binghong Chen[1*], Tianzhe Wang[1,5*], Chengtao Li[2], Hanjun Dai[3], Le Song[4]

[1]Georgia Institute of Technology, [2]Galixir, [3]Google Research, Brain Team,
[4]Mohamed bin Zayed University of AI, [5]Shanghai Qi Zhi Institute

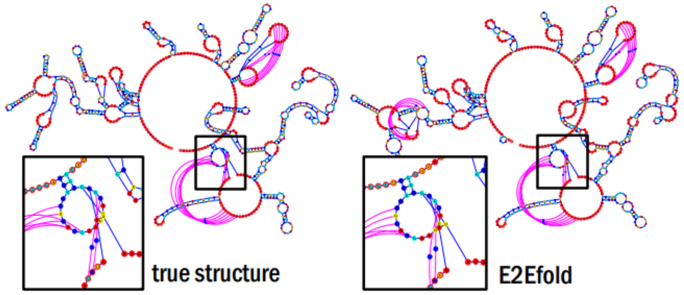
Machine Learning + Drug Design



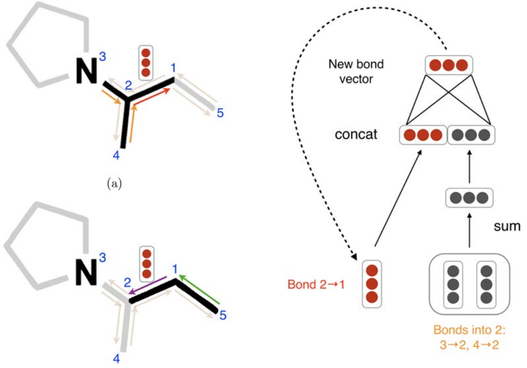
Protein Folding
(AlphaFold 2, 2020)



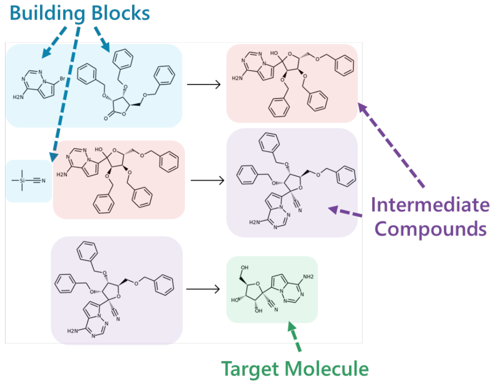
Molecule Generation
(Dai et al., 2018; Batra et al., 2020)



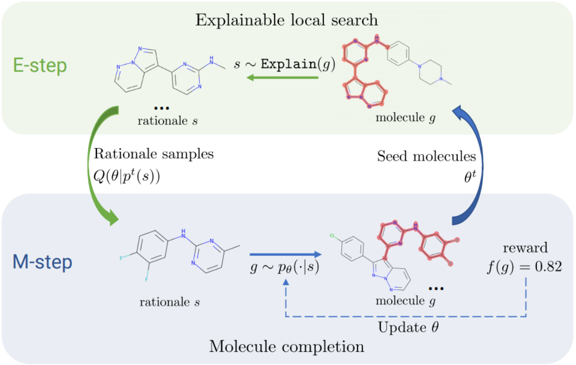
RNA Secondary Structure Prediction
(Chen et al., 2020)



Molecular Property Prediction
(Dai et al., 2016; Yang et al., 2019)



Retrosynthesis
(Dai et al., 2019; Chen et al., 2020)

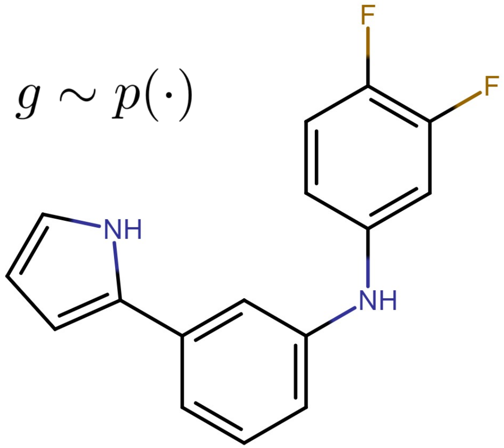


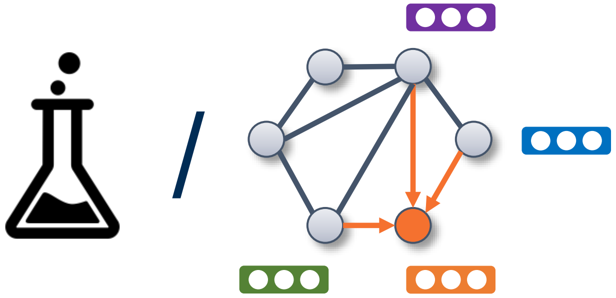
Molecule Optimization
(this work)

Molecule Optimization

- Design new molecules with desired properties:
  - Property scoring function f (potent, non-toxic, easy to synthesize, …)
  - Challenges: searching over the vast space of $> 10^{60}$ molecules.

$g \sim p(\cdot)$





f

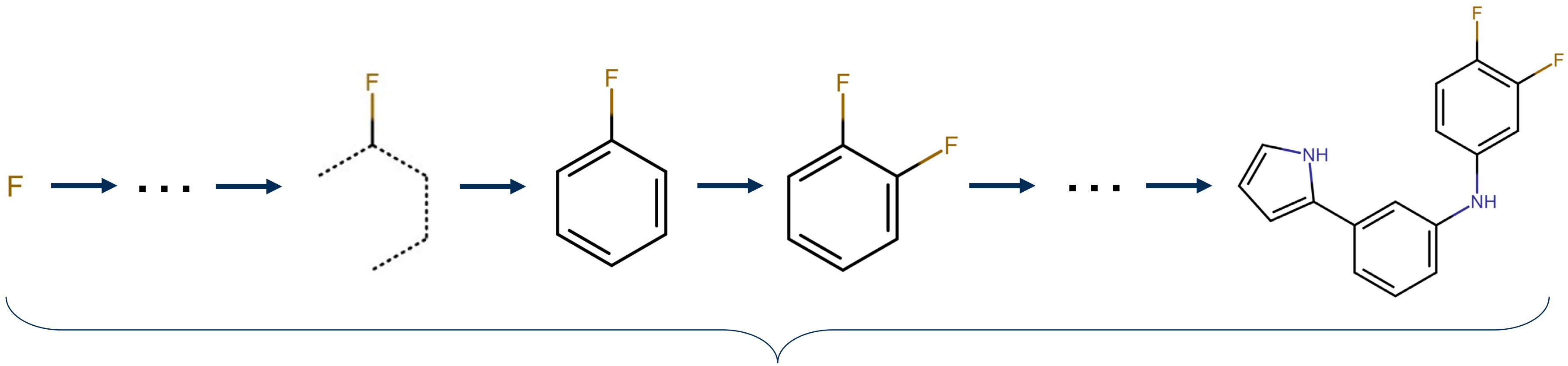
Property Scoring

Score: $f(g) = 0.61$

- Task: Learn a molecule generative model $p(\cdot)$ to maximize $\max_{p(\cdot)} \mathbb{E}_{g \sim p(\cdot)}[f(g)]$

RL for Generative Design

- Generation policy p: decide a new atom (and bonds) to add to the current partial molecule.

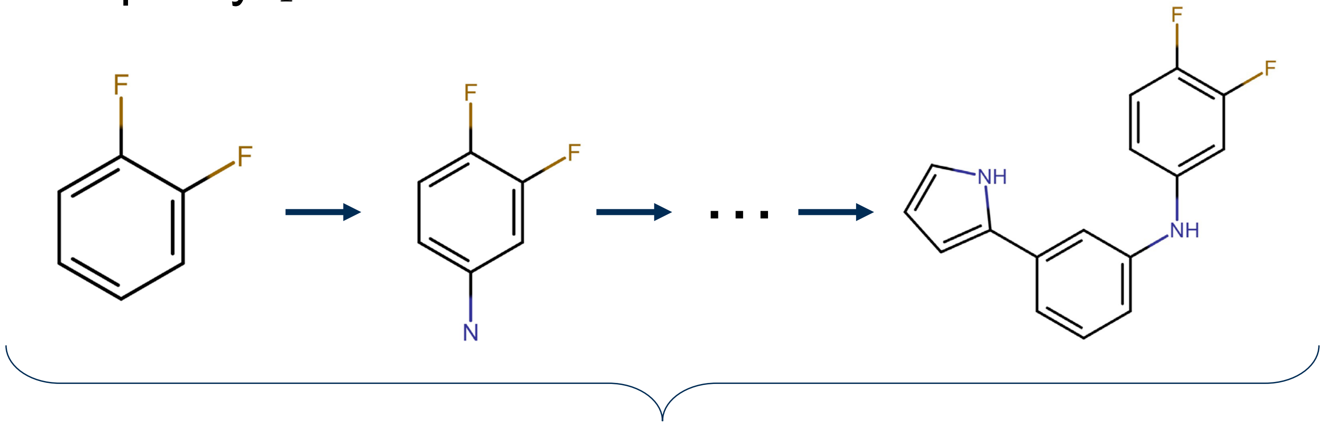


Autoregressive generative process

- Use RL to optimize p:
  - Reward $r = f(g)$ only obtained at the end.
  - Sparse reward, long horizon $\longrightarrow$ hard to optimize.

Conditioning on Substructures

- Rationales – substructures that most contributes to the desired molecular properties.
- Conditioning generation policy p on rationales.



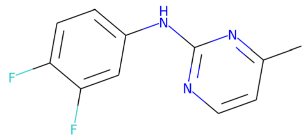
Conditioning autoregressive generative process

- Use RL to optimize p:
  - Shorter horizon $\longrightarrow$ easier to optimize.
  - Obtaining rationales is hard
    - Designed manually: require human effort.
    - MCTS (Jin et al., 2020): unable to optimize rationales jointly with p.

Our Approach: MolEvol

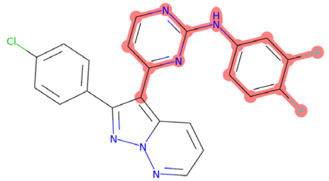
- Hierarchical Generative Model

Rationale Distribution $\xrightarrow{s \sim p(s)}$



rationale s

$\xrightarrow{g \sim p_\theta(\cdot|s)}$

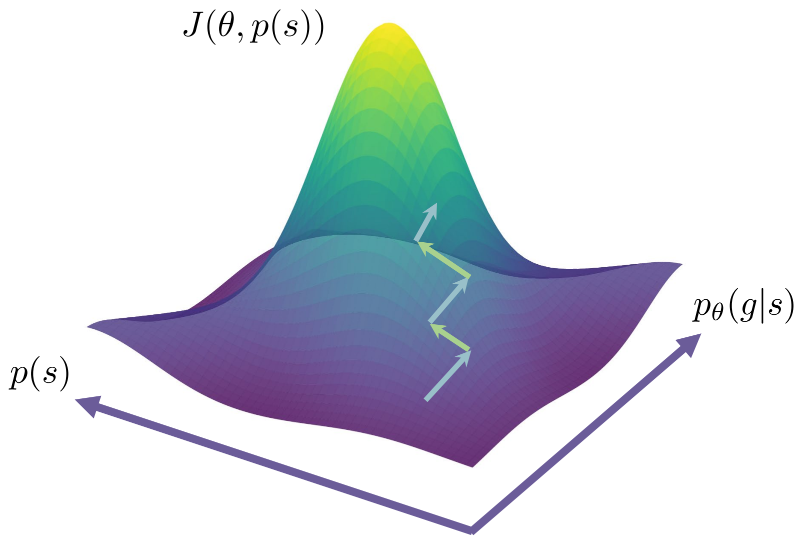


molecule g

- Alternating Optimization (EM-style)

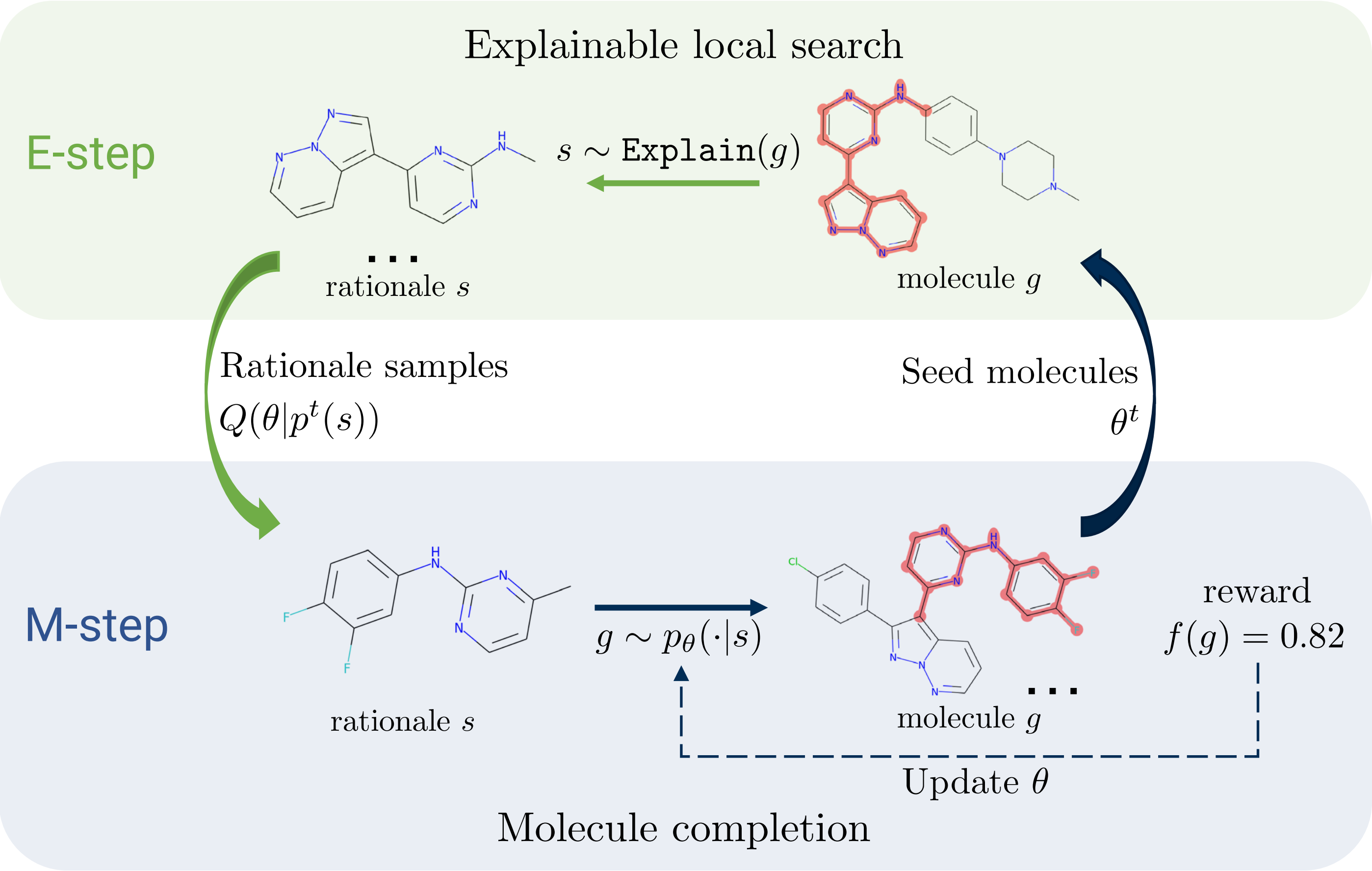
$$J(\theta, p(s)) = \mathbb{E}_{g \sim p_\theta(\cdot)}[f(g)] + \lambda \cdot \mathbb{H}[p(s)]$$

- E-step
  - Fix $p_\theta(g|s)$, update $p(s)$.
- M-step
  - Fix $p(s)$, update $p_\theta(g|s)$.



MolEvol: Algorithm Overview

- Init
  - A set of seed molecules are given.
  - Parameter θ^0.
- E-step
  - Produce a set of rationales with explainable graph model.
  - Optimize $p(s)$ (closed form).
- M-step
  - Produce a set of seed molecules.
  - Optimize $p_\theta(g|s)$ (RL).



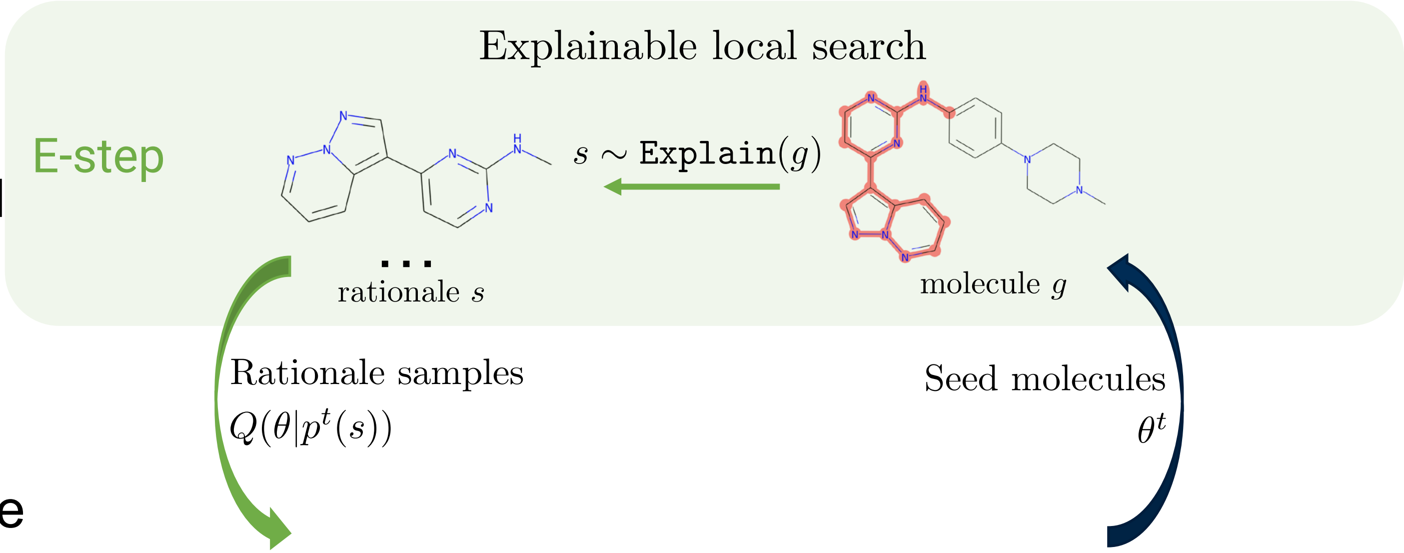
MolEvol: E-step

- In the t-th round, given
  - Parameter θ^{t-1},
  - Seed molecules $\mathcal{G}^{t-1}$,
- The rationale distribution has a closed form:

$$p^t(s) = \frac{1}{Z_\theta} \exp\left(\frac{1}{\lambda}\mathbb{E}_{g\sim p_{\theta^{t-1}}(\cdot|s)}[f(g)]\right)$$

- The support set of $p^t(s)$ is given by the explainable local search:

$$\mathcal{S}^t = \bigcup_{i=1}^{t} \left\{\mathtt{Explain}(g) : g \in \mathcal{G}^{i-1}\right\}$$

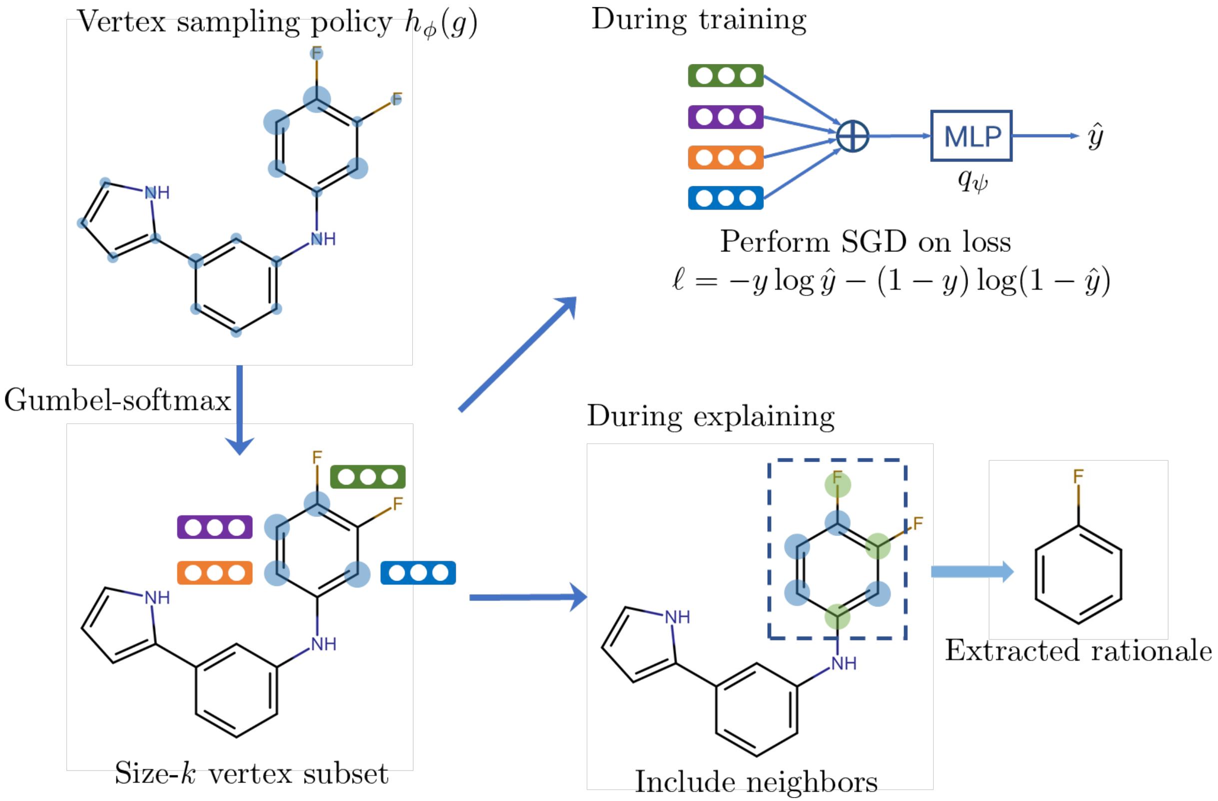


MolEvol: Explainable Graph Model

- To explain $\mathbb{P}(Y=1|g) \triangleq f(g)$, we maximize the mutual information between Y and rationale s.
- Variational objective:

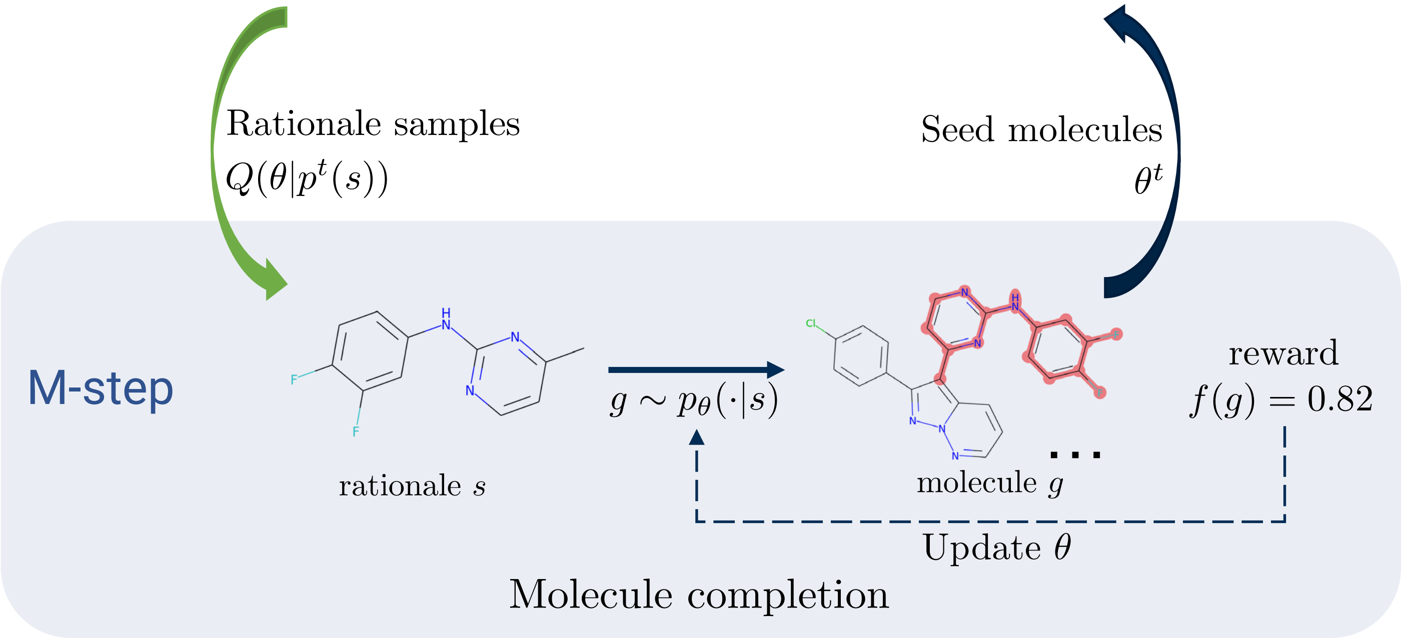
$$\max_{h_\phi(\cdot),\mathbb{Q}} \quad \underbrace{\mathbb{E}_{Y\sim\mathbb{P}(\cdot|g)}}_{\text{Sample from } \mathcal{G}^{t-1}} \left[\underbrace{\log \mathbb{Q}(Y \mid s)}_{\text{Variational approximation}} \right],$$

$$\text{such that } \underbrace{s = (\mathcal{U}, \mathcal{E}_g^{\mathcal{U}})}_{\text{Subgraph induced by vertex subset}} \text{ and } \mathcal{U} \sim \underbrace{h_\phi(g)}_{\text{Vertex sampler}}.$$



MolEvol: M-step

- In the t-th round, given
  - Distribution $p^t(s)$,
  - Parameter θ^{t-1},
- We update θ^t from θ^{t-1} using RL,
  - Init state $s \sim p^t(s)$,
  - Reward $r = f(g)$.



Experiments

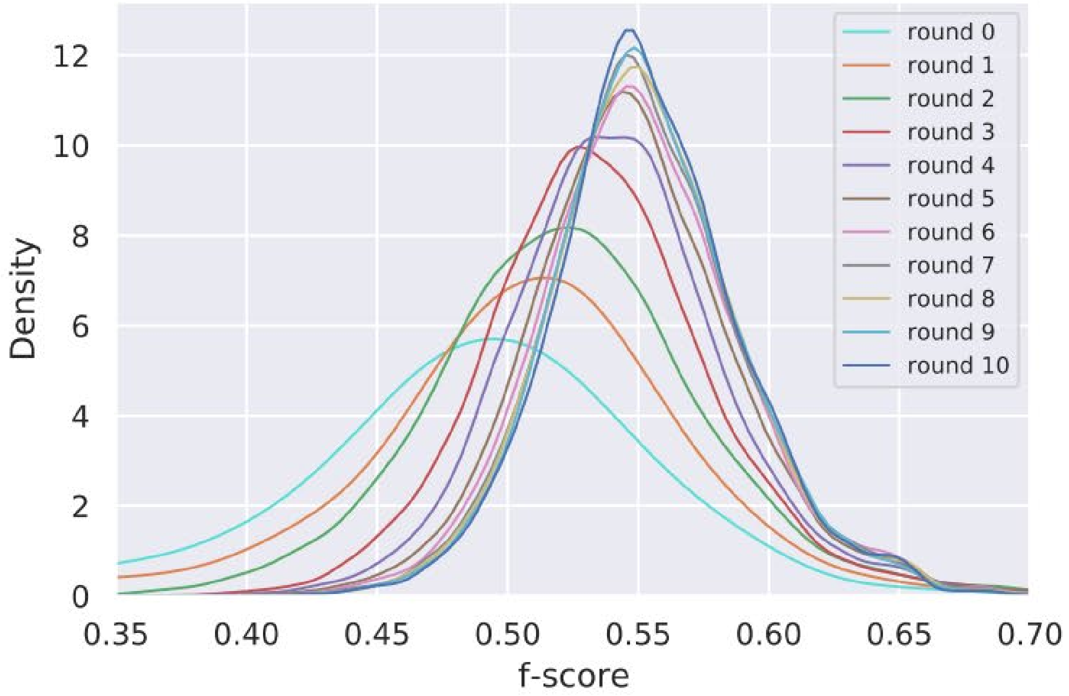
- Task: multi-property molecular optimization (Li et al., 2018; Jin et al.,2020)
  - $\underbrace{\text{GSK-3}\beta \text{ (Li et al., 2018), JNK3 (Li et al., 2018),}}_{\text{Potential targets in the treatment of Alzheimer's disease.}}$ $\underbrace{\text{QED (Bickerton et al., 2012),}}_{\text{Quantitative estimate of drug-likeness.}}$ $\underbrace{\text{SA (Ertl et al., 2009)}}_{\text{Synthetic accessibility.}}$
  - Scoring function: $f(g) = \left[\text{GSK-3}\beta(g) \cdot \text{JNK3}(g) \cdot \text{QED}(g) \cdot \text{SA}(g)\right]^{\frac{1}{4}}.$
- Metrics: we generate $N = 20K$ molecules from each method and measure,
  - $\underbrace{\sum_g \mathbb{I}[g \text{ is } \texttt{Qualified}]/N}_{\text{Success Rate.}}$ $\underbrace{\sum_g \mathbb{I}[g \text{ is } \texttt{Novel}]/\sum_g \mathbb{I}[g \text{ is } \texttt{Qualified}]}_{\text{Novelty.}}$ $\underbrace{1 - \frac{2}{n(n-1)} \sum_{g \neq g'} \texttt{Sim}(g, g')}_{\text{Diversity.}}$
  - $\underbrace{\texttt{card}(\{g \mid g \text{ is } \texttt{Qualified} \text{ and } \texttt{Novel}\})/N}_{\text{QNU score.}}$ $g \text{ is } \texttt{Qualified} \iff f(g) > 0.5$

Comparing to Baselines

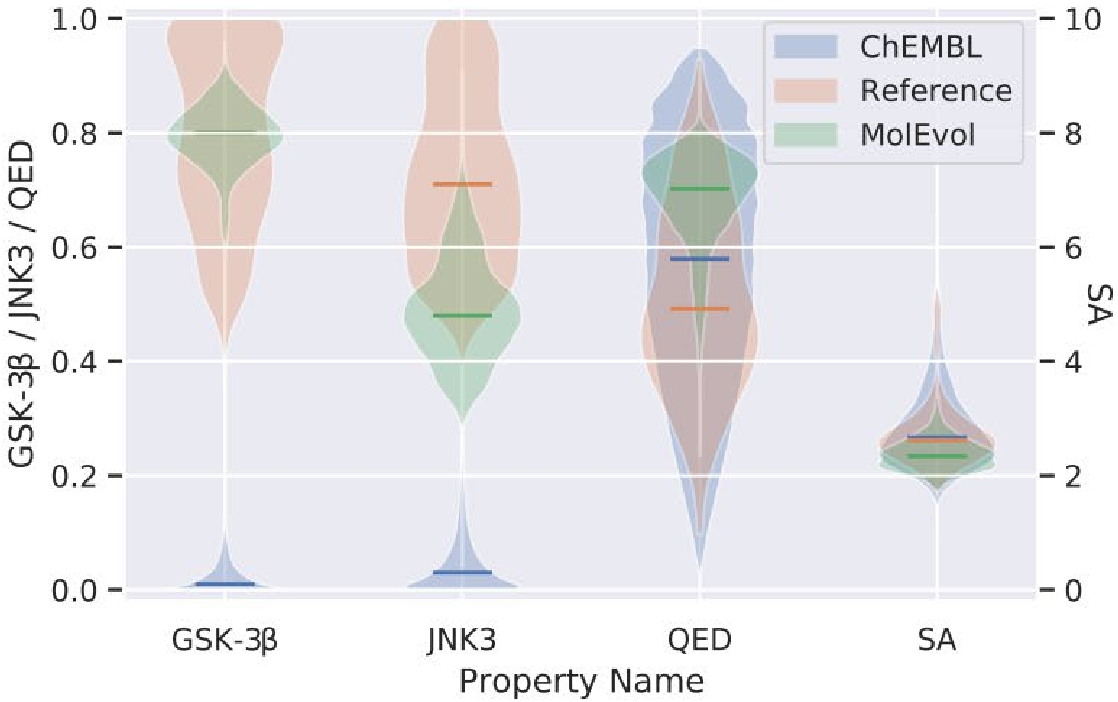
- Baselines:
  - RationaleRL (Jin et al., 2020): MCTS for rationale extraction + RL.
  - REINVENT (Olivecrona et al., 2017): RL on SMILES string.
  - MSO (Winter et al., 2019): Particle Swarm Optimization (PSO) in latent space.
  - GA-D(t) (Nigam et al., 2020): neural network-enhanced genetic algorithm.

Algorithm	MolEvol	[MCTS]	[FixM]	[FixR]	RationaleRL	REINVENT	MSO	GA-D(t)
Success rate	**93.0%**	77.7%	67.3%	66.3%	61.1%	46.6%	57.7%	62.0%
Novelty	**75.7%**	72.5%	67.4%	54.6%	57.4%	66.4%	28.6%	19.4%
Diversity	0.681	0.707	0.723	0.727	**0.749**	0.666	-	-
QNU	**52.7%**	47.4%	39.3%	28.3%	29.5%	7.4%	16.4%	12.0%

Property Score Distributions

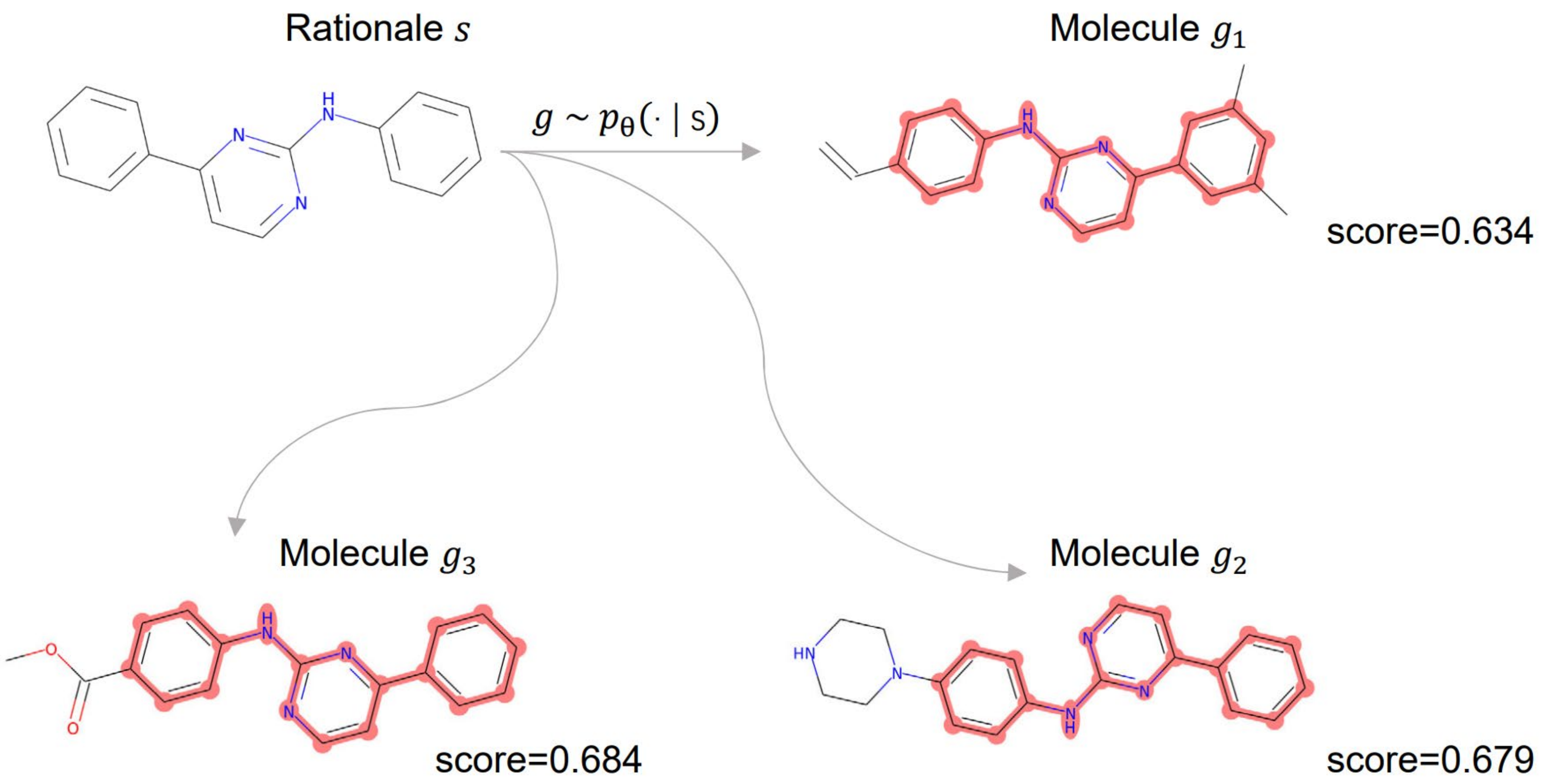


Distribution of $f(g)$ from each iteration.



Distribution of the property scores.

Sample Rationale



Future Work

- Generalized methodology
  1. First identify useful structural elements,
  2. Then improve the design based on these elements.
  3. Reiterate the process.

- Discrete structure optimization in other domains
  - Program synthesis
  - AutoML

Thanks for listening!

For more details, please refer to our paper/full slides/poster/repo:

Paper   Full Slides   Poster   GitHub