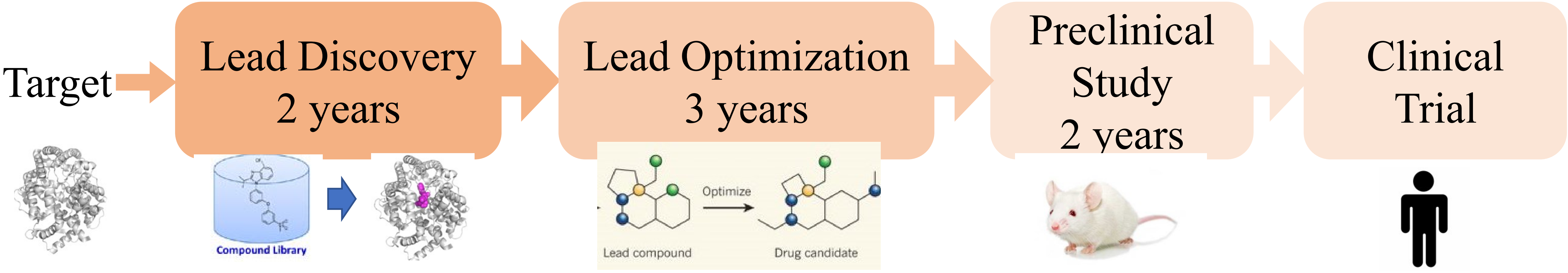
Graph Representation Learning for Drug Discovery

Jian Tang
Mila-Quebec AI Institute
CIFAR AI Research Chair
HEC Montreal
www.jian-tang.com

The Process of Drug Discovery

- A very long and costly process
  - On average takes more than 10 years and $2.5B to get a drug approved



Target

Lead Discovery
2 years

Lead Optimization
3 years

Preclinical
Study
2 years

Clinical
Trial

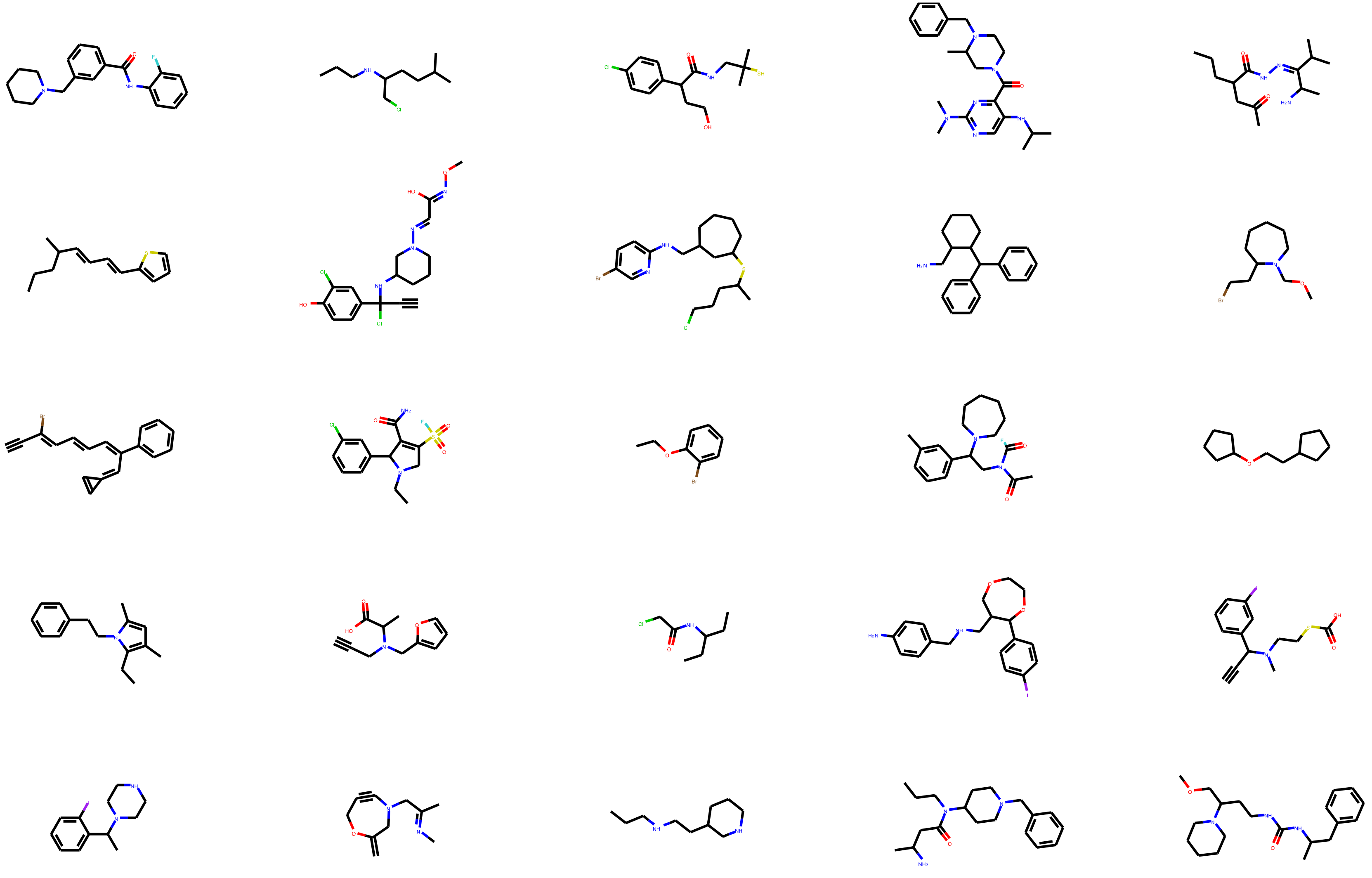
Screen millions of functional molecules; Found by serendipity: Penicillin

Modify the molecule to improve specific properties.
e.g. toxicity, SA

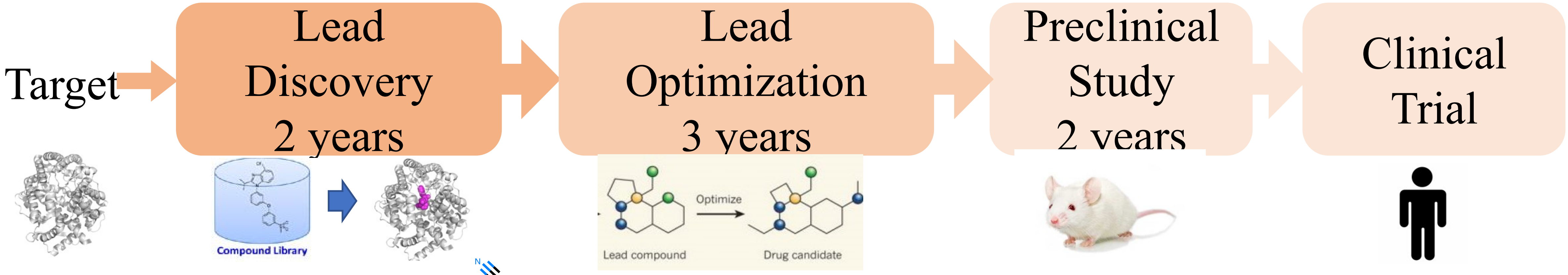
In-vitro and in-vivo experiments; synthesis

Multiple Phases

Molecules

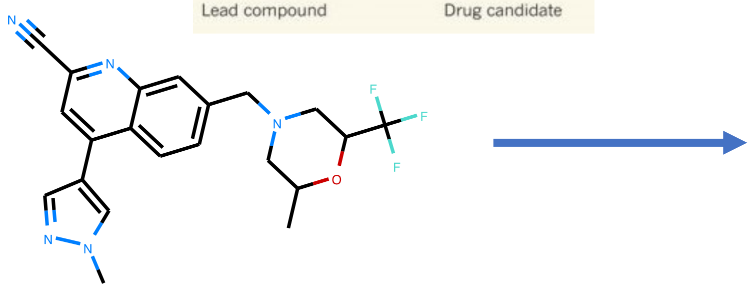


Research Problems



Target → Lead Discovery 2 years → Lead Optimization 3 years → Preclinical Study 2 years → Clinical Trial

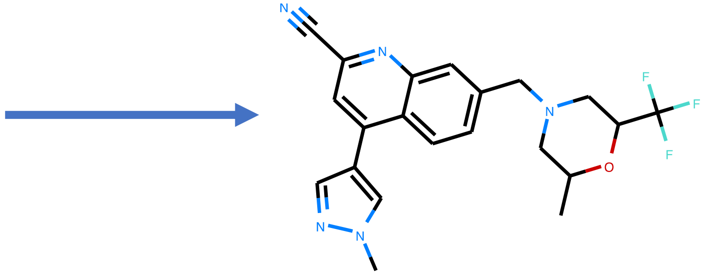
Molecule Property Prediction



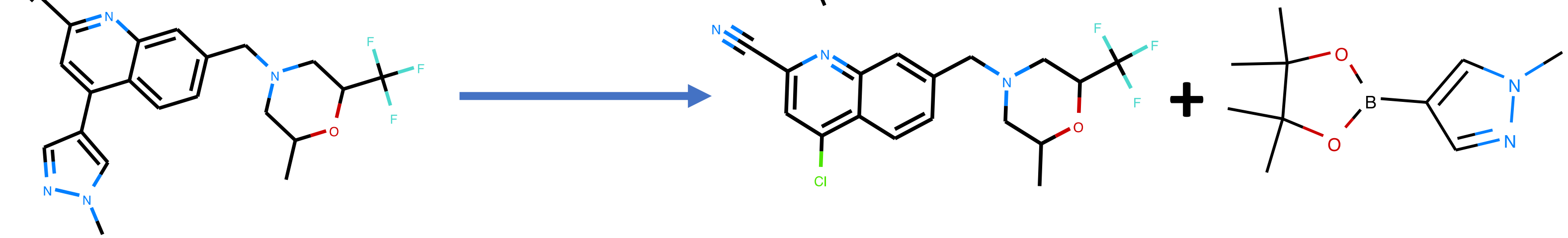
Property

Molecule Design and Optimization

Property

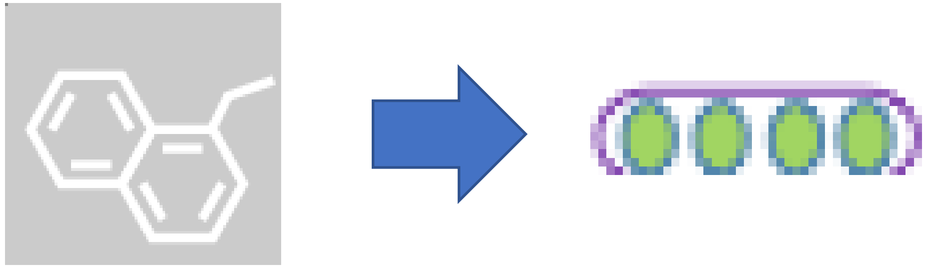


Retrosynthesis Prediction



Molecule Properties Prediction

- Predicting the properties of molecules or compounds is a fundamental problem in drug discovery
  - E.g., in the stage of virtual screening
- Each molecule is represented as a graph
- The fundamental problem: how to represent **a whole molecule (graph)**



Graph Neural Networks

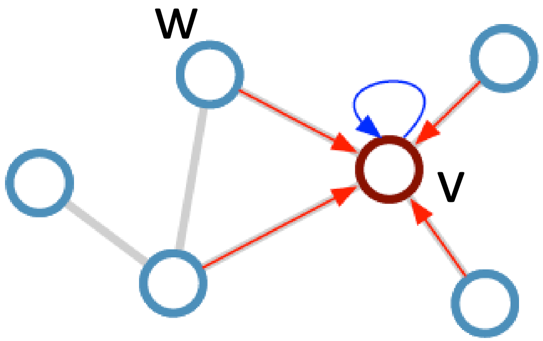
- Techniques for learning node/graph representations
  - Graph convolutional Networks (Kipf et al. 2016)
  - Graph attention networks (Veličković et al. 2017)
- Neural Message Passing (Gilmer et al. 2017)

MESSAGE PASSING: $M_k(h_v^k, h_w^k, e_{vw})$

AGGREGATE : $m_v^{k+1} = \mathrm{AGGREGATE}\{M_k\left(h_v^k, h_w^k, e_{vw}\right): w \in N(v)\}$

COMBINE : $h_v^{k+1} = \mathrm{COMBINE}(h_v^k, m_v^{k+1})$

READOUT: $g = \mathrm{READOUT}\{h_v^K: v \in G\}$



InfoGraph: Unsupervised and Semi-supervised Whole-Graph Representation Learning (Sun et al. ICLR'20)

- For supervised methods based on graph neural networks, a large number of labeled data are required for training
- The number of labeled data are very limited in drug discovery
  - A large amount of unlabeled data (molecules) are available
- This work: how to effectively learn whole graph representations in unsupervised or semi-supervised fashion

Fanyun Sun, Jordan Hoffman, Vikas Verma and Jian Tang. **InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization**. ICLR'20.

InfoGraph: Unsupervised Whole-Graph Representation Learning (Sun et al. ICLR'20)

- Maximizing the **mutual information** between the whole graph representation $H_\varphi(G)$ and all the sub-structure representation h_φ^i .
  - Ensure the graph representation capture the predominant information among all the substructures
- K-layer graph neural networks:

$$h_v^{(k)} = \text{COMBINE}^{(k)} \left(h_v^{(k-1)}, \text{AGGREGATE}^{(k)} \left(\left\{ \left(h_v^{(k-1)}, h_u^{(k-1)}, e_{uv} \right) : u \; 2 \; N(v) \right\} \right) \right)$$

- Summarize the local structure information at every node *i*:

$$h_\varphi^i = \text{CONCAT}(\{h_i^{(k)}\}_{k=1}^K)$$

- Summarize the information of the whole graph:

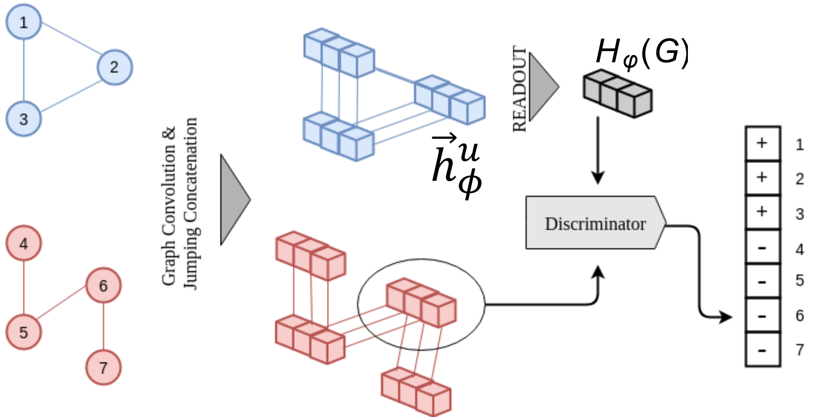
$$H_\varphi(G) = \text{READOUT}(\{h_\varphi^i\}_{i=1}^N)$$

Fanyun Sun, Jordan Hoffman, Vikas Verma and Jian Tang. **InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization**. ICLR'20.

InfoGraph: Unsupervised Whole-Graph Representation Learning

- Maximizing the **mutual information** between the whole graph representation $H_\varphi(G)$ and all the sub-structure representation $\vec{h}_\phi^u$

$$\hat{\phi}, \hat{\psi} = \arg\max_{\phi,\psi} \sum_{G\in\mathbf{G}} \frac{1}{|G|} \sum_{u\in G} I_{\phi,\psi}(\vec{h}_\phi^u; H_\phi(G)).$$



- We use the Jensen-Shannon MI estimator:

$$I_{\phi,\psi}(h_\phi^i(G); H_\phi(G)) :=$$
$$\mathbb{E}_{\mathbb{P}}[-\text{sp}(-T_{\phi,\psi}(\vec{h}_\phi^i(x), H_\phi(x)))] - \mathbb{E}_{\mathbb{P}\times\tilde{\mathbb{P}}}[\text{sp}(T_{\phi,\psi}(\vec{h}_\phi^i(x'), G_\phi(x)))]$$

- Where x is an input sample, x' is a negative graph sample, $\text{sp}(z) = \log(1 + e^z)$, $T(,)$ is a neural network

InfoGraph*: Semi-supervised Graph Representation Learning

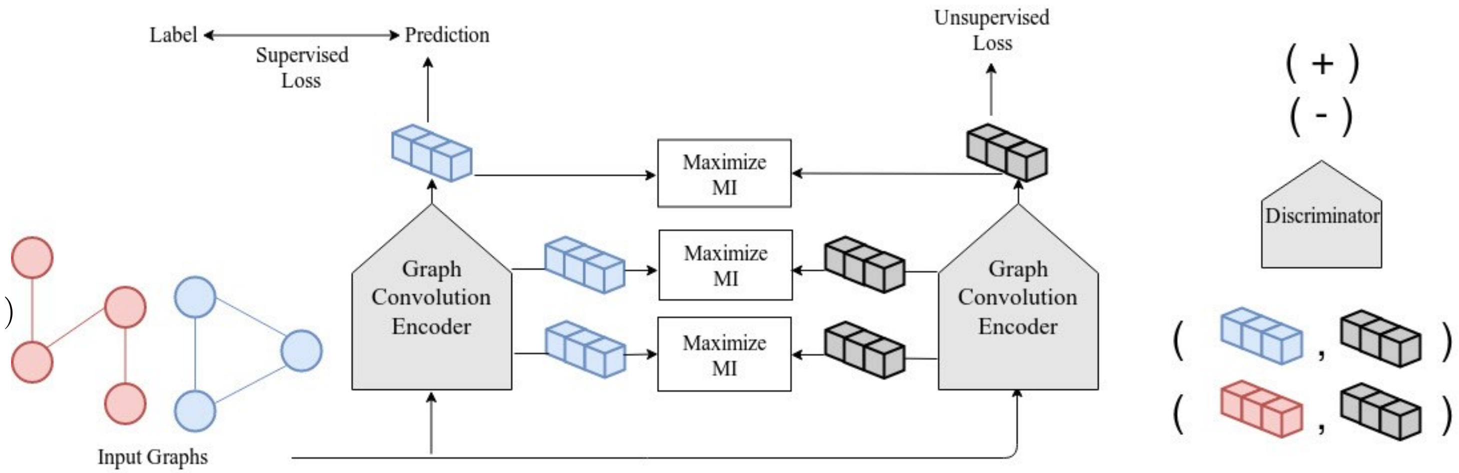
- Two objective functions:
  - Supervised loss
  - Unsupervised loss
- Simply combining the two objectives using the same encoder may lead to ”negative transfer”
  - The two objectives may favor different information

$$\mathrm{L_{total}} = \sum_{\mathrm{i=1}}^{|\mathbb{G}^{\mathrm{L}}|} \mathrm{L_{supervised}}(\mathrm{y}_\phi(\mathrm{G_i}), \mathrm{o_i}) + \lambda \sum_{\mathrm{j=1}}^{|\mathbb{G}^{\mathrm{L}}|+|\mathbb{G}^{\mathrm{U}}|} \mathrm{L_{unsupervised}}(\mathrm{h}_\phi(\mathrm{G_j}); \mathrm{H}_\phi(\mathrm{G_j}))$$

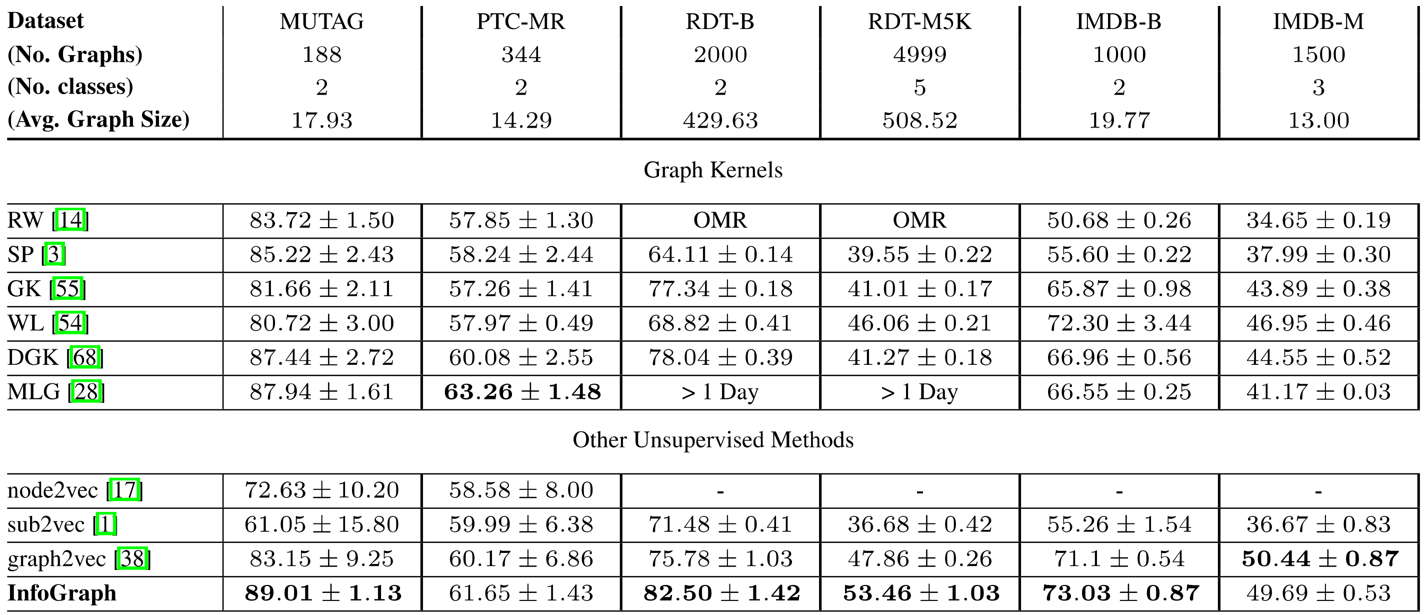
InfoGraph*: Semi-supervised Graph Representation Learning

- Two different encoders for the supervised and unsupervised tasks
- Maximize the mutual information of the representations learned by the two encoders at all levels (or layers)

$$\mathrm{L_{total}} = \sum_{\mathrm{i}=1}^{|\mathbb{G}^{\mathrm{L}}|} \mathrm{L_{supervised}}(\mathrm{y}_\phi(\mathrm{G_i}), \mathrm{o_i}) + \sum_{j=1}^{|\mathbb{G}^{L}|+|\mathbb{G}^{U}|} \mathrm{L_{unsupervised}}(h_\varphi(G_j); H_\varphi(G_j)) \\ - \lambda \sum_{j=1}^{|\mathbb{G}^{L}|+|\mathbb{G}^{U}|} \frac{1}{|G_j|} \sum_{k=1}^{K} I(H_\phi^k(G_j); H_\varphi^k(G_j)$$

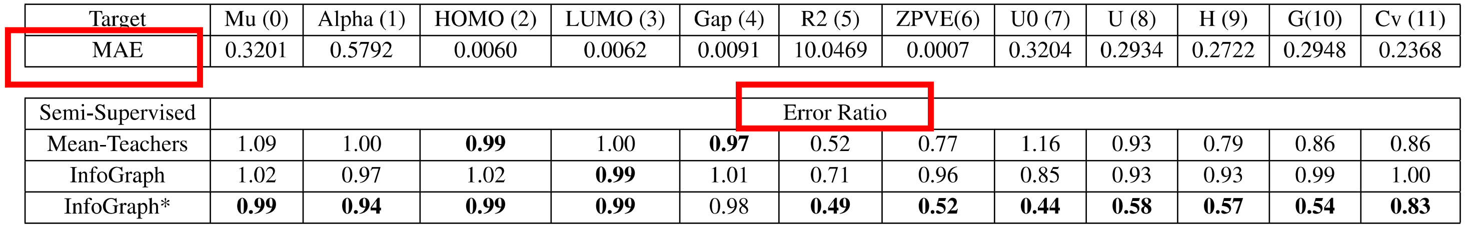


Results on Graph Classification and Regression



Dataset	MUTAG	PTC-MR	RDT-B	RDT-M5K	IMDB-B	IMDB-M
(No. Graphs)	188	344	2000	4999	1000	1500
(No. classes)	2	2	2	5	2	3
(Avg. Graph Size)	17.93	14.29	429.63	508.52	19.77	13.00
Graph Kernels						
RW [14]	83.72 ± 1.50	57.85 ± 1.30	OMR	OMR	50.68 ± 0.26	34.65 ± 0.19
SP [3]	85.22 ± 2.43	58.24 ± 2.44	64.11 ± 0.14	39.55 ± 0.22	55.60 ± 0.22	37.99 ± 0.30
GK [55]	81.66 ± 2.11	57.26 ± 1.41	77.34 ± 0.18	41.01 ± 0.17	65.87 ± 0.98	43.89 ± 0.38
WL [54]	80.72 ± 3.00	57.97 ± 0.49	68.82 ± 0.41	46.06 ± 0.21	72.30 ± 3.44	46.95 ± 0.46
DGK [68]	87.44 ± 2.72	60.08 ± 2.55	78.04 ± 0.39	41.27 ± 0.18	66.96 ± 0.56	44.55 ± 0.52
MLG [28]	87.94 ± 1.61	**63.26 ± 1.48**	> 1 Day	> 1 Day	66.55 ± 0.25	41.17 ± 0.03
Other Unsupervised Methods						
node2vec [17]	72.63 ± 10.20	58.58 ± 8.00	-	-	-	-
sub2vec [1]	61.05 ± 15.80	59.99 ± 6.38	71.48 ± 0.41	36.68 ± 0.42	55.26 ± 1.54	36.67 ± 0.83
graph2vec [38]	83.15 ± 9.25	60.17 ± 6.86	75.78 ± 1.03	47.86 ± 0.26	71.1 ± 0.54	**50.44 ± 0.87**
InfoGraph	**89.01 ± 1.13**	61.65 ± 1.43	**82.50 ± 1.42**	**53.46 ± 1.03**	**73.03 ± 0.87**	49.69 ± 0.53

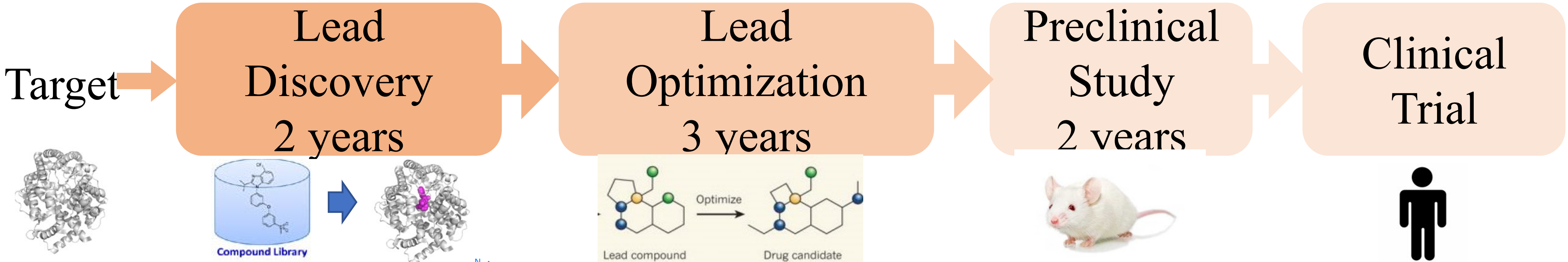
Table 1: Graph classification accuracy with unsupervised methods



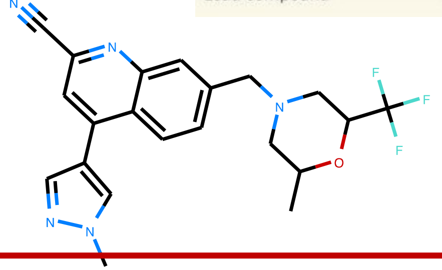
Target	Mu (0)	Alpha (1)	HOMO (2)	LUMO (3)	Gap (4)	R2 (5)	ZPVE(6)	U0 (7)	U (8)	H (9)	G(10)	Cv (11)
MAE	0.3201	0.5792	0.0060	0.0062	0.0091	10.0469	0.0007	0.3204	0.2934	0.2722	0.2948	0.2368
Semi-Supervised	Error Ratio											
Mean-Teachers	1.09	1.00	**0.99**	1.00	**0.97**	0.52	0.77	1.16	0.93	0.79	0.86	0.86
InfoGraph	1.02	0.97	1.02	**0.99**	1.01	0.71	0.96	0.85	0.93	0.93	0.99	1.00
InfoGraph*	**0.99**	**0.94**	**0.99**	**0.99**	0.98	**0.49**	**0.52**	**0.44**	**0.58**	**0.57**	**0.54**	**0.83**

Table 2: Results of semi-supervised experiments on QM9 data set.

Research Problems



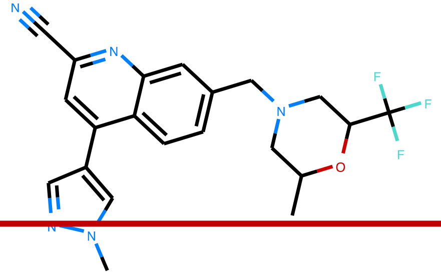
Molecule Property Prediction



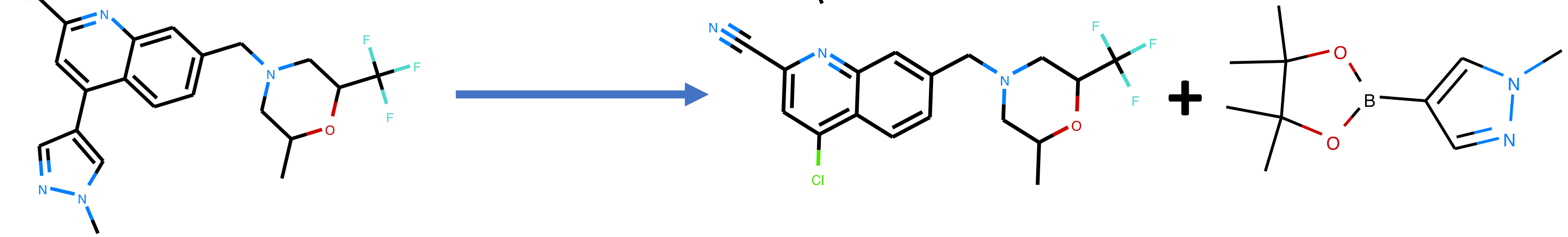
→ Property

Molecule Design and Optimization

Property →



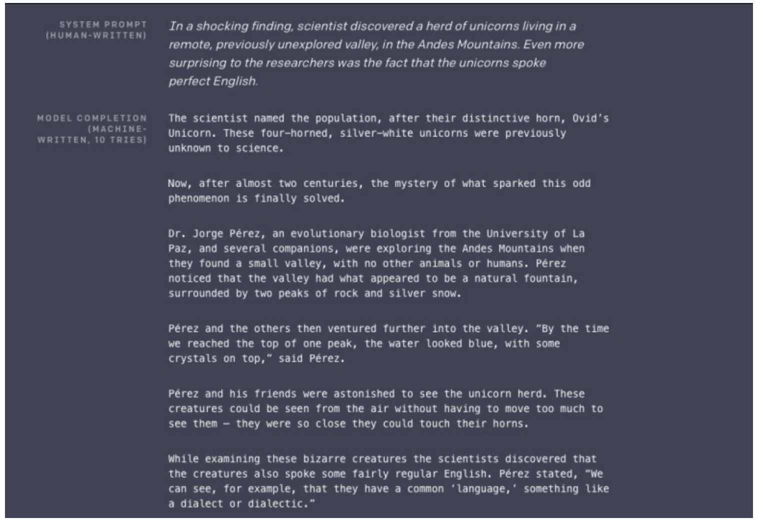
Retrosynthesis Prediction



Molecule Generation and Optimization

- Deep generative models for data generation

Image generation
(by StyleGAN, From Internet)



SYSTEM PROMPT (HUMAN-WRITTEN)

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

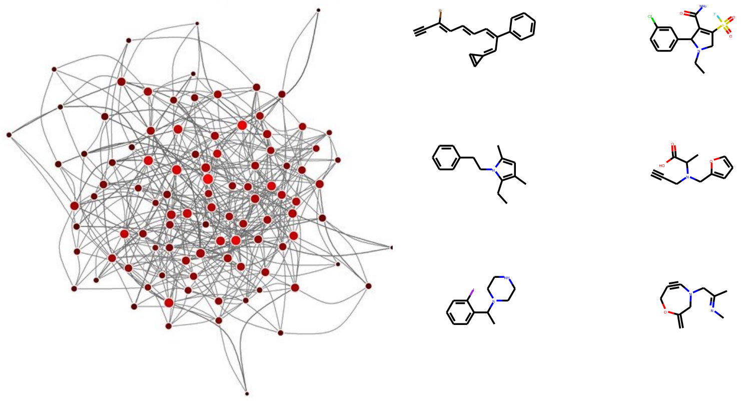
Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

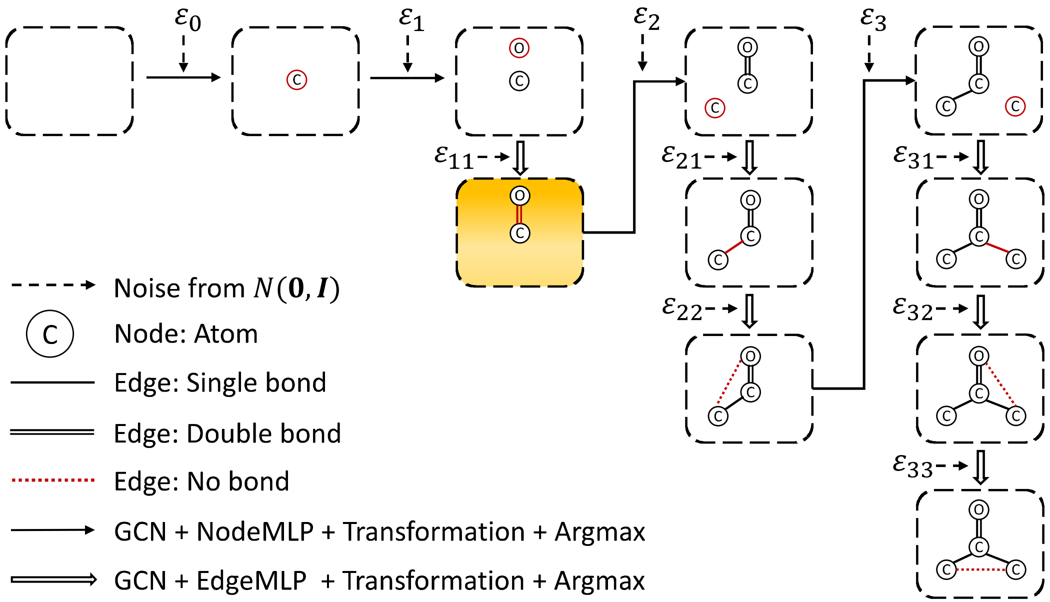
Text generated by by GPT-2,
Examples from Internet



Graphs?

GraphAF: an Autoregressive Flow for Molecular Graph Generation (Shi & Xu ICLR'20)

- Formulate graph generation as a sequential decision process
  - In each step, generate a new atom
  - Determine the bonds between the new atoms and existing atoms



(a) Sampling Framework

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang.
"GraphAF: a Flow-based Autoregressive Model for Molecular Graph Generation." ICLR'20.

Normalizing Flows (Dinh et al. 2016)

- Defines an invertible mapping from a base distribution (e.g. Gaussian Distribution) to observation space $f\colon \mathcal{Z} \to \mathcal{X}$

Data space $\mathcal{X}$

Latent space $\mathcal{Z}$

Inference

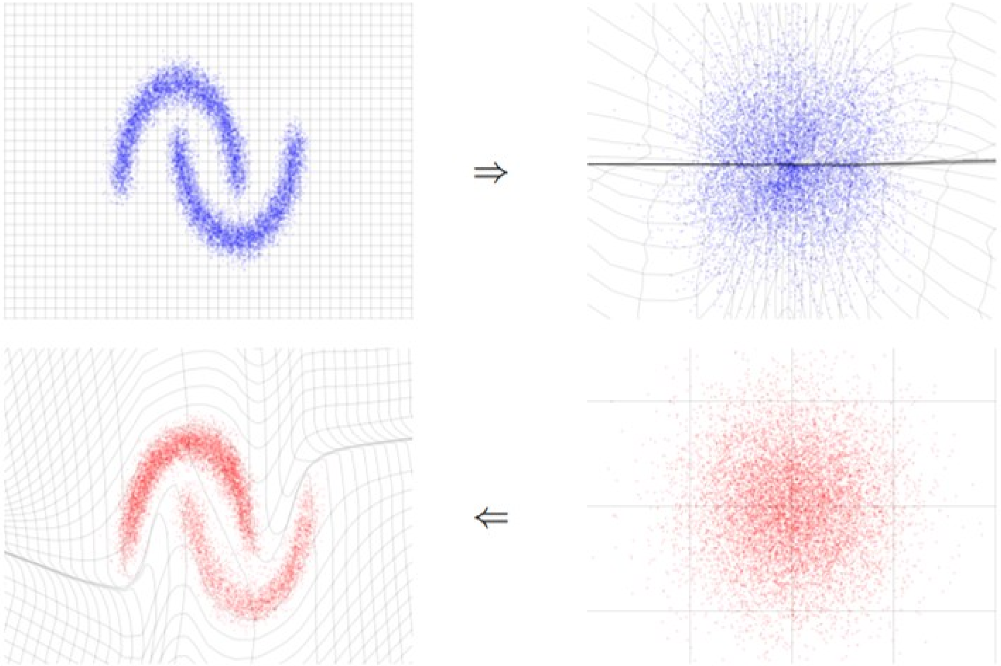
$x \sim \hat{p}_x$

$\mathrm{z} = f(x)$

Generation

$z \sim p_z$

$x = f(z)$



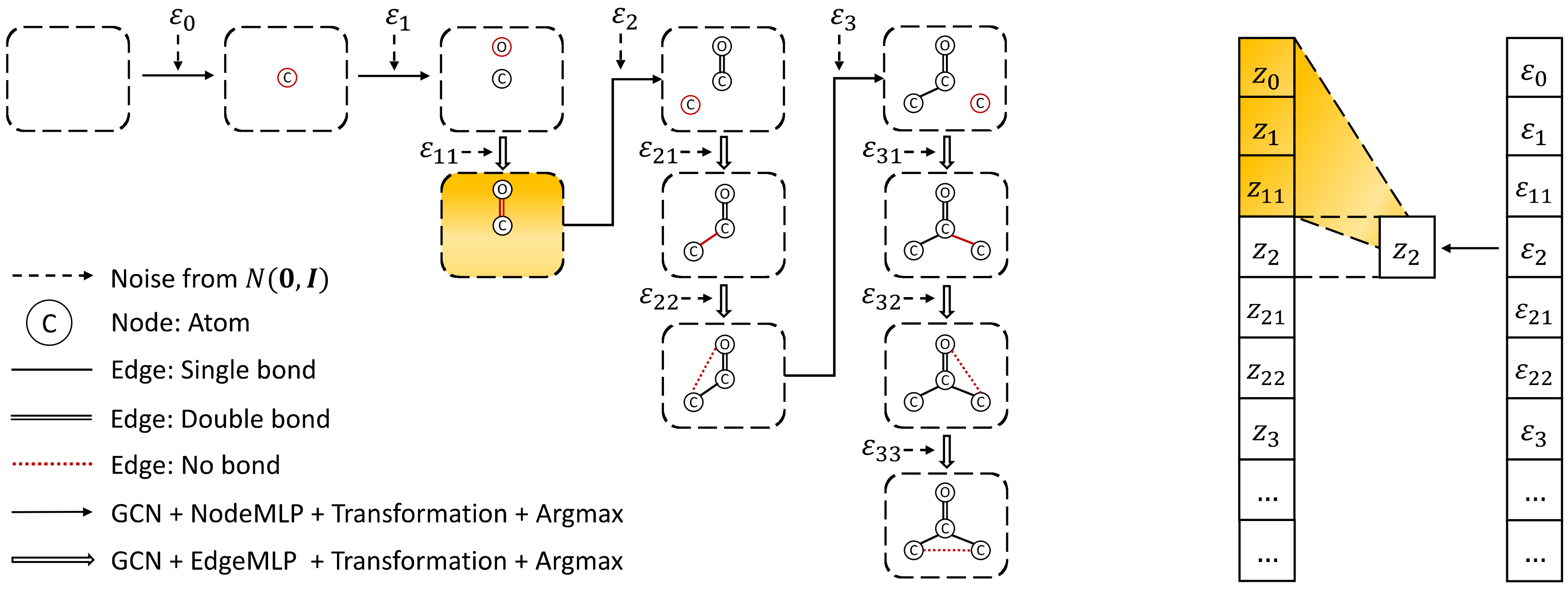
Change-of-Variables

$$p_X(x) = p_Z\left(f_\theta^{-1}(x)\right)\left|\det\frac{\partial f_\theta^{-1}(x)}{\partial x}\right|$$

Density estimation using Real NVP (2016)

GraphAF: an Autoregressive Flow for Molecular Graph Generation

- Traverse a graph through BFS-order
  - Transform each graph into a sequence of nodes and edges
- Defines an invertible mapping from a base distribution (Gaussian distribution) to the observations (graph nodes and edge sequences)

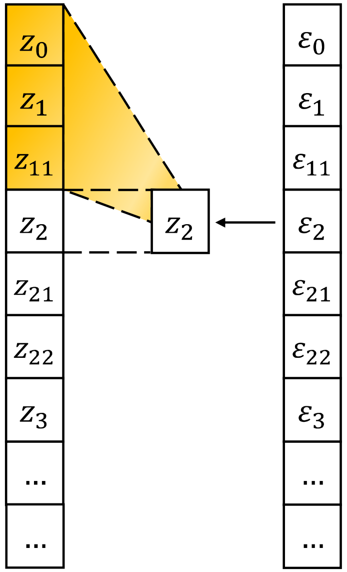


(a) Sampling Framework

(b) Autoregressive Flow

Advantages of GraphAF

- Strong capacity for data density modeling
  - Thanks to normalizing flow-based framework
- Training (from z to ϵ): parallel
  - Efficient training process
- Sampling (from ϵ to z): sequential
  - Effectively capture the graph structure
  - Feasible to incorporate chemical rules

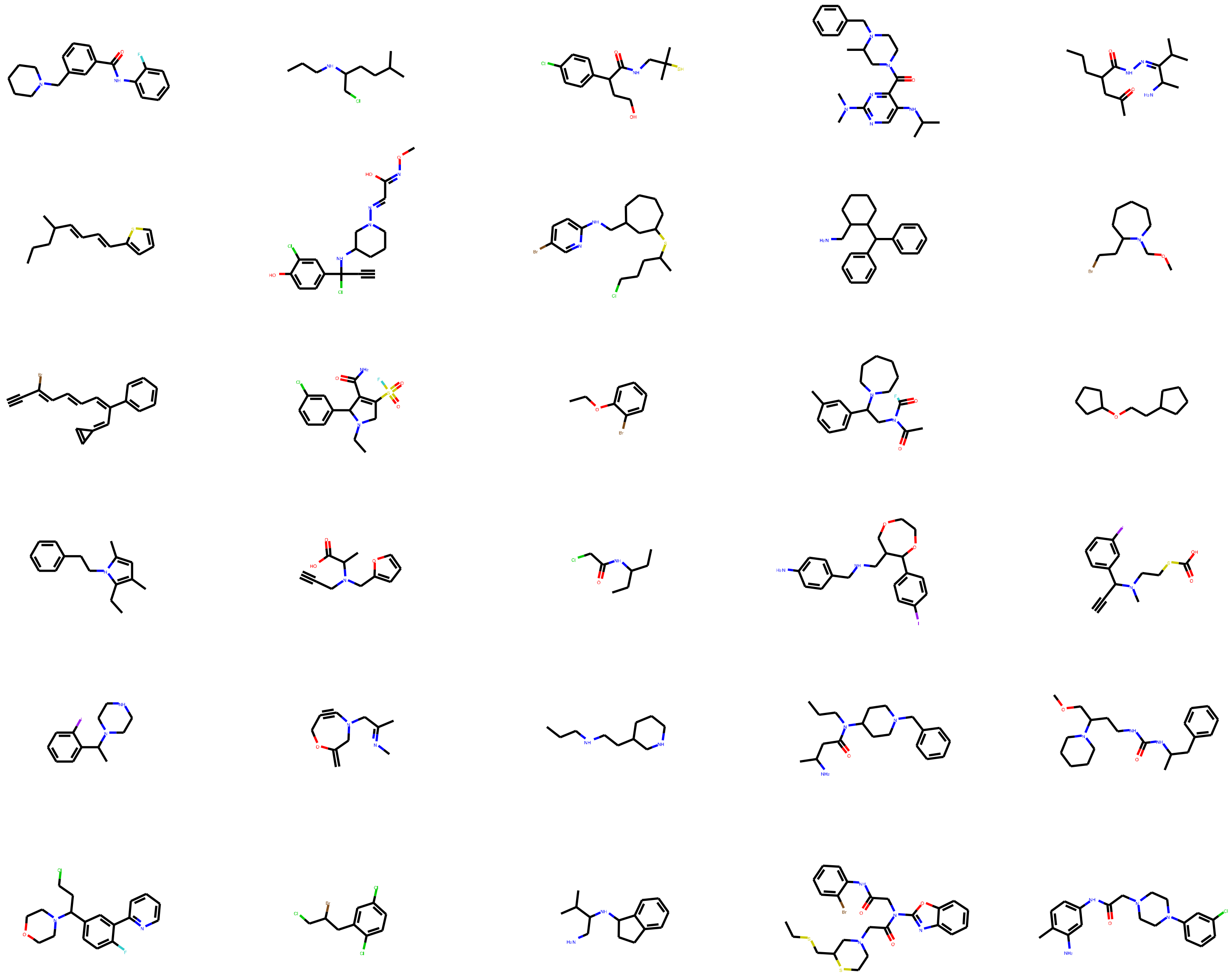


(b) Autoregressive Flow

Molecule Generation

- Training Data: ZINC250K
  - 250K drug-like molecules with a maximum atom number of 38
  - 9 atom types and 3 edge types

Method	Validity	Validity w/o check	Uniqueness	Novelty	Reconstruction
JT-VAE	100%	—	100%[‡]	100%[‡]	76.7%
GCPN	100%	20%[†]	99.97%[‡]	100%[‡]	—
MRNN	100%	65%	99.89%	100%	—
GraphNVP	42.60%	—	94.80%	100%	100%
GraphAF	100%	68%	99.10%	100%	100%

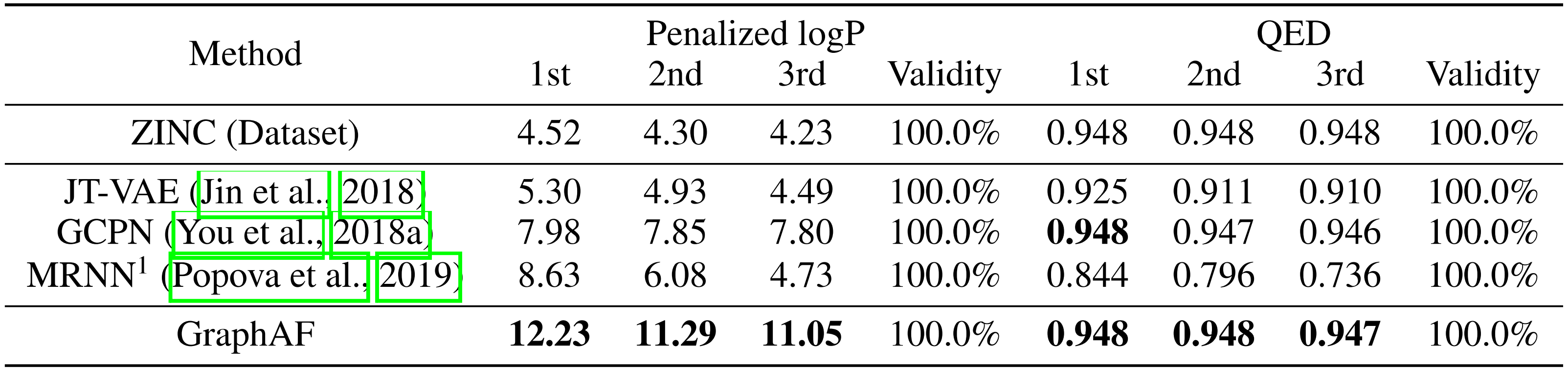


Goal-Directed Molecule Generation with Reinforcement Learning

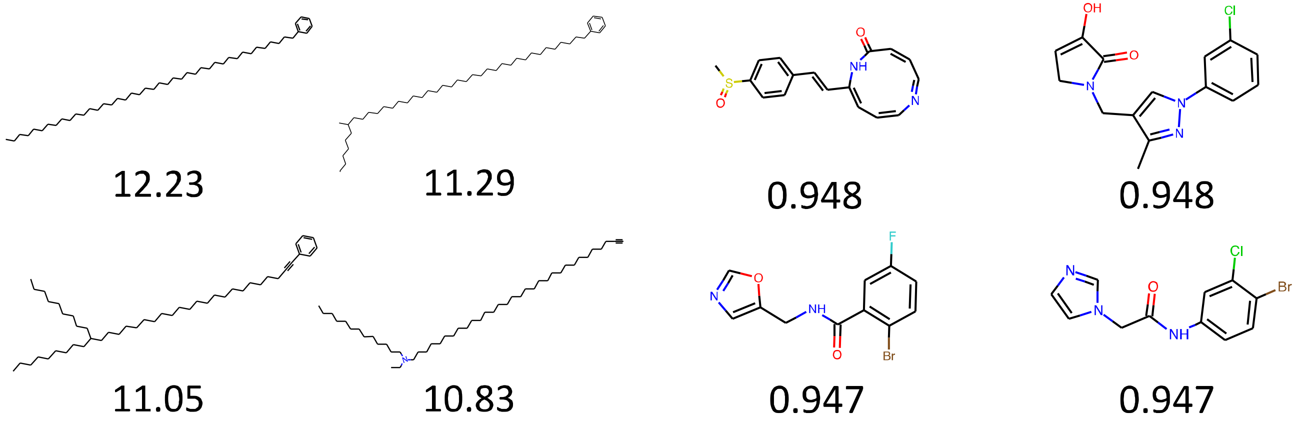
- Fine tune the generation policy with reinforcement learning to optimize the properties of generated molecules
- **State**: current subgraph G_i
- **Action**: generating a new atom (i.e. $\mathrm{p}(X_i|G_i)$) or a new edge ($\mathrm{p}(A_{ij}|G_i, X_i, A_{i,1:j-1})$).
- **Reward Design**: the properties of molecules (final reward) and chemical validity (intermediate and final reward)

Molecule Optimization

- Properties
  - Penalized logP
  - QED (druglikeness)

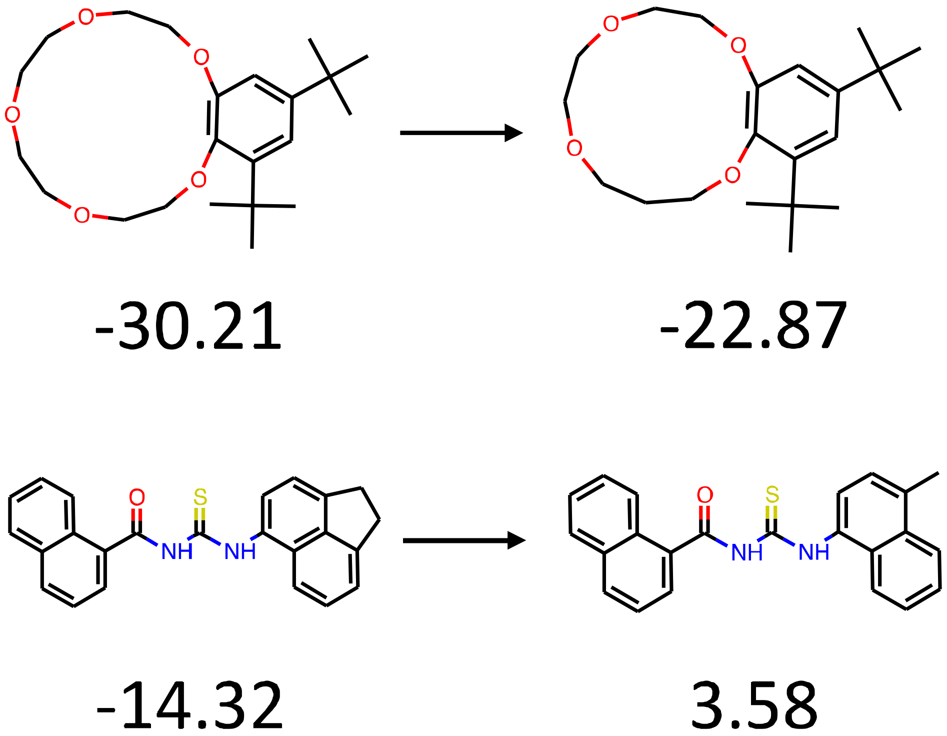


Method	Penalized logP 1st	2nd	3rd	Validity	QED 1st	2nd	3rd	Validity
ZINC (Dataset)	4.52	4.30	4.23	100.0%	0.948	0.948	0.948	100.0%
JT-VAE (Jin et al., 2018)	5.30	4.93	4.49	100.0%	0.925	0.911	0.910	100.0%
GCPN (You et al., 2018a)	7.98	7.85	7.80	100.0%	**0.948**	0.947	0.946	100.0%
MRNN[1] (Popova et al., 2019)	8.63	6.08	4.73	100.0%	0.844	0.796	0.736	100.0%
GraphAF	**12.23**	**11.29**	**11.05**	100.0%	**0.948**	**0.948**	**0.947**	100.0%



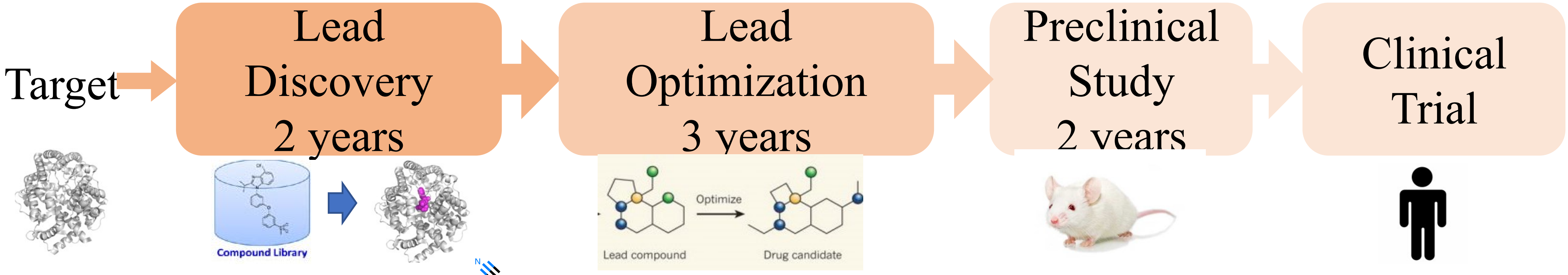
(a) Penalized logP optimization (b) QED optimization

Constrained Optimization



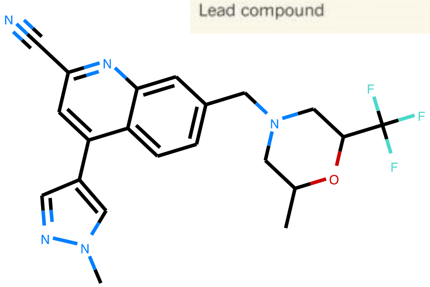
(c) Constrained optimization

Research Problems



Target → Lead Discovery 2 years → Lead Optimization 3 years → Preclinical Study 2 years → Clinical Trial

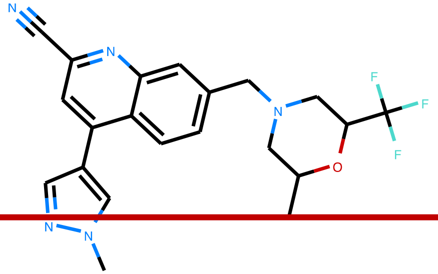
Property Prediction



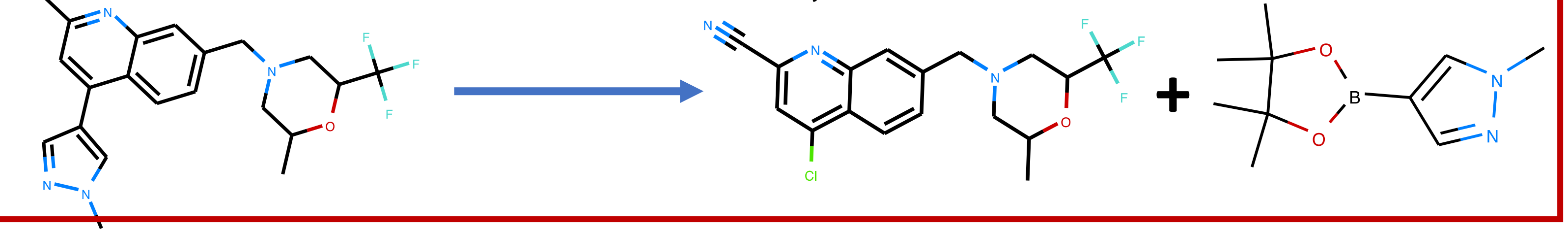
→ Property

Molecule Design and Optimization

Property →

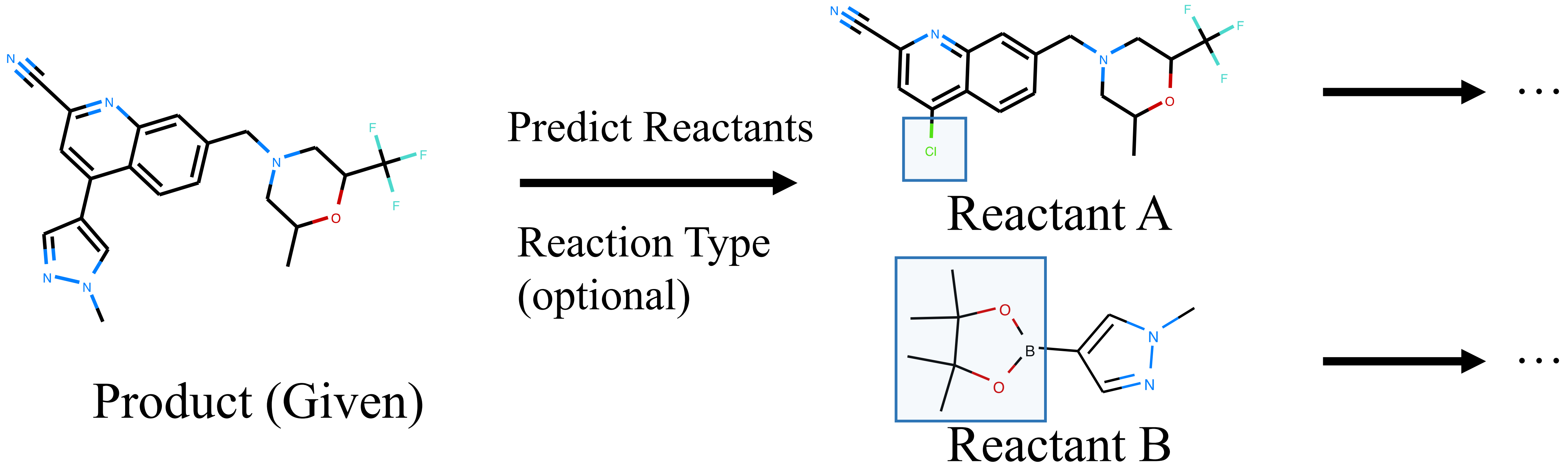


Retrosynthesis Prediction



Retrosynthesis Prediction

- Once a molecular structure is designed, how to synthesize it?
- Retrosynthesis planning/prediction
  - Identify a set of reactants to synthesize a target molecule



Product (Given)

Predict Reactants

Reaction Type (optional)

Reactant A

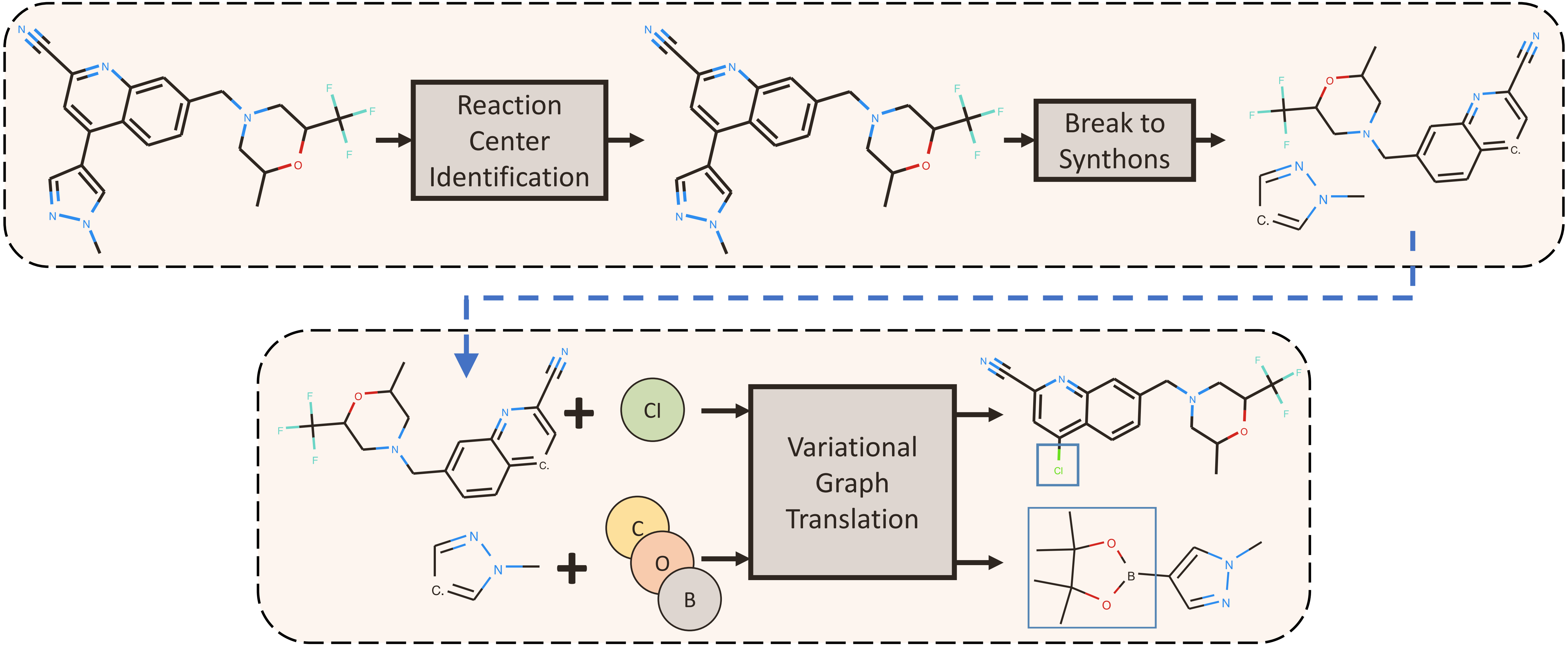
Reactant B

A Graph to Graphs Framework for Retrosynthesis Prediction (Shi et al. 2020)

- Each molecule is represented as a molecular graph
- Formulate the problem as a graph (product molecule) to a set of graphs (reactants)
- The whole framework are divided into two stages
  - Reaction center identification
  - Graph Translation

Chence Shi, Minkai Xu, Hongyu Guo, Ming Zhang and Jian Tang. **A Graph to Graphs Framework for Retrosynthesis Prediction**. ICML, 2020.

The G2Gs Framework (Shi et al. 2020)



Reaction Center Identification

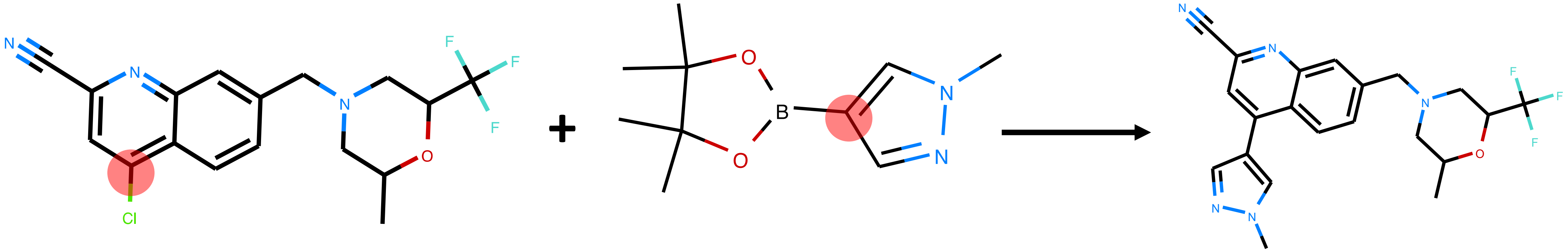
Break to Synthons

Variational Graph Translation

Shi et al., 2020, A Graph to Graphs Framework for Retrosynthesis Prediction

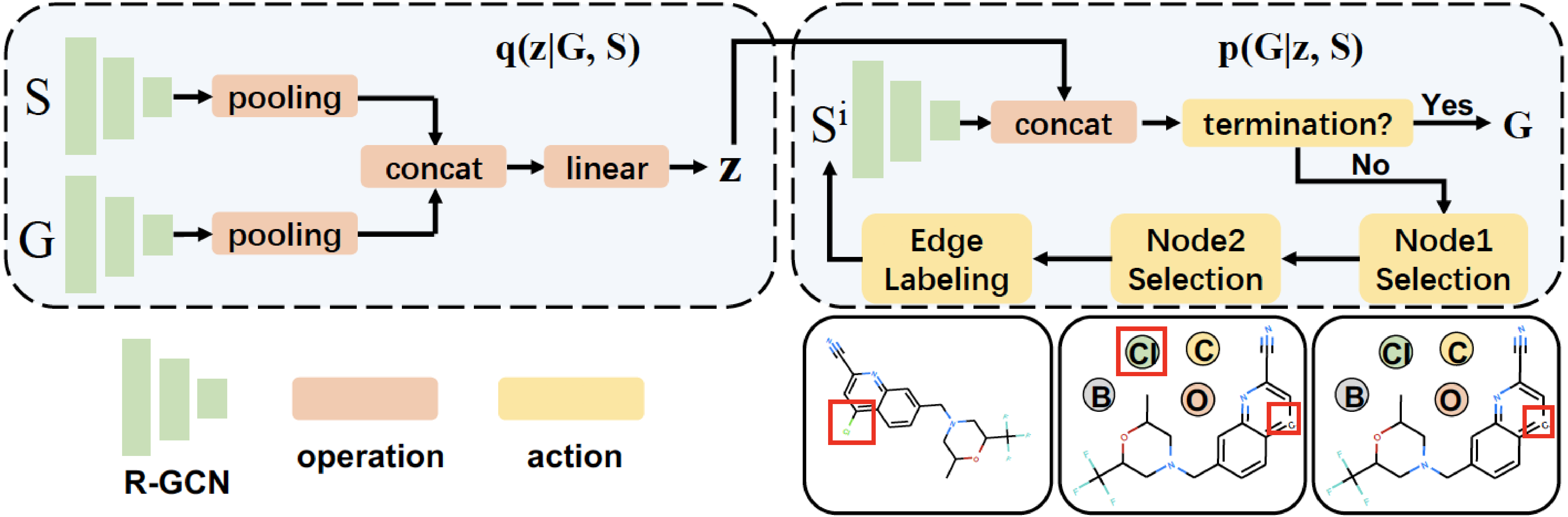
Reaction Center Prediction

- An atom pair *(i, j)* is a reaction center if:
  - There is a bond between atom *i* and atom *j* in product
  - There is no bond between atom *i* and atom *j* in reactants
- A supervised classification problem
  - Encode each edge with a graph neural network



Graph Translation

- Translate the incomplete synthon to the final reactant
- A variational graph to graph framework
  - A latent variable z is introduced to capture the uncertainty during translation



Experiments

- Experiment Setup
  - Benchmark data set USPTO-50K, containing 50k atom-mapped reactions
  - Evaluation metrics: top-k exact match (based on canonical SMILES) accuracy

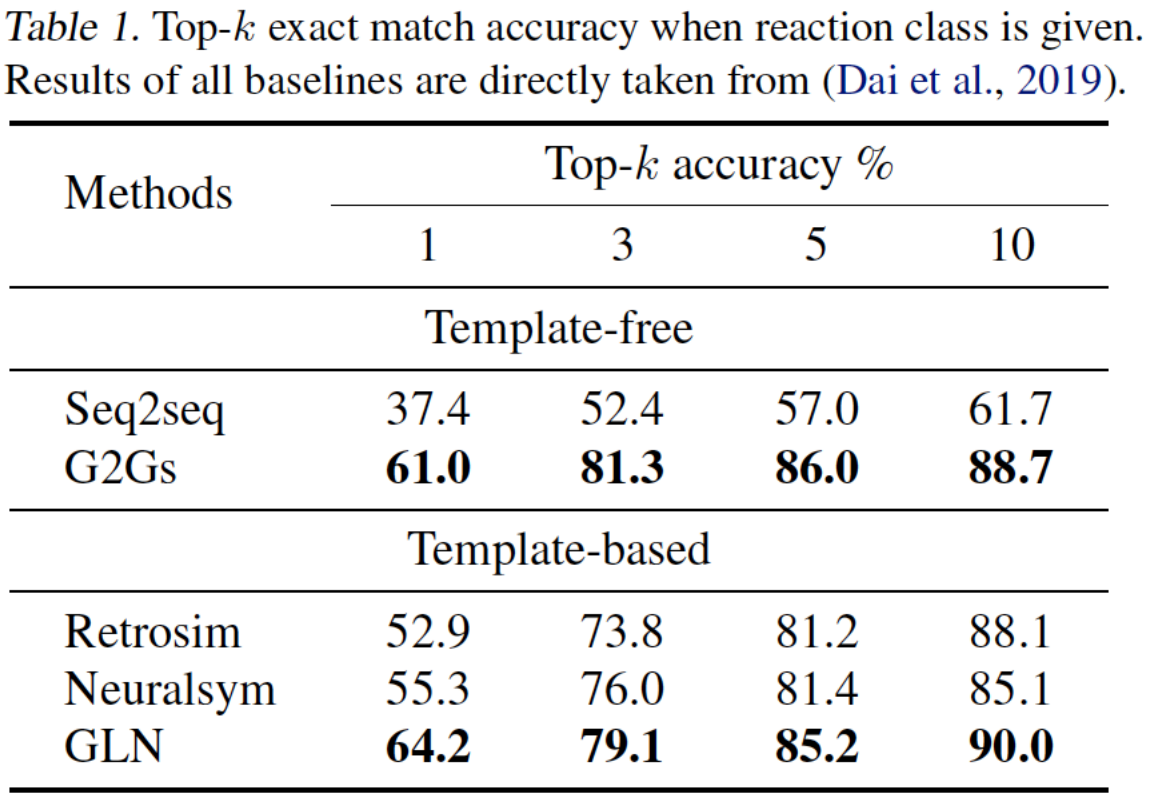


Table 1. Top-k exact match accuracy when reaction class is given. Results of all baselines are directly taken from (Dai et al., 2019).

Methods	Top-k accuracy %			
	1	3	5	10
Template-free				
Seq2seq	37.4	52.4	57.0	61.7
G2Gs	**61.0**	**81.3**	**86.0**	**88.7**
Template-based				
Retrosim	52.9	73.8	81.2	88.1
Neuralsym	55.3	76.0	81.4	85.1
GLN	**64.2**	**79.1**	**85.2**	**90.0**

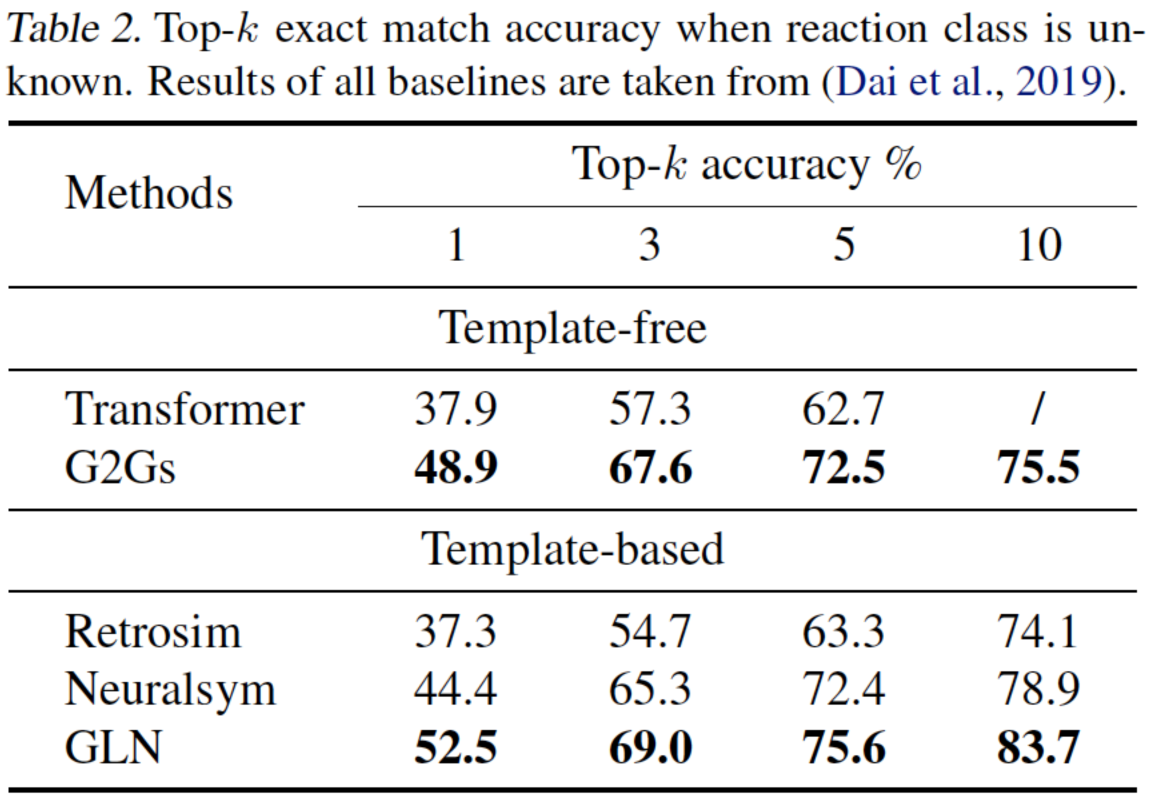


Table 2. Top-k exact match accuracy when reaction class is unknown. Results of all baselines are taken from (Dai et al., 2019).

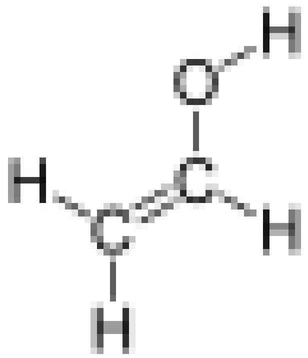
Methods	Top-k accuracy %			
	1	3	5	10
Template-free				
Transformer	37.9	57.3	62.7	/
G2Gs	**48.9**	**67.6**	**72.5**	**75.5**
Template-based				
Retrosim	37.3	54.7	63.3	74.1
Neuralsym	44.4	65.3	72.4	78.9
GLN	**52.5**	**69.0**	**75.6**	**83.7**

Going Beyond 2D Graphs: 3D Structures

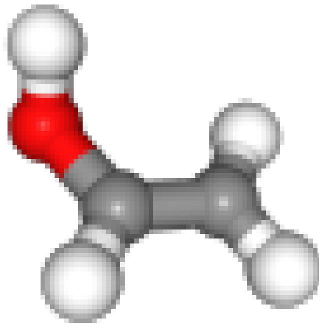
- A more natural and intrinsic representations of molecules: 3D conformations
  - Determines its biological and physical activities
  - E.g., charge distribution, steric constraints, and interaction with other molecules

C1CO

1D SMILES



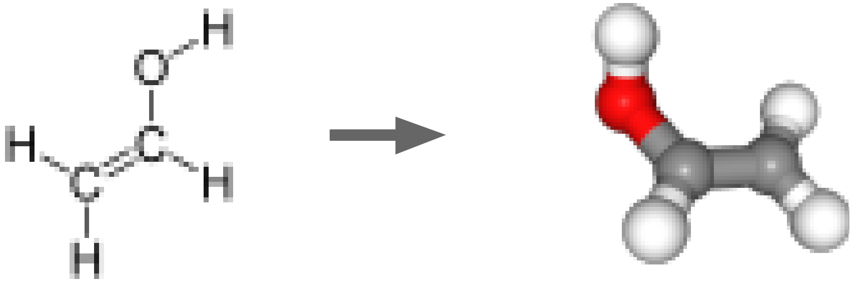
2D Graph



3D Conformation

Conformation Prediction

- For most molecules, their 3D structure are not available
- How to predict valid and stable conformations?
  - Each atom is represented as its 3D coordinates

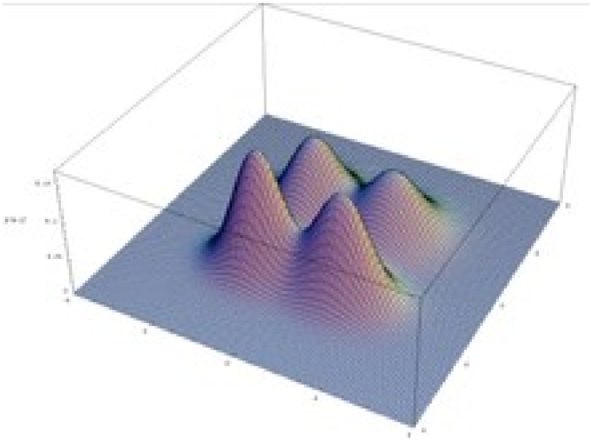


Traditional Approaches

- Experimental methods
  - Crystallography
  - Expensive and time consuming
- Computational methods
  - Molecular dynamics, Markov chain Monte Carlo
  - Very computational expensive, especially for large molecules

Machine Learning Approaches

- Train a model to predict molecular conformations $\boldsymbol{R}$ given the molecular graph $\mathcal{G}$, i.e., modeling $\mathrm{p}(\boldsymbol{R}|\mathcal{G})$ (Mansimov et al. 2019, Simm and Hernandez-Lobato 2020)
- Challenges
  - Conformations are rotation and translation equivalent
  - The distribution $\mathrm{p}(\boldsymbol{R}|\mathcal{G})$ is multimodal and very complex



Our Solution (Xu et al. 2020)

- A flexible generative model $p_\theta(\boldsymbol{R}|\mathcal{G})$ based on normalizing flows
  - Treating pairwise distances $\boldsymbol{d}$ as intermediate variables
  - First generating the distance $\boldsymbol{d}$ based $\mathcal{G}$, i. e. $p_\theta(\boldsymbol{d}|\mathcal{G})$
  - Generating conformations based on $\boldsymbol{d}$ and $\mathcal{G}$, i.e. $p_\theta(\boldsymbol{R}|\boldsymbol{d},\mathcal{G})$

$$p_\theta(\boldsymbol{R}|\mathcal{G}) = \int p(\boldsymbol{R}|\boldsymbol{d},\mathcal{G}) \cdot p_\theta(\boldsymbol{d}|\mathcal{G})\,\mathrm{d}\boldsymbol{d}$$

- Further correct $p_\theta(\boldsymbol{R}|\mathcal{G})$ with an energy-based tilting term $E_\phi(\boldsymbol{R},\mathcal{G})$

$$p_{\checkmark,\varphi}(R|G) \, / \;\; p_{\checkmark}(R|G) \cdot \exp(-E_\varphi(R,G))$$

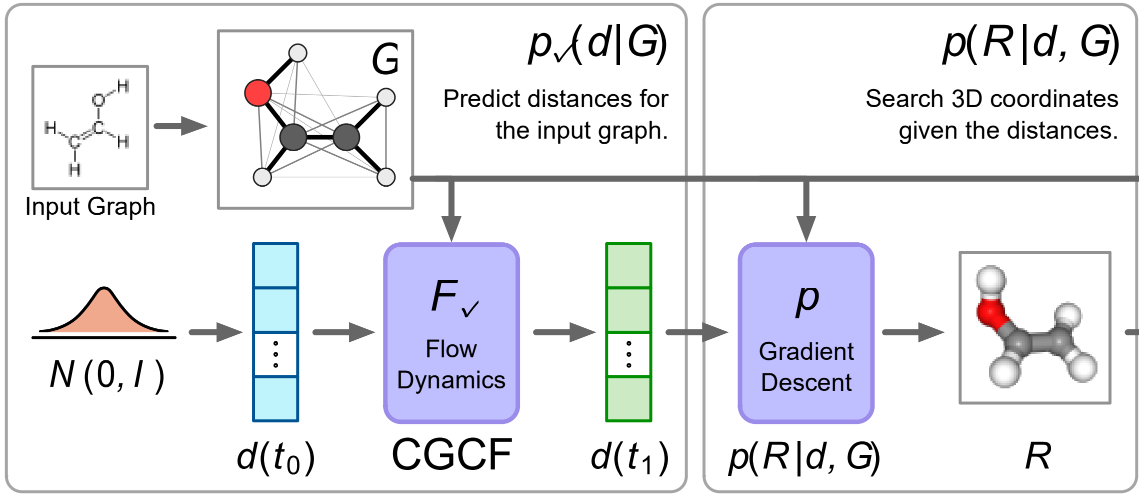
Minkai Xu*, Shitong Luo*, Yoshua Bengio, Jian Peng, Jian Tang. **Learning Neural Generative Dynamics for Molecular Conformation Generation**. In Submission.

Distance Geometry Generation $p_\theta(d|\mathcal{G})$

- Conditional Graph Continuous Flow (CGCF)
  - Defines an invertible mapping between a base distribution and the pairwise atom distance ***d*** conditioning on the molecular graph $\mathcal{G}$
  - Defines the continuous dynamics of distance ***d*** with Neural Ordinary Differential Equations (ODEs):

$$d = F_\theta(d(t_0), \mathcal{G}) = d(t_0) + \int_{t_0}^{t_1} \boxed{f_\theta(d(t), t; \mathcal{G})} dt, \quad d(t_0) \sim \mathcal{N}(0, I)$$

Graph Neural Networks

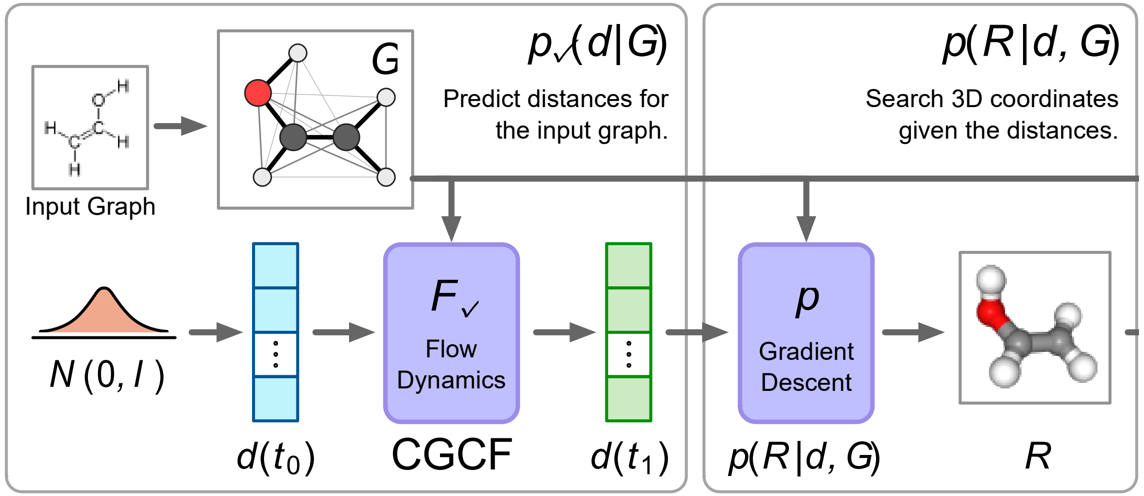


Conformation Prediction $p(\boldsymbol{R}|\boldsymbol{d}, \mathcal{G})$

- Defines the distribution of conformation R given the molecular graph $\mathcal{G}$ and the pairwise atom distance $\boldsymbol{d}$

$$p(\boldsymbol{R}|\boldsymbol{d}, \mathcal{G}) = \frac{1}{Z} \exp\left\{ - \sum_{e_{uv} \in \mathcal{E}} \alpha_{uv} \left(\|\boldsymbol{r}_u - \boldsymbol{r}_v\|_2 - d_{uv} \right)^2 \right\}$$

- Trying to find the conformations $\boldsymbol{R}$ that satisfy the distance constraints



$p_✓(d|G)$

Predict distances for the input graph.

$p(R|d, G)$

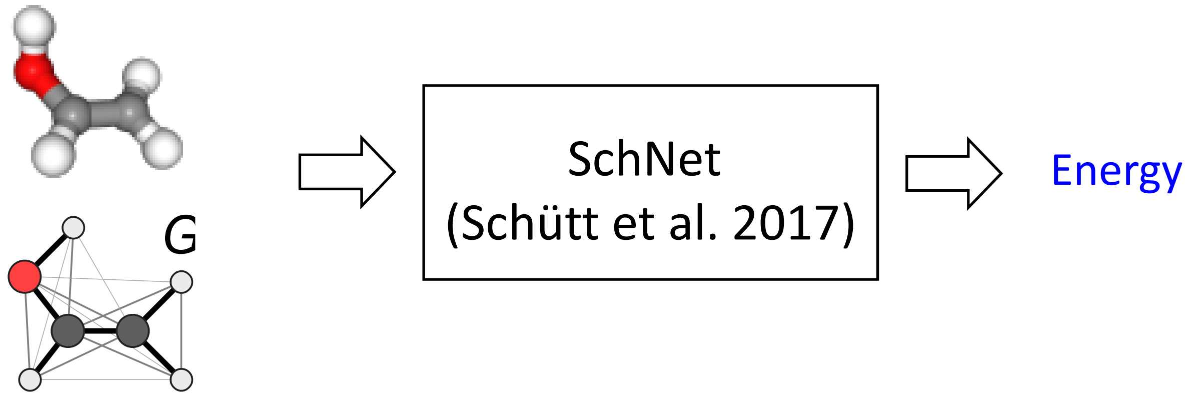
Search 3D coordinates given the distances.

Energy-based Tilting Model

- Further correct $p_\theta(\boldsymbol{R}|\mathcal{G})$ with an energy-based tilting term $E_\phi(\boldsymbol{R},\mathcal{G})$

$$p_{\checkmark,\varphi}(R|G) \ / \ \ p_{\checkmark}(R|G) \cdot \exp(-E_\varphi(R, G))$$

- Explicitly learn an energy function $E_\phi(\boldsymbol{R},\mathcal{G})$ with SchNet (Schütt et al. 2017)
  - Neural message passing in 3D space



SchNet
(Schütt et al. 2017)

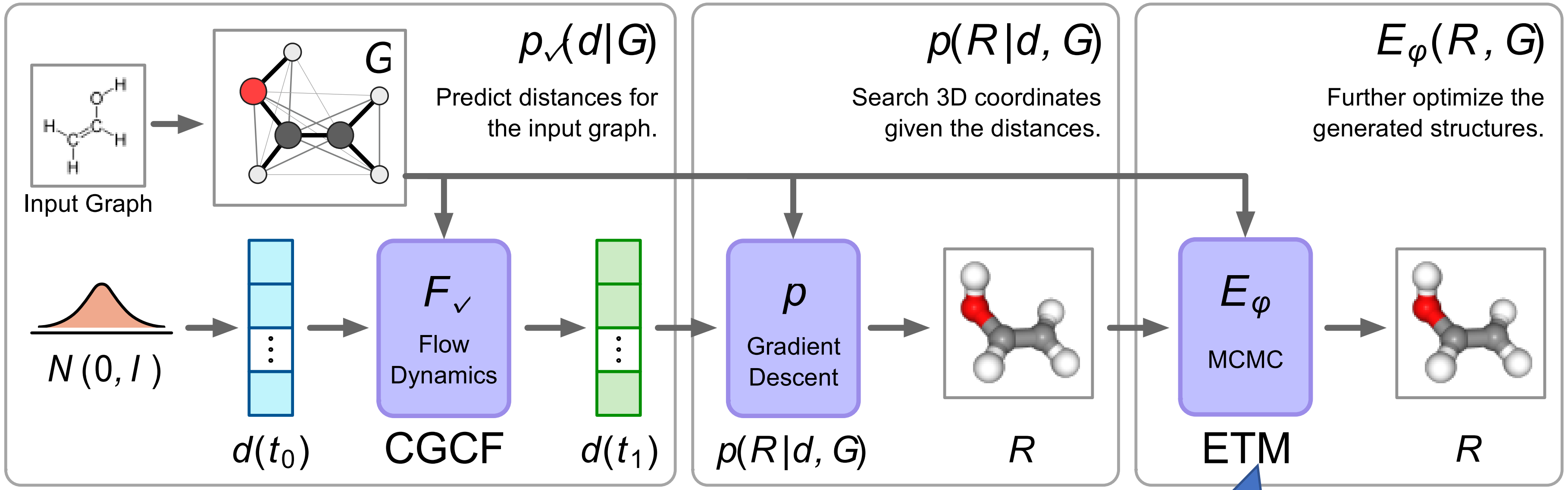
Energy

Training Energy Model

- Directly training EBMs with maximum likelihood is difficult
  - Involving a slow sampling process from the model distribution (e.g. with Langevin dynamics)
- Training EBMs with negative sampling
  - Treating observed conformations as positive examples
  - Generating negative conformations through the flow-based model $p_\theta(\boldsymbol{R}|\mathcal{G})$

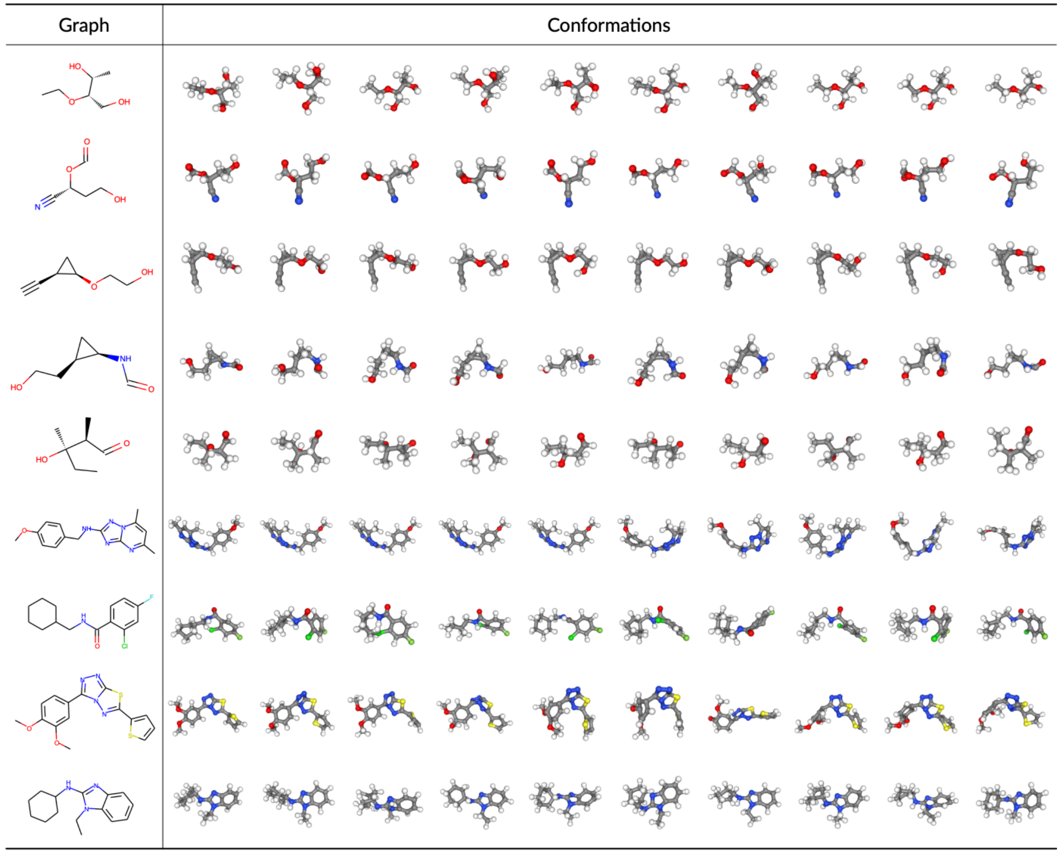
$$\mathcal{L}_{\text{nce}}(\boldsymbol{R},\mathcal{G};\phi) == -\,\mathbb{E}_{p_{\text{data}}}\Big[\log\frac{1}{1+\exp(E_\phi(\boldsymbol{R},\mathcal{G}))}\Big] - \mathbb{E}_{p_\theta}\Big[\log\frac{1}{1+\exp(-E_\phi(\boldsymbol{R},\mathcal{G}))}\Big]$$

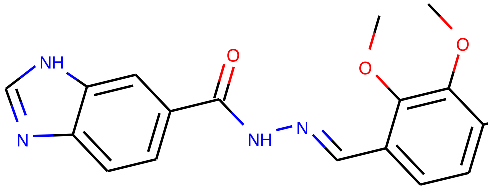
The Final Sampling Process:

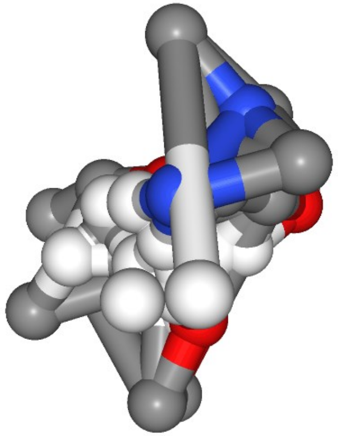

Input Graph
G
p✓(d|G)
Predict distances for the input graph.
p(R|d, G)
Search 3D coordinates given the distances.
Eφ(R, G)
Further optimize the generated structures.
N (0, I)
d(t0)
F✓
Flow Dynamics
CGCF
d(t1)
p
Gradient Descent
p(R|d, G)
R
Eφ
MCMC
ETM
R

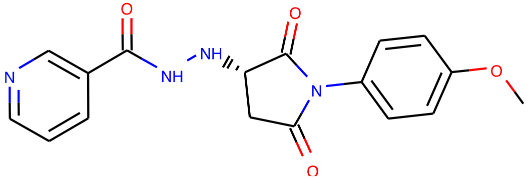
K-steps of Langevin Dynamics

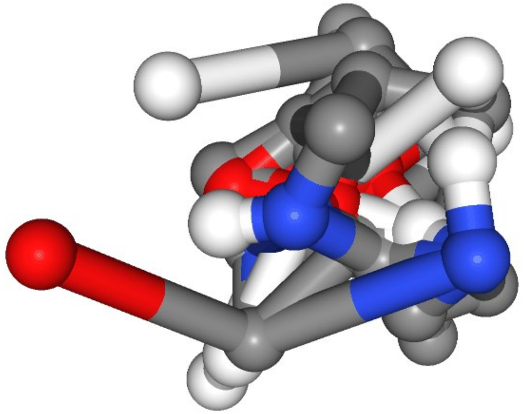
Examples











Medical Knowledge Graph Construction (Ongoing)

- >7M Entities, ~300M facts
  - Disease
  - Drug
  - Phenotype
  - Gene
  - Protein
  - Side effect
- Biomedical literature

PubMed

DrugBank

ctd

Comparative Toxicogenomics Database

STITCH

SIDER
Side Effect Resource

DISEASE ONTOLOGY

OMIM

GENEONTOLOGY
Unifying Biology

e!Ensembl

BRENDA
The Comprehensive Enzyme Information System

Drug Repurposing with Biomedical Knowledge Graphs (Ongoing)

- Drug repurposing: identifying effective drugs for a disease from existing approved list
- Predicting the links between diseases and drugs on biomedical knowledge graphs

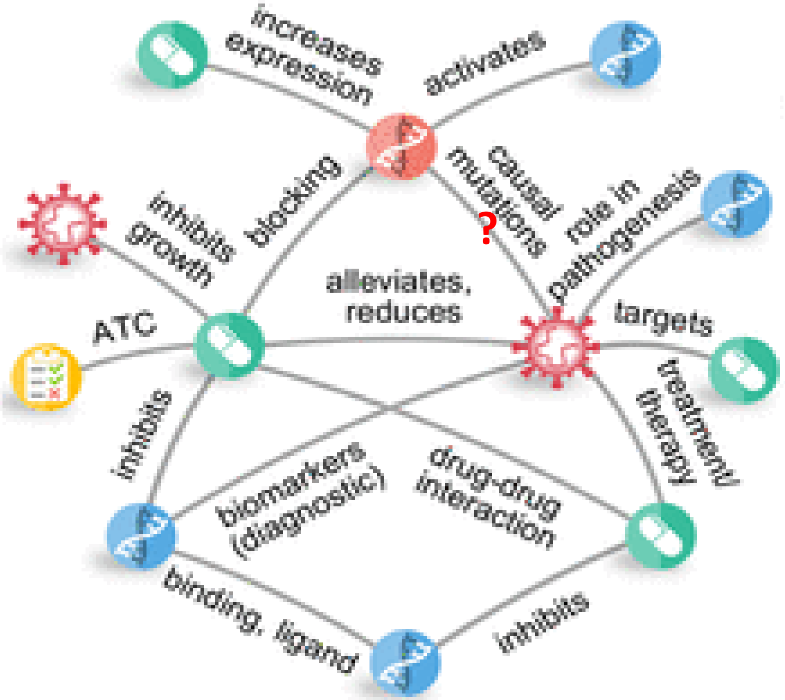


Figure borrowed from Zeng et al. 2020

Summary

- Great potential of AI to drug discovery
  - Extracting evidence from a huge amount of biomedical data
- Many data in this domain are graph-structured
  - Molecules, biomedical knowledge graphs
- Great representation learning for drug discovery
  - Molecule properties prediction
  - De novo molecule design and optimization
  - Retrosynthesis prediction
  - Drug repurposing

Future Directions

- Going beyond from 2D graphs to 3D structures

- Drug Discovery with Limited Labeled Data
  - Active Learning
  - Self-supervised Learning
  - Multi-task/Transfer Learning
  - Few-shot Learning

AAAI’21 Tutorial on
Artificial Intelligence for Drug Discovery

- **Date**: 8:30 am – 11:45 am, Feb. 03, 2021
- **Speakers**

Jian Tang
Mila-Quebec AI Institute

Fei Wang
Weill Cornell Medicine

Feixiong Cheng
Cleveland Clinic

Thanks!

- **Current Students**
  - Meng Qu
  - Zhaocheng Zhu
  - Andreea Deac
  - Louis-Pascal Xhonneux
  - Shengchao Liu
  - Chence Shi
  - Minkai Xu
- **Collaborators and previous students**:, Yoshua Bengio, Jian Peng, Fei Wang, Feixiong Cheng, Ming Zhang, Fanyun Sun, Hongyu Guo, Jordan Hoffmann, Vikas Verma,….

NSERC CRSNG

CIFAR

Microsoft Research

amazon

IVADO
INSTITUT DE VALORISATION DES DONNÉES

Canada
NRC·CNRC

Tencent AI Lab

SAMSUNG

References

- Fanyun Sun, Jordan Hoffman, Vikas Verma and Jian Tang. **InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization**. ICLR’20.
- Chence Shi*, Minkai Xu*, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. **GraphAF: a Flow-based Autoregressive Model for Molecular Graph Generation.** ICLR’20.
- Chence Shi, Minkai Xu, Hongyu Guo, Ming Zhang and Jian Tang. **A Graph to Graphs Framework for Retrosynthesis Prediction**. ICML, 2020.
- Minkai Xu*, Shitong Luo*, Yoshua Bengio, Jian Peng, Jian Tang. **Learning Neural Generative Dynamics for Molecular Conformation Generation**. In Submission.
- Thomas Gaudelet, Ben Day, Arian R Jamasb, Jyothish Soman, Cristian Regep, Gertrude Liu, Jeremy BR Hayter, Richard Vickers, Charles Roberts, Jian Tang, David Roblin, Tom L Blundell, Michael M Bronstein, Jake P Taylor-King**. Utilising Graph Machine Learning within Drug Discovery and Development.** arXiv:2012.05716
- Yadi Zhou, Fei Wang, Jian Tang, Ruth Nussinov, Feixiong Cheng. **Artificial intelligence in COVID-19 drug repurposing**. The Lancet Digital Health.

GraphAF: an Autoregressive Flow for Molecular Graph Generation

- G=(A, X), where A is the adjacency matrix, X is the atom type
- Dequantize a discrete graph G into continuous data

$$z_i^X = X_i + u,\ u \sim U[0,1)^d;\ z_{ij}^A \quad = A_{ij} + u,\ u \sim U[0,1)^{b+1}$$

- Define the conditional distributions as:

Node generation: $p(z_i^X | G_i) = \mathcal{N}(\mu_i^X, (\alpha_i^X)^2),$

$$\text{where } \mu_i^X = g_{\mu^X}(G_i), \alpha_i^X = g_{\alpha^X}(G_i),$$

Edge generation: $p(z_{ij}^A | G_i, X_i, A_{i,1:j-1}) = \mathcal{N}(\mu_{ij}^A, (\alpha_{ij}^A)^2),\ j \in \{1, 2, \ldots, i-1\},$

$$\text{where } \mu_{ij}^A = g_{\mu^A}(G_i, X_i, A_{i,1:j-1}), \alpha_{ij}^A = g_{\alpha^A}(G_i, X_i, A_{i,1:j-1})$$

$\boldsymbol{G_i}$: current graph substructure, encoded with graph neural networks