

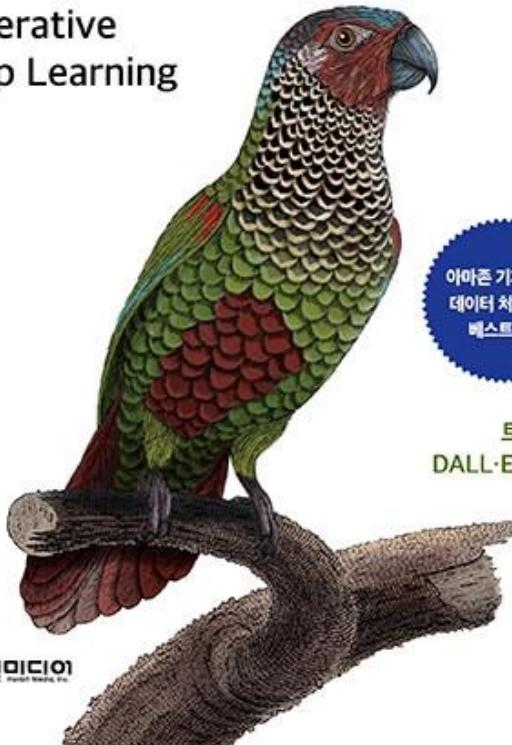
•생성 모델을 이용한 신약개발

O'REILLY®

2nd edition

만들면서 배우는 생성 AI [2판]

Generative
Deep Learning



트랜스포머부터 GPT,
DALL·E 2, 스테이블 디퓨전,
플라밍고까지

데이비드 포스터 지음
박해선 옮김

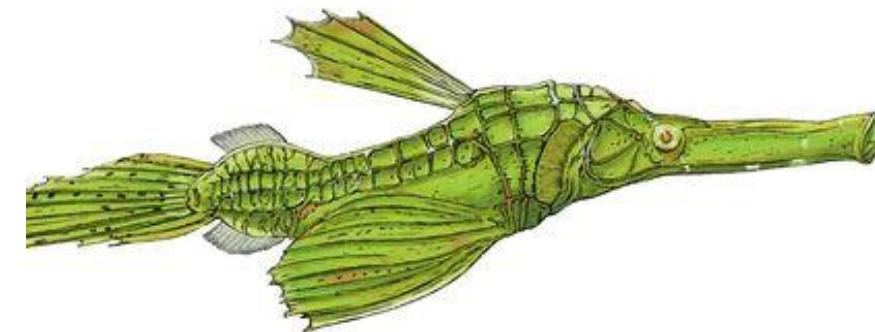
한빛미디어
Hanbit Media, Inc.

O'REILLY®

밑바닥부터 시작하는 딥러닝 5

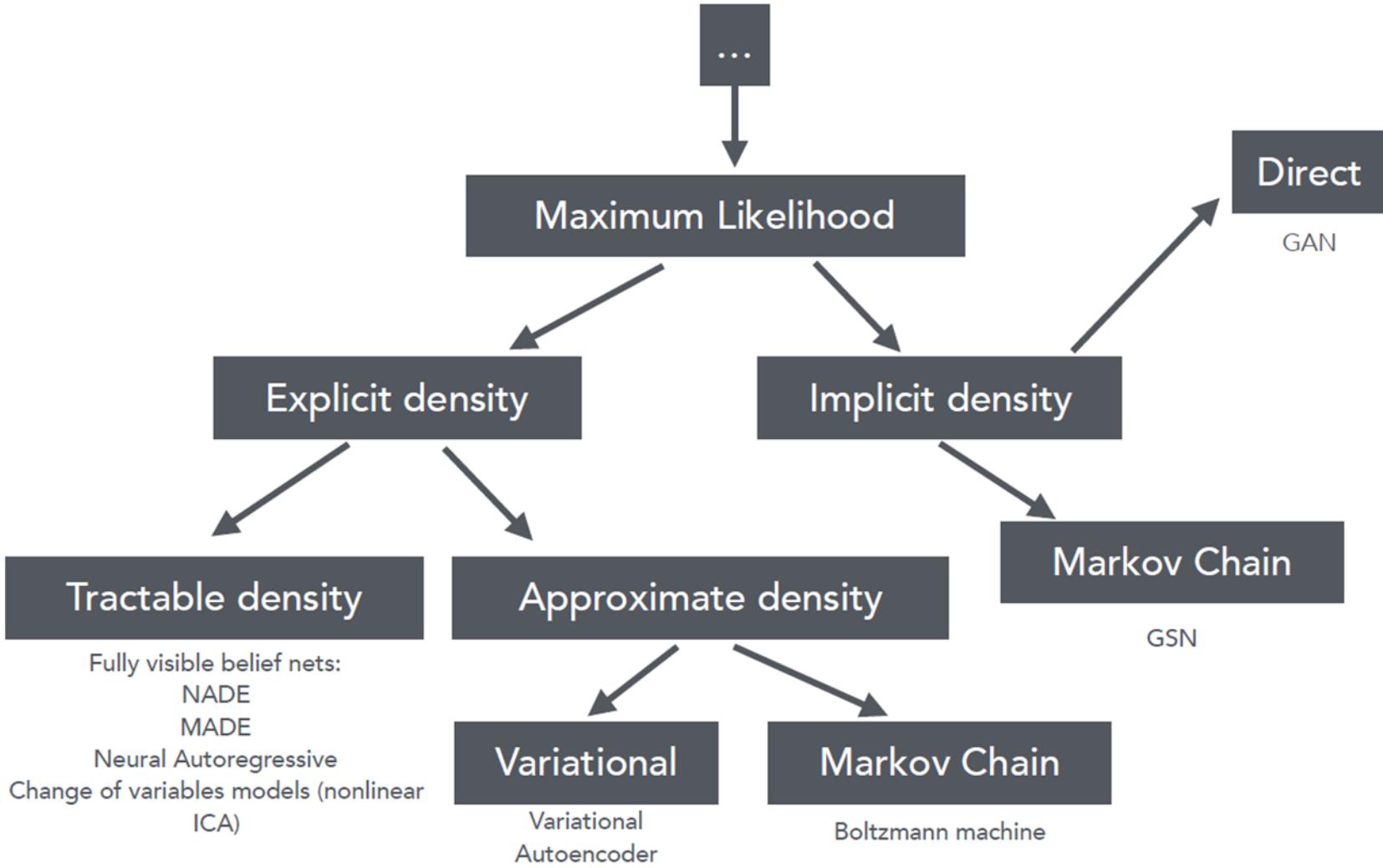
Deep Learning from Scratch 5

10단계로 익히는
이미지 생성 모델의 원리



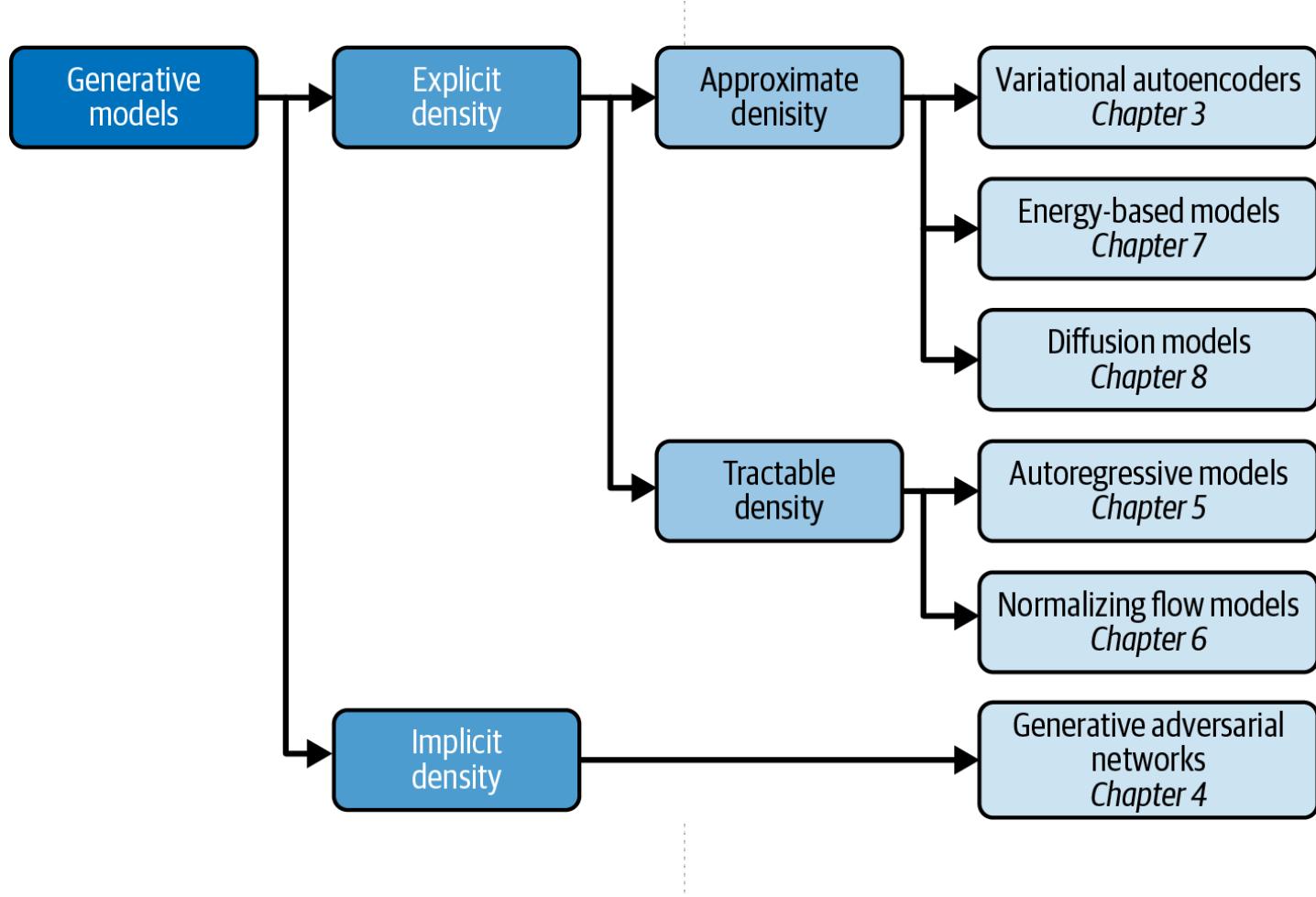
한빛미디어
Hanbit Media, Inc.

사이로 고카 지음
개일웹시 옮김

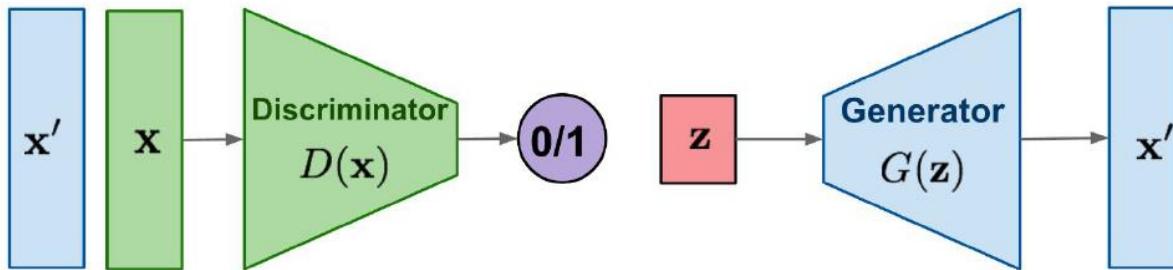


Source: Tutorial on Generative Adversarial Networks, 2017

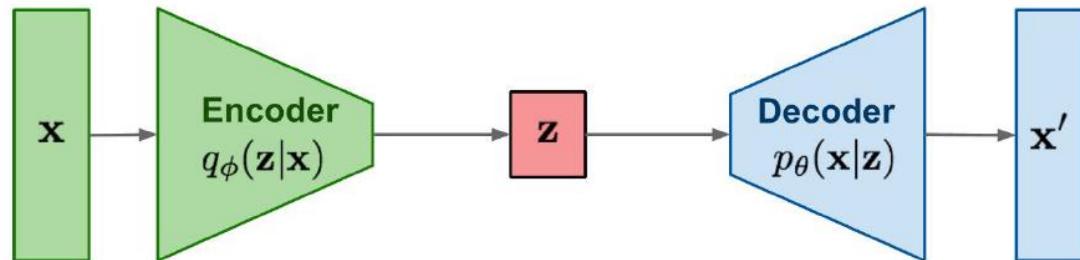
Generative Models



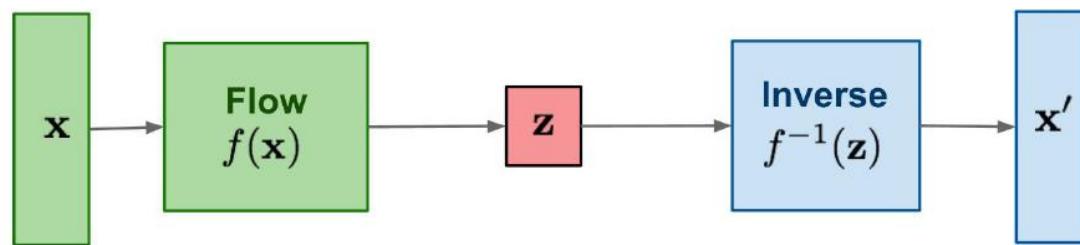
GAN: Adversarial training



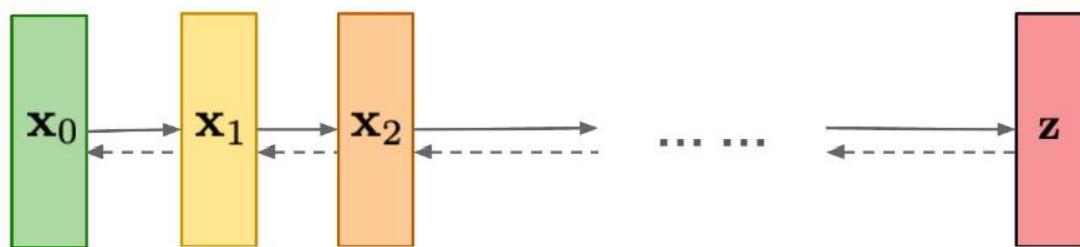
VAE: maximize variational lower bound



Flow-based models:
Invertible transform of distributions



Diffusion models:
Gradually add Gaussian noise and then reverse

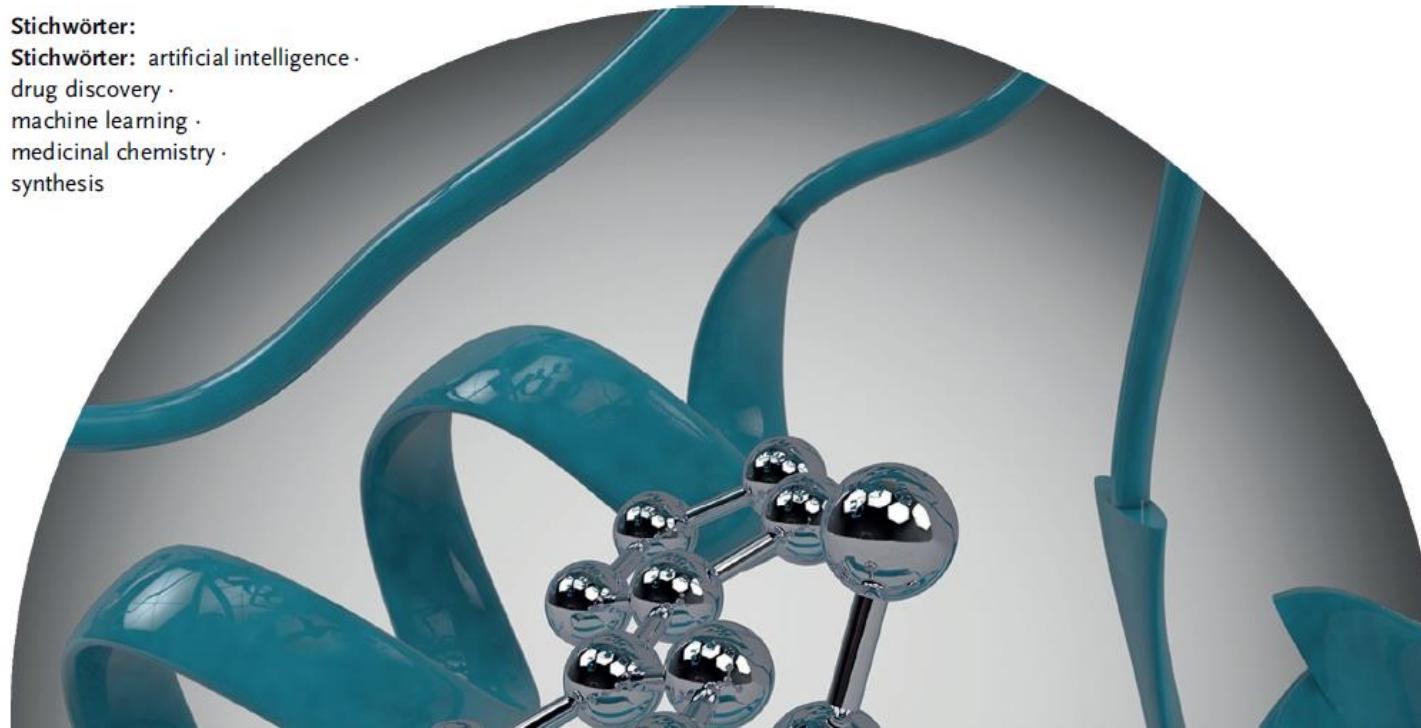


Automated De Novo Drug Design: Are We Nearly There Yet?

Gisbert Schneider und David E. Clark*

Stichwörter:

Stichwörter: artificial intelligence ·
drug discovery ·
machine learning ·
medicinal chemistry ·
synthesis



- Generation of new chemical structures
 - Variational Autoencoder (VAE)
 - Generative Adversarial Network (GAN)
 - Flow-based models
 - Diffusion-based models
- Optimization
 - Iterative optimization
 - Bayesian optimization
 - Reinforcement Learning

Molecular design in drug discovery: a comprehensive review of deep generative models

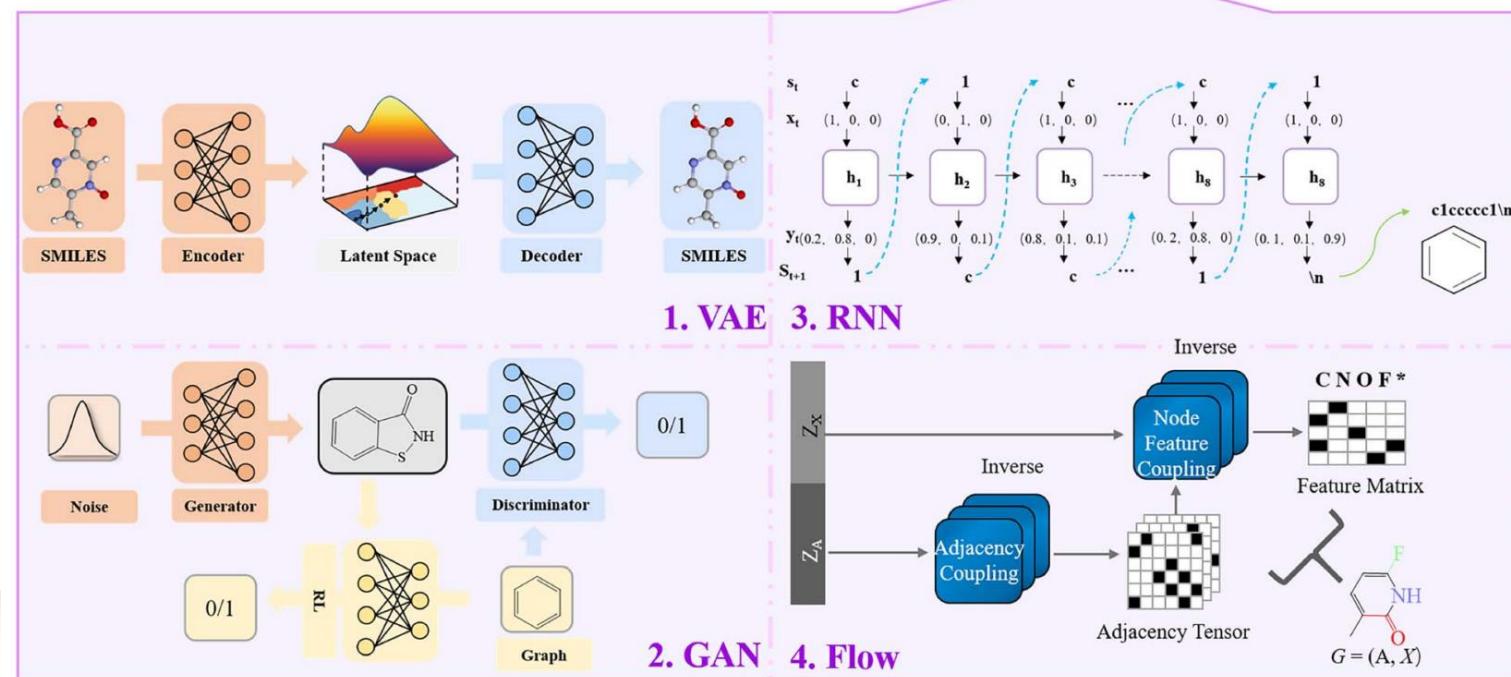
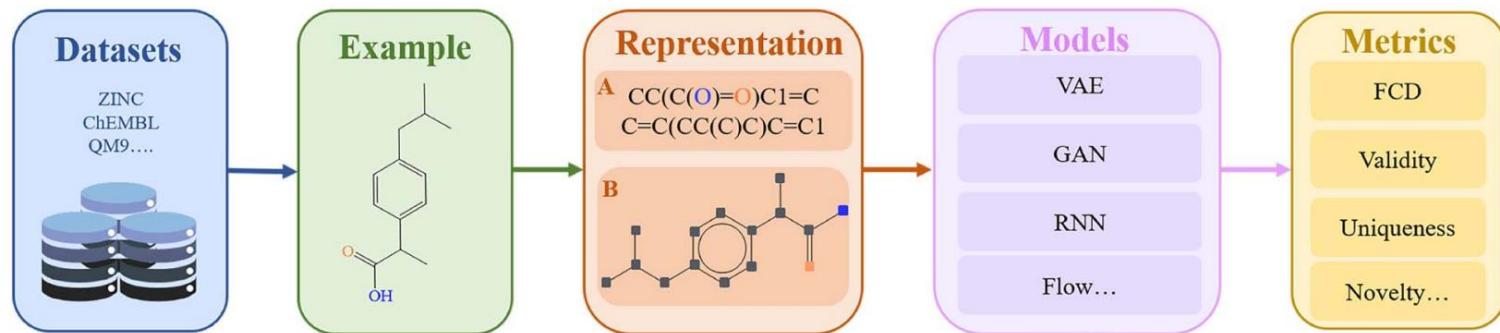
Yu Cheng, Yongshun Gong, Yuansheng Liu, Bosheng Song and Quan Zou

Corresponding authors: Yuansheng Liu, College of Information Science and Engineering, Hunan University, 2 Lushan S Rd, Yuelu District, 410086 Changsha, China. E-mail: yuanshengliu@hnu.edu.cn; Bosheng Song. E-mail: boshengsong@hnu.edu.cn

Abstract

Deep generative models have been an upsurge in the deep learning community since they were proposed. These models are designed for generating new synthetic data including images, videos and texts by fitting the data approximate distributions. In the last few years, deep generative models have shown superior performance in drug discovery especially *de novo* molecular design. In this study, deep generative models are reviewed to witness the recent advances of *de novo* molecular design for drug discovery. In addition, we divide those models into two categories based on molecular representations in silico. Then these two classical types of models are reported in detail and discussed about both pros and cons. We also indicate the current challenges in deep generative models for *de novo* molecular design. *De novo* molecular design automatically is promising but a long road to be explored.

Key words: deep generative model; deep learning; *de novo* drug design; molecular design



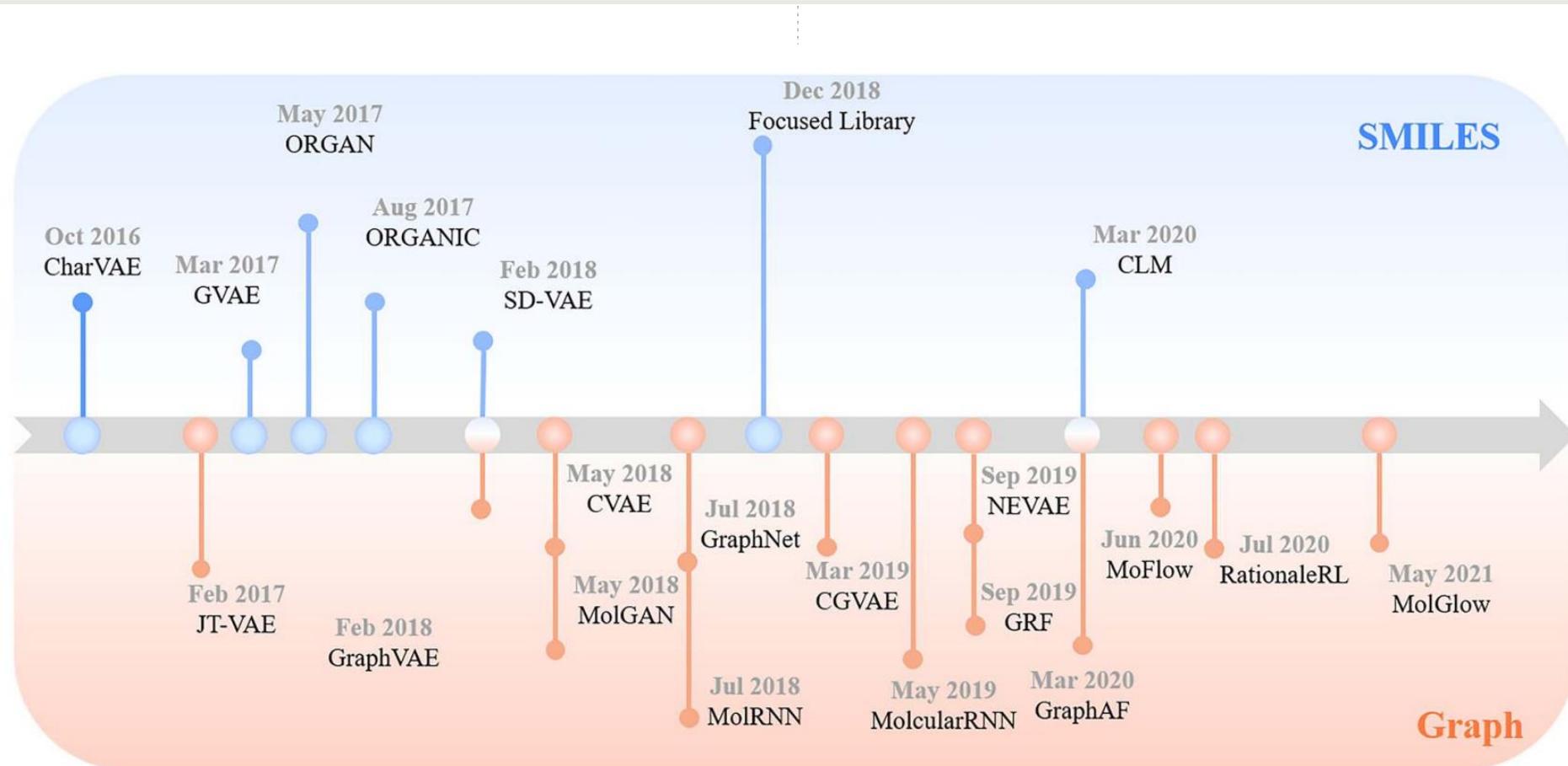


Figure 3. The timeline of deep generative models for molecular design.



FOUNDATION (PURPLE)

Unleashing the power of generative AI in drug discovery

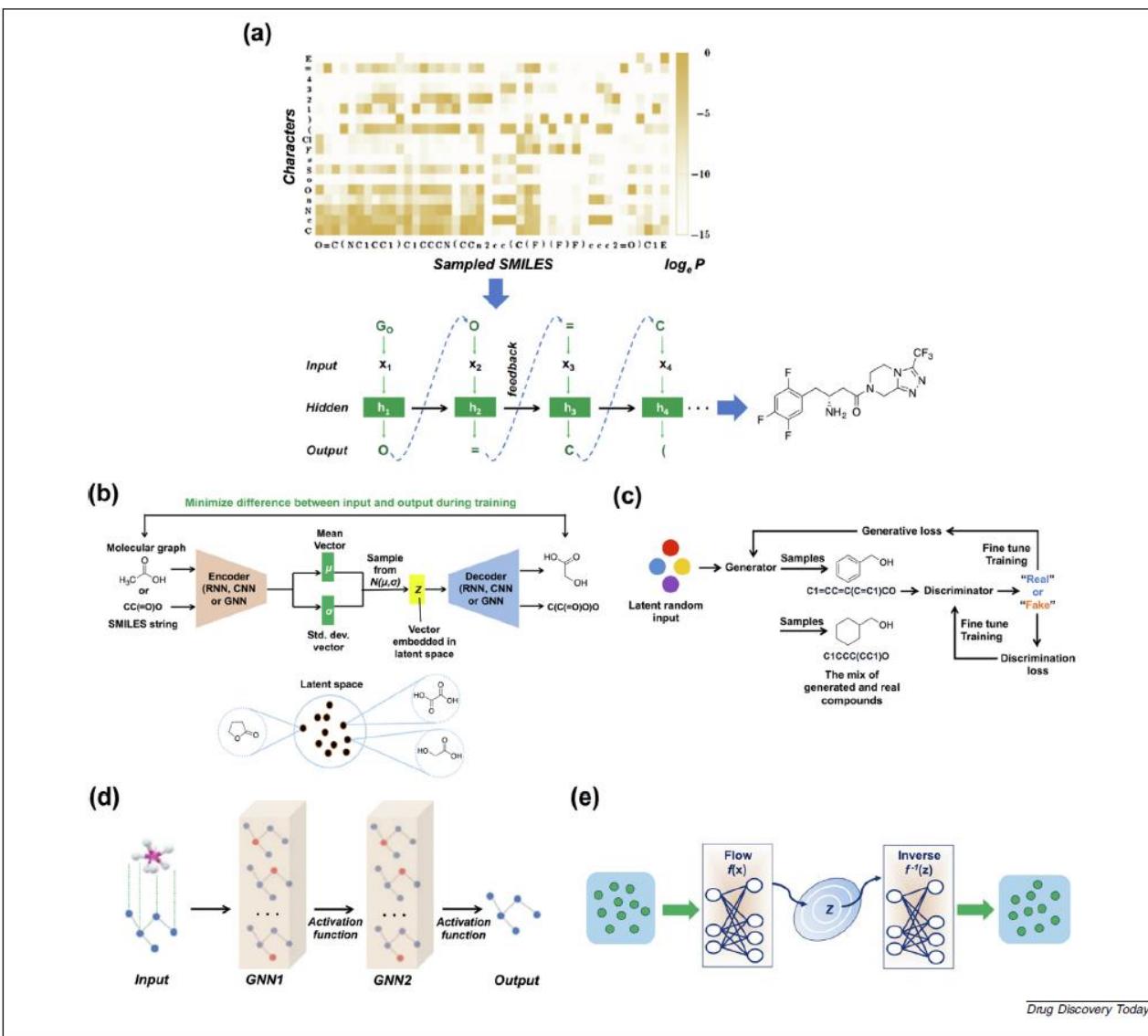
Amit Gangwal¹, Antonio Lavecchia^{2,*}

¹Department of Natural Product Chemistry, Shri Vile Parle Kelavani Mandal's Institute of Pharmacy, Dhule 424001, Maharashtra, India

²"Drug Discovery" Laboratory, Department of Pharmacy, University of Naples Federico II, I-80131 Naples, Italy



Amit Gangwal served as a professor and as principal at Shri Vithal Education & Research Institute's College of Pharmacy, Pandharpur Maharashtra. At present, he is an associate professor at the prestigious Shri Vile Parle Kelavani Mandal's Institute of Pharmacy, Dhule. He is an academic-researcher with >18 years of teaching and



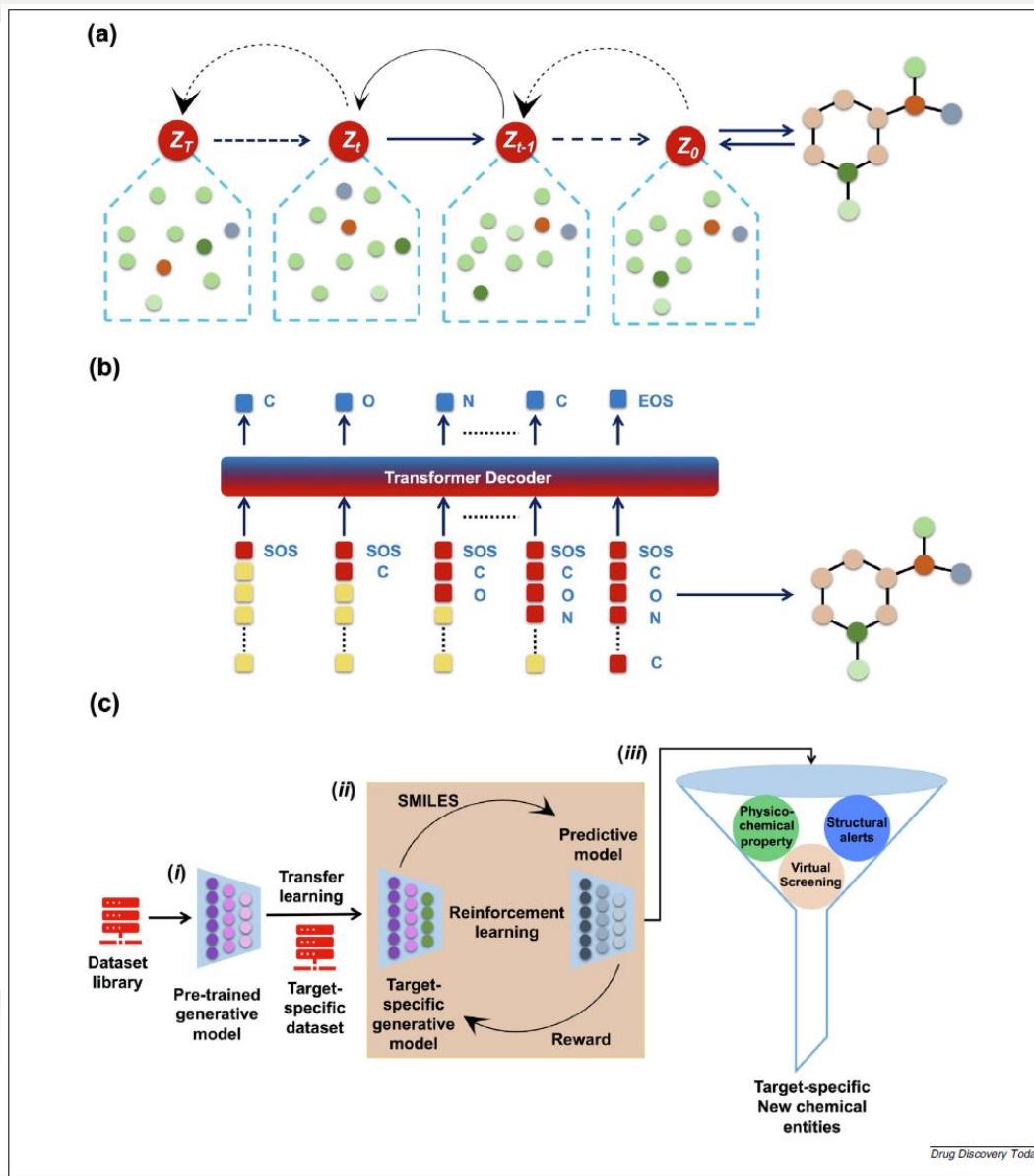
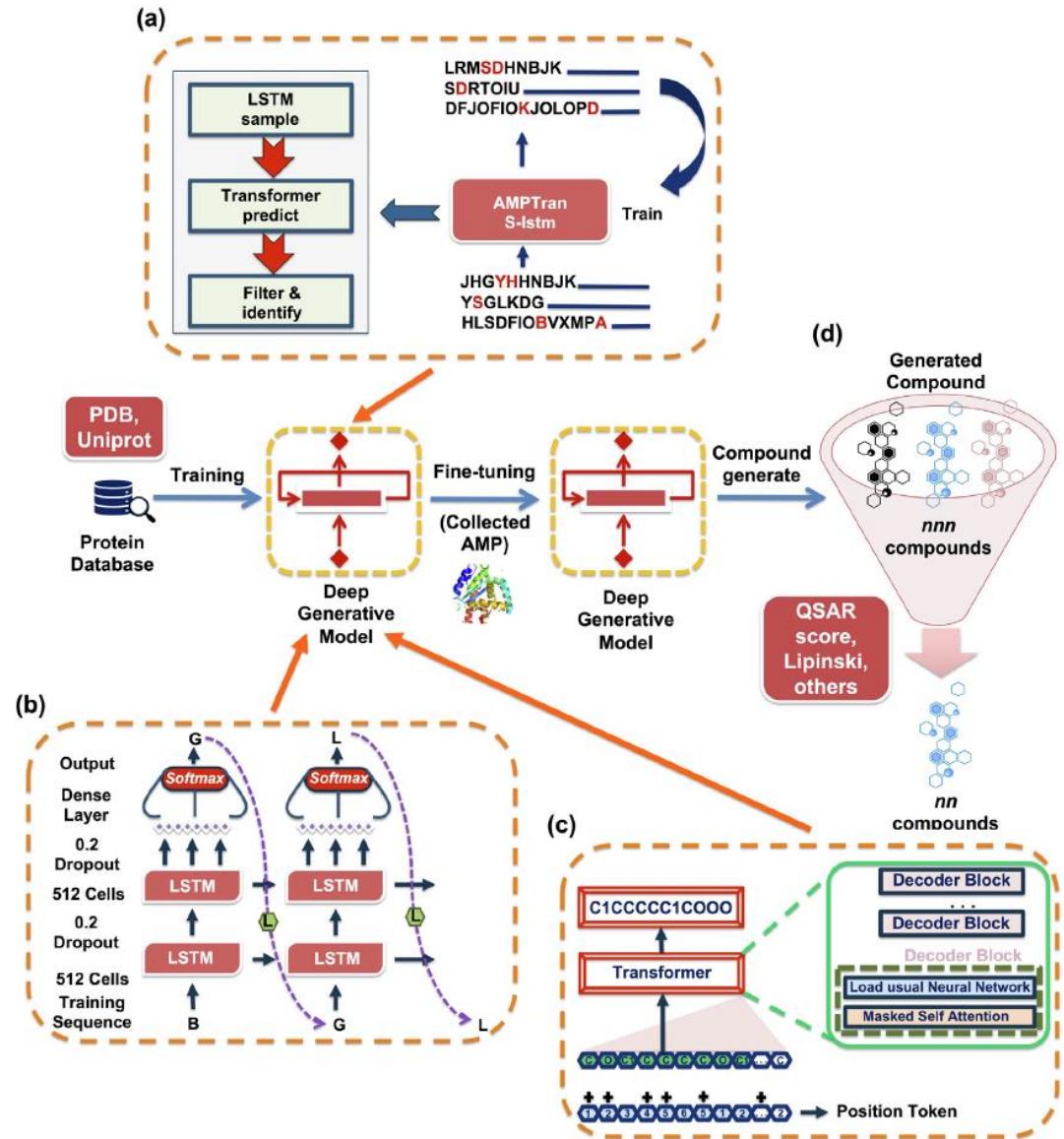


TABLE 1

Comparison of different DGMs in terms of efficacy, accuracy, strengths, weaknesses and applications in drug discovery

Model	Efficacy	Accuracy	Strengths	Weaknesses	Drug discovery applications
RNNs	Medium	Medium	Sequential data modeling, temporal dependencies	Gradient vanishing, limited context	Drug–target interaction prediction, sequence generation
VAEs	Medium to high	High	Latent space representation, generative modeling	Mode collapse, blurry images	Molecular generation, drug design optimization
CVAE	Medium to high	High	Conditional generation, improved VAEs	Mode collapse, computational complexity	Conditional molecular generation, property prediction
AAE	Medium to High	High	Adversarial training, disentangled representations	Mode collapse, training instability	Data augmentation, molecule generation
GANs	High	High	High-quality image generation, diversity	Mode collapse, training instability	Molecular design, property prediction, generative modeling
GNNs	Medium to high	High	Graph structure modeling, node embeddings	Overfitting, limited scalability	Drug–protein interaction prediction, molecular graph analysis
Normalizing flows	High	High	Exact likelihood, invertibility	Computational cost, complex architectures	Molecular generation, density estimation
Diffusion models	High	High	Modeling complex data distributions, generative modeling	Computational cost, training complexity	Molecular generation, generative modeling
Transformers	High	High	Attention mechanism, parallel processing	Training data size dependency, interpretability	Molecular sequence generation, property prediction
TL	Medium to high	Medium to high	Knowledge transfer, leveraging pre-trained models	Task-specific fine-tuning, domain mismatch	Molecular property prediction, model generalization
RL	Medium to high	Medium to high	Sequential decision making, exploration–exploitation	Training instability, sample inefficiency	Molecular design optimization, drug discovery process



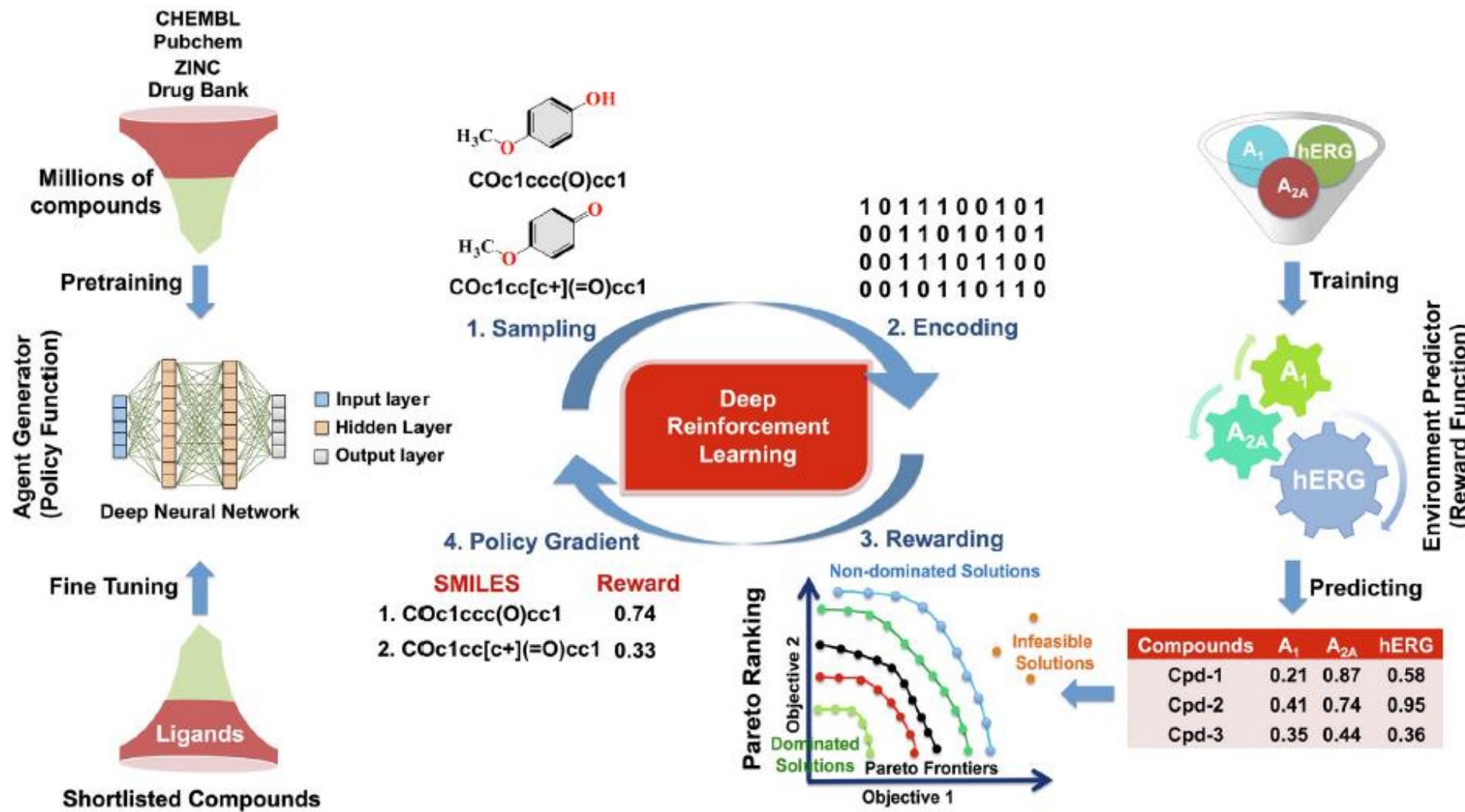


TABLE 3**Some common parameters used to assess DGMs**

Type	Assessment parameters
Molecule set	Validity Novelty Uniqueness Controllability Nearest neighbor similarity Scaffold similarity Internal diversity Fragment similarity Fréchet ChemNet Distance ^(p146) Completeness, uniformity, closedness ^(p147)
Molecules	Physicochemical property Synthetic accessibility score (SA score) Natural product likeness score QED Jointly score ^(p148)
Integrated benchmark	GuacaMol ^(p33) MOSES ^(p149)

TABLE 2

Experimental endorsement of *de novo*-generated molecules

Receptor	Method type	Activity confirmation			Reference
		<i>In silico</i>	<i>In vitro</i>	<i>In vivo</i>	
PPAR, RXR	TL	SPiDER	Out of five synthesized molecules, four were found to be active, with two molecules inhibiting both PPAR and RXR.	—	(p140)
RXR	TL	SPiDER WHALES	Reported: four synthesized; two active.	—	(p106)
JK3 selective	RL	Docking	One synthesized molecule was found to be selectively active for JK3.	—	(p45)
Inhibitors of kinases	RL		Out of five purchased molecules, seven were reported to be active.	—	(p141)
DRD2, 5-HT1A, 5-HT2A	TL	MT-DNN on ECFP4	One molecule, synthesized along with six analogs, was found to be active for three targets.	One molecule was tested and found to be active and safe.	(p142)
VEGFR-2	Train on actives	Docking	Out of five synthesized molecules, three were reported to be active and noncytotoxic.	—	(p143)
DDR1	RL	SOM Pharmacophore	Only two actives were reported out of a total of six synthesized.	One molecule was tested, and its half-life was found to be 3.5 h.	(p21)
p300/CBP inhibitors	TL	Docking	One molecule, synthesized along with 26 analogs, was found to be active and selective.	Good bioavailability, efficacy, and safety	(p144)
LXR agonists	TL		Besides synthesizing 25 compounds, three were purchased. A total of 12 compounds showed a positive response.	—	(p145)



Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models

Daniil Polykovskiy^{1}, Alexander Zhebrak¹, Benjamin Sanchez-Lengeling², Sergey Golovanov³, Oktai Tatanov³, Stanislav Belyaev³, Rauf Kurbanov³, Aleksey Artamonov³, Vladimir Aladinskiy¹, Mark Veselov¹, Artur Kadurin¹, Simon Johansson⁴, Hongming Chen⁴, Sergey Nikolenko^{1,3,5*}, Alán Aspuru-Guzik^{6,7,8,9*} and Alex Zhavoronkov^{1*}*

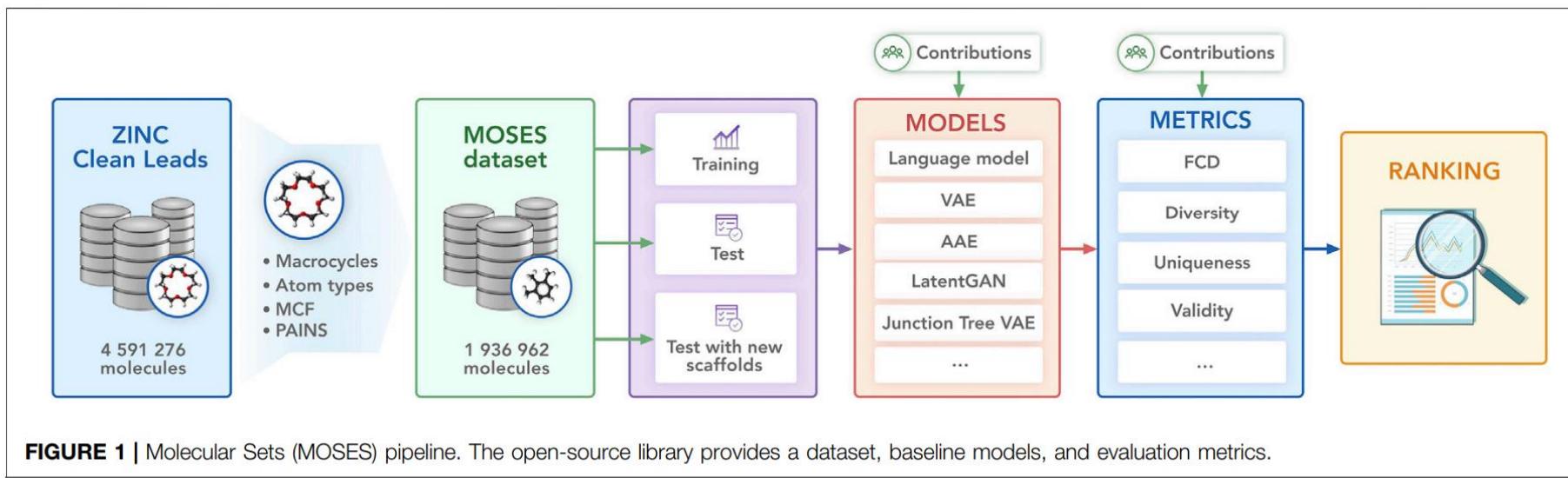
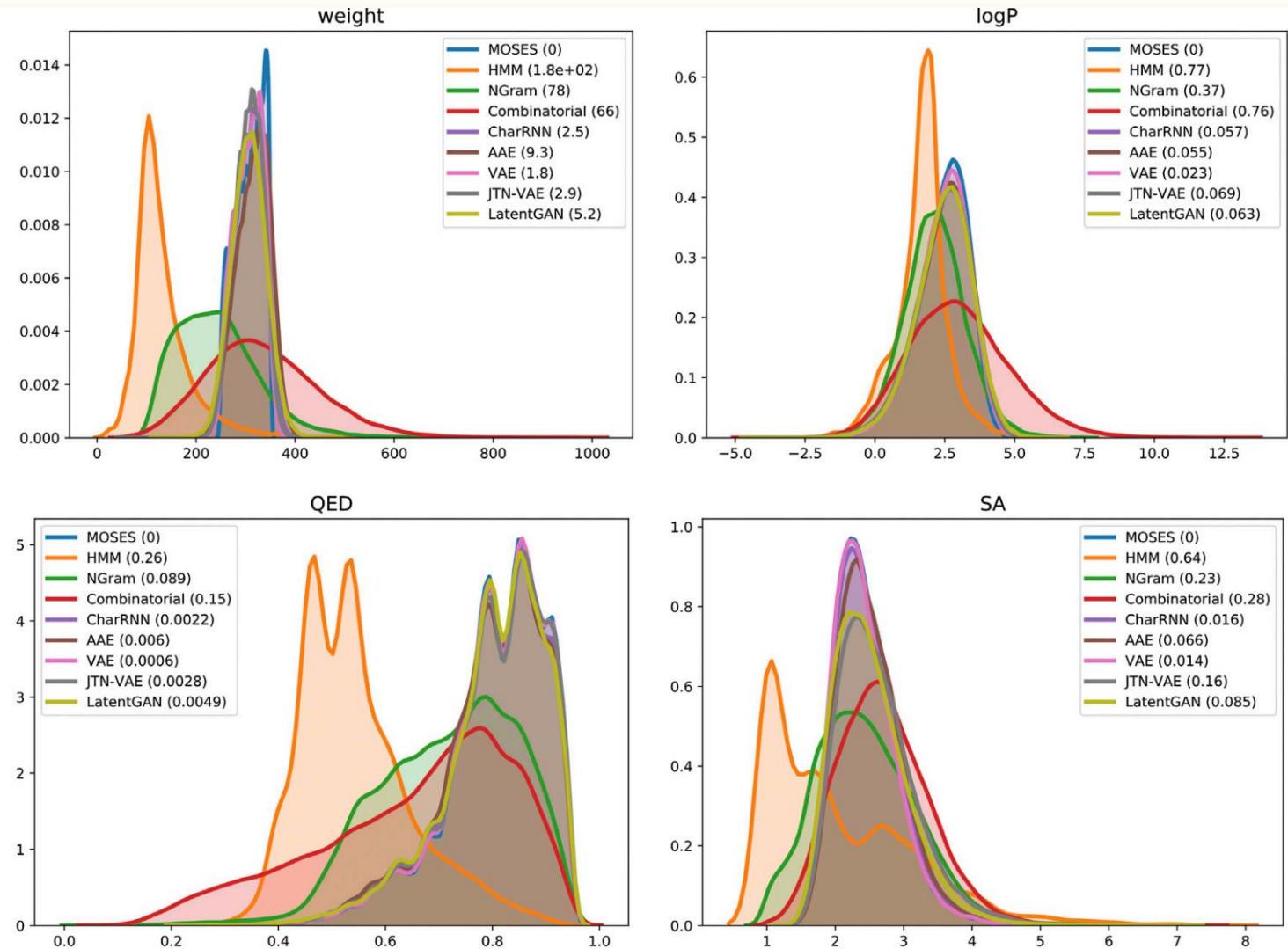


FIGURE 1 | Molecular Sets (MOSES) pipeline. The open-source library provides a dataset, baseline models, and evaluation metrics.

TABLE 1 | Performance metrics for baseline models: fraction of valid molecules, fraction of unique molecules from and molecules.

Model	Valid (↑)	Unique@1k (↑)	Unique@10k (↑)
<i>Train</i>	1.0	1.0	1.0
HMM	0.076 ± 0.0322	0.623 ± 0.1224	0.5671 ± 0.1424
NGram	0.2376 ± 0.0025	0.974 ± 0.0108	0.9217 ± 0.0019
Combinatorial	1.0 ± 0.0	0.9983 ± 0.0015	0.9909 ± 0.0009
CharRNN	0.975 ± 0.026	1.0 ± 0.0	0.999 ± 0.0
VAE	0.977 ± 0.001	1.0 ± 0.0	0.998 ± 0.001
AAE	0.937 ± 0.034	1.0 ± 0.0	0.997 ± 0.002
JTN-VAE	1.0 ± 0.0	1.0 ± 0.0	0.9996 ± 0.0003
LatentGAN	0.897 ± 0.002	1.0 ± 0.0	0.997 ± 0.005



OptiMol: Optimization of Binding Affinities in Chemical Space for Drug Discovery

Jacques Boitreauaud,^{II} Vincent Mallet,^{II} Carlos Oliver, and Jérôme Waldspühl*



Cite This: *J. Chem. Inf. Model.* 2020, 60, 5658–5666



Read Online

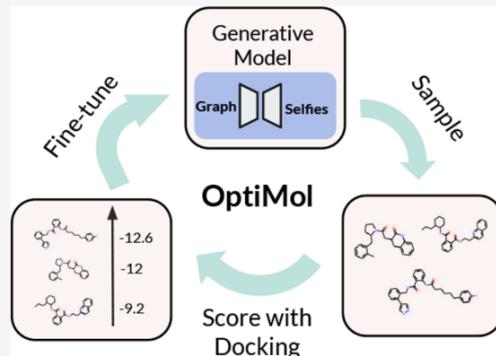
ACCESS |

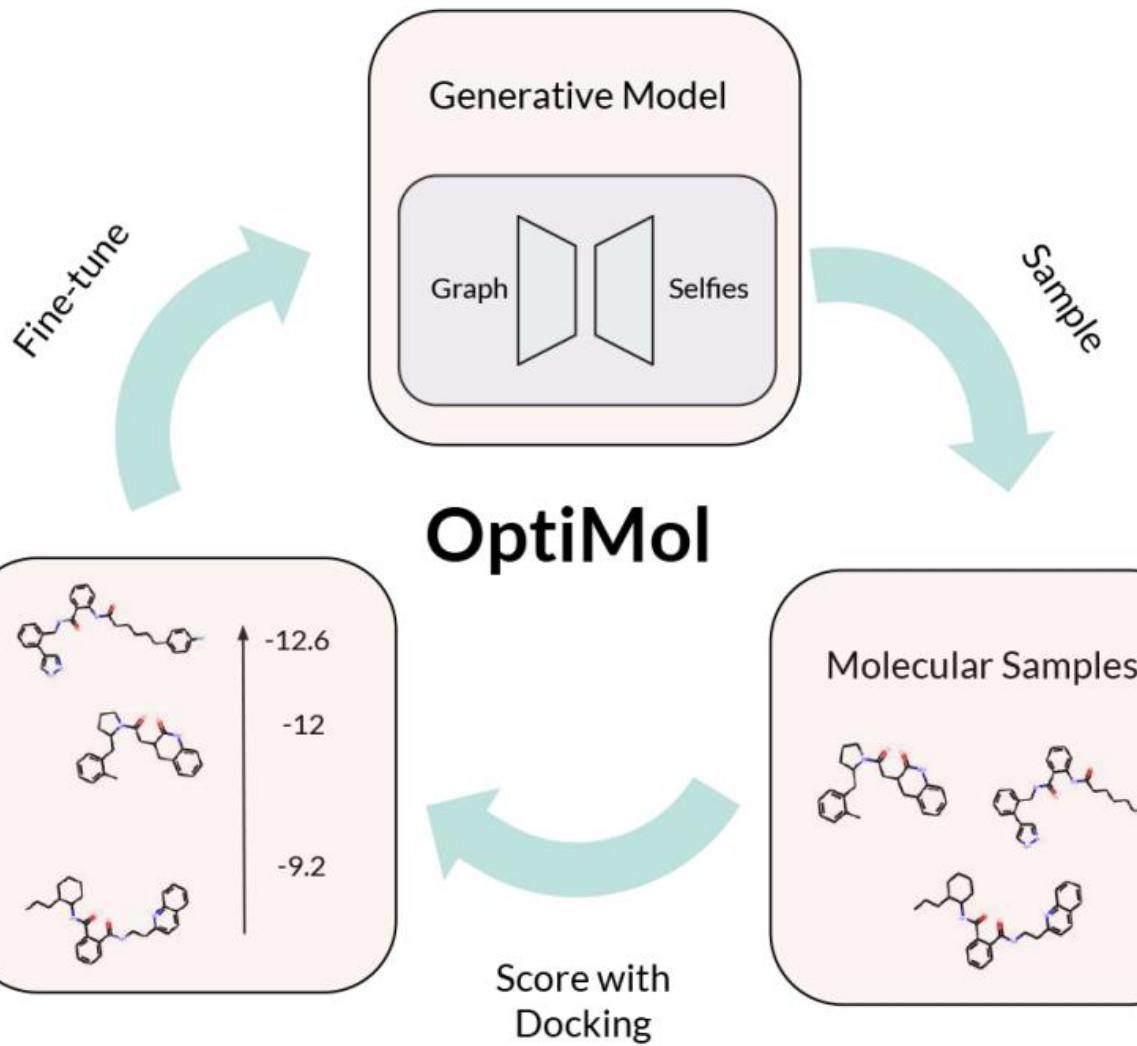
Metrics & More

Article Recommendations

Supporting Information

ABSTRACT: Ligand-based drug design has recently benefited from the development of deep generative models. These models enable extensive explorations of the chemical space and provide a platform for molecular optimization. However, the vast majority of current methods does not leverage the structure of the binding target, which potentiates the binding of small molecules and plays a key role in the interaction. We propose an optimization pipeline that leverages complementary structure-based and ligand-based methods. Instead of performing docking on a fixed chemical library, we iteratively select promising compounds in the full chemical space using a ligand-centered generative model. Molecular docking is then used as an oracle to guide compound optimization. This allows for iterative generation of compounds that fit the target structure better and better, without prior knowledge about bioactives. For this purpose, we introduce a new graph to Selfies Variational Autoencoder (VAE) which benefits from an 18-fold faster decoding than the graph to graph state of the art, while achieving a similar performance. We then successfully optimize the generation of molecules toward high docking scores, enabling a 10-fold enrichment of high-scoring compounds found with a fixed computational cost.





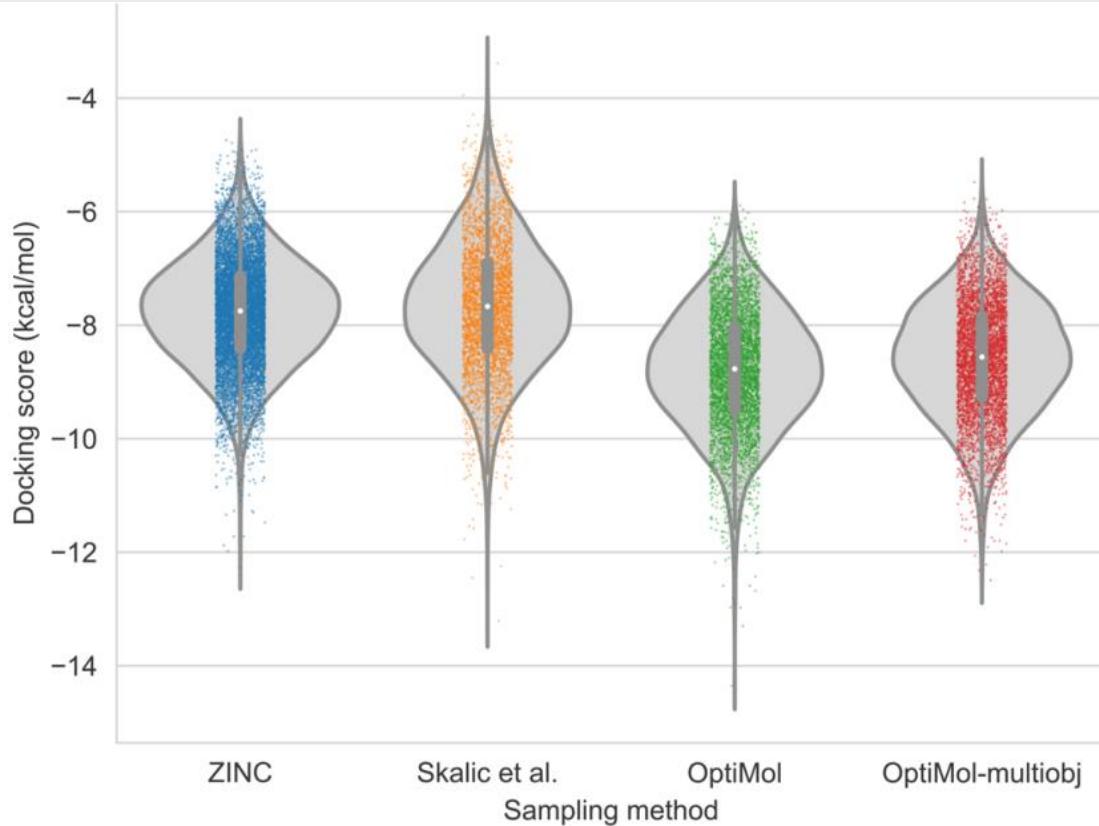


Figure 6. Distributions of docking scores on DRD3 for 2k random ZINC compounds, samples obtained with Skalic et al., OptiMol and OptiMol-multiobjective generative models.

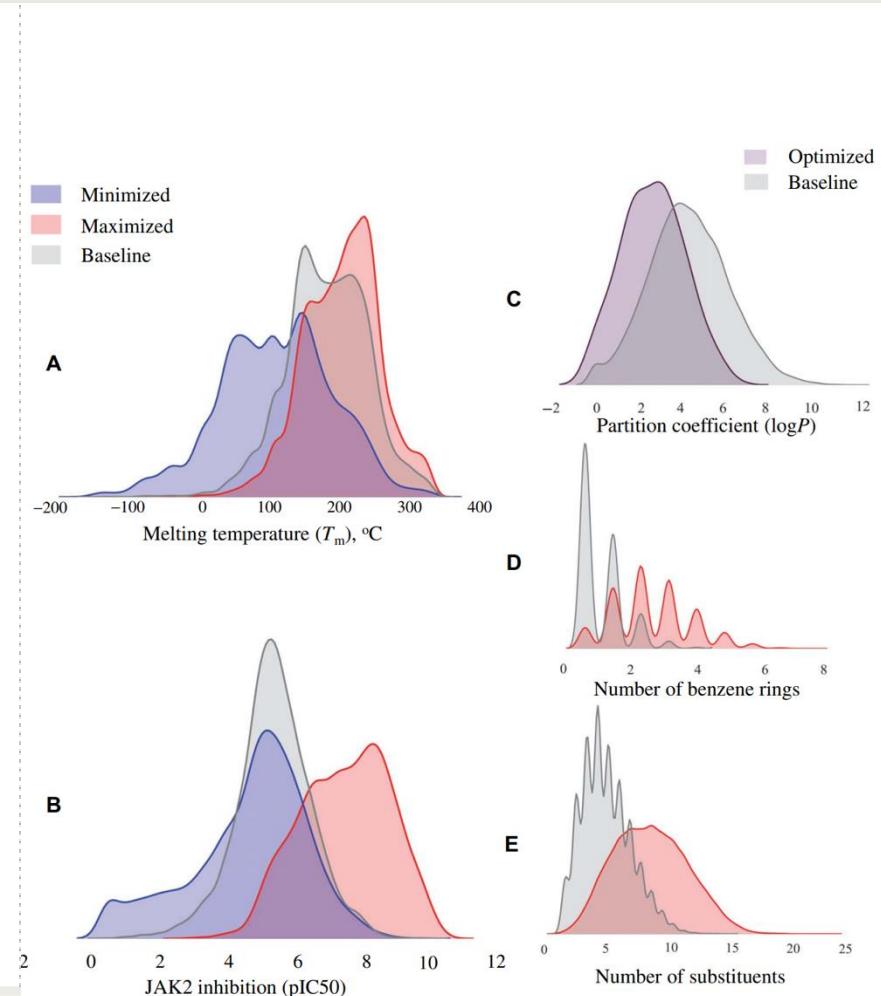
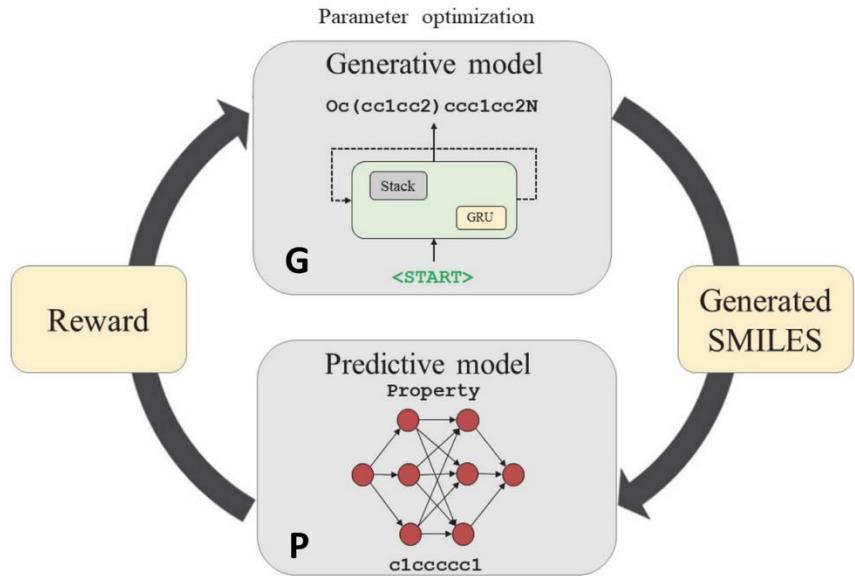
COMPUTATIONAL BIOLOGY

Deep reinforcement learning for de novo drug design

Mariya Popova^{1,2,3}, Olexandr Isayev^{1*}, Alexander Tropsha^{1*}

We have devised and implemented a novel computational strategy for de novo design of molecules with desired properties termed ReLeaSE (Reinforcement Learning for Structural Evolution). On the basis of deep and reinforcement learning (RL) approaches, ReLeaSE integrates two deep neural networks—generative and predictive—that are trained separately but are used jointly to generate novel targeted chemical libraries. ReLeaSE uses simple representation of molecules by their simplified molecular-input line-entry system (SMILES) strings only. Generative models are trained with a stack-augmented memory network to produce chemically feasible SMILES strings, and predictive models are derived to forecast the desired properties of the de novo-generated compounds. In the first phase of the method, generative and predictive models are trained separately with a supervised learning algorithm. In the second phase, both models are trained jointly with the RL approach to bias the generation of new chemical structures toward those with the desired physical and/or biological properties. In the proof-of-concept study, we have used the ReLeaSE method to design chemical libraries with a bias toward structural complexity or toward compounds with maximal, minimal, or specific range of physical properties, such as melting point or hydrophobicity, or toward compounds with inhibitory activity against Janus protein kinase 2. The approach proposed herein can find a general use for generating targeted chemical libraries of novel compounds optimized for either a single desired property or multiple properties.

M. Popova, “Deep reinforcement learning for de novo drug design”, Science Advanced, 4:eaap7885 (2018).





There are amendments to this paper

OPEN

Optimization of Molecules via Deep Reinforcement Learning

Zhenpeng Zhou^{ID}^{1,3}, Steven Kearnes^{ID}², Li Li², Richard N. Zare¹ & Patrick Riley^{ID}²

Received: 5 March 2019

Accepted: 10 July 2019

Published online: 24 July 2019

We present a framework, which we call Molecule Deep Q-Networks (MolDQN), for molecule optimization by combining domain knowledge of chemistry and state-of-the-art reinforcement learning techniques (double Q-learning and randomized value functions). We directly define modifications on molecules, thereby ensuring 100% chemical validity. Further, we operate without pre-training on any dataset to avoid possible bias from the choice of that set. MolDQN achieves comparable or better performance against several other recently published algorithms for benchmark molecular optimization tasks. However, we also argue that many of these tasks are not representative of real optimization problems in drug discovery. Inspired by problems faced during medicinal chemistry lead optimization, we extend our model with multi-objective reinforcement learning, which maximizes drug-likeness while maintaining similarity to the original molecule. We further show the path through chemical space to achieve optimization for a molecule to understand how the model works.

Action

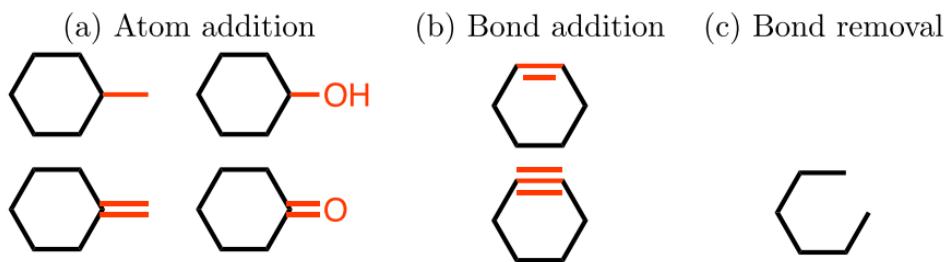
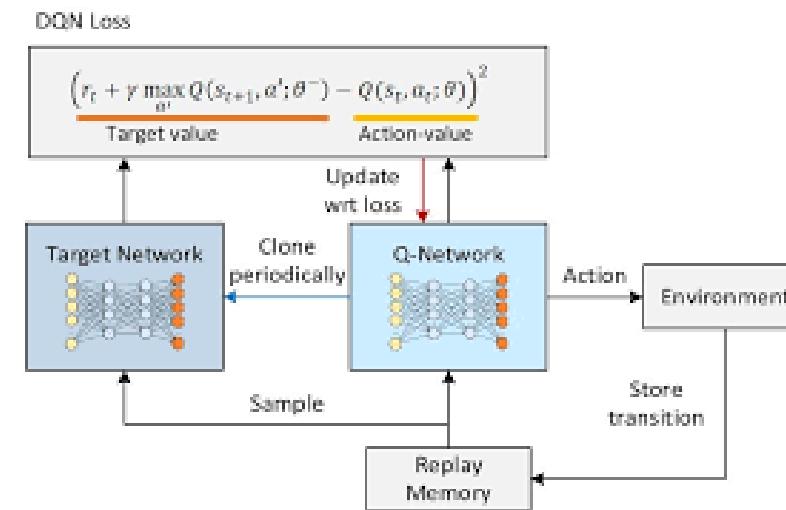
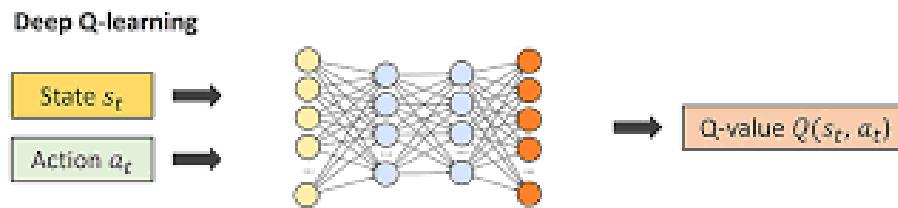
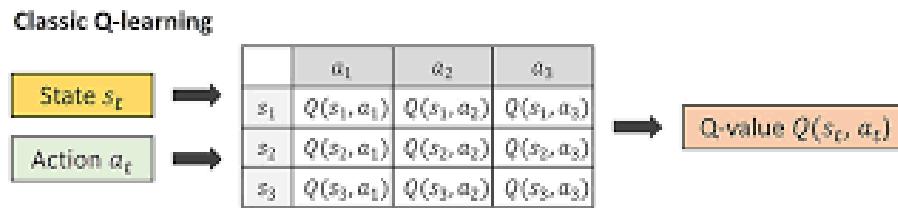
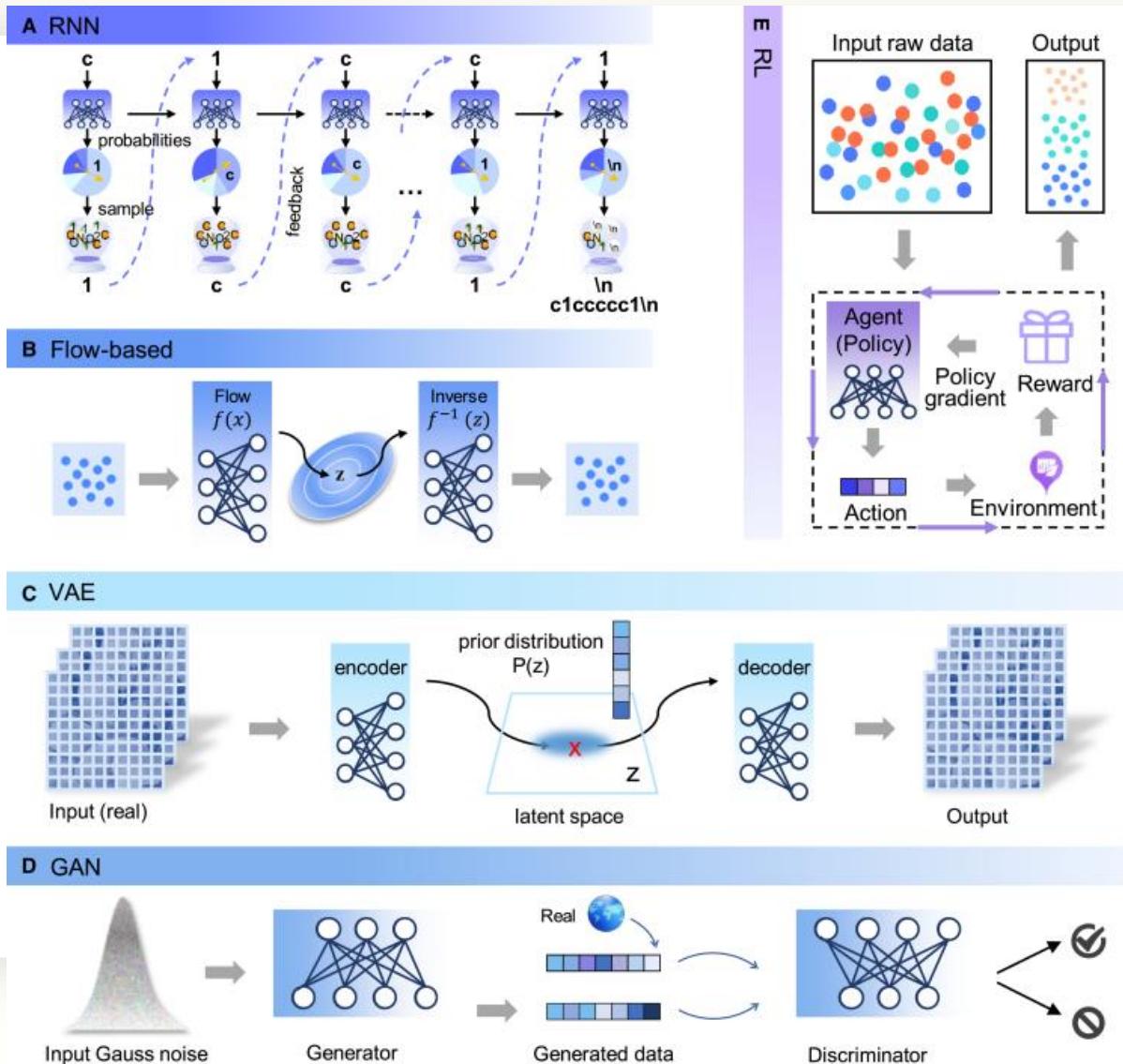
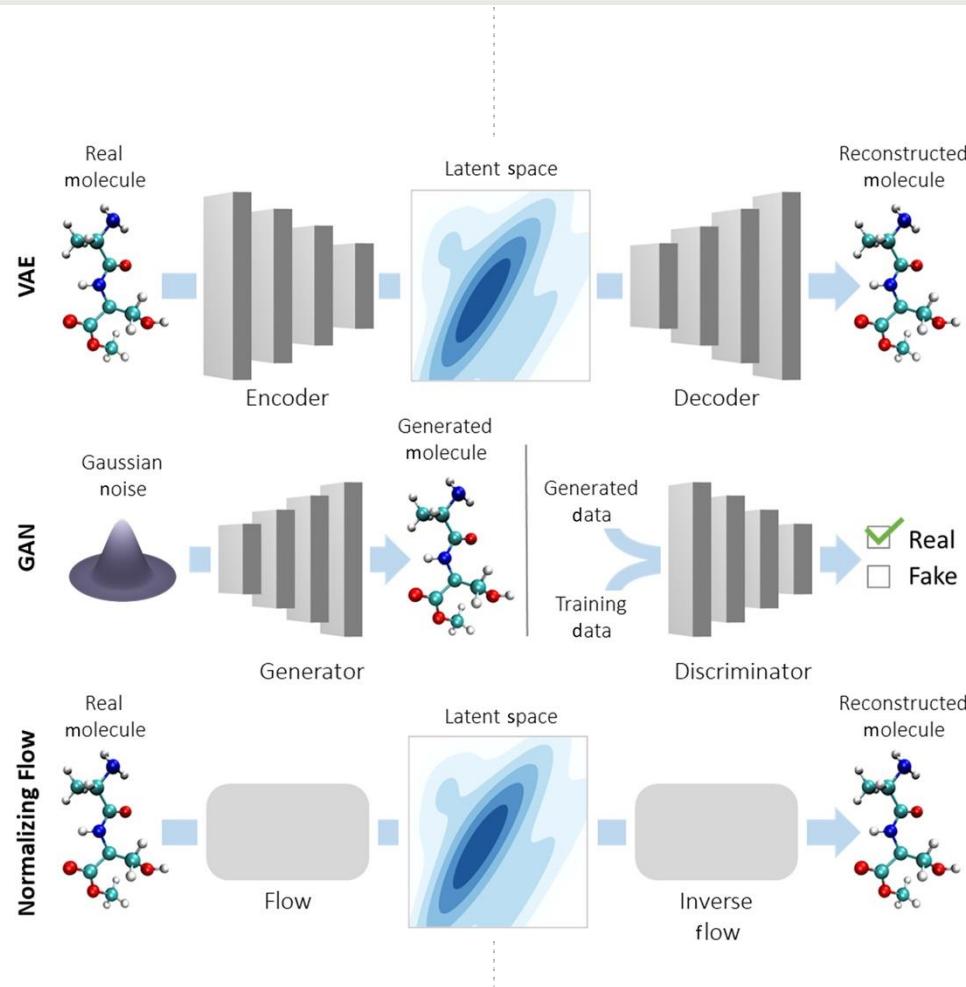


Figure 1. Valid actions on the state of cyclohexane. Modifications are shown in red. Invalid bond additions which violate the heuristics explained in Section 2.1 are not shown.

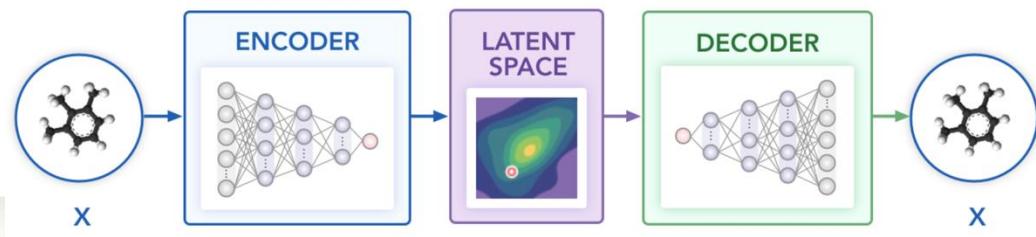


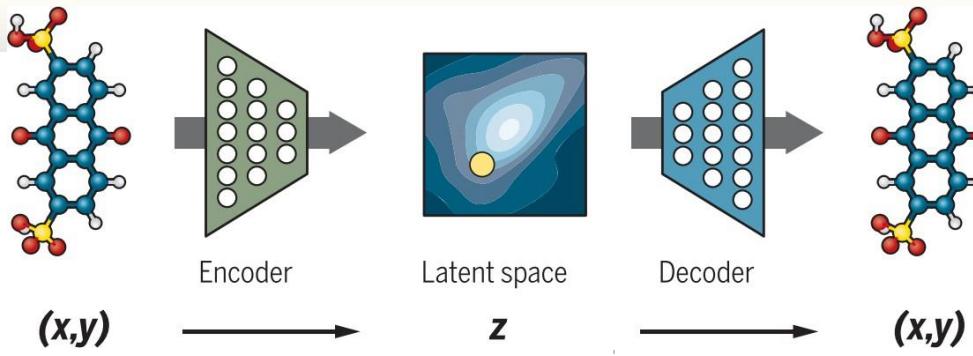


Generative models for molecular discovery: Recent advances and challenges

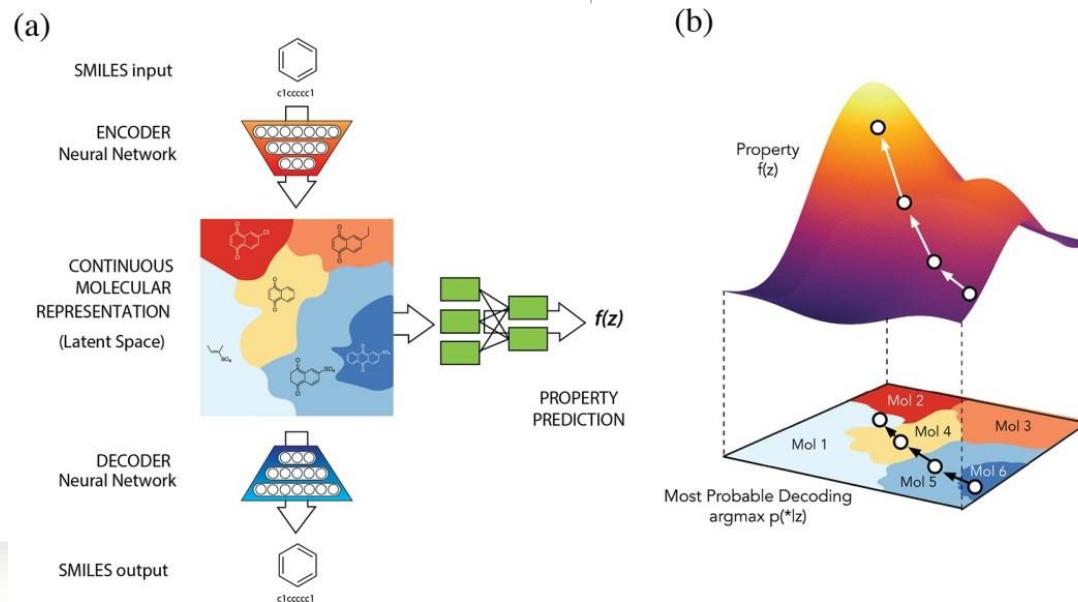


- Variational Autoencoder
- Autoencoder는 정보를 압축하는 encoder 와 압축된 정보로부터 원본을 복원하는 decoder로 이루어진 모델
- 압축 과정에서 각 물질의 핵심적인 특징을 **latent feature**로 나타낼 수 있게 됨
- Variational Autoencoder 는 latent feature 의 값들이 normal distribution 을 이루므로, 다양한 구조를 만들어내는 생성기로 활용할 수 있음.



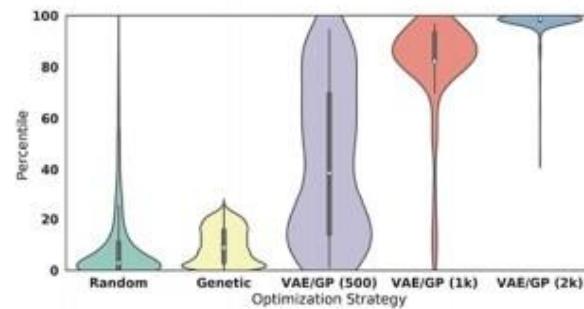


Science 27 Jul 2018, Vol. 361. pp.360-365

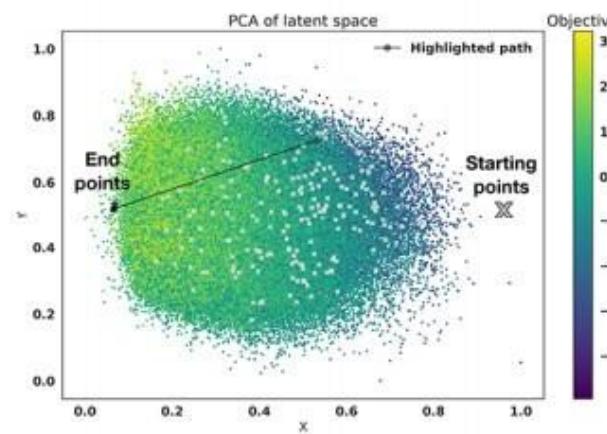


ACS Cent. Sci. 2018, 4, 2, 268-276

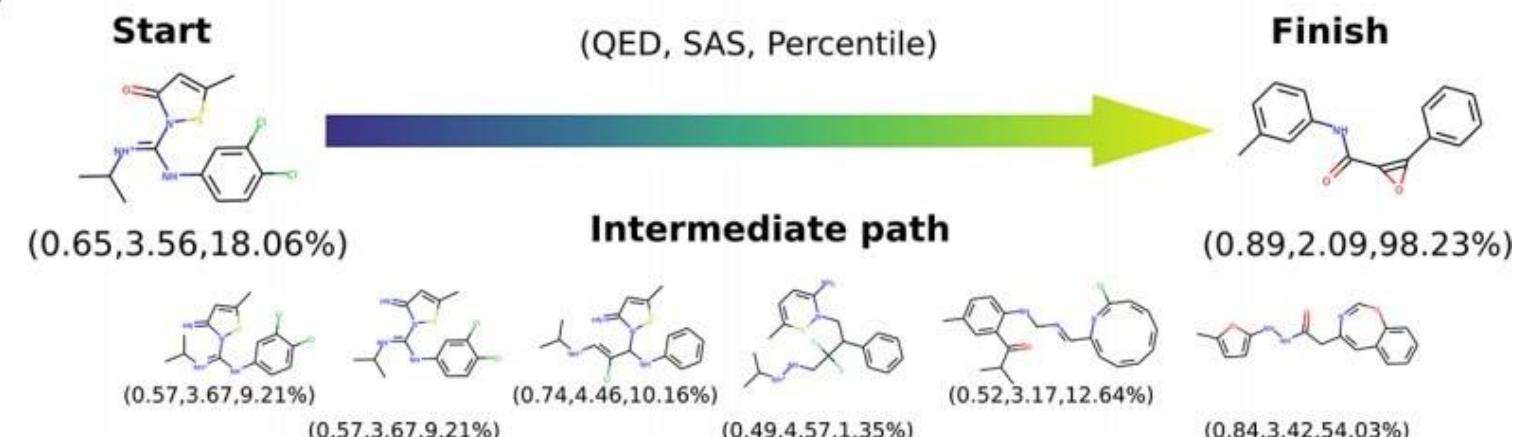
(a)

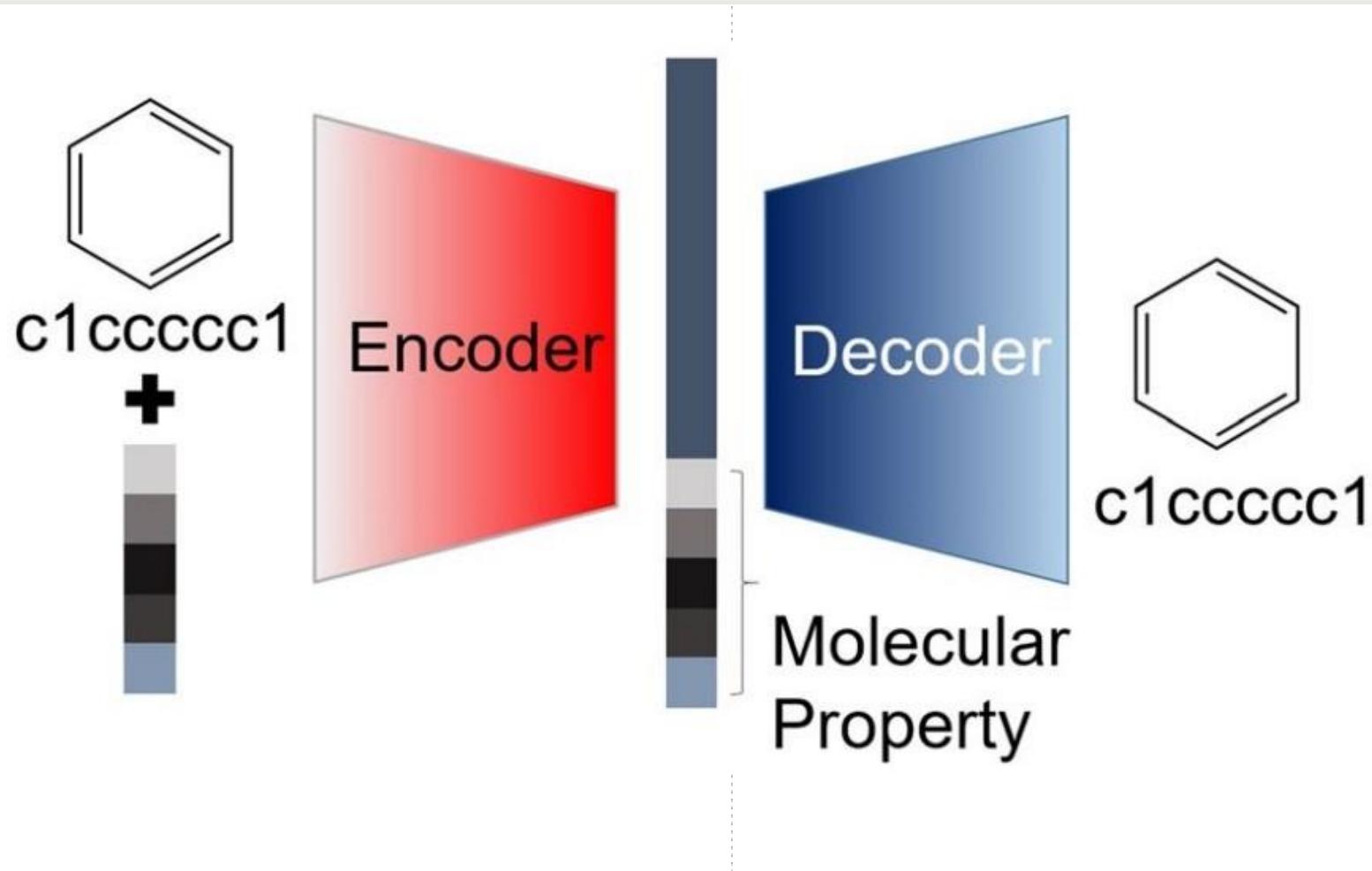


(b)

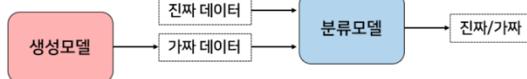


(c)

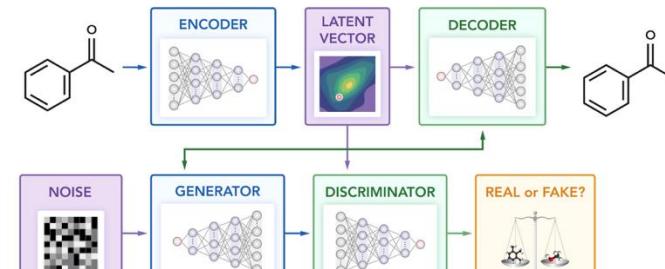




- Generative Adversarial Network
- 이미지 생성을 위해 개발된 기술
- 생성자(Generator) 와 판별자(Discriminator)가 경쟁하여 실제와 유사한 결과를 생성



GAN



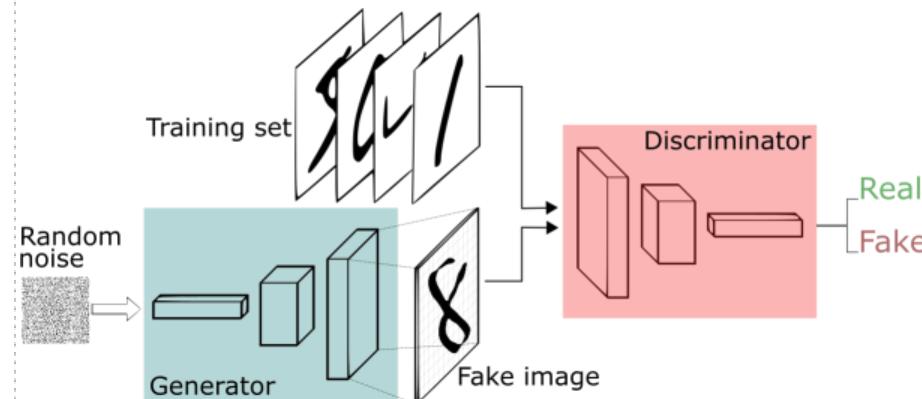
Latent GAN

“Rat race between fake bill maker vs. police”

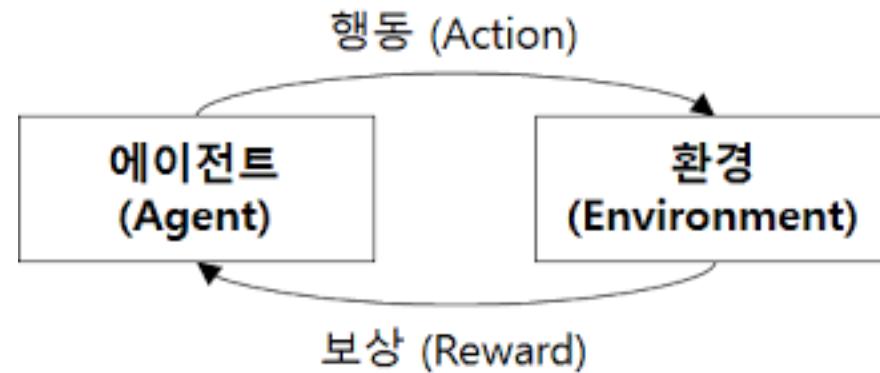
- generator : generate data as resemble as possible dataset samples
 - **discriminator** : distinguish real / fake data as precise as possible
- train two modules alternately

do not calculate actual distribution

→ danger of mode collapse



- 강화학습은 알파고의 사례로 유명해짐
- 강화학습을 위해서는 agent 와 environment 가 필요함
- Agent는 action (행동)을 취하면서 다양한 선택지를 탐색하고, Environment는 agent의 행동에 대한 보상 (reward) 을 제공한다
- Agent는 보상에 따라 자신의 행동을 조정하여 최적화된 선택을 하게 됨



- De novo design에서 강화학습의 agent는 물질, reward는 물성에 대한 보상으로 선택함.
- 매 step마다 물질은 합성을 통해 다른 구조로 변환되는 행동 (action)을 선택
- 합성 이후의 구조가 원하는 물성(drug-like, synthesis accessible)을 갖출 수록 보상이 높아진다.
- 여러 차례의 반복 학습을 통해 물성이 좋아지는 방향으로 합성을 하게 된다

- 강화학습의 대표적인 방법인 DQN (Deep Q-Networks) 를 구조 변경에 접목한 생성 모델
- Action 으로 원자 추가, 결합 추가, 결합 제거 중 하나를 선택하게 함
- 강화학습 기반의 de novo design의 시작을 알렸으나 실제 합성과는 차이가 커서 합성 불가능한 구조가 생성되기 쉬움

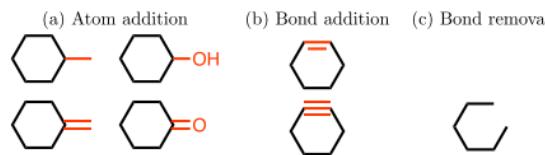
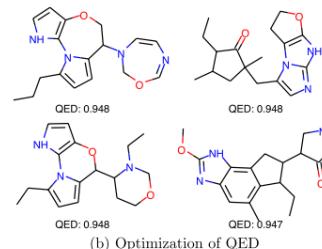
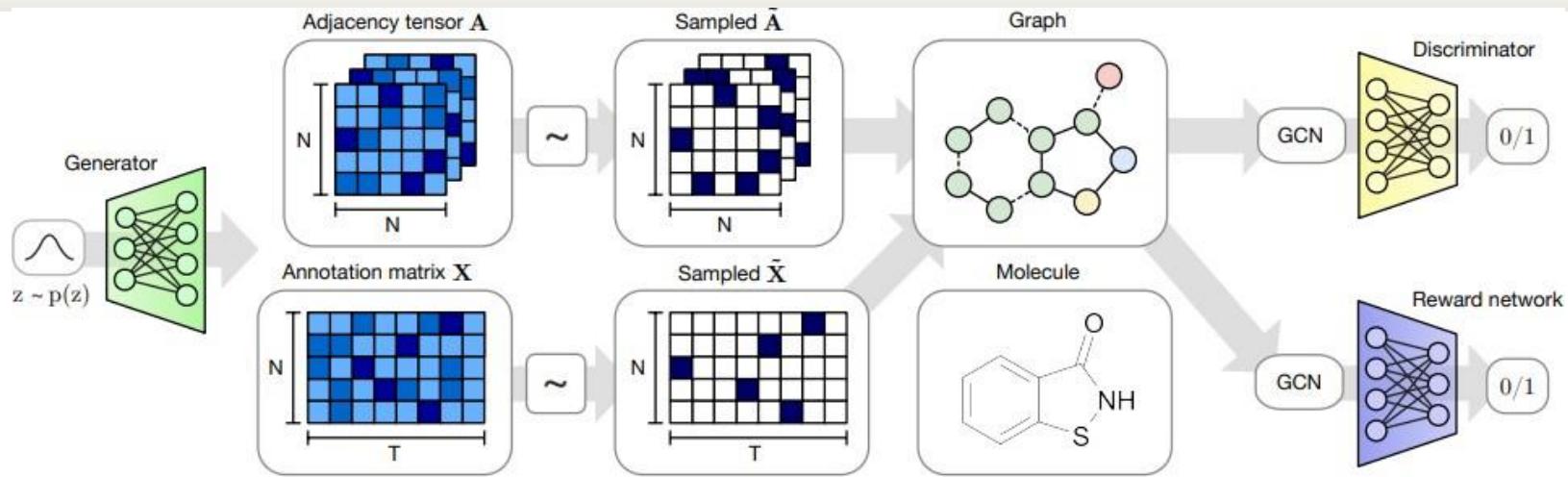


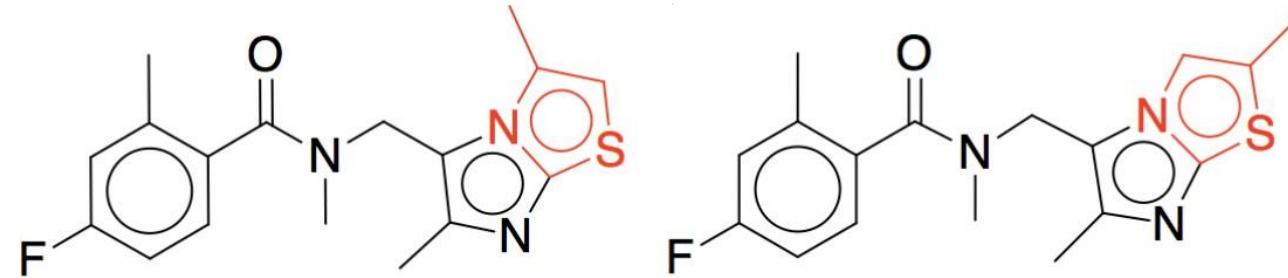
Figure 1. Valid actions on the state of cyclohexane. Modifications are shown in red. Invalid bond additions which violate the heuristics explained in Section 2.1 are not shown.





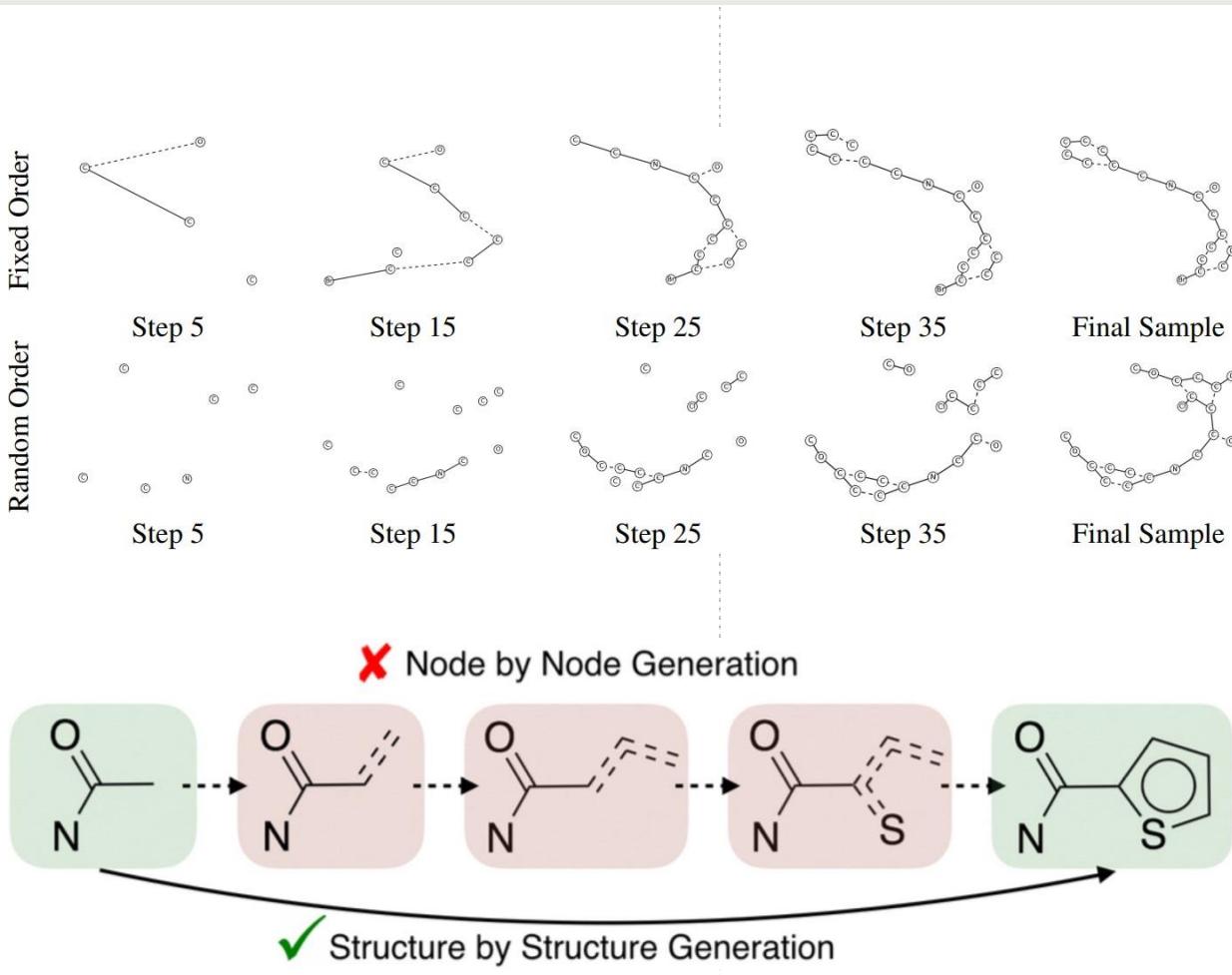
Algorithm	Valid	Unique	Novel
CharacterVAE	10.3	67.5	90.0
GrammarVAE	60.2	9.3	80.9
GraphVAE	55.7	76.0	61.6
GraphVAE/imp	56.2	42.0	75.8
GraphVAE NoGM	81.0	24.1	61.0
MolGAN	98.1	10.4	94.2

arXiv:1805.11973



arXiv:1802.04364

- Graph 기반과 Fragment 기반

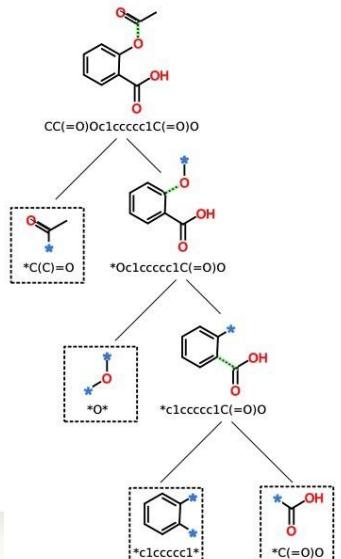


A Deep Generative Model for Fragment-Based Molecule Generation

Marco Podda
University of Pisa,
Largo Bruno Pontecorvo 3,
56127 Pisa, Italy

Davide Bacci
University of Pisa,
Largo Bruno Pontecorvo 3,
56127 Pisa, Italy

Alessio Micheli
University of Pisa,
Largo Bruno Pontecorvo 3,
56127 Pisa, Italy



Algorithm 1 Fragmentation

Require: Molecule M , Fragment List $F \leftarrow []$

- 1: **procedure** FRAGMENT(M, F)
- 2: **declare** Bond b
- 3: $b \leftarrow \text{GETFIRSTBRICSBOND}(M)$
- 4: **if** ISEMPTY(b) **then**
- 5: **return** F
- 6: **else**
- 7: **declare** Fragment f
- 8: **declare** Molecule M'
- 9: $f, M' \leftarrow \text{BREAKMOLATBOND}(M, b)$
- 10: $F \leftarrow \text{APPEND}(F, f)$
- 11: $M \leftarrow M'$
- 12: FRAGMENT(M, F)
- 13: **end if**
- 14: **end procedure**

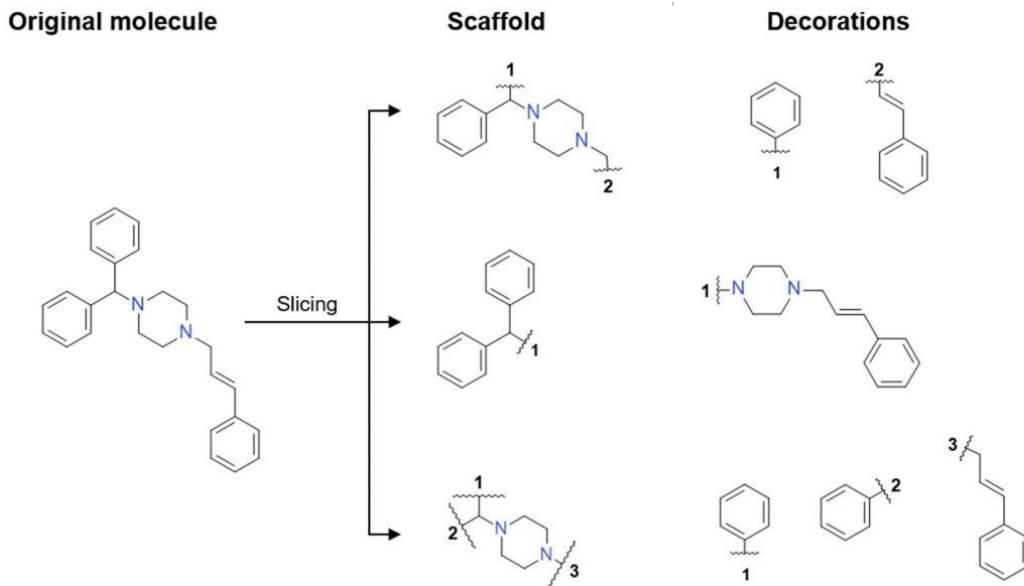
RESEARCH ARTICLE

Open Access

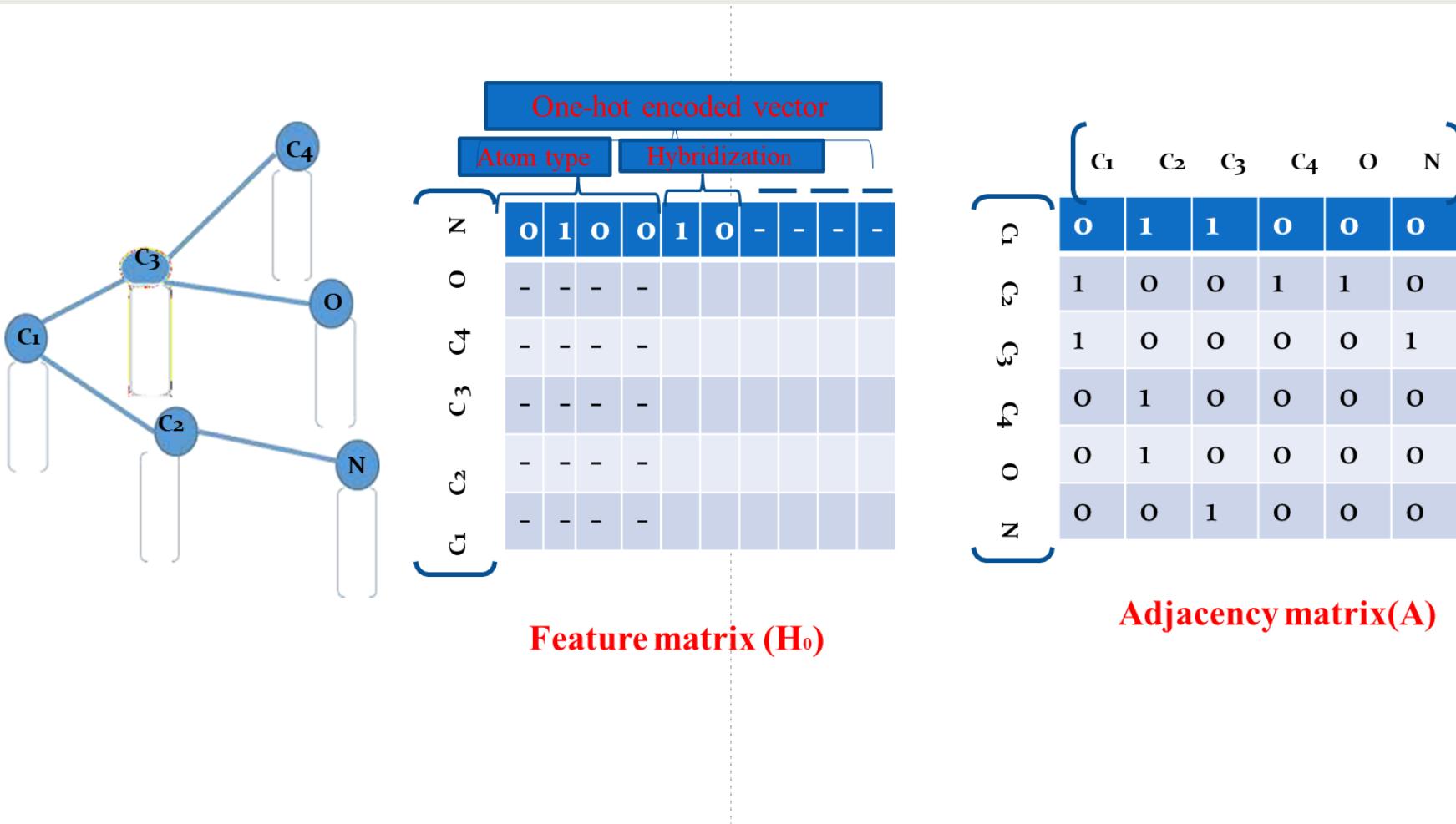


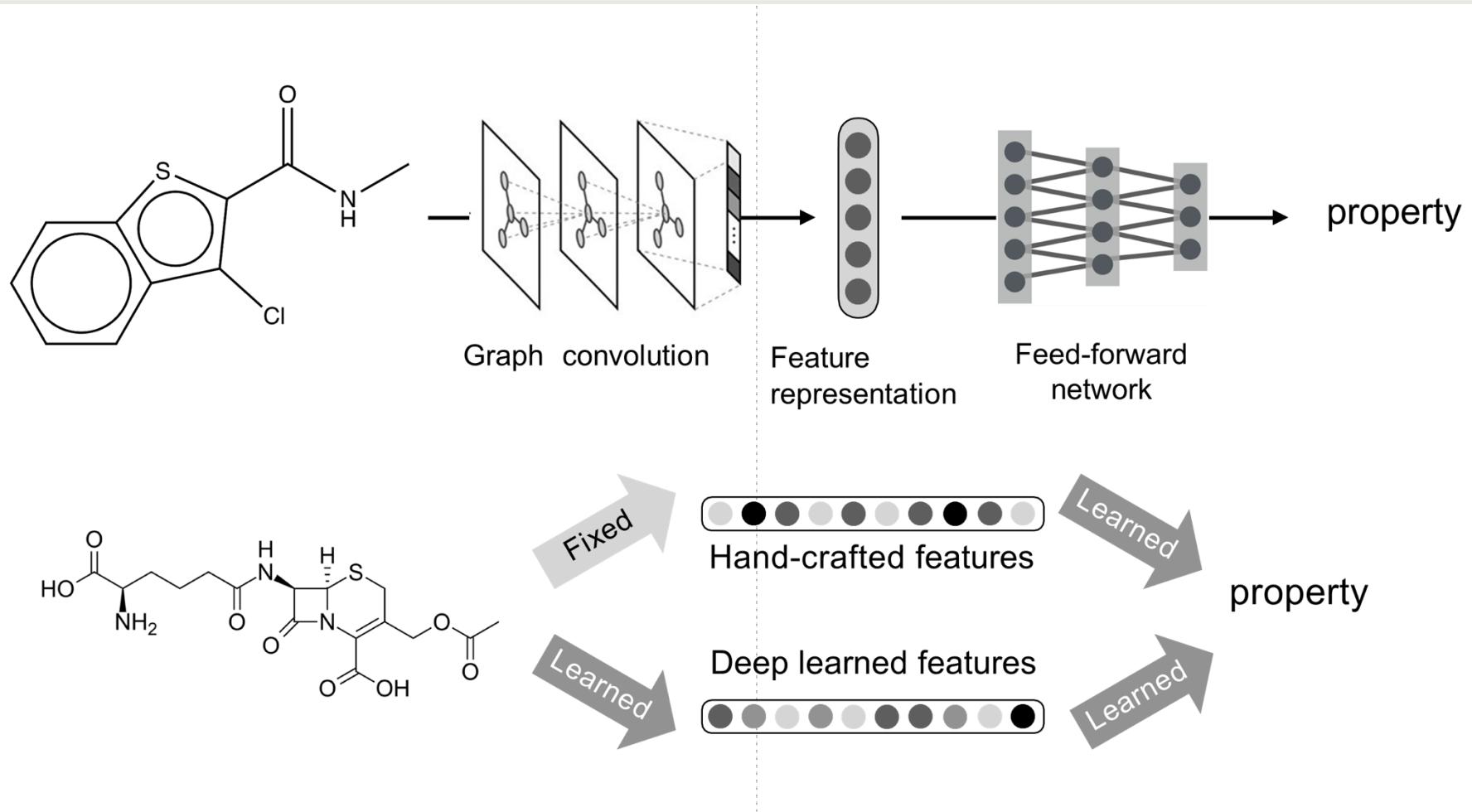
SMILES-based deep generative scaffold decorator for de-novo drug design

Josep Arús-Pous^{1,3*} , Atanas Patronov¹, Esben Jannik Bjerrum¹, Christian Tyrchan², Jean-Louis Reymond³, Hongming Chen⁴ and Ola Engkvist¹

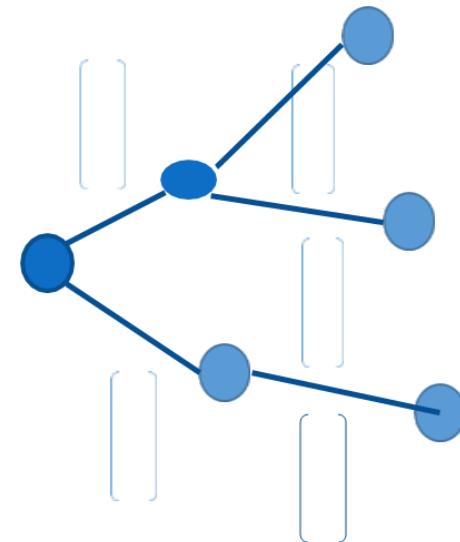


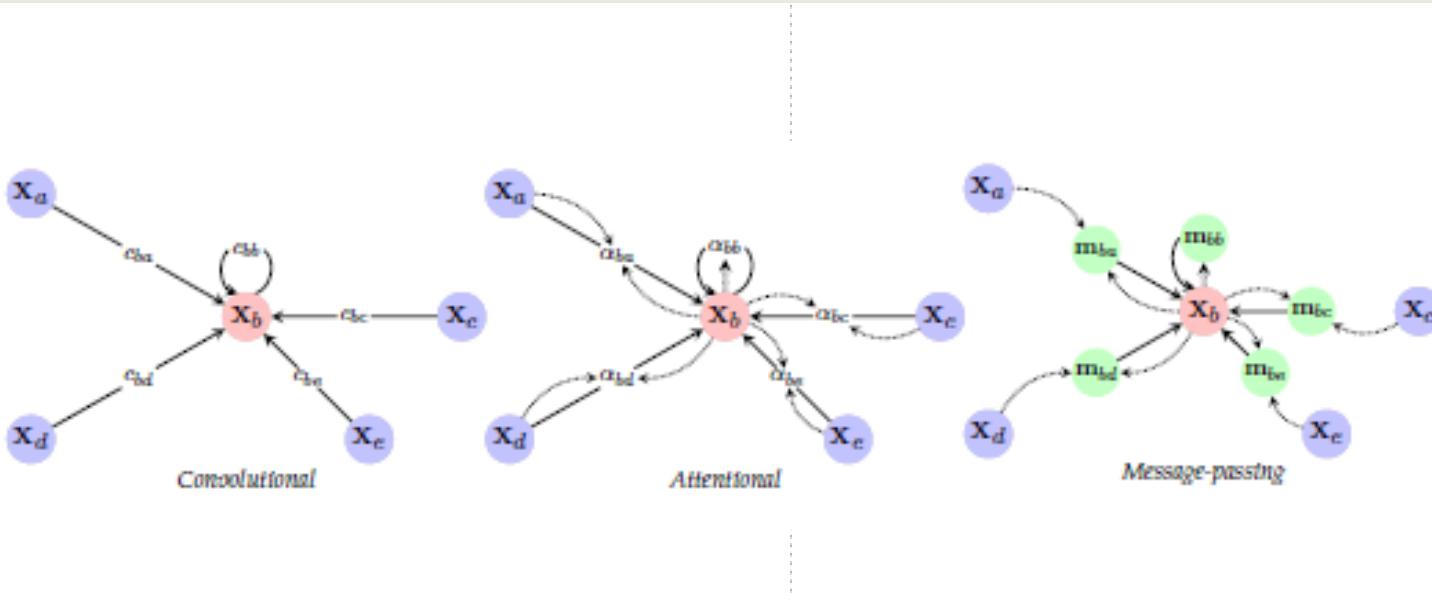
- Graph Neural Network





Features	Description	Size
Bond Type	Single, double, triple, or aromatic	4
Conjugated	Whether the bond is conjugated.	1
In-ring	Whether the bond is part of a ring.	1
Stereo	None, any, E/Z or cis/trans	6





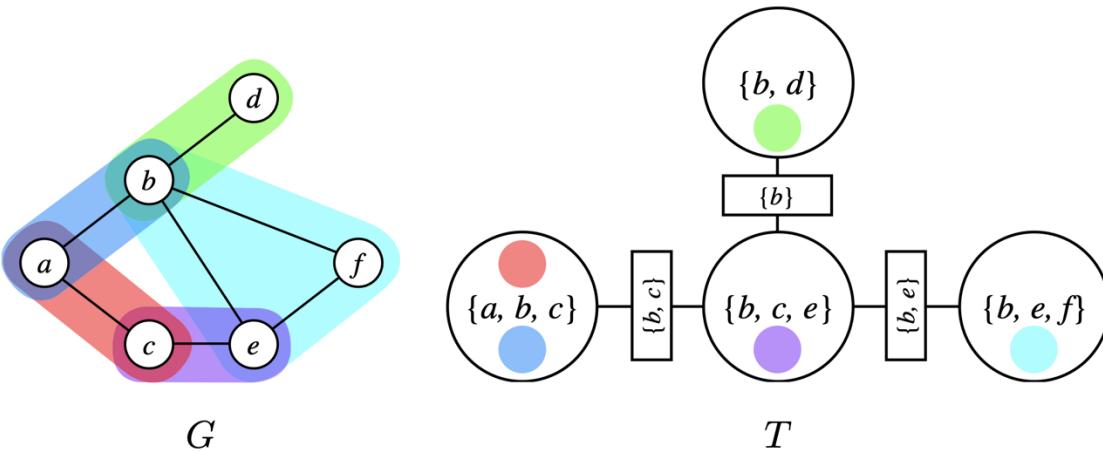
$$\text{NP: } h_i^{(l+1)} = \sum_{j \in \mathcal{N}(i) \cup \{i\}} c_{ij} h_j^{(l)}$$

$$\text{GCN: } h_i^{(l+1)} = \phi \left(h_i^{(l)}, \bigoplus_{j \in \mathcal{N}(i)} c_{ij} \psi(h_j^{(l)}) \right)$$

$$\text{GAT: } h_i^{(l+1)} = \phi \left(h_i^{(l)}, \bigoplus_{j \in \mathcal{N}(i)} \alpha(h_i^{(l)}, h_j^{(l)}) \psi(h_j^{(l)}) \right)$$

$$\text{MPN: } h_i^{(l+1)} = \phi \left(h_i^{(l)}, \bigoplus_{j \in \mathcal{N}(i)} \psi(h_i^{(l)}, h_j^{(l)}) \right)$$

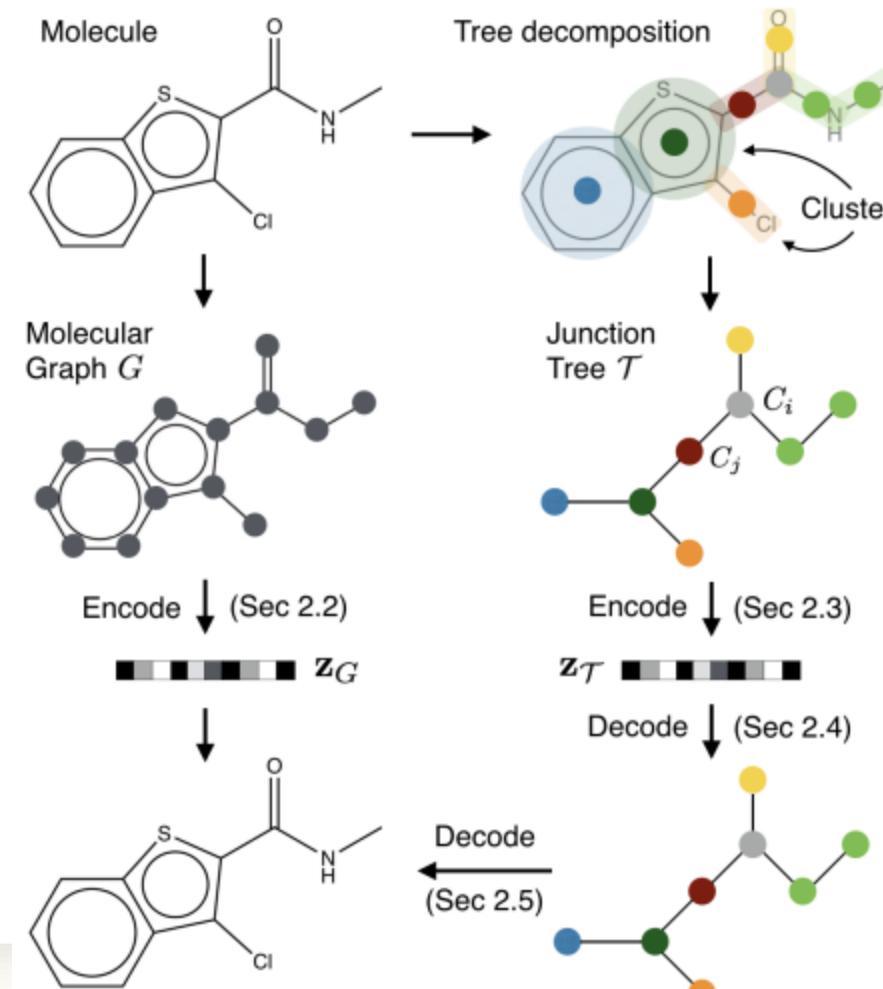
Junction trees



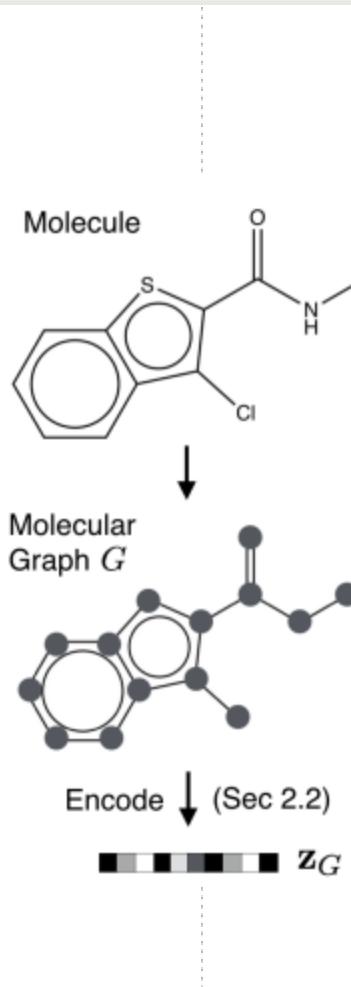
A cluster graph T is a **junction tree** for G if it has these three properties:

1. **singly connected**: there is exactly one path between each pair of clusters.
2. **covering**: for each clique A of G there is some cluster C such that $A \subseteq C$.
3. **running intersection**: for each pair of clusters B and C that contain i , each cluster on the unique path between B and C also contains i .

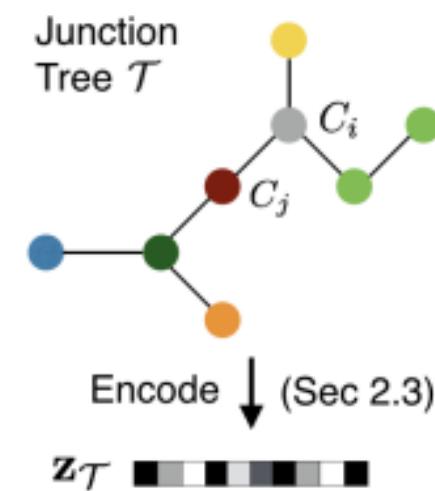
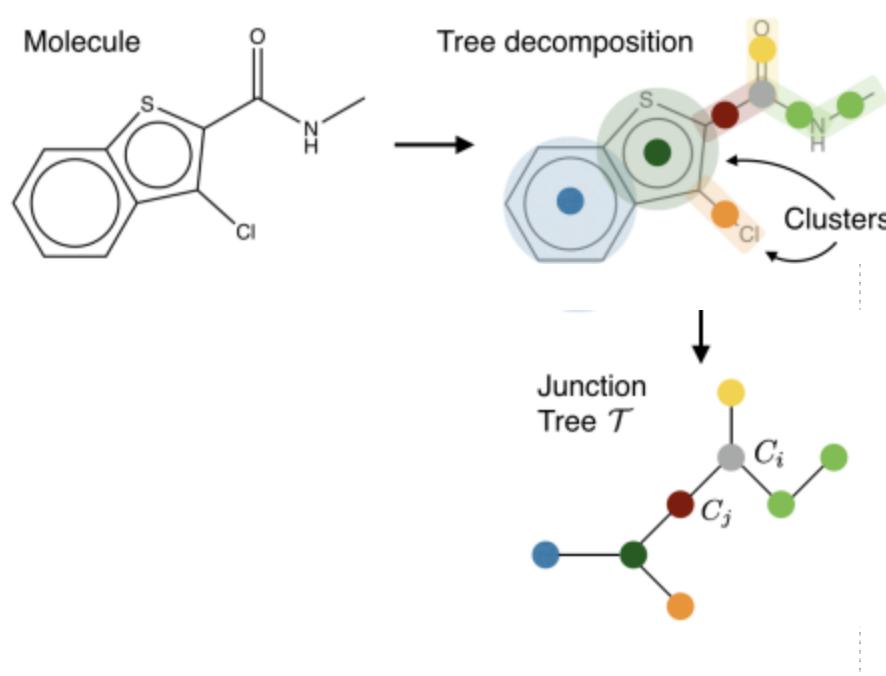
- JT-VAE



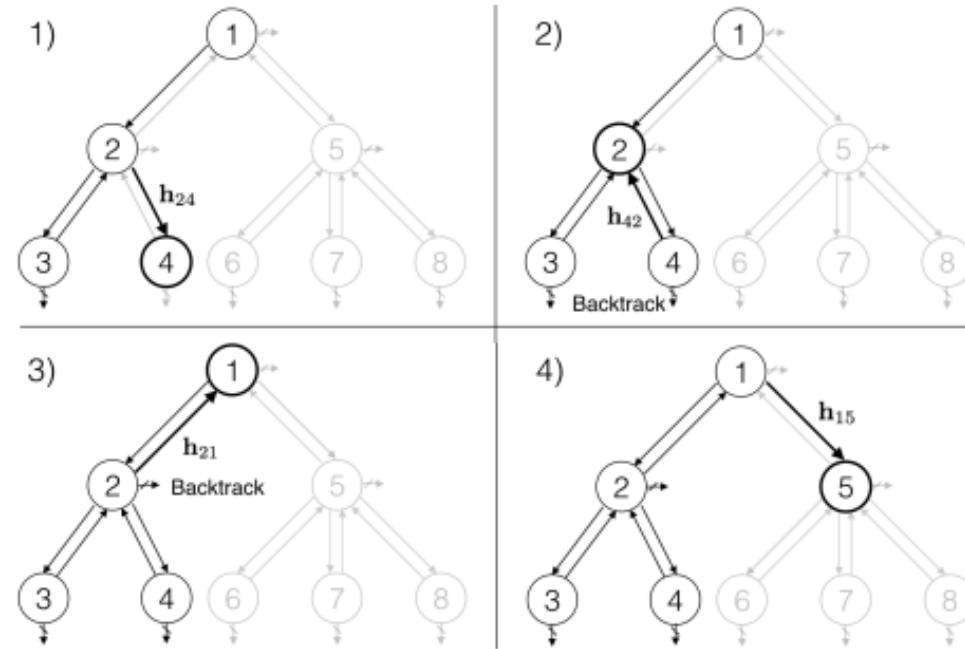
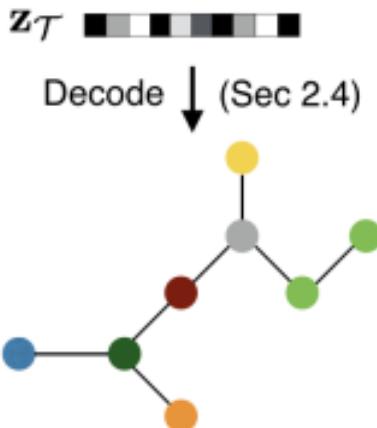
Graph encoder



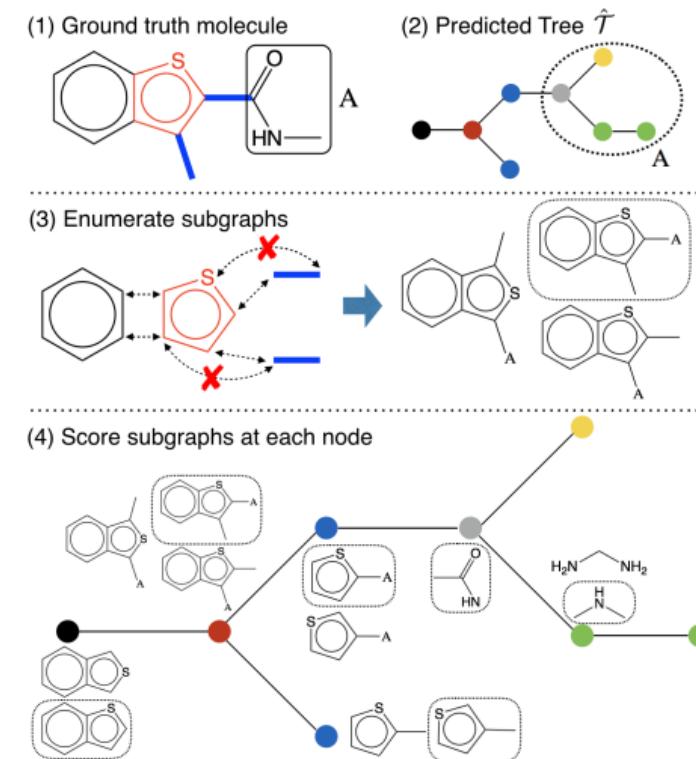
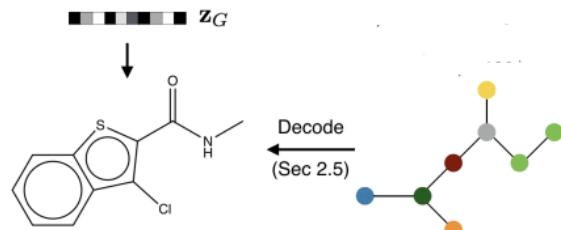
Tree encoder



Tree decoder



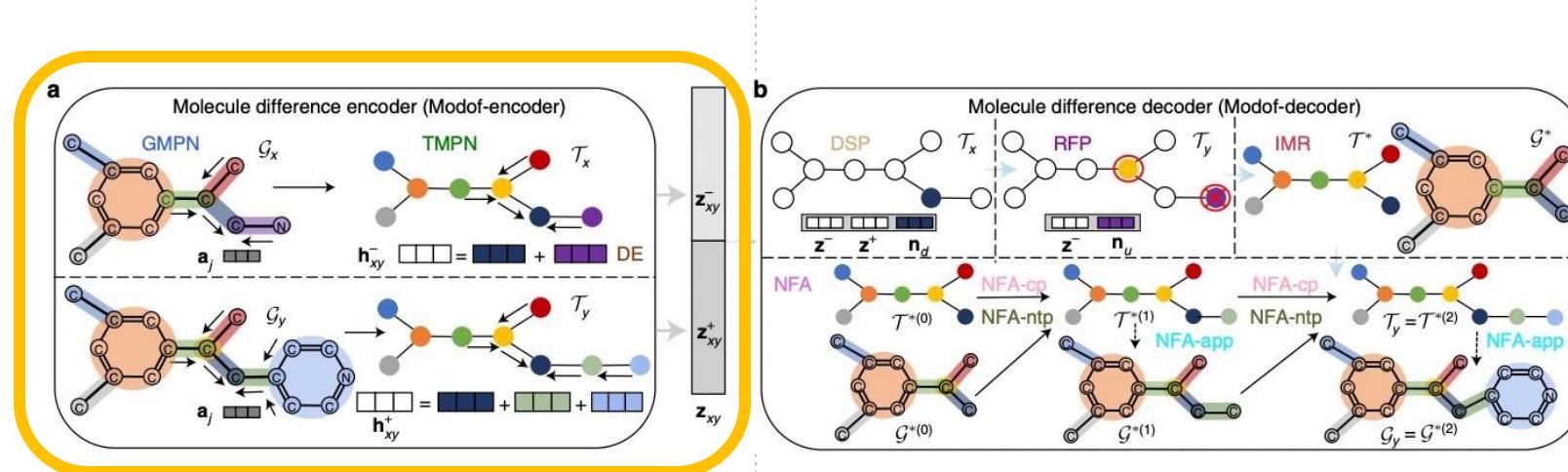
Graph decoder



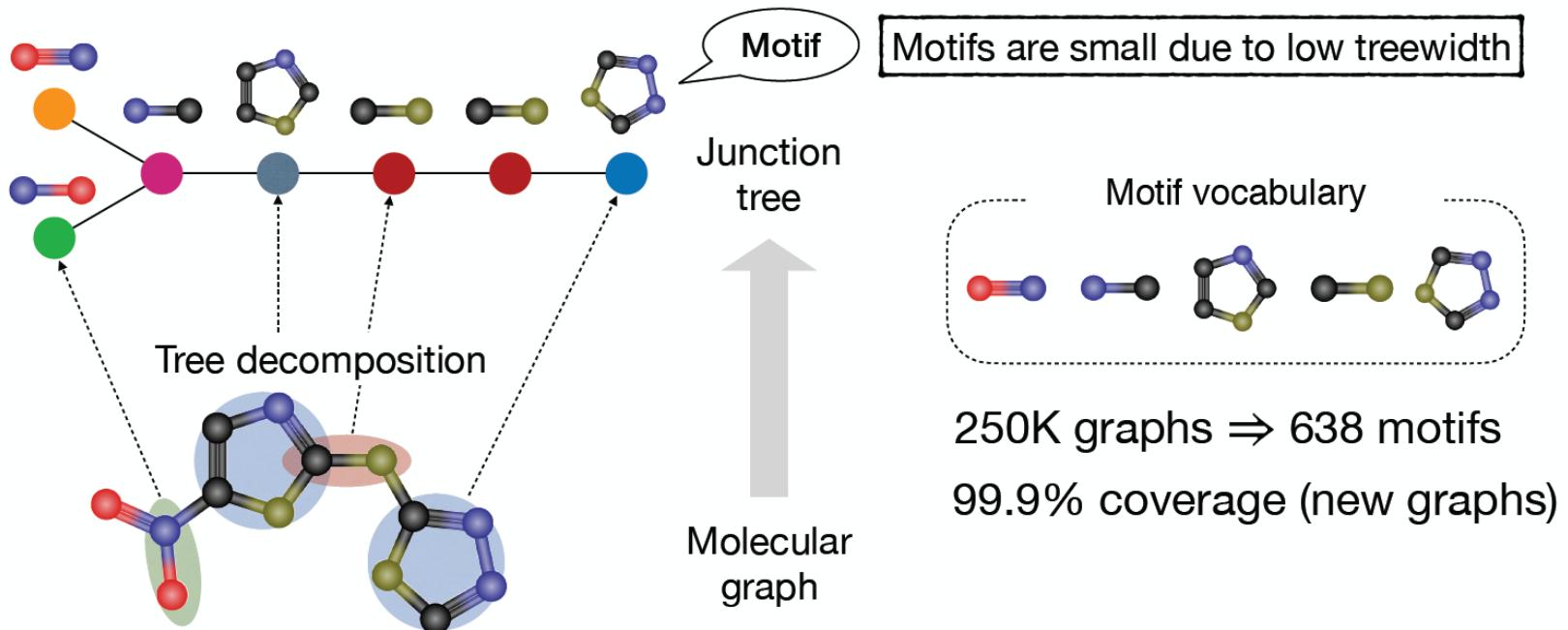
Method	Reconstruction	Validity
CVAE	44.6%	0.7%
GVAE	53.7%	7.2%
SD-VAE	76.2%	43.5%
GraphVAE	-	13.5%
JT-VAE (w/o check)	76.4%	93.5%
JT-VAE (full)	76.7%	100.0%

A deep generative model for molecule optimization via one fragment modification

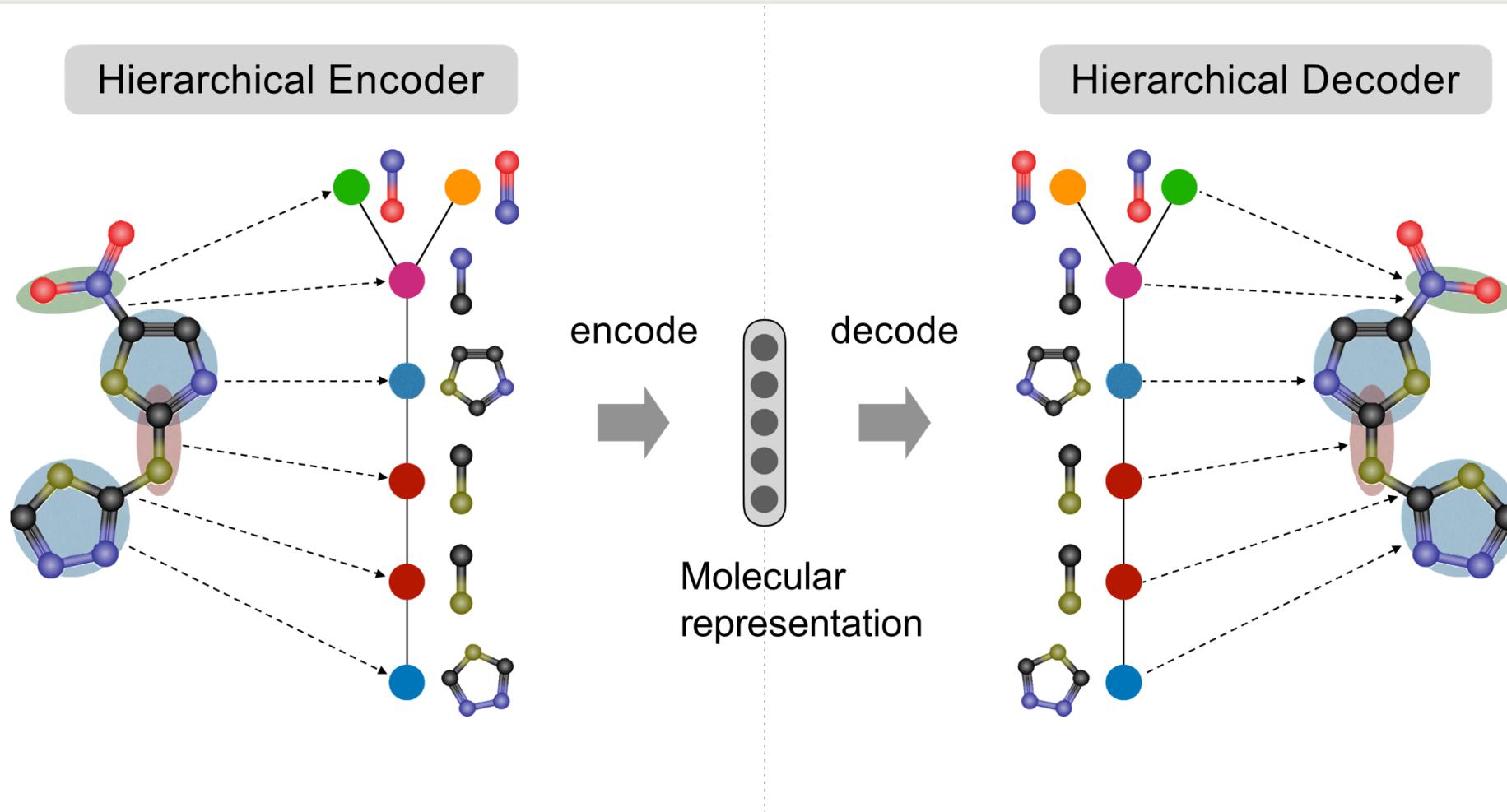
Ziqi Chen¹, Martin Renqiang Min², Srinivasan Parthasarathy^{1,3} and Xia Ning^{1,3,4}  



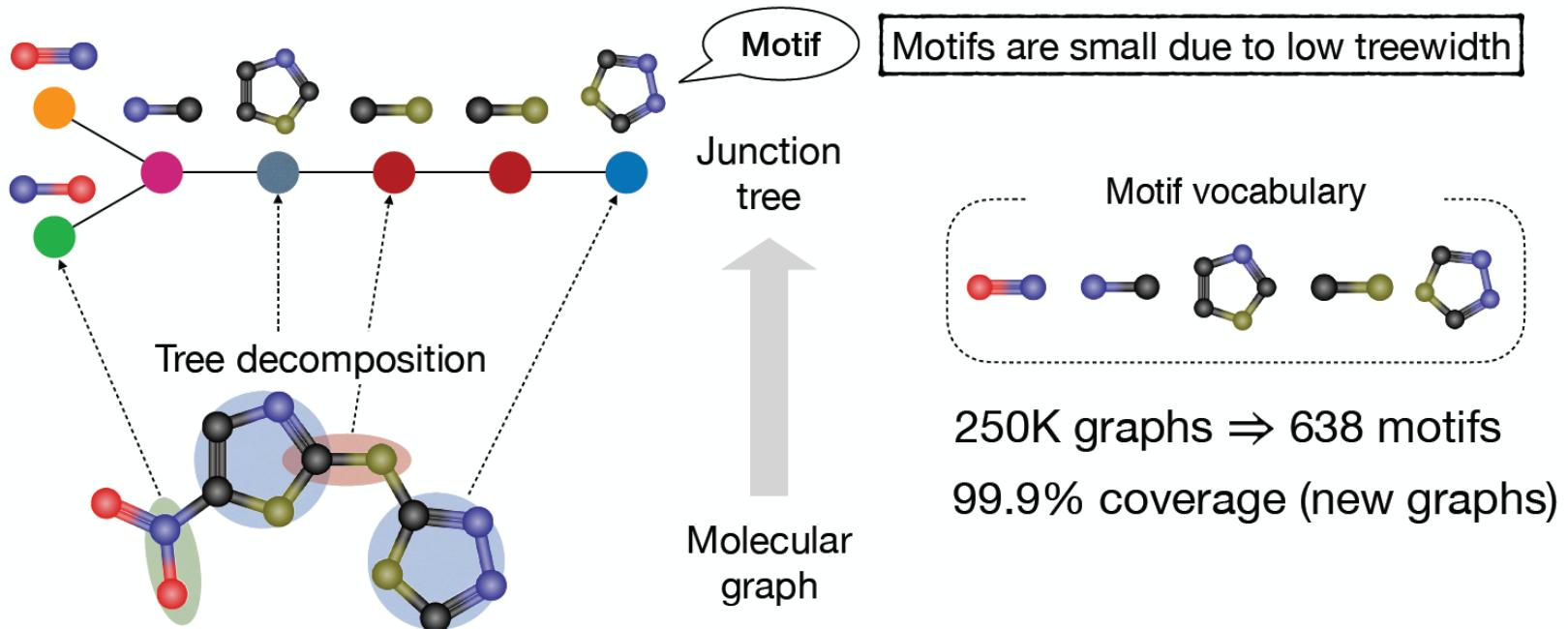
- Motif 들의 junction tree를 Graph neural net을 사용하여 학습



Inspired by the junction tree algorithm in graphical models.

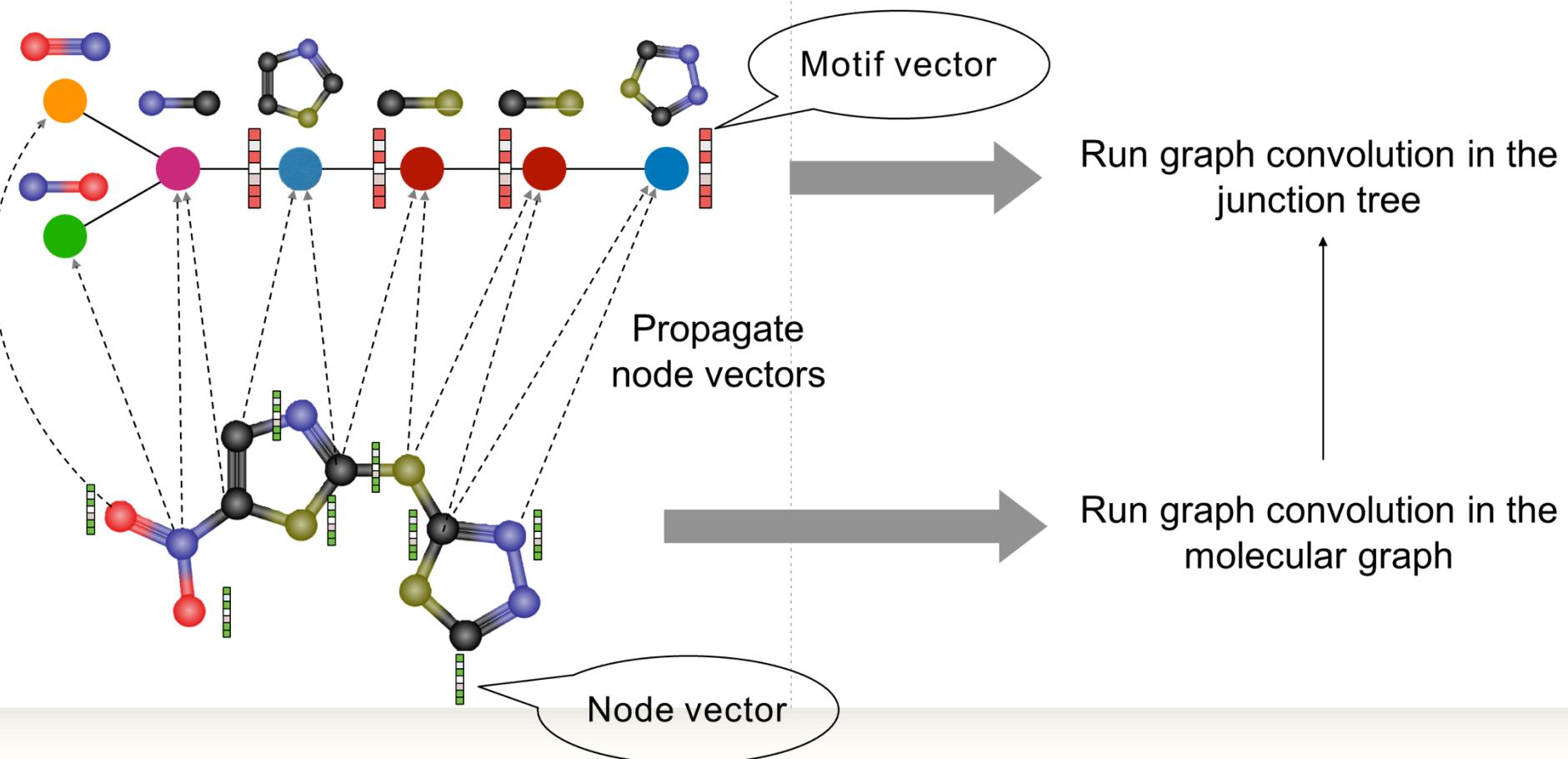


- Motif 들의 junction tree를 Graph neural net을 사용하여 학습

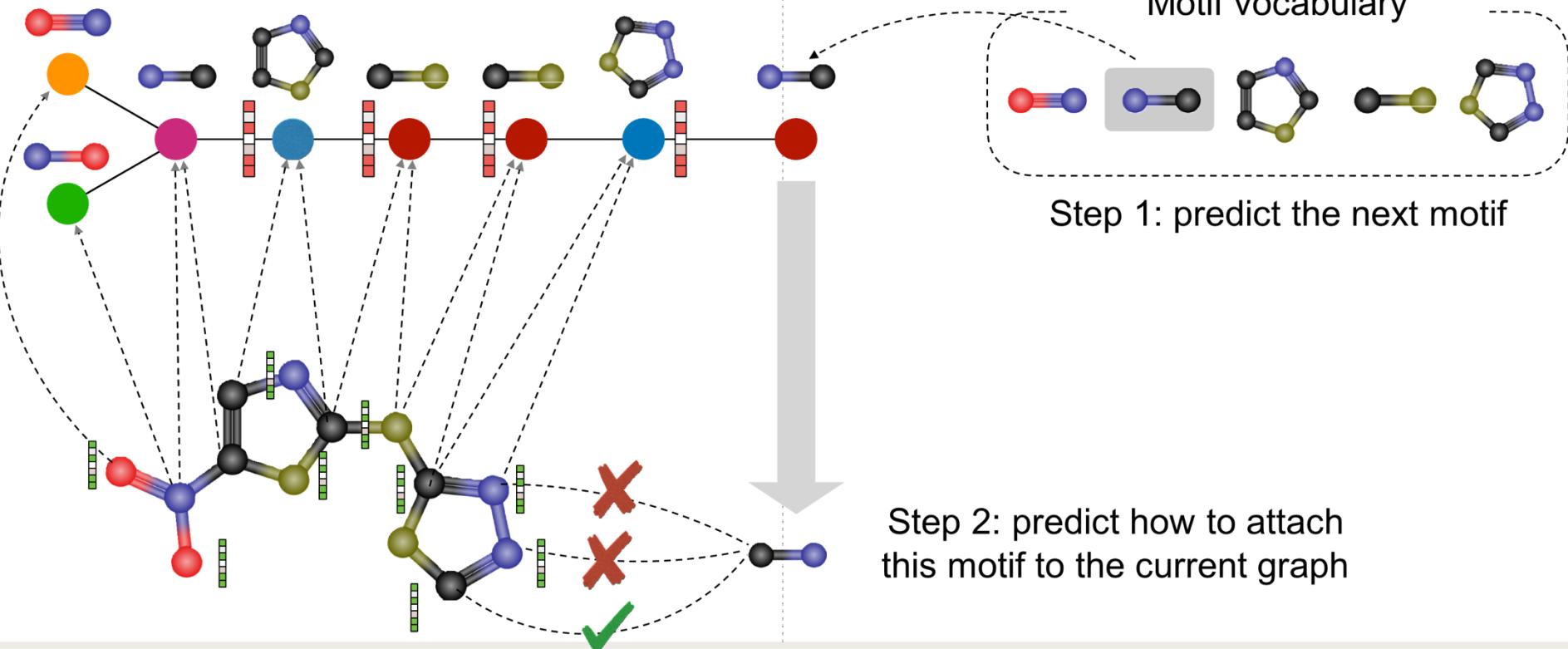


Inspired by the junction tree algorithm in graphical models.

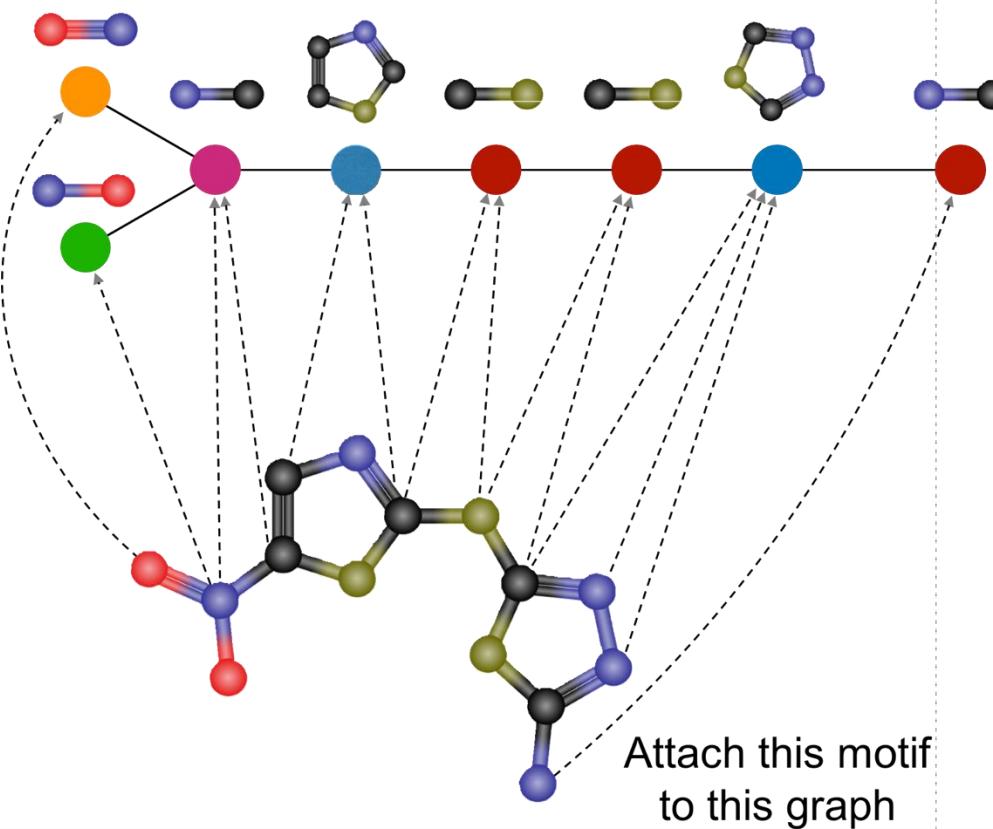
Hierarchical graph encoder



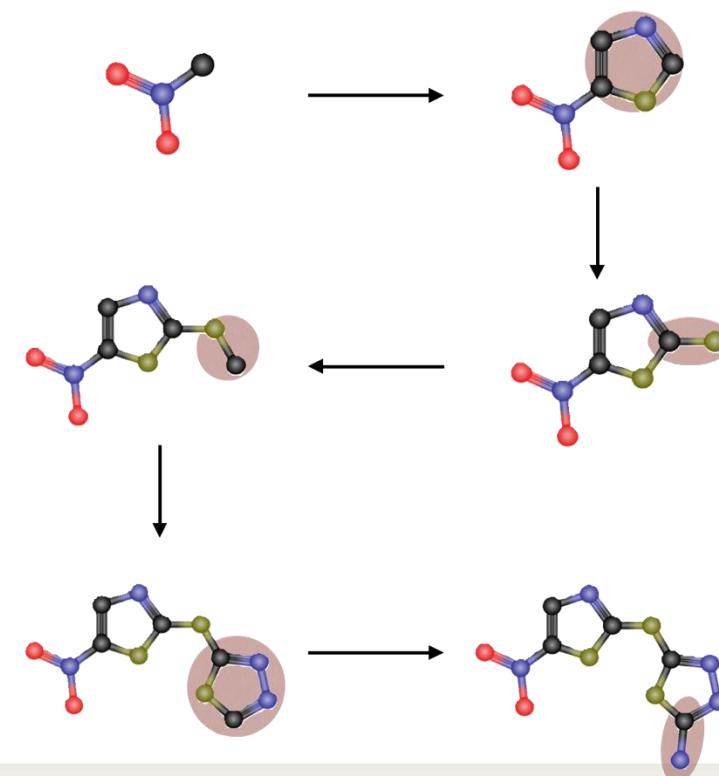
Hierarchical graph decoder

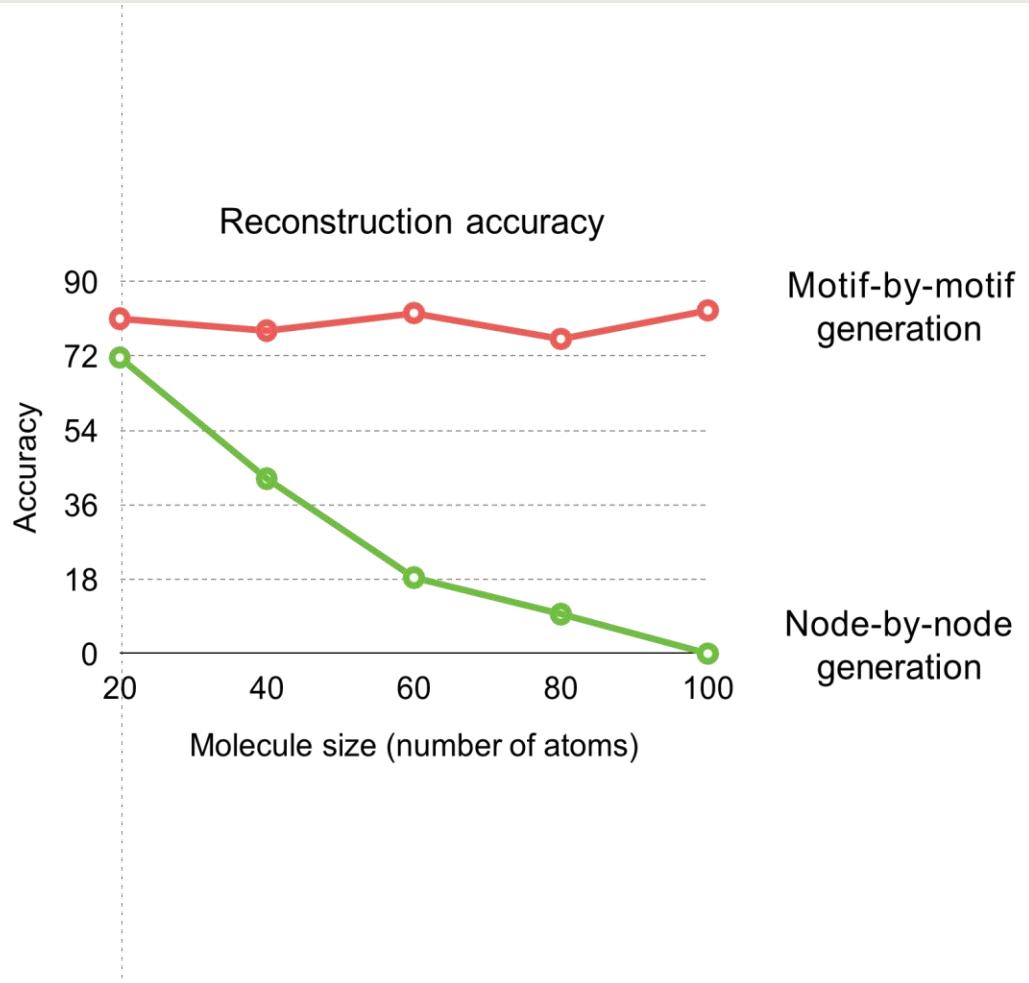
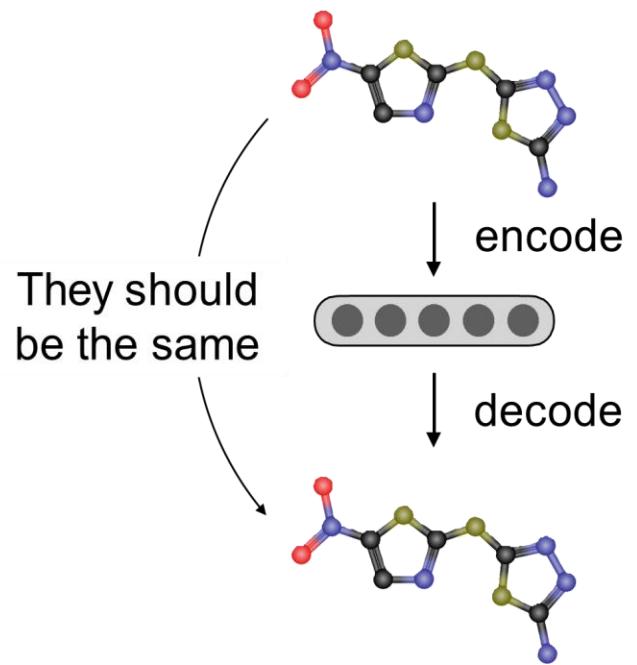


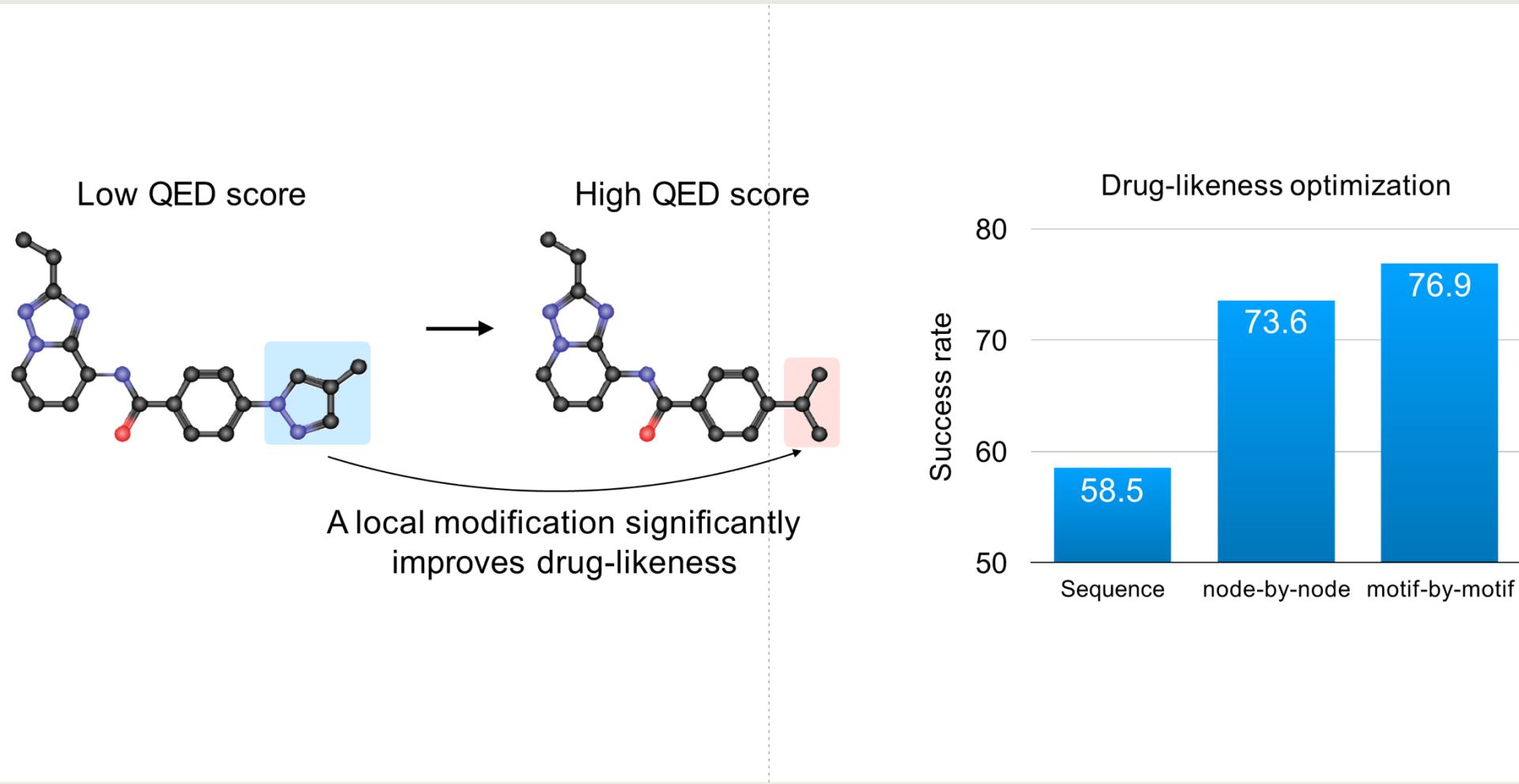
Hierarchical graph decoder



Motif-by-motif generation







RESEARCH

Open Access



COMA: efficient structure-constrained molecular generation using contractive and margin losses

Jonghwan Choi^{1,2}, Sangmin Seo^{1,2} and Sanghyun Park^{1*}

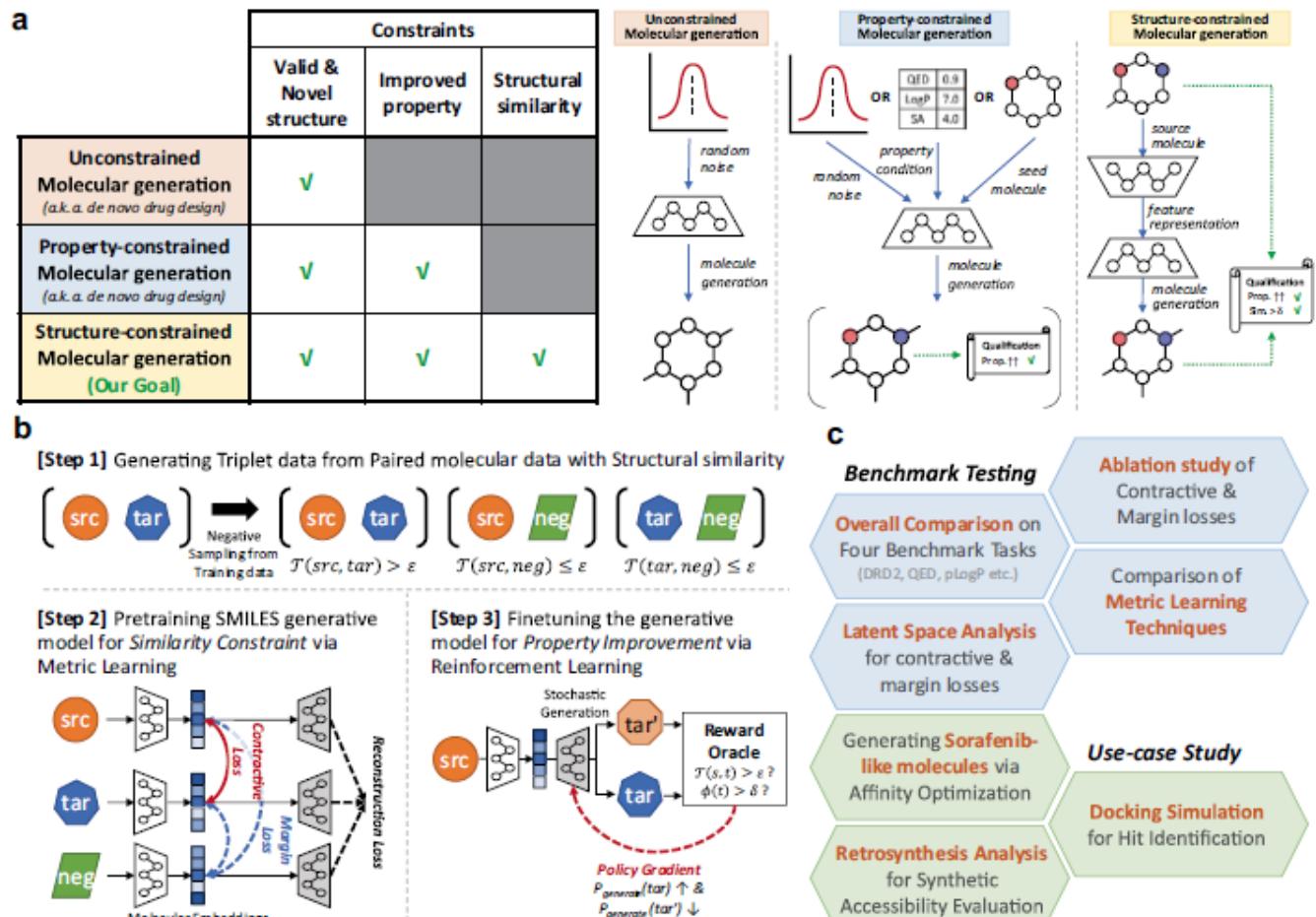
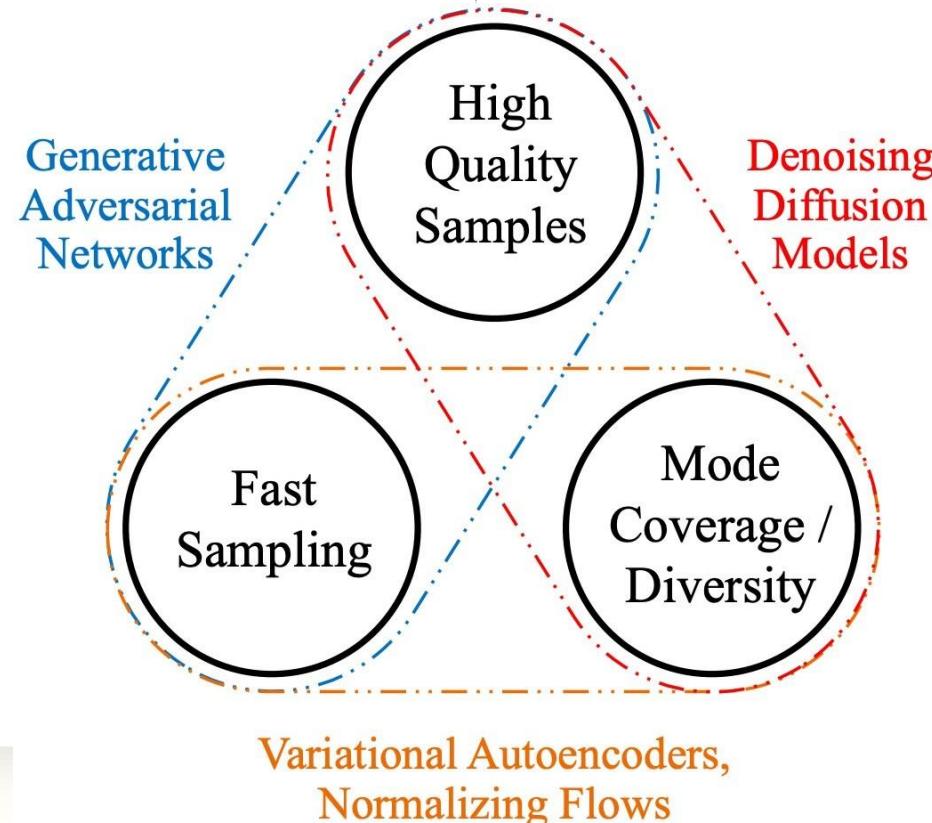


Fig. 1 Overviews of structure-constrained molecular generation and COMA. **a** Descriptions of three types of goal-directed molecular optimisation and examples. **b** Overviews of model architecture and training scheme of COMA. The generative model of COMA is based on a variational autoencoder, and the training process consists of following two steps: metric learning with contractive and margin losses to achieve the structural similarity constraint and reinforcement learning to produce molecules satisfying both constraints of structural similarity and property improvement. **c** Overview of experiments. The goals of four benchmark tasks are to enhance each molecular property score, and the goal of use case study for sorafenib resistance is to decrease an affinity score against the ABCG2 protein.

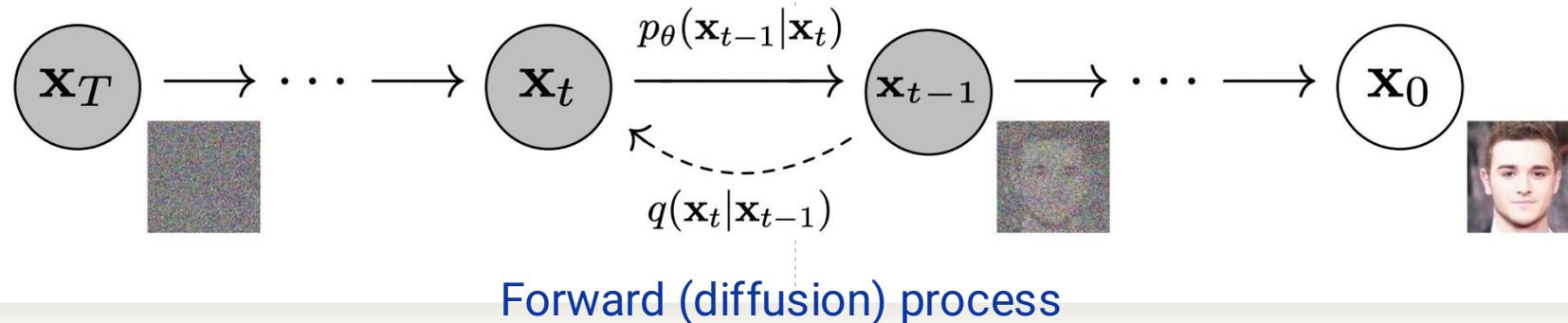
- Diffusion Model is All We Need?
 - **Trilemma of generative models: Quality vs. Diversity vs. Speed**
 - Diffusion model produces **diverse** and **high-quality** samples, but generations is **slow**



- Diffusion Probabilistic Model

Diffusion model aims to learn the reverse of noise generation procedure

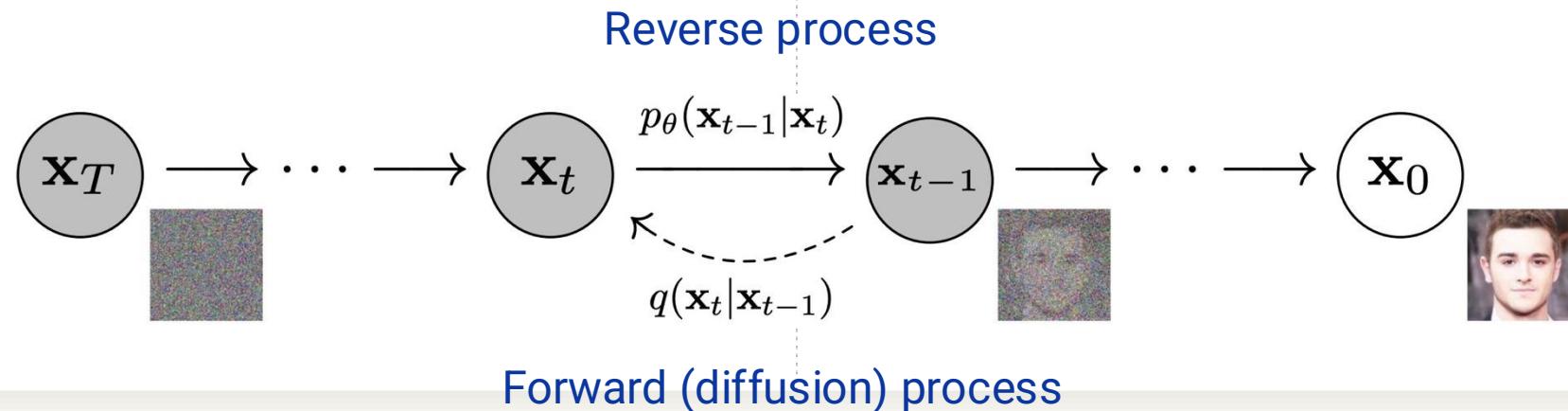
- **Forward step:** (Iteratively) Add noise to the original sample
→ The sample x_t converges to the **complete noise** x^{noise} (e.g., $\sim \mathcal{N}(0, I)$)



- Diffusion Probabilistic Model

Diffusion model aims to learn the reverse of noise generation procedure

- **Forward step:** (Iteratively) Add noise to the original sample
→ The sample x_t converges to the **complete noise** x^* (e.g., $\sim \mathcal{N}(0, I)$)
- **Reverse step:** Recover the original sample from the noise
→ Note that it is the “**generation**” procedure



- Diffusion Probabilistic Model

Diffusion model aims to learn the reverse of noise generation procedure

- **Forward step:** (Iteratively) Add noise to the original sample
→ Technically, it is a product of **conditional noise** distributions $q(\mathbf{x}_t | \mathbf{x}_{t-1})$
- Usually, the parameters β_t are **fixed** (one can jointly learn, but not beneficial)
- **Noise annealing** (i.e., reducing noise scale $\beta_t < \beta_{t-1}$) is crucial to the performance

$$q(\mathbf{x}_{1:T} | \mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1}), \quad q(\mathbf{x}_t | \mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

- Diffusion Probabilistic Model

Diffusion model aims to learn the reverse of noise generation procedure

- **Forward step:** (Iteratively) Add noise to the original sample

→ Technically, it is a product of **conditional noise** distributions $q(\mathbf{x}_t|\mathbf{x}_{t-1})$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

Reverse step: Recover the original sample from the noise

→ It is also a product of **conditional (de)noise** distributions $p(\mathbf{x}_{0:T}|\mathbf{x}_0)$

- Use the **learned** parameters: **denoiser** μ_θ (main part) and **randomness** Σ_θ

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

- Diffusion Probabilistic Model

- Diffusion model aims to learn the **reverse of noise generation procedure**

Forward step: (Iteratively) Add noise to the original sample

Reverse step: Recover the original sample from the noise

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t),$$

- **Training:** Minimize **variational lower bound** of the model $p_\theta(\mathbf{x}_0)$

$$\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q \left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right]$$

- Diffusion Probabilistic Model

- Diffusion model aims to learn the **reverse of noise generation procedure**

Forward step: (Iteratively) Add noise to the original sample

Reverse step: Recover the original sample from the noise

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t),$$

- **Training:** Minimize variational lower bound of the model $p_\theta(\mathbf{x}_0)$

$$\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q \left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right]$$

→ It can be decomposed to the **step-wise losses** (for each step t)

$$\mathbb{E}_q \left[\underbrace{D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \right]$$

- Diffusion Probabilistic Model

- Diffusion model aims to learn the **reverse of noise generation procedure**
- **Training:** Minimize **variational lower bound** of the model $p_\theta(\mathbf{x}_0)$
 - It can be decomposed to the **step-wise** losses (for each step t)

$$\mathbb{E}_q \left[\underbrace{D_{\text{KL}}(q(\mathbf{x}_T | \mathbf{x}_0) \| p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t))}_{L_{t-1}} - \underbrace{\log p_\theta(\mathbf{x}_0 | \mathbf{x}_1)}_{L_0} \right]$$

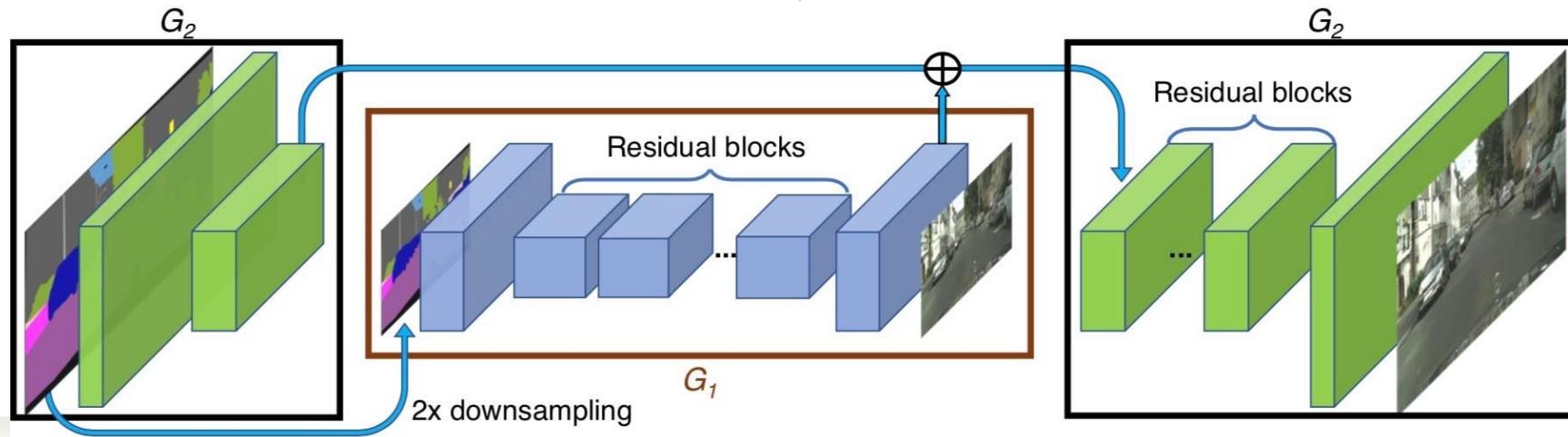
- Here, the **true reverse step** $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$ can be computed as a **closed form** of β_t
 - Note that we only define the true forward step $q(\mathbf{x}_t | \mathbf{x}_{t-1})$

$$q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\boldsymbol{\beta}}_t^3 \mathbf{I})$$

where $\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) := \tilde{\beta}_t^1 \mathbf{x}_0 + \tilde{\beta}_t^2 \mathbf{x}_t$
- Since all distributions above are Gaussian, the KL divergences are tractable

- Diffusion Probabilistic Model

- Diffusion model aims to learn the reverse of noise generation procedure
 - Network: Use the image-to-image translation (e.g., U-Net) architectures
 - Recall that input is x_t and output is x_{t-1} , both are images
 - It is expensive since both input and output are high-dimensional
 - Note that the denoiser $\mu_\theta(x_t, t)$ shares weights, but conditioned by step t



- Denoising Diffusion Probabilistic Model (DDPM)
 - DDPM **reparametrizes** the reverse distributions of diffusion models
 - **Key idea:** The original reverse step **fully creates** the denoiser $\mu_\theta(x_t, t)$ from x_t
 - However, x_{t-1} and x_t share most information, and thus it is redundant
→ Instead, create the **residual** $\epsilon_\theta(x_t, t)$ and add to the original x_t

- Denoising Diffusion Probabilistic Model (DDPM)

- DDPM **reparametrizes** the reverse distributions of diffusion models

- **Key idea:** The original reverse step **fully creates** the denoiser $\mu_\theta(\mathbf{x}_t, t)$ from \mathbf{x}_t
 - However, \mathbf{x}_{t-1} and \mathbf{x}_t share most information, and thus it is redundant
→ Instead, create the **residual** $\epsilon_\theta(\mathbf{x}_t, t)$ and add to the original \mathbf{x}_t
 - Formally, DDPM **reparametrizes** the learned reverse distribution as¹

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right)$$

and the step-wise objective L_{t-1} can be reformulated as²

$$\mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[\left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) \right\|^2 \right]$$

1. α_t are some constants determined by β_t

2. Note that we need no “intermediate” samples, and only compare the forward noise ϵ and reverse noise ϵ_θ conditioned on \mathbf{x}_0
Ho et al. Denoising Diffusion Probabilistic Models. NeurIPS'20

- Denoising Diffusion Implicit Model (DDIM)
 - DDIM **roughly sketches** the final sample, then **refine** it with the reverse process
 - Motivation:
Diffusion model is slow due to the **iterative procedure**
 - GAN/VAE creates the sample by **one-shot** forward operation
 - ⇒ Can we combine the advantages for **fast sampling** of diffusion models?
 - **Technical spoiler:**
Instead of naively applying diffusion model upon GAN/VAE,
DDIM proposes a **principled approach** of rough sketch + refinement

- Denoising Diffusion Implicit Model (DDIM)
- DDIM roughly sketches the final sample, then refine it with the reverse process

- Key idea:

- Given \mathbf{x}_t , generate the rough sketch \mathbf{x}_0 and refine $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ ¹
- Unlike original diffusion model, it is not a Markovian structure



Figure 1: Graphical models for diffusion (left) and non-Markovian (right) inference models.

1. Recall that the original diffusion model uses $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
 Song et al. Denoising Diffusion Implicit Models. ICLR'21

- Denoising Diffusion Implicit Model (DDIM)

- DDIM roughly sketches the final sample, then refine it with the reverse process

- Key idea: Given \mathbf{x}_t , generate the rough sketch \mathbf{x}_0 and refine $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$



- Formulation: Define the forward distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ as

$$q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}\mathbf{x}_0 + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1 - \alpha_t}}, \sigma_t^2 \mathbf{I}\right)$$

then, the forward process is derived from Bayes' rule

$$q_\sigma(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) = \frac{q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)q_\sigma(\mathbf{x}_t|\mathbf{x}_0)}{q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_0)}$$

- Denoising Diffusion Implicit Model (DDIM)
- DDIM roughly sketches the final sample, then refine it with the reverse process
 - Key idea: Given \mathbf{x}_t , generate the rough sketch \mathbf{x}_0 and refine $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$



- Formulation: Forward process is $q_\sigma(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) = \frac{q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)q_\sigma(\mathbf{x}_t|\mathbf{x}_0)}{q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_0)}$

and reverse process is

$$\mathbf{x}_{t-1} = \underbrace{\sqrt{\alpha_{t-1}} \left(\frac{\mathbf{x}_t - \sqrt{1 - \alpha_t} \epsilon_\theta^{(t)}(\mathbf{x}_t)}{\sqrt{\alpha_t}} \right)}_{\text{"predicted } \mathbf{x}_0\text{"}} + \underbrace{\sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta^{(t)}(\mathbf{x}_t)}_{\text{"direction pointing to } \mathbf{x}_t\text{"}} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}}$$

- Denoising Diffusion Implicit Model (DDIM)
 - DDIM roughly sketches the final sample, then refine it with the reverse process
 - Key idea: Given \mathbf{x}_t , generate the rough sketch \mathbf{x}_0 and refine $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$

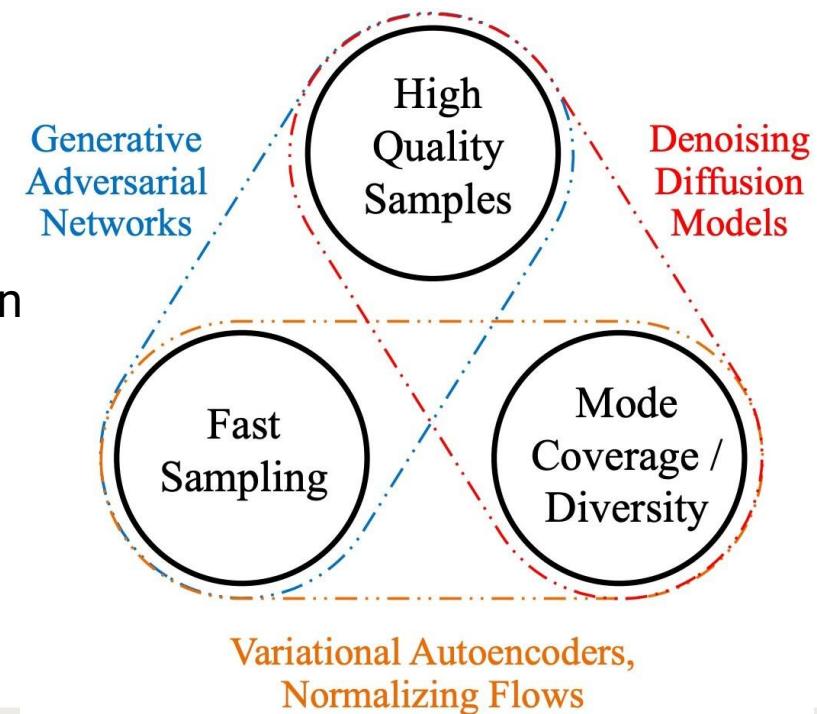


- Formulation: Forward process is $q_\sigma(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) = \frac{q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)q_\sigma(\mathbf{x}_t|\mathbf{x}_0)}{q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_0)}$
- and reverse process is $\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}} \left(\underbrace{\frac{\mathbf{x}_t - \sqrt{1 - \alpha_t} \epsilon_\theta^{(t)}(\mathbf{x}_t)}{\sqrt{\alpha_t}}}_{\text{"predicted } \mathbf{x}_0\text{"}} \right) + \underbrace{\sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta^{(t)}(\mathbf{x}_t)}_{\text{"direction pointing to } \mathbf{x}_t\text{"}} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}}$
- Training: The variational lower bound of DDIM is identical to the one of DDPM1
 - It is surprising since the forward/reverse formulation is totally different

1. Precisely, the bound is different, but the solution is identical under some assumption (though violated in practice)
Song et al. Denoising Diffusion Implicit Models. ICLR'21

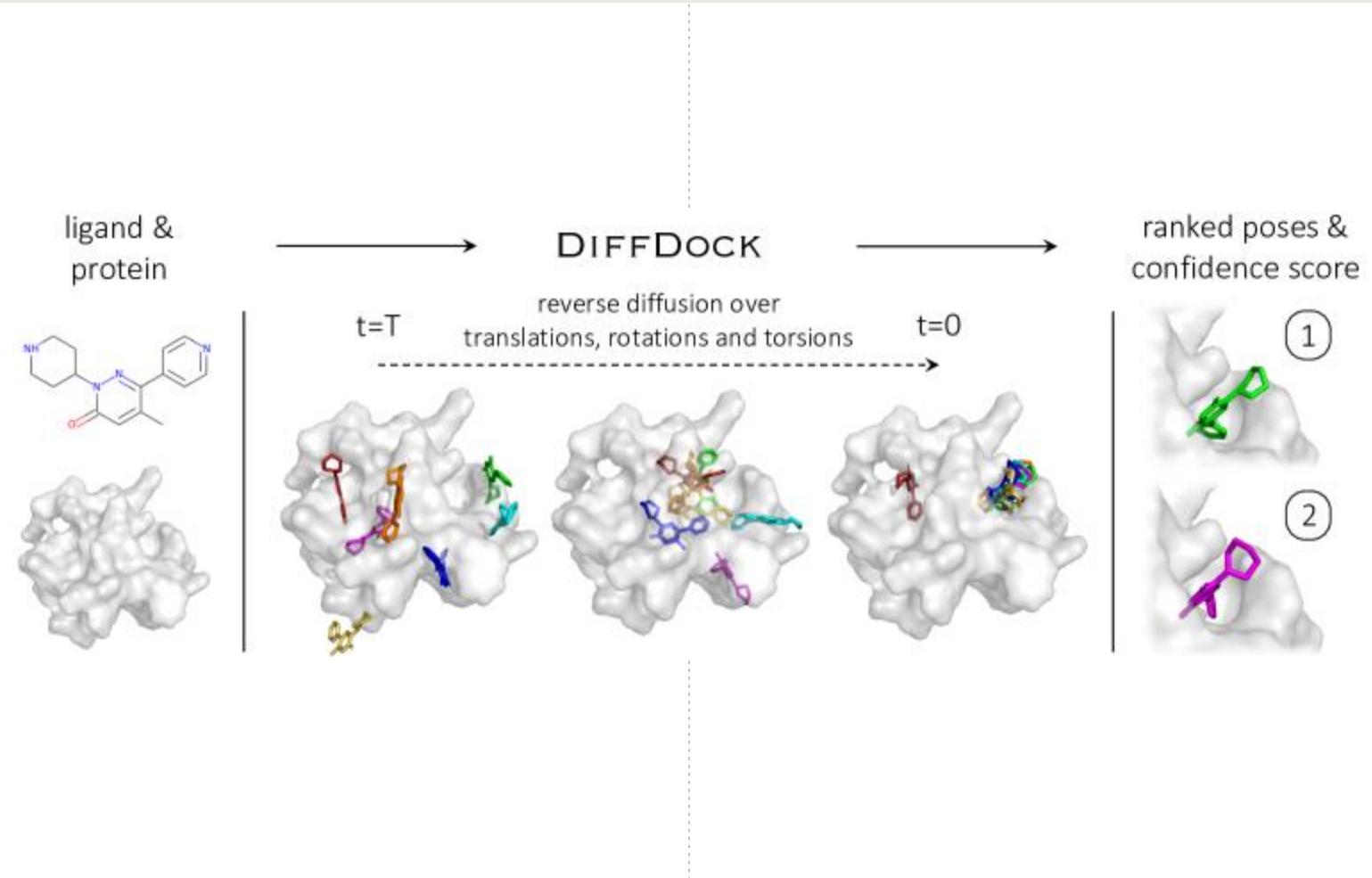
- Take Home Message

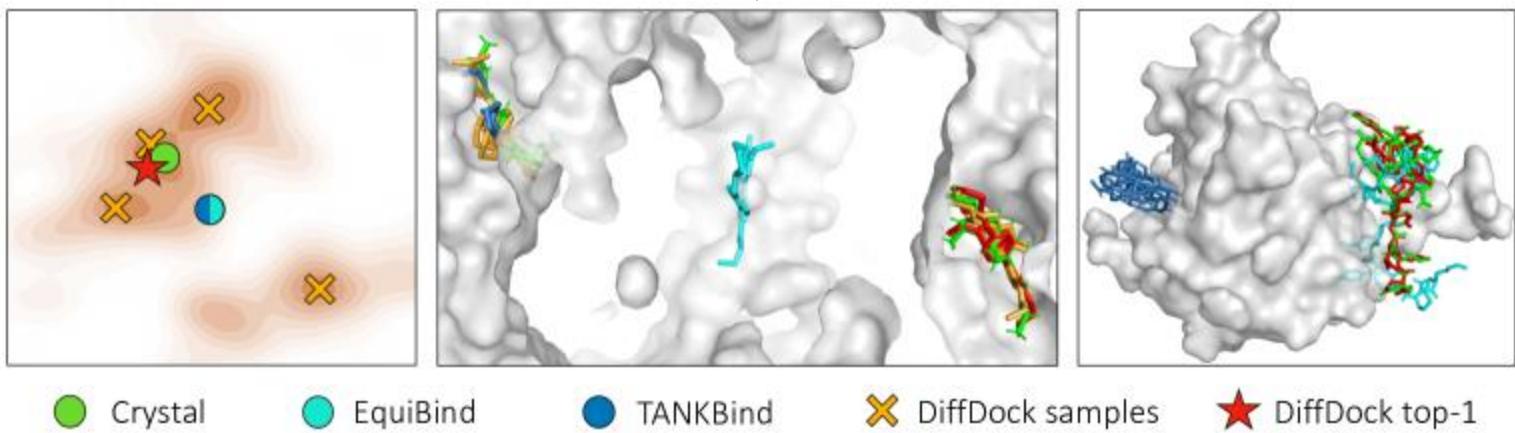
- **New golden era of generative models**
 - Competition of various approaches: GAN, VAE, flow, diffusion model1
 - Also, lots of hybrid approaches (e.g., score SDE = diffusion + continuous flow)
- **Which model to use?**
 - **Diffusion model** seems to be a nice option for **high-quality** generation
 - However, **GAN** is (currently) still a more practical solution which needs **fast sampling** (e.g., real-time apps.)



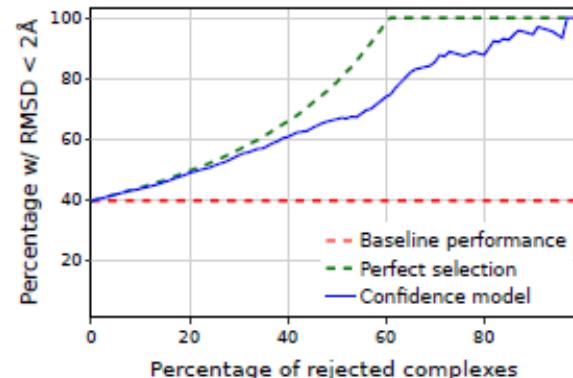
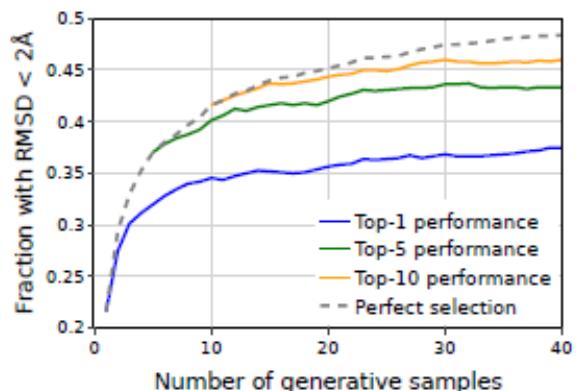
1. VAE also shows promising generation performance (see NVAE, very deep VAE)

- DiffDock

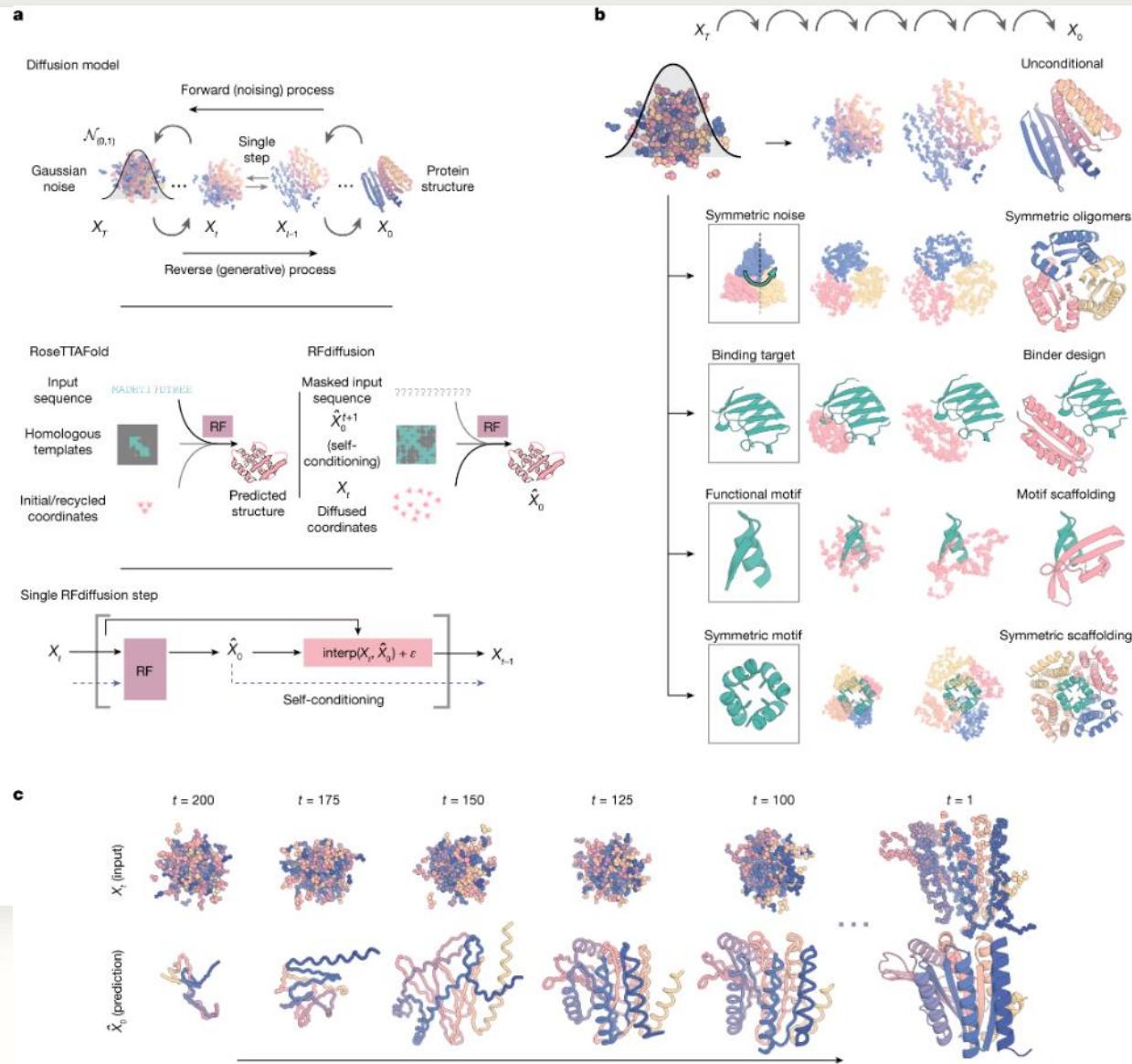


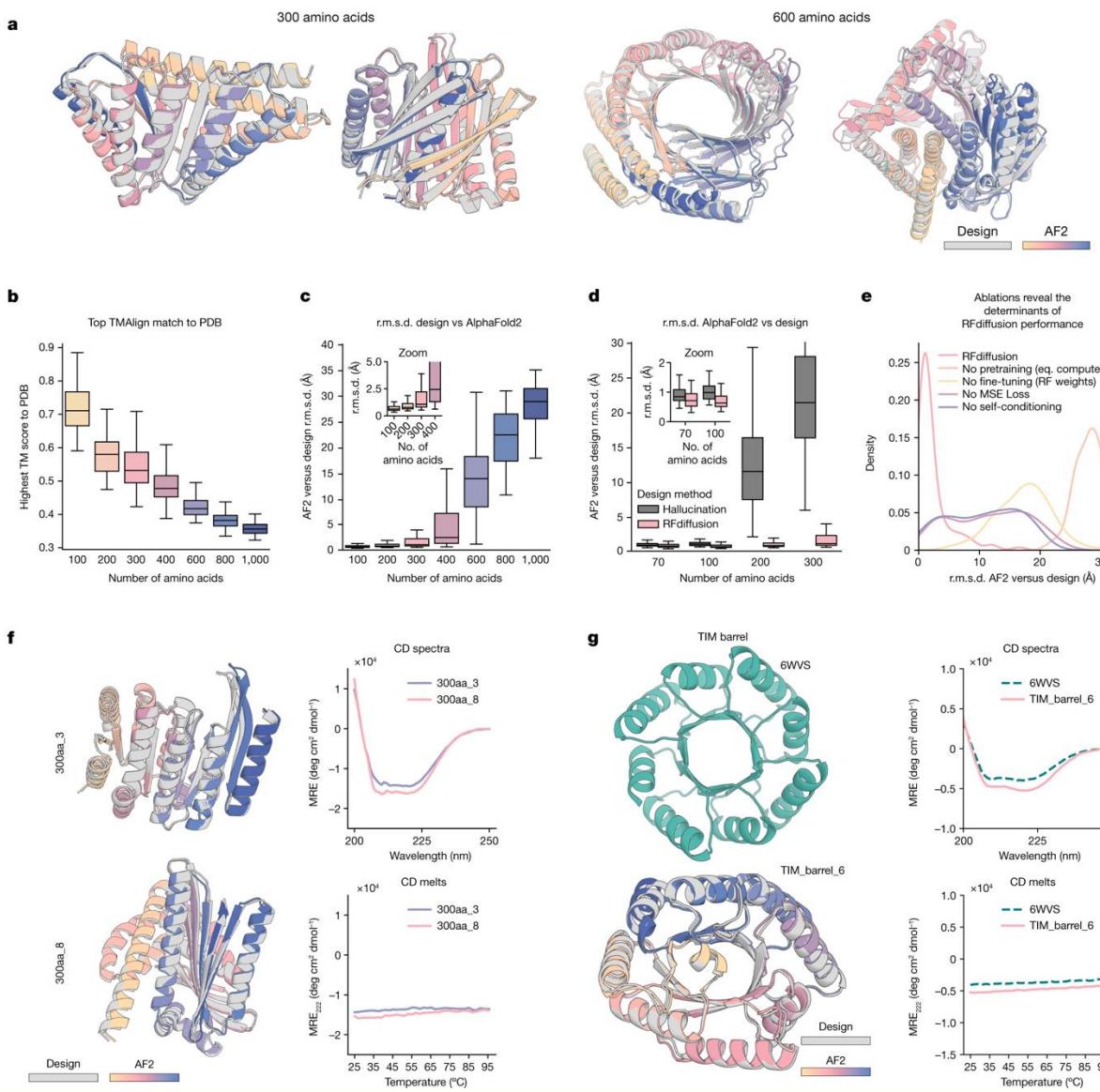


Method	Holo crystal proteins				Apo ESMFold proteins				Average Runtime (s)
	Top-1 RMSD %<2	Med.	Top-5 RMSD %<2	Med.	Top-1 RMSD %<2	Med.	Top-5 RMSD %<2	Med.	
GNINA	22.9	7.7	32.9	4.5	2.0	22.3	4.0	14.22	127
SMINA	18.7	7.1	29.3	4.6	3.4	15.4	6.9	10.0	126*
GLIDE	21.8	9.3	-	-	-	-	-	-	1405*
EQUIBIND	5.5	6.2	-	-	1.7	7.1	-	-	0.04
TANKBIND	20.4	4.0	24.5	3.4	10.4	5.4	14.7	4.3	0.7/2.5
P2RANK+SMINA	20.4	6.9	33.2	4.4	4.6	10.0	10.3	7.0	126*
P2RANK+GNINA	28.8	5.5	38.3	3.4	8.6	11.2	12.8	7.2	127
EQUIBIND+SMINA	23.2	6.5	38.6	3.4	4.3	8.3	11.7	5.8	126*
EQUIBIND+GNINA	28.8	4.9	39.1	3.1	10.2	8.8	18.6	5.6	127
DIFFDOCK (10)	35.0	3.6	40.7	2.65	21.7	5.0	31.9	3.3	10
DIFFDOCK (40)	38.2	3.3	44.7	2.40	20.3	5.1	31.3	3.3	40



- RFdiffusion







A dual diffusion model enables 3D molecule generation and lead optimization based on target pockets

Received: 13 February 2023

Accepted: 1 March 2024

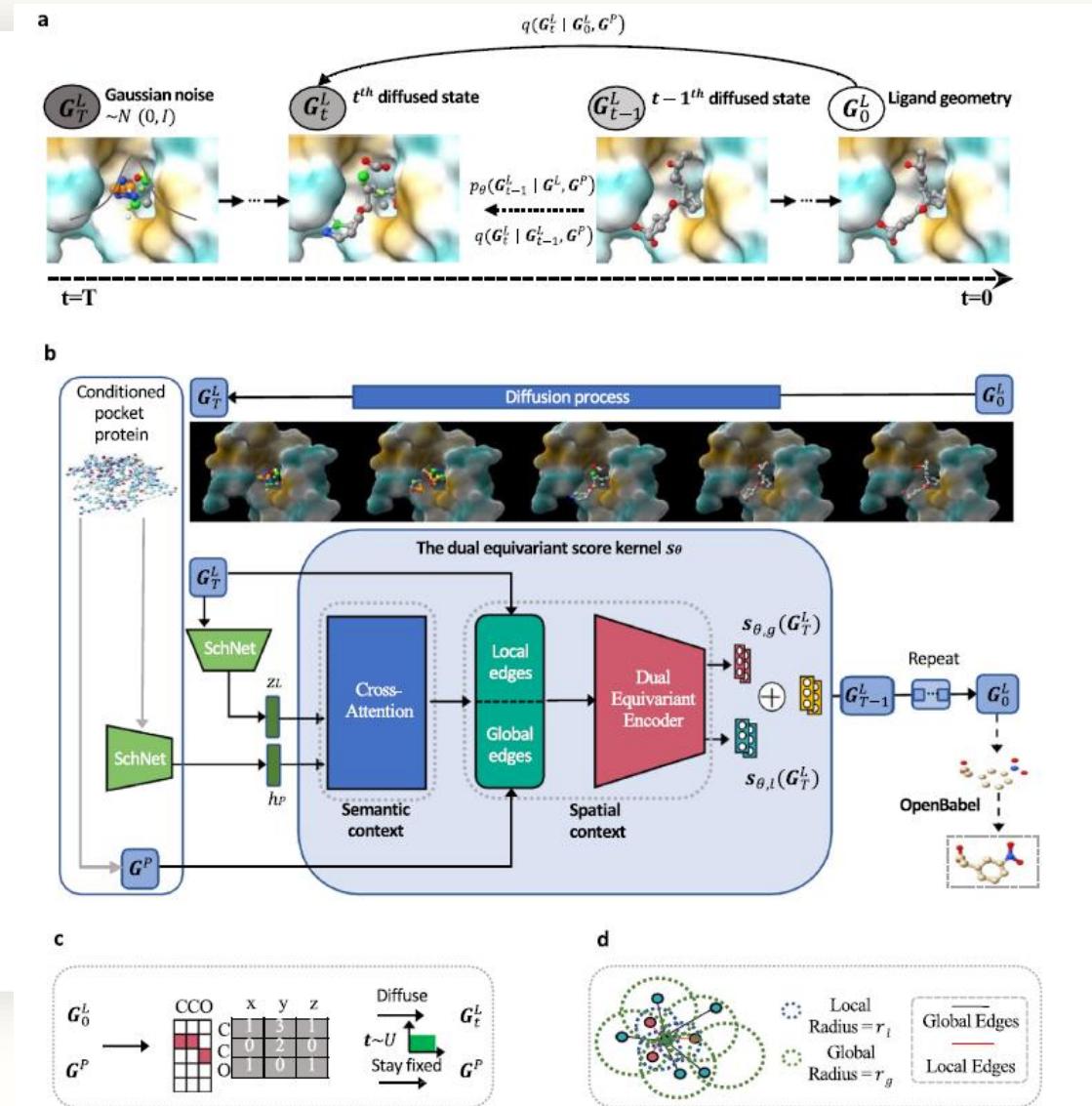
Published online: 26 March 2024

Check for updates

Lei Huang ^{1,2}, Tingyang Xu², Yang Yu ², Peilin Zhao², Xingjian Chen³, Jing Han⁴, Zhi Xie⁴, Hailong Li ⁴✉, Wenge Zhong⁴, Ka-Chun Wong ¹✉ & Hengtong Zhang ²✉

Structure-based generative chemistry is essential in computer-aided drug discovery by exploring a vast chemical space to design ligands with high

- PMDM: pocket based molecular diffusion model



- Alphafold 3

Article

Accurate structure prediction of biomolecular interactions with AlphaFold 3

<https://doi.org/10.1038/s41586-024-07487-w>

Received: 19 December 2023

Accepted: 29 April 2024

Published online: 8 May 2024

Open access

 Check for updates

Josh Abramson^{1,7}, Jonas Adler^{1,7}, Jack Dunger^{1,7}, Richard Evans^{1,7}, Tim Green^{1,7}, Alexander Pritzel^{1,7}, Olaf Ronneberger^{1,7}, Lindsay Willmore^{1,7}, Andrew J. Ballard¹, Joshua Bambrick², Sebastian W. Bodensteln¹, David A. Evans¹, Chia-Chun Hung², Michael O'Neill¹, David Relman¹, Kathryn Tunyasuvunakool¹, Zachary Wu¹, Akvilė Žemgulytė¹, Eirni Arvanitī³, Charles Beattie³, Ottavia Bertoli³, Alex Bridgland³, Alexey Cherepanov⁴, Miles Congreve⁴, Alexander I. Cowen-Rivers³, Andrew Cowle³, Michael Fligurnov³, Fabian B. Fuchs³, Hannah Gladman³, Rishabh Jain³, Yousuf A. Khan^{3,5}, Caroline M. R. Low⁴, Kuba Perlin³, Anna Potapenko³, Pascal Savy⁴, Sukhdeep Singh³, Adrian Stecula⁴, Ashok Thillaisundaram³, Catherine Tong⁴, Sergei Yakneen⁴, Ellen D. Zhong^{3,6}, Michal Zielinski³, Augustin Žídek³, Victor Bapst^{1,8}, Pushmeet Kohli^{1,8}, Max Jaderberg^{2,8}, Demis Hassabis^{1,2,8} & John M. Jumper^{1,8}

The introduction of AlphaFold 2¹ has spurred a revolution in modelling the structure of proteins and their interactions, enabling a huge range of applications in protein modelling and design^{2–6}. Here we describe our AlphaFold 3 model with a substantially updated diffusion-based architecture that is capable of predicting the joint structure of complexes including proteins, nucleic acids, small molecules, ions and modified residues. The new AlphaFold model demonstrates substantially improved accuracy over many previous specialized tools: far greater accuracy for protein–ligand interactions compared with state-of-the-art docking tools, much higher accuracy for protein–nucleic acid interactions compared with nucleic-acid-specific predictors and substantially higher antibody–antigen prediction accuracy compared with AlphaFold-Multimer v.2.3^{7,8}. Together, these results show that high-accuracy modelling across biomolecular space is possible within a single unified deep-learning framework.

