41st IEEE International Conference
on Data Engineering
HONG KONG SAR, CHINA | MAY 19 – 23, 2025

AIGC for Graphs: Current Techniques and Future Trends

Contributors: Hanchen Wang, Dawei Cheng, Ying Zhang and Wenjie Zhang

Contributors

Hanchen Wang is a Lecturer and ARC DECRA Fellow at Australian Artificial Intelligence Institute, University of Technology Sydney.

Dawei Cheng (程大伟) is currently an associate professor appointed at the School of Computer Science and Technology of Tongji University (同济大学), Shanghai, China.

Ying Zhang is a Professor at the School of Computer Science and Technology, Zhejiang Gongshang University.

Wenjie Zhang is a full Professor and ARC Future Fellow in the School of Computer Science and Engineering, the University of New South Wales, Australia.

41st IEEE International Conference on Data Engineering
HONG KONG SAR, CHINA | MAY 19 – 23, 2025

AIGC for Graphs: Current Techniques and Future Trends

Introduction

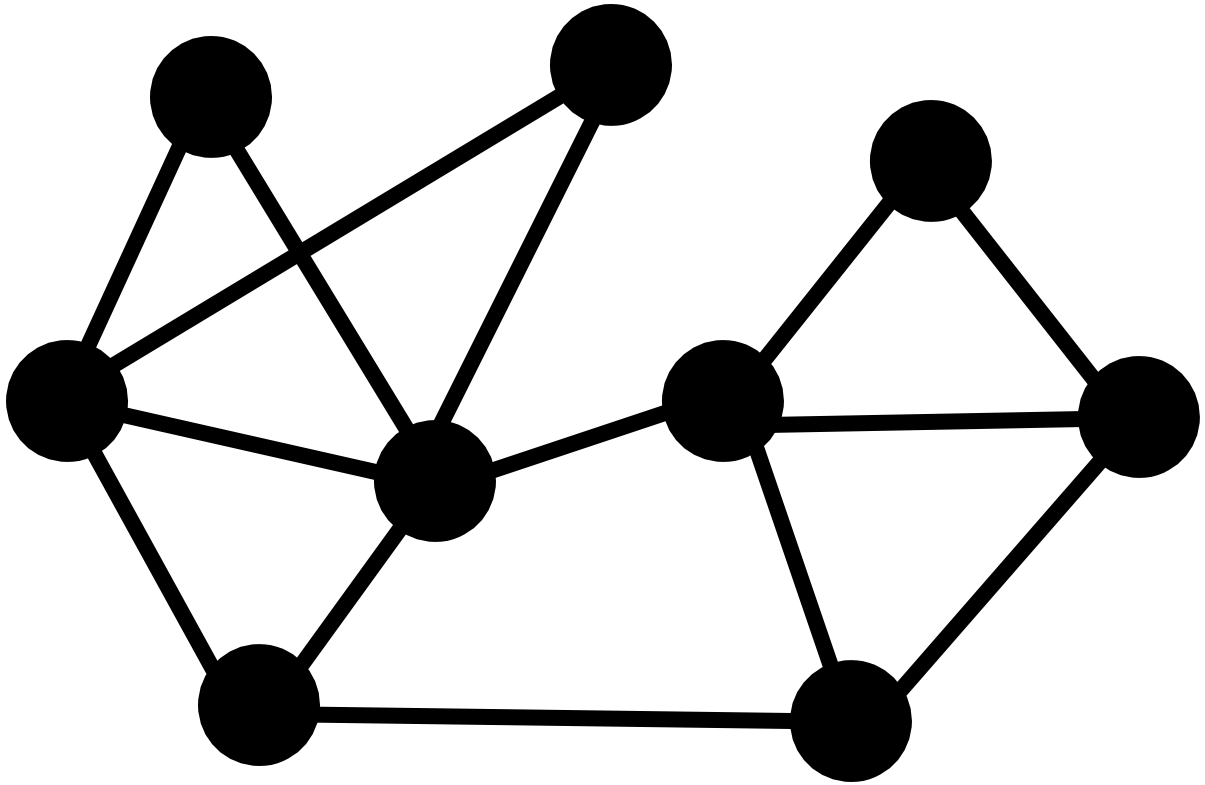
Speaker:
Hanchen Wang

Lecturer & ARC DECRA Fellow
Australian Artificial Intelligence Institute,
University of Technology Sydney

Contributors: Hanchen Wang, Dawei Cheng, Ying Zhang and Wenjie Zhang

Graphs

Graphs are a general language for describing and analyzing **entities with relations/interactions**.



Many types of data are graphs

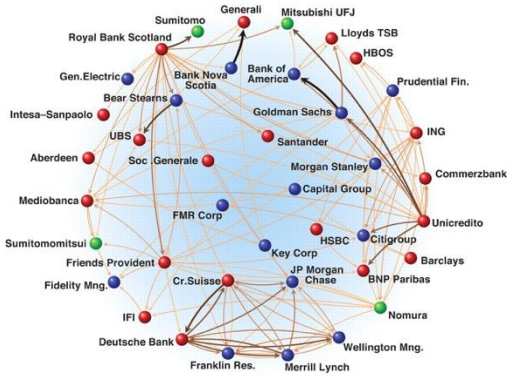
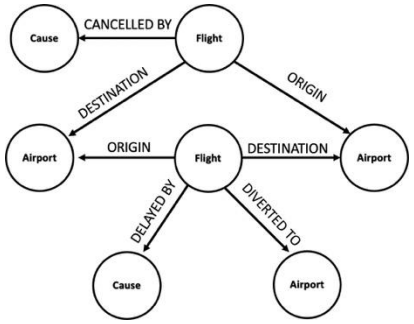


Image credit: Science

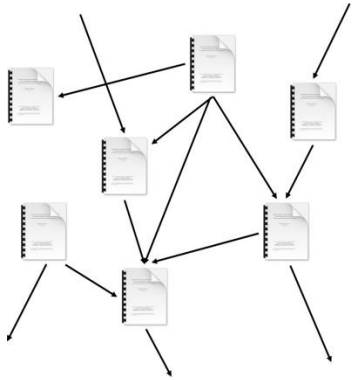
Economic Networks

Image credit: Lumen Learning

Communication Networks



Event Graphs



Citation Networks

Image credit: Missoula Current News

Internet

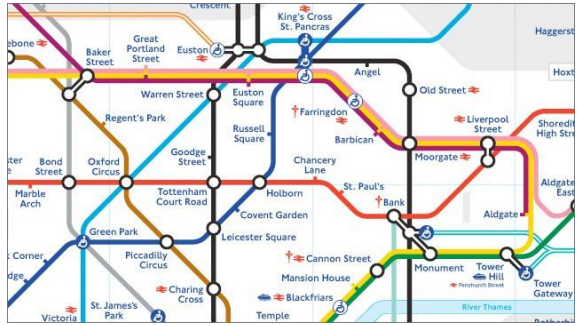


Image credit: visitlondon.com

Underground Networks

Many types of data are graphs

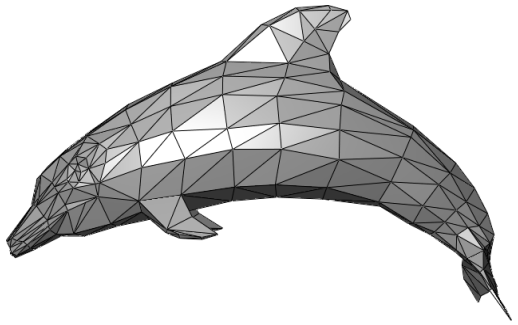
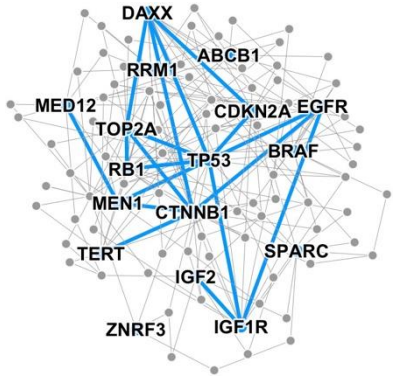


Image credit: Wikipedia

3D Shapes

Image credit: SalientNetworks

Computer Networks



Disease Pathways

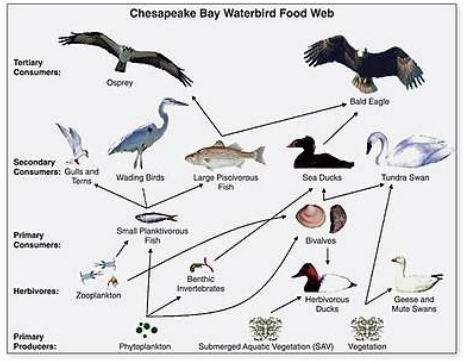


Image credit: Wikipedia

Food Webs

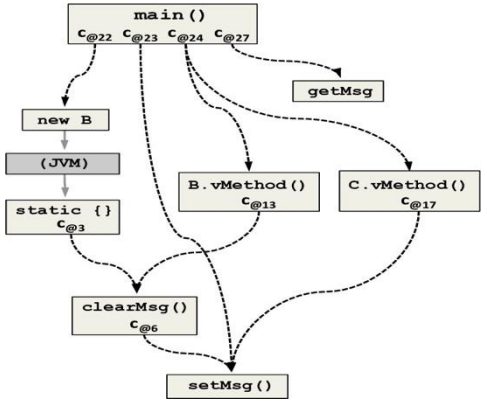


Image credit: ResearchGate

Code Graphs

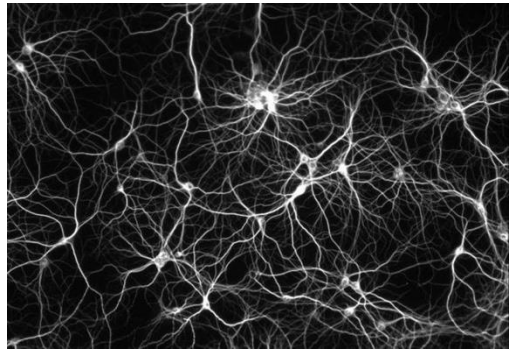
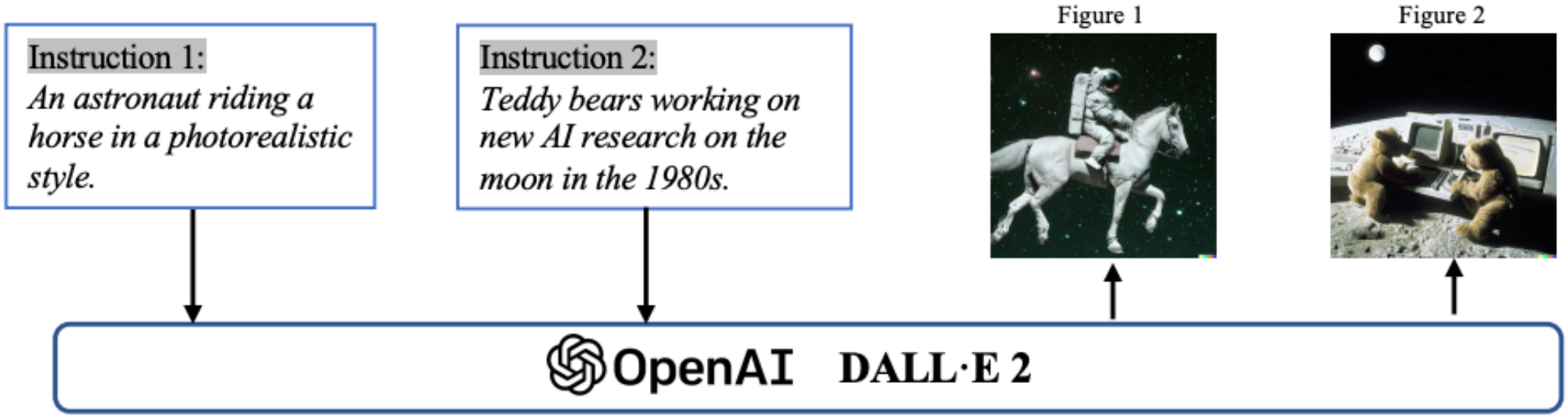


Image credit: The Conversation

Networks of Neurons

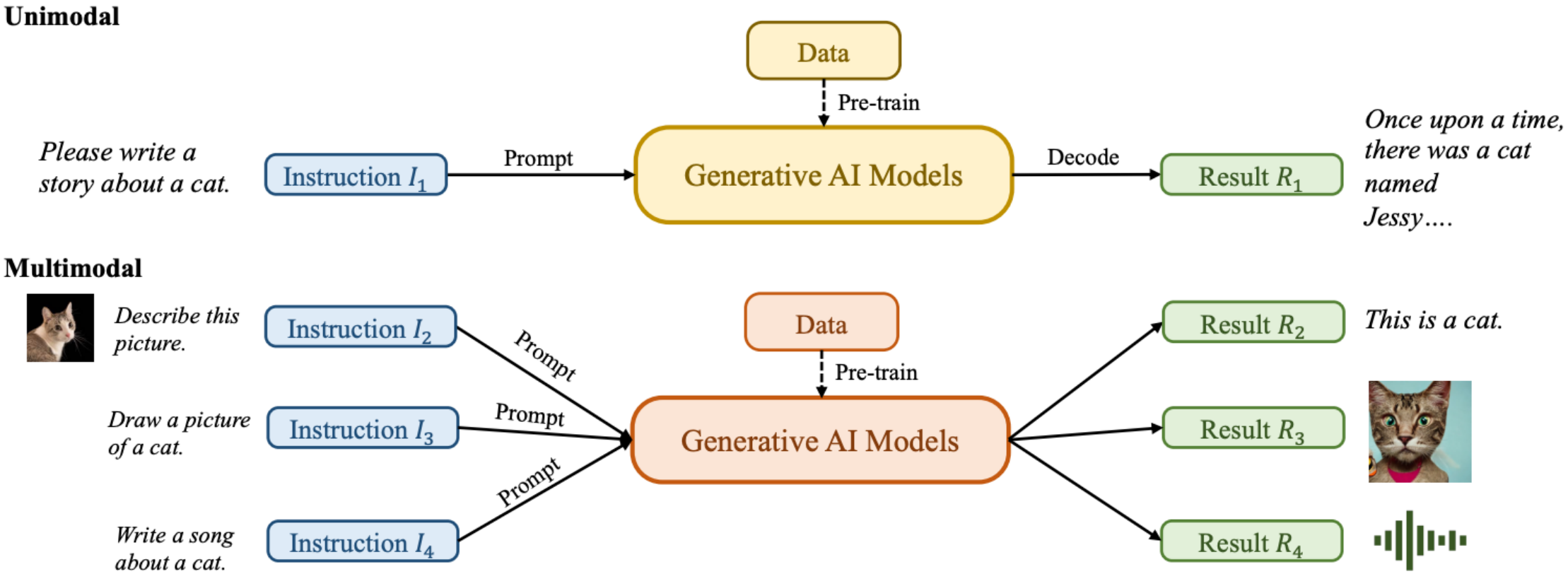
Artificial Intelligence-Generated Content (AIGC)

AIGC refers to content that is generated using **advanced Generative AI (GAI)** techniques, as opposed to being created by human authors, which can automate the creation of large amounts of content in a short amount of time.



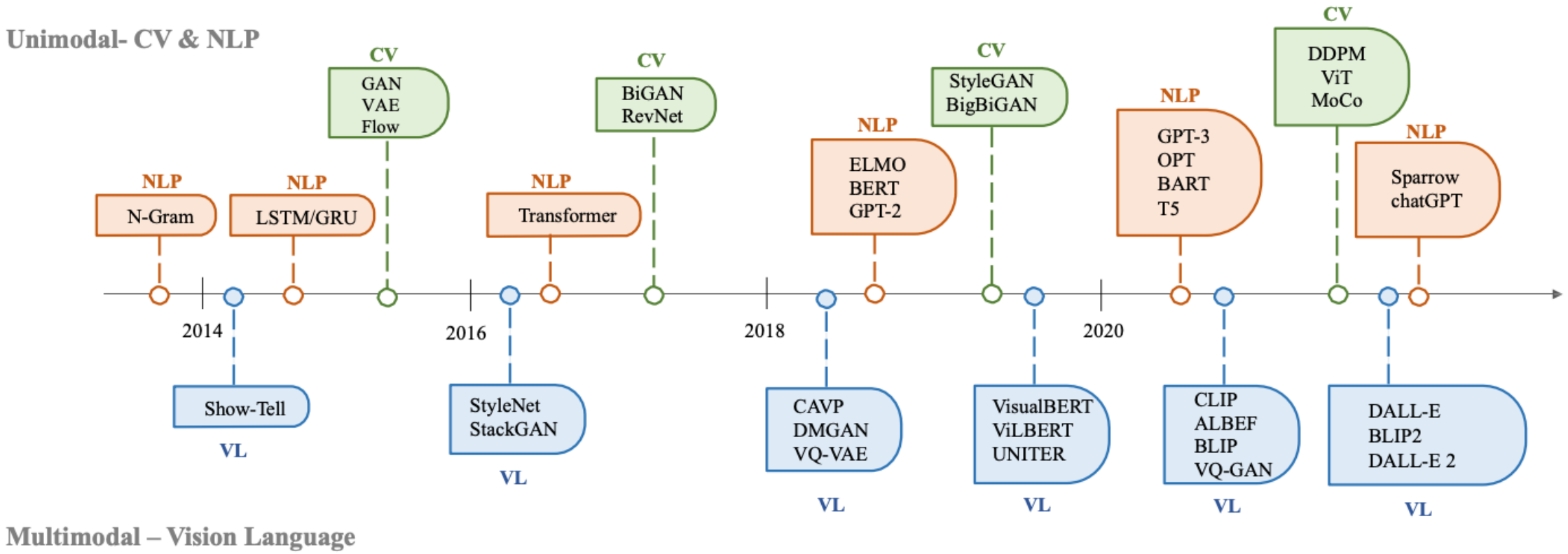
Cao, Yihan, et al. "A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt." *arXiv preprint arXiv:2303.04226* (2023).

Introduction about Generative AI Models



Cao, Yihan, et al. "A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt." *arXiv preprint arXiv:2303.04226* (2023).

History of Generative AI Models

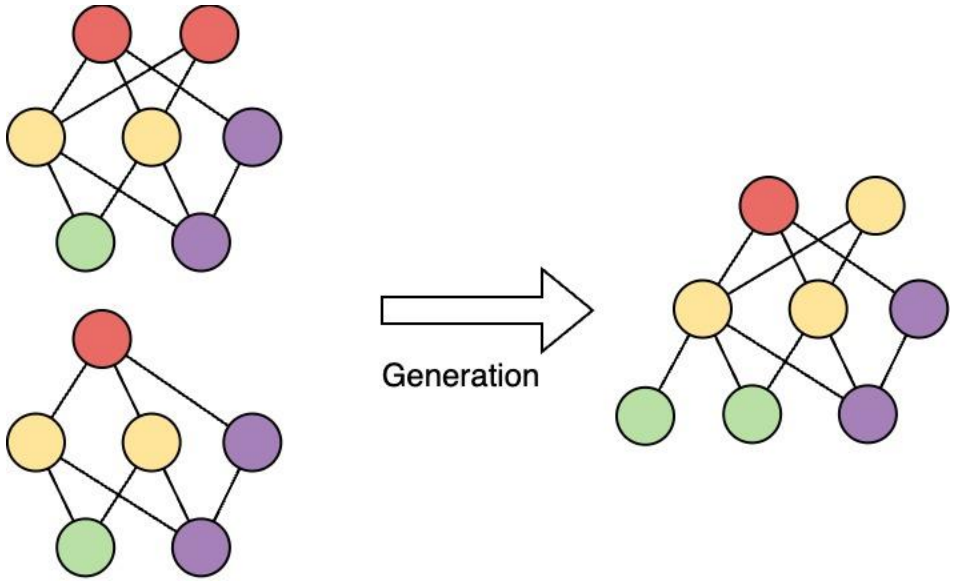


Cao, Yihan, et al. "A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt." *arXiv preprint arXiv:2303.04226* (2023).

Graph Generation

Definition of Graph Generation

Given a set of observed graphs {G}, **graph generation** aims to construct a generative model $p_\theta(G)$ to capture the distribution of these graphs, from which new graphs can be sampled $\hat{G} \sim p_\theta(G)$. The generation process can be conditioned on additional information s, i.e., conditional graph generation $\hat{G} \sim p_\theta(G|s)$ to apply specific constraints on the graph generation results.



Xiang, S., Wen, D., Cheng, D., Zhang, Y., Qin, L., Qian, Z. and Lin, X., 2022. General graph generators: experiments, analyses, and improvements. The VLDB Journal, pp.1-29.

Traditional Graph Generators

(Rule-based) Graph Generator

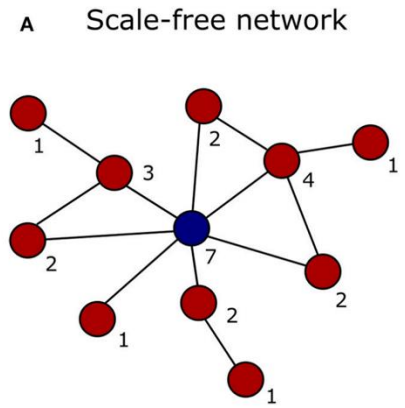
Number of nodes ***N***, edge probability ***P***

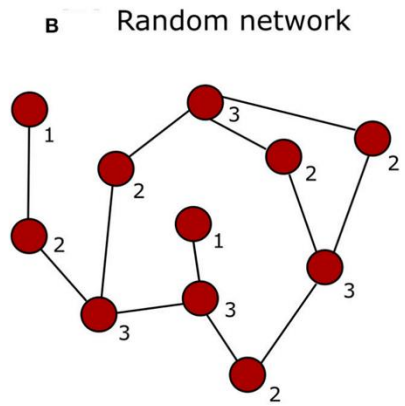
Number of nodes ***N***, edges per node ***m***

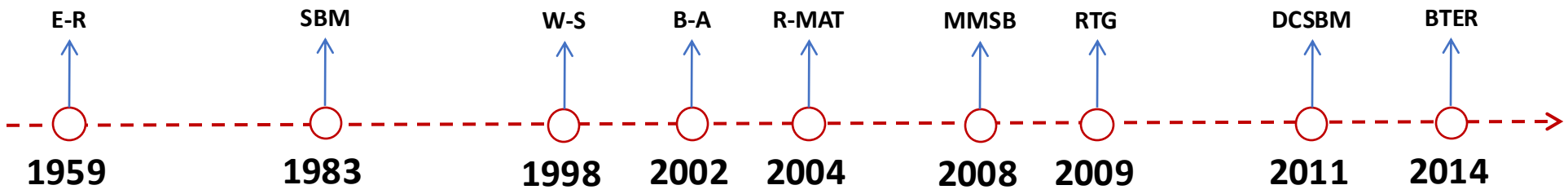
B-A Graph Model

E-R Graph Model

Graph Generator

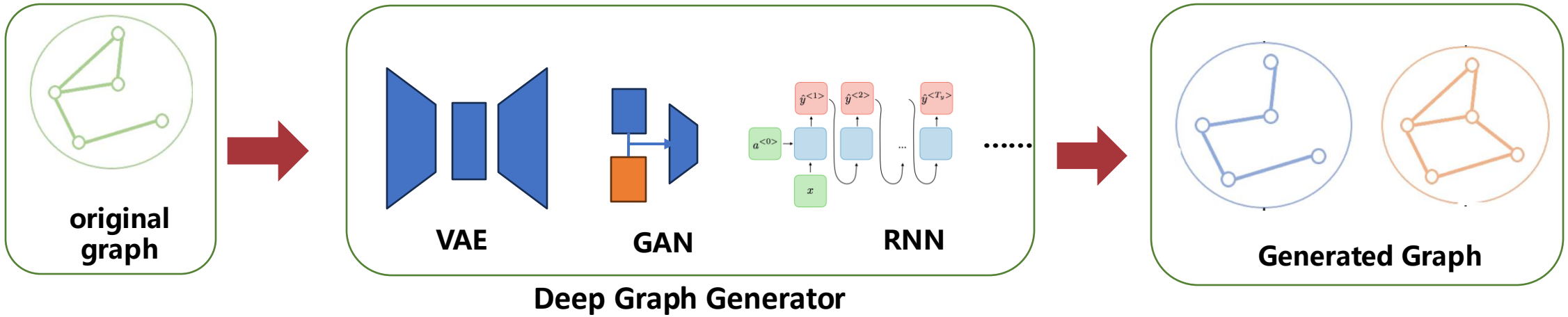


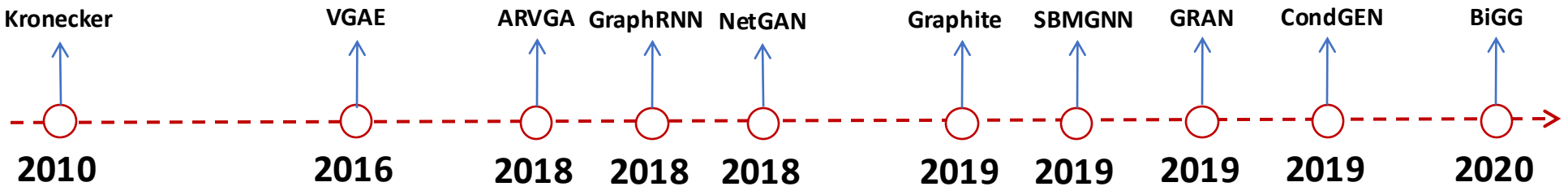




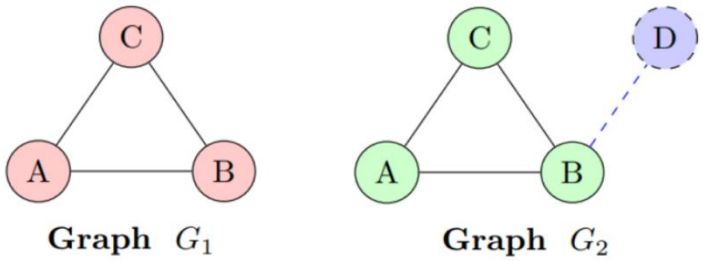
Learning-based Graph Generators

(Learning-based) Graph Generator

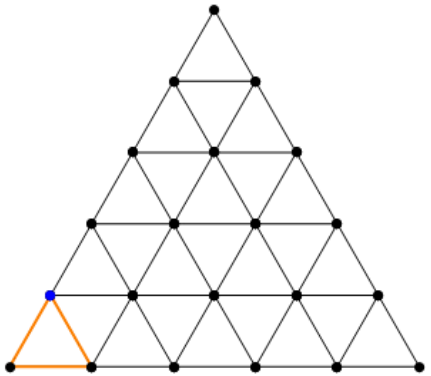




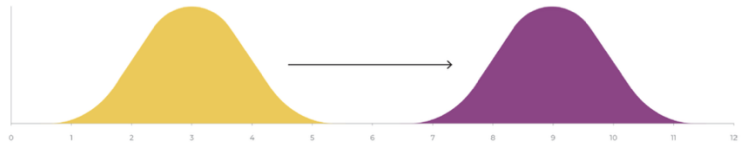
Graph Similarity



1. Graph Edit Distance(GED)



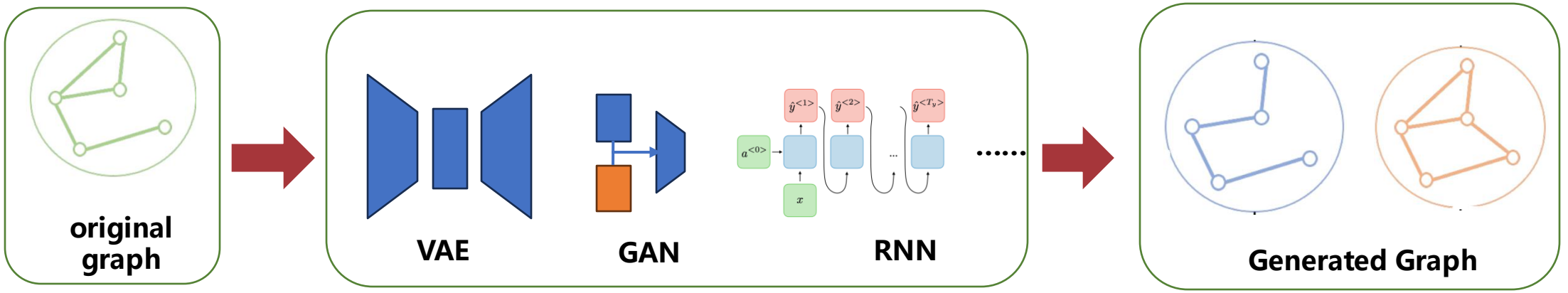
2. Graph Statistics
(e.g., triangle counts)



3. Node Distribution
(e.g., Wasserstein Distance)

Similarity-based Graph Generation

Generated Graph Data: According to the data format, generated data can be divided into: text data, image data, table data, and graph structure data. Simulated graph structure data has the good properties of the original data and does not contain detailed information of the original data.



Similarity-based deep graph generators: these algorithms have the following advantages: (1) the generation performance is significantly improved; (2) input data in different formats can be simulated end-to-end; (3) the implicit data distribution on the graph can be effectively captured.

Challenge:

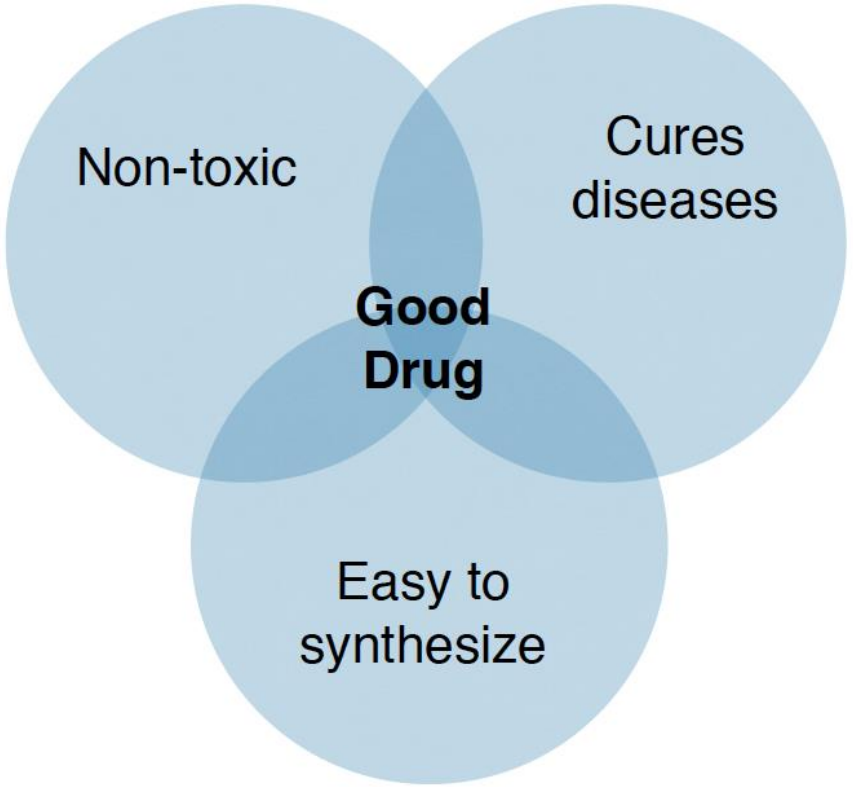
- ➢ can hardly simulate large-scale graph data
- ➢ real-world graph data have community structure

[1] Xiang, S., Wen, D., Cheng, D., Zhang, Y., Qin, L., Qian, Z. and Lin, X., 2022. General graph generators: experiments, analyses, and improvements. The VLDB Journal, pp.1-29.

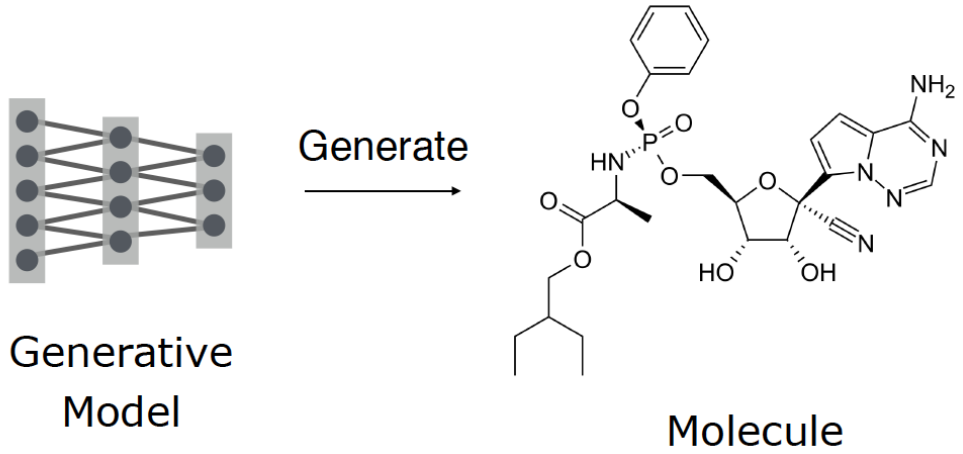
Application: Generated Molecule Graph

Drug Discovery: finding molecules with desired chemical properties.

A good drug needs to satisfy multiple objectives:



- The scale of potential drug-like molecules: $10^{33} \sim 10^{60}$
- The scale of existing chemical database: 10^{6}
- A huge gap!

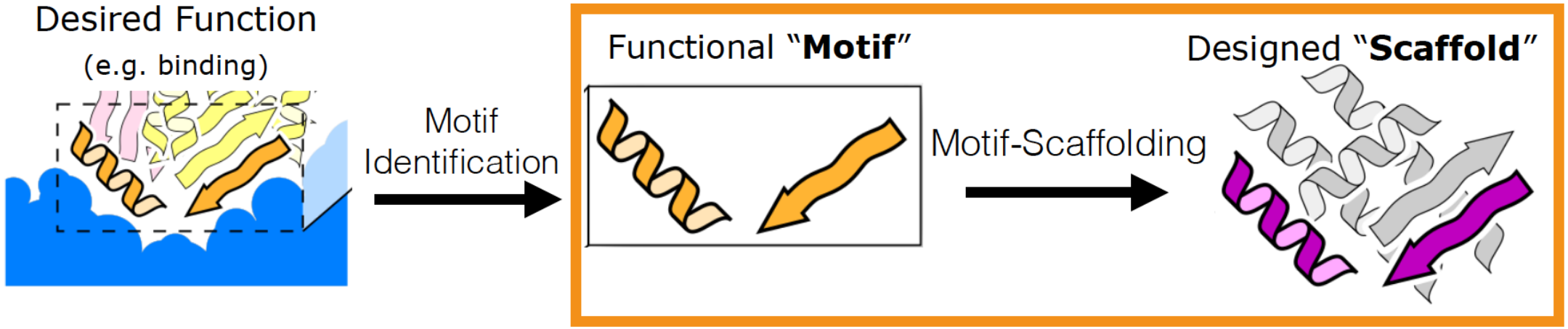


Jin, Wengong, Regina Barzilay, and Tommi Jaakkola. "Multi-objective molecule generation using interpretable substructures." *ICML*. PMLR, 2020.
Zang, Chengxi, and Fei Wang. "Moflow: an invertible flow model for generating molecular graphs." *Proceedings of the 26th ACM SIGKDD*. 2020.

Application: Generated Protein Structure

Protein design is also a fundamental application of graph generation.

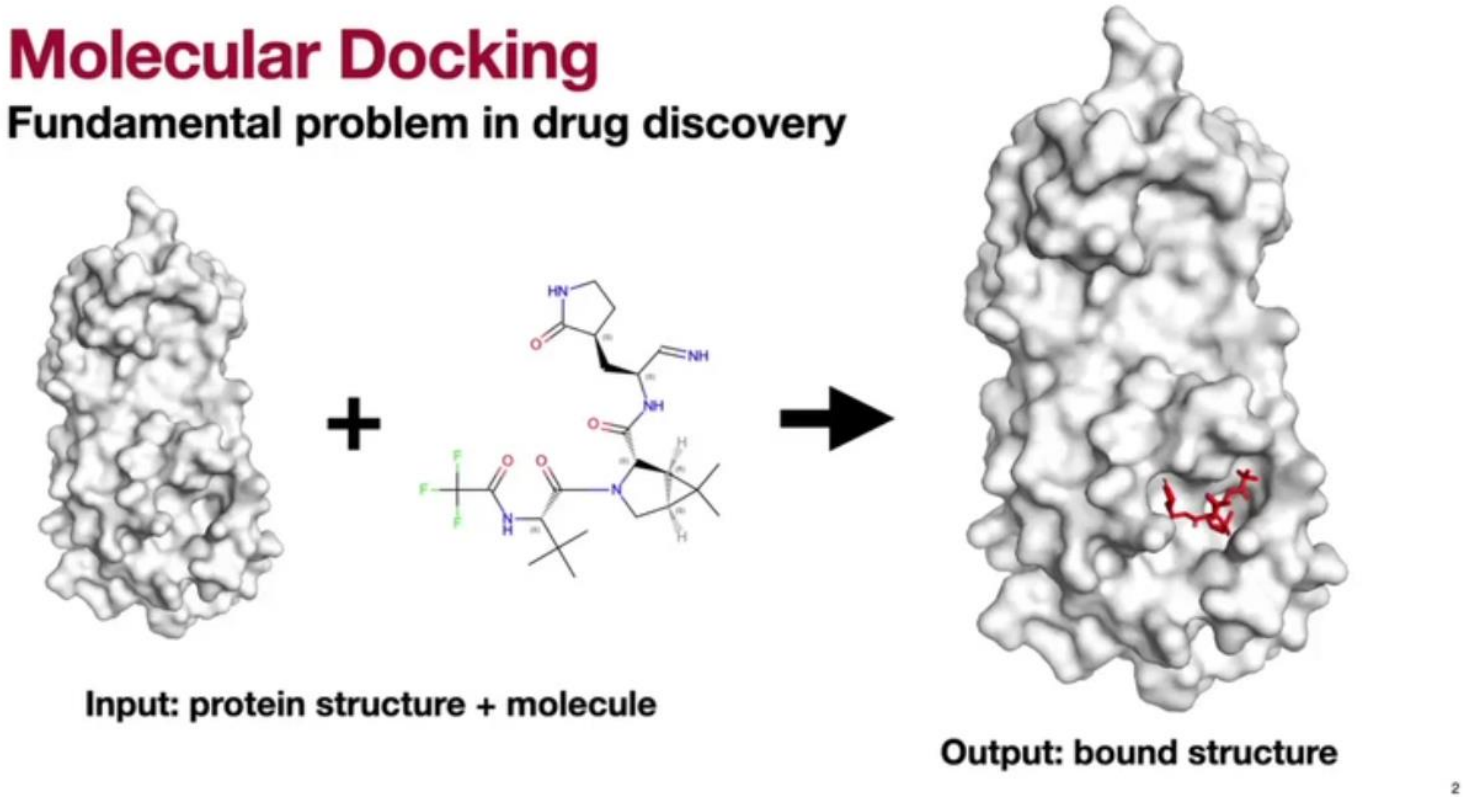
Motif-Scaffolding: design of protein structures ("scaffolds") that can support a specific functional motif (a short, conserved sequence or structural element with a defined biological function).



Trippe, Brian L., et al. "Diffusion Probabilistic Modeling of Protein Backbones in 3D for the motif-scaffolding problem." *ICLR 2023*.

Application: Generated Protein Structure

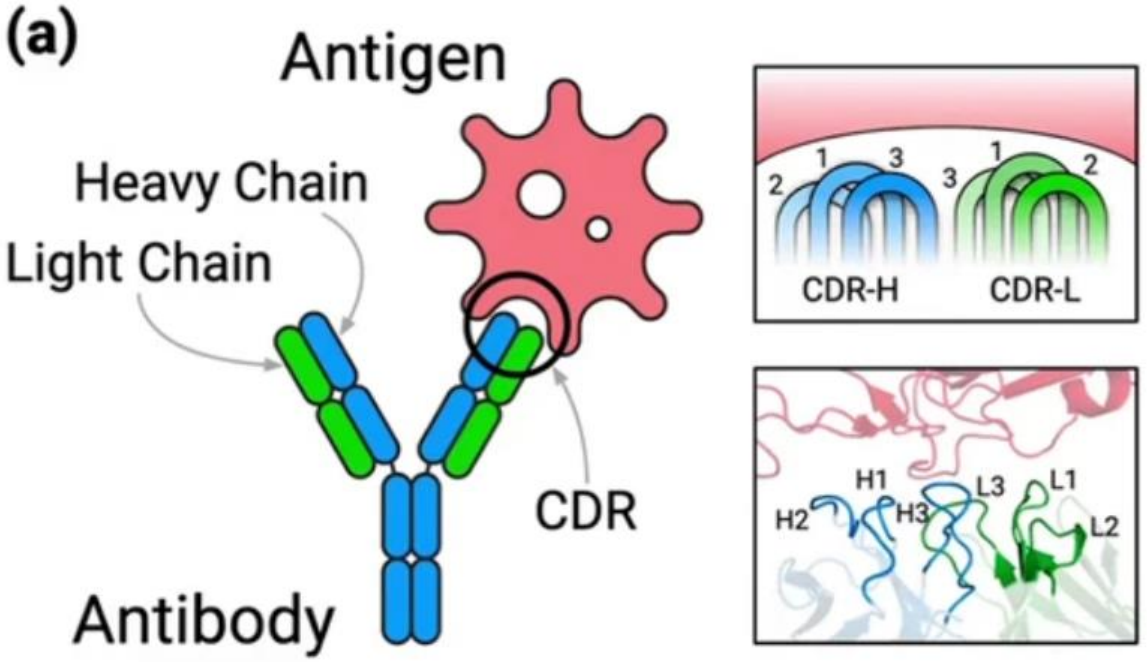
Molecule docking: predict how two molecules (e.g., a protein and a ligand) interact and bind to each other. It is widely used in **drug discovery**, **protein-protein interaction studies**, and **structural biology** to identify potential binding poses, affinity, and binding sites.



Corso, Gabriele, et al. "DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking." *International Conference on Learning Representations (ICLR 2023)*. 2023.

Application: Generated Protein Structure

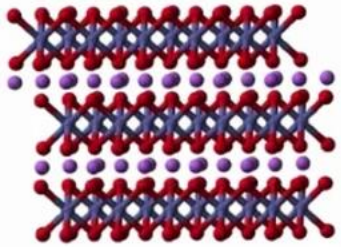
Antibody Design: Design of antibodies (a class of proteins produced by the immune system) to bind specific targets (antigens) with high affinity and specificity. This field combines **immunology, bioinformatics, and protein engineering** to develop therapeutic antibodies, diagnostic tools, and research reagents.



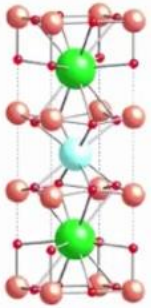
Luo, Shitong, et al. "Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures." *NeurIPS* (2022): 9754-9767.

Application: Generated Material Structure Graph

Periodic Material Generation: Design of materials with repeating structural patterns (periodicity) at the atomic, molecular, or mesoscopic scale. Graph generation is used to discover or optimize materials with desired properties, such as mechanical strength, thermal conductivity, or electronic behavior.

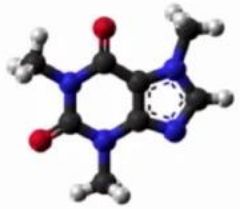


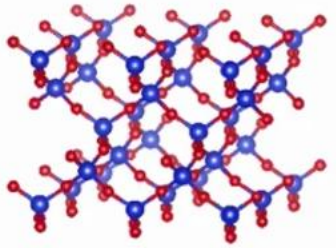
$LiCoO_2$
Cathode material for Li-ion battery
2019 Nobel Prize in Chemistry



$YBa_2Cu_3O_7$
First high-T superconductor
1987 Nobel Prize in Physics

Small molecules	Materials
• Non-periodic, finite	• Periodic, infinite
• 5 – 10 elements	• All 94 naturally occurring elements
• Simple 2D graph	• Graph difficult to define
• Relatively simple valency rules	• No general valency rules





3D material structures must be directly generated, rather than relying on intermediate graphs.

Xie, Tian, et al. "Crystal Diffusion Variational Autoencoder for Periodic Material Generation." *International Conference on Learning Representations 2022*.

Outline

- Similarity-based Graph Generation
- Function-driven Graph Generation
- Current Trend and Future Directions

41st IEEE International Conference
on Data Engineering
HONG KONG SAR, CHINA | MAY 19 – 23, 2025

AIGC for Graphs: Current Techniques and Future Trends

Similarity-based Graph Generation

Speaker:
Hanchen Wang

Contributors: Hanchen Wang, Dawei Cheng, Ying Zhang and Wenjie Zhang

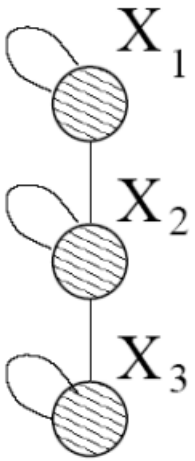
UTS UNIVERSITY OF TECHNOLOGY SYDNEY
TONGJI UNIVERSITY
ZHEJIANG GONGSHANG UNIVERSITY
UNSW THE UNIVERSITY OF NEW SOUTH WALES

Similarity-Based Graph Generation

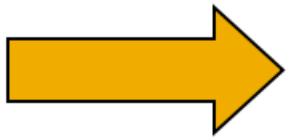
- **Recursive: Kronecker**
- Autoregressive: GraphRNN
- Random Walk: NetGAN
- Diffusion Model: DiGress
- Dynamic Graph Generation
- Similarity and Graph Generation

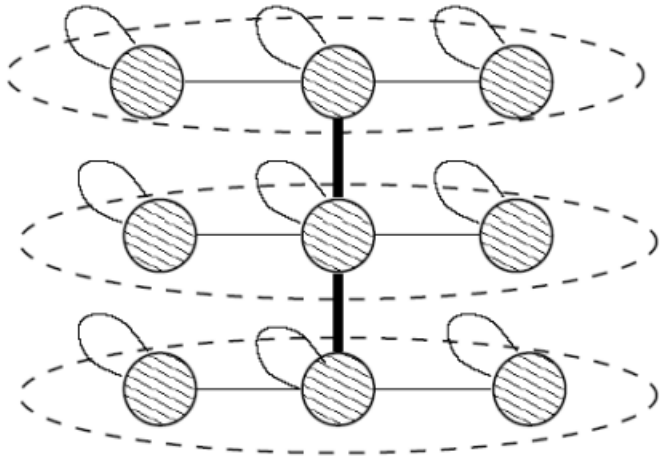
Recursive: Kronecker

Decomposing Graph Generation into Recursive Expansion:



Initial graph

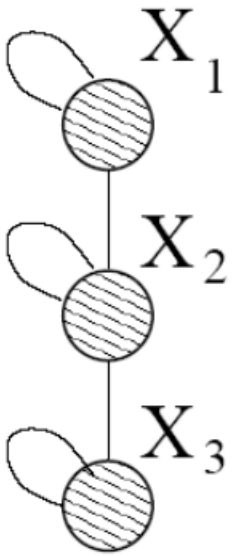


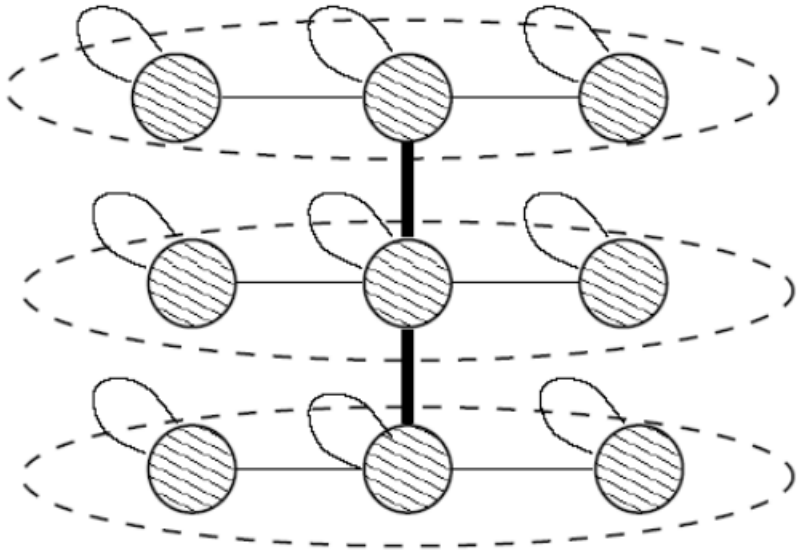


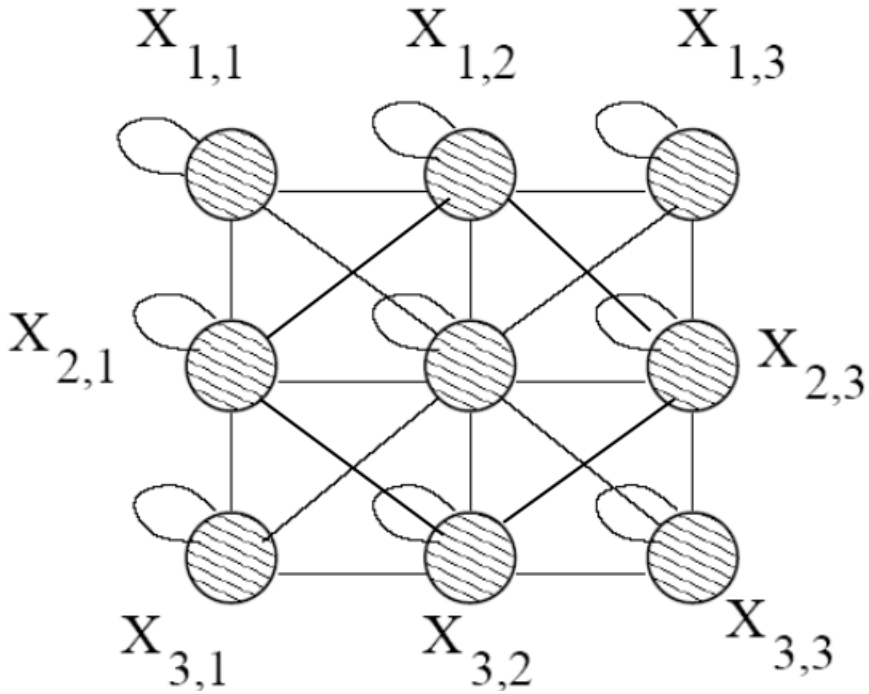
Recursive expansion

Recursive: Kronecker

Decomposing Graph Generation into Recursive Expansion:







Recursive: Kronecker

Decomposing Graph Generation into Recursive Expansion:

1	1	0
1	1	1
0	1	1

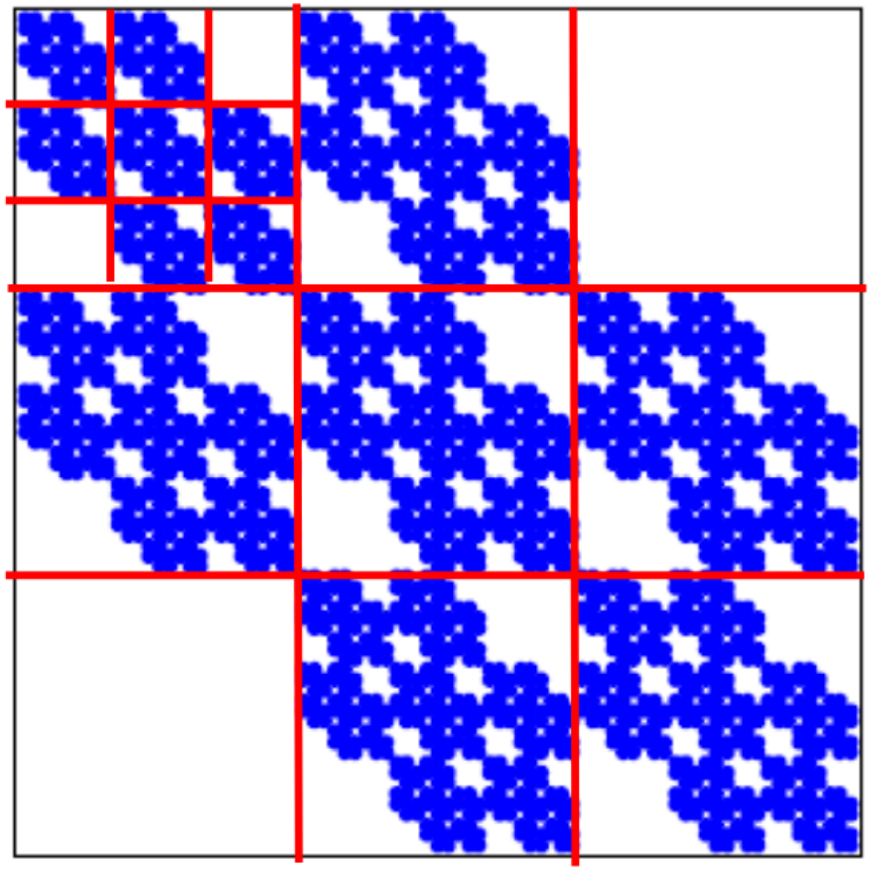
K_1

3 x 3

K_1	K_1	0
K_1	K_1	K_1
0	K_1	K_1

$K_2 = K_1 \otimes K_1$

9 x 9



81 x 81 adjacency matrix

Recursive: Kronecker

Decomposing Graph Generation into Recursive Expansion:

Kronecker product of matrices A and B is given by

$$\underset{N \times M}{\mathbf{C}} = \underset{K \times L}{\mathbf{A} \otimes \mathbf{B}} \doteq \begin{pmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \dots & a_{1,m}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \dots & a_{2,m}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}\mathbf{B} & a_{n,2}\mathbf{B} & \dots & a_{n,m}\mathbf{B} \end{pmatrix}$$

$N*K \times M*L$

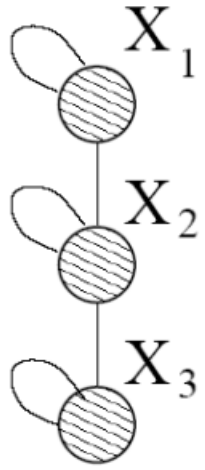
Recursive: Kronecker

Decomposing Graph Generation into Recursive Expansion:

- **Kronecker graph:** a growing sequence of graphs by iterating the **Kronecker product:**

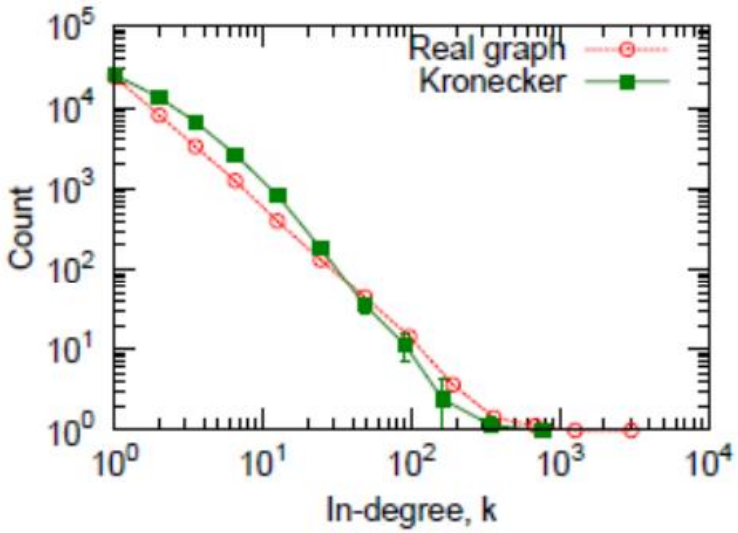
$$K_1^{[m]} = K_m = \underbrace{K_1 \otimes K_1 \otimes \ldots K_1}_{m \ times} = K_{m-1} \otimes K_1$$

1	1	0
1	1	1
0	1	1

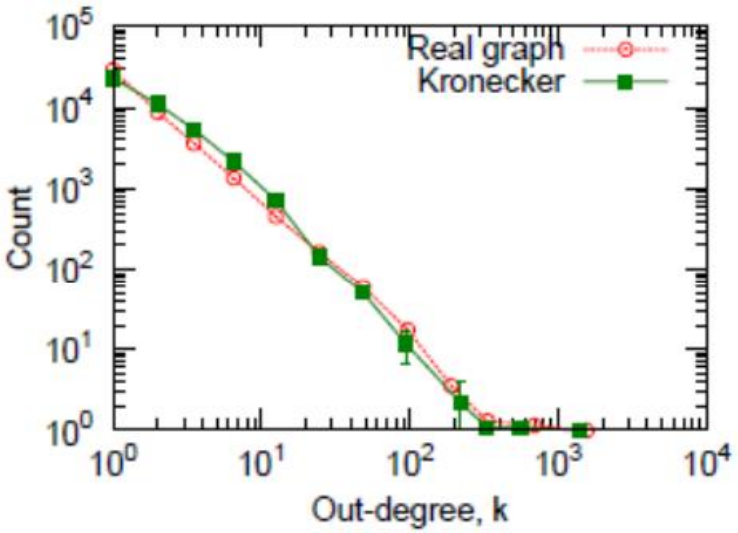


K_1

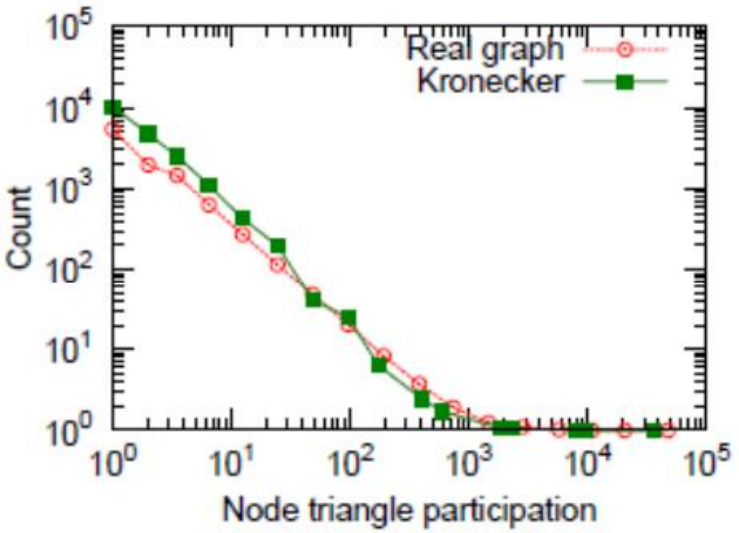
Recursive: Kronecker



(a) In-Degree



(b) Out-degree



(c) Triangle participation

Similarity-Based Graph Generation

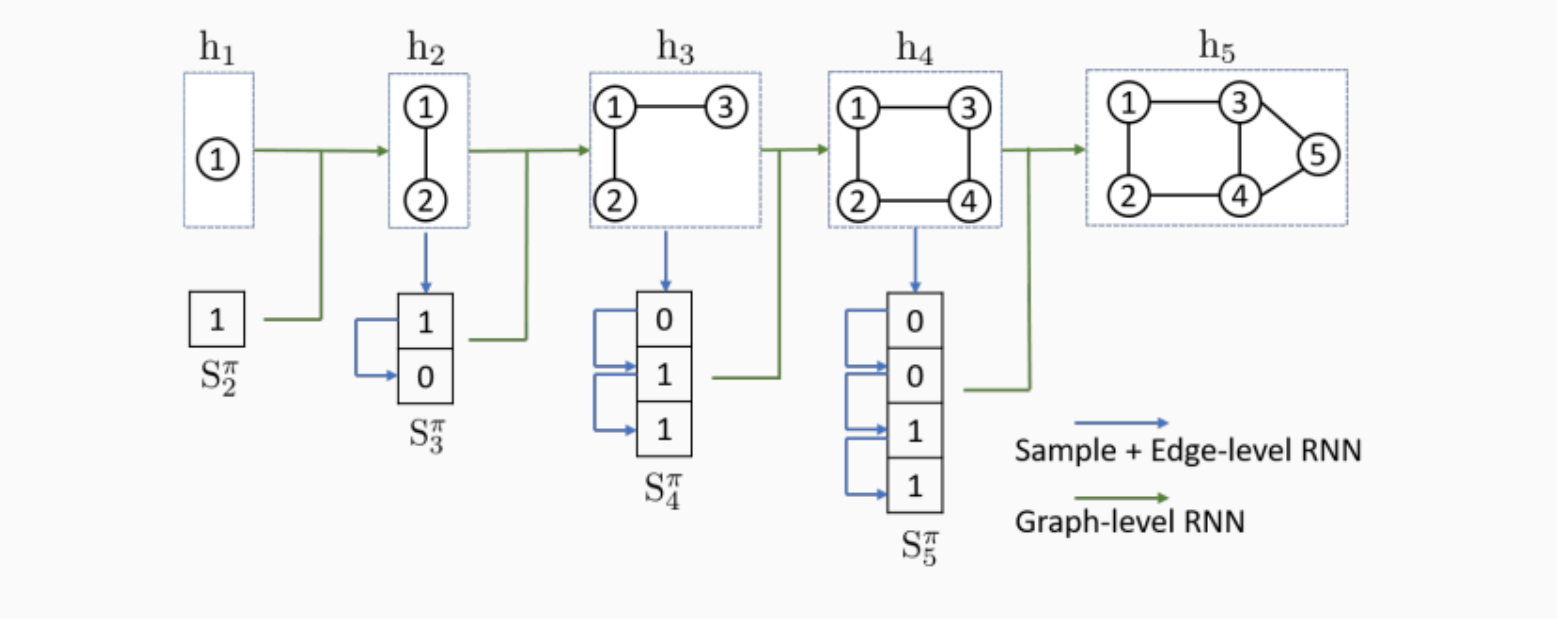
- Recursive: Kronecker
- **Autoregressive: GraphRNN**
- Random Walk: NetGAN
- Diffusion Model: DiGress
- Dynamic Graph Generation
- Similarity and Graph Generation

Autoregressive: GraphRNN

Decomposing Graph Generation into two RNNs:

- Graph-level: generates sequence of nodes

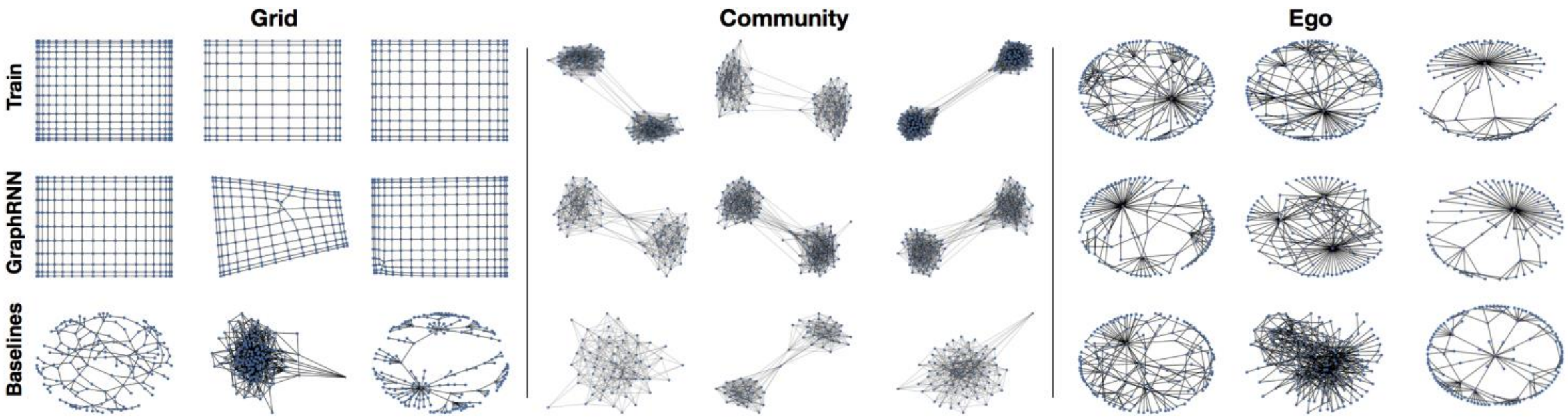
- Edge-level: generates sequence of edges for each new node



You, Jiaxuan, et al. "Graphrnn: Generating realistic graphs with deep auto-regressive models." International conference on machine learning. PMLR, 2018.

Autoregressive: GraphRNN

Visualization of input graphs and generated graphs



You, Jiaxuan, et al. "Graphrnn: Generating realistic graphs with deep auto-regressive models." International conference on machine learning. PMLR, 2018.

Autoregressive: GraphRNN

Quantitative Comparison on Generative Performance

Table 1. Comparison of GraphRNN to traditional graph generative models using MMD. $(\max(|V|), \max(|E|))$ of each dataset is shown.

	Community (160,1945)			Ego (399,1071)			Grid (361,684)			Protein (500,1575)		
	Deg.	Clus.	Orbit	Deg.	Clus.	Orbit	Deg.	Clus.	Orbit	Deg.	Clus.	Orbit
E-R	0.021	1.243	0.049	0.508	1.288	0.232	1.011	0.018	0.900	0.145	1.779	1.135
B-A	0.268	0.322	0.047	0.275	0.973	0.095	1.860	0	0.720	1.401	1.706	0.920
Kronecker	0.259	1.685	0.069	0.108	0.975	0.052	1.074	0.008	0.080	0.084	0.441	0.288
MMSB	0.166	1.59	0.054	0.304	0.245	0.048	1.881	0.131	1.239	0.236	0.495	0.775
GraphRNN-S	0.055	0.016	0.041	0.090	**0.006**	0.043	0.029	10^{-5}	0.011	0.057	**0.102**	**0.037**
GraphRNN	**0.014**	**0.002**	**0.039**	**0.077**	0.316	**0.030**	$\mathbf{10^{-5}}$	**0**	$\mathbf{10^{-4}}$	**0.034**	0.935	0.217

You, Jiaxuan, et al. "Graphrnn: Generating realistic graphs with deep auto-regressive models." International conference on machine learning. PMLR, 2018.

Similarity-Based Graph Generation

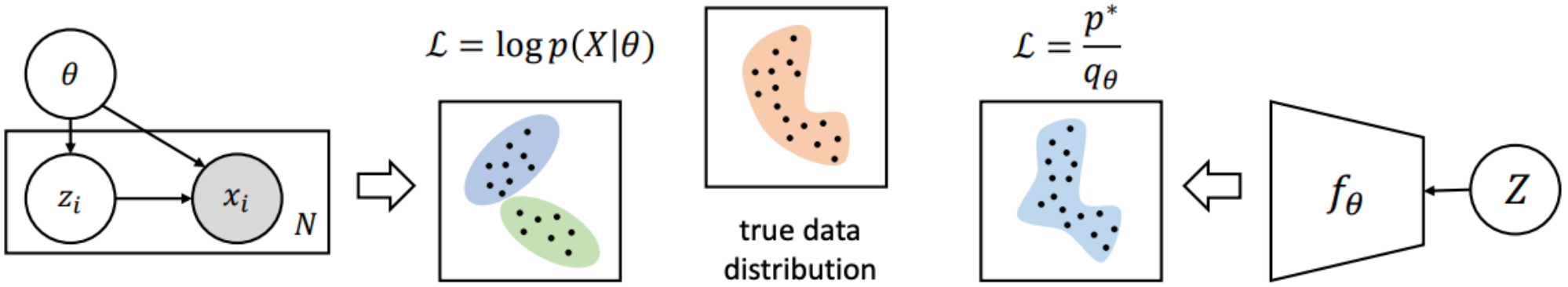
- Recursive: Kronecker
- Autoregressive: GraphRNN
- **Random Walk: NetGAN**
- Diffusion Model: DiGress
- Dynamic Graph Generation
- Similarity and Graph Generation

Random Walk: NetGAN

Explicit vs. implicit models

- **Explicit models** have a parametric specification of the data distribution
- Observe patterns and manually specify a model to capture them
- Learn via MLE, …

- **Implicit models** define a stochastic process that directly generates data
- Likelihood free: learn by comparison with the true data distribution (e.g. class probability estimation, GANs)



Bojchevski, Aleksandar, et al. "Netgan: Generating graphs via random walks." *International conference on machine learning*. PMLR, 2018.

Random Walk: NetGAN

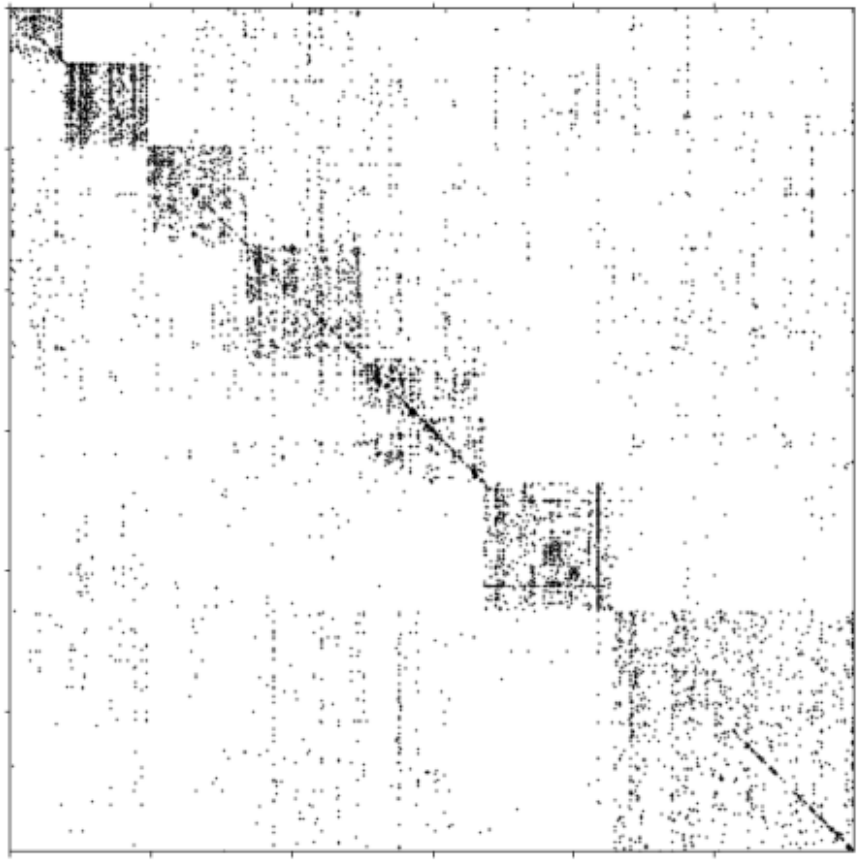
Challenges

1. Single large graph as input
• Compared to e.g. many images in computer vision

2. Quadratic scaling and sparsity
• For N nodes there are N^2 possible edges
• Real graphs have $|E| \ll N^2$ significantly fewer edges

3. Discrete output samples
• Can't easily backpropagate through sampling step

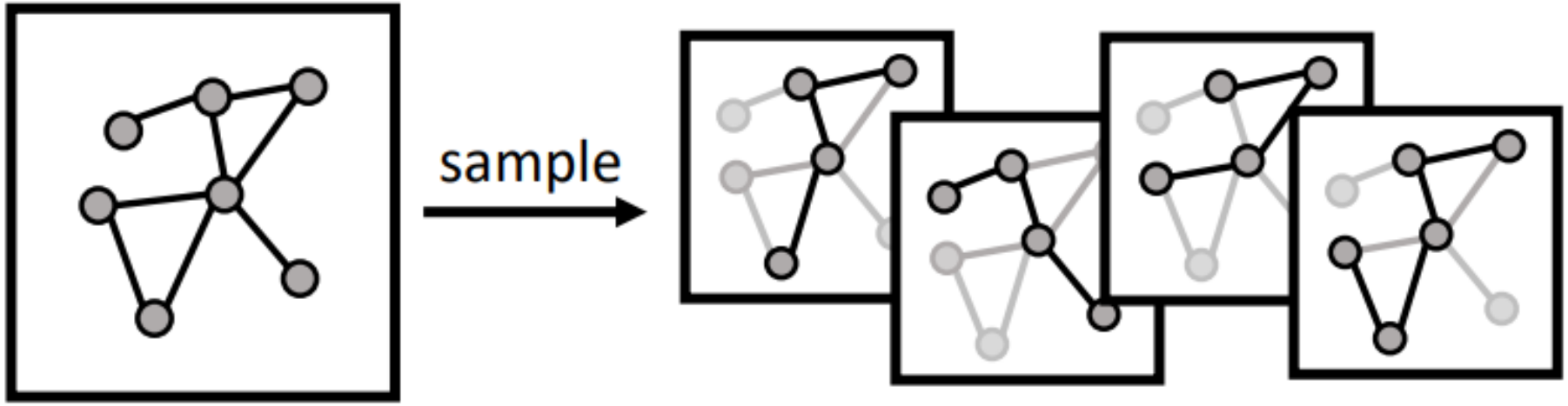
4. Permutation invariance



Bojchevski, Aleksandar, et al. "Netgan: Generating graphs via random walks." *International conference on machine learning*. PMLR, 2018.

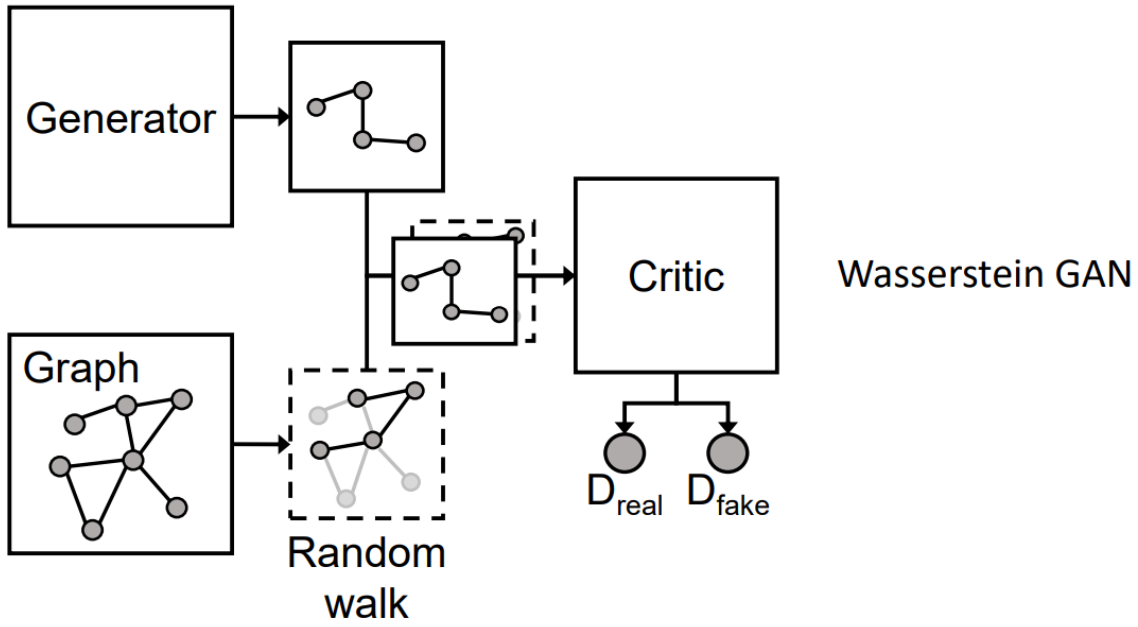
Random Walk: NetGAN

Decomposing Graph Generation into learning a distribution of random walks over the graph

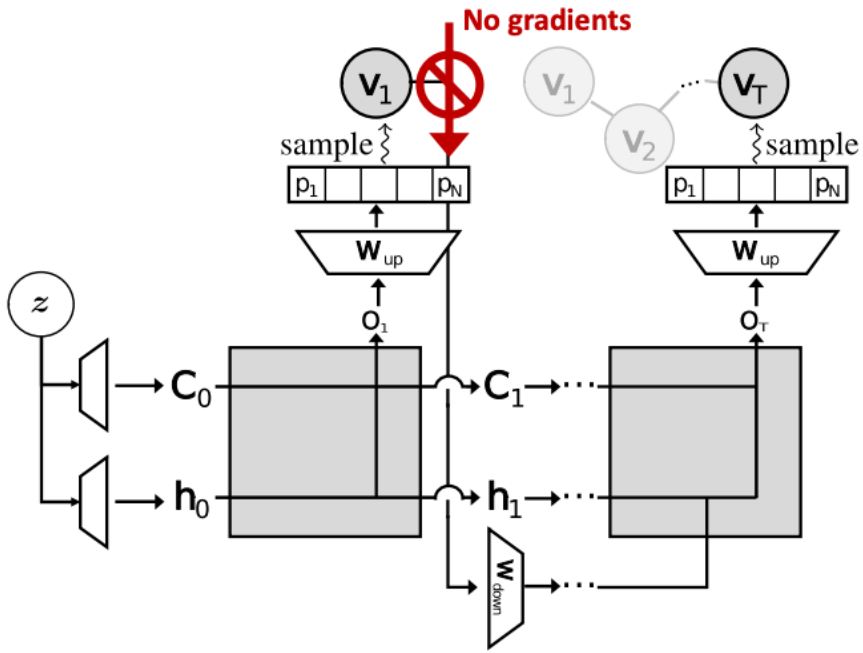


Bojchevski, Aleksandar, et al. "Netgan: Generating graphs via random walks." *International conference on machine learning*. PMLR, 2018.

Random Walk: NetGAN



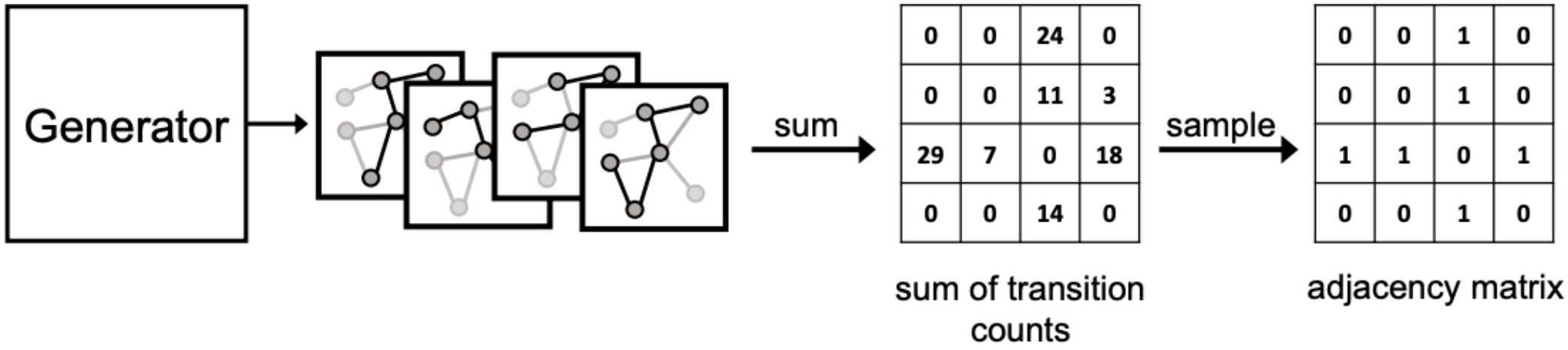
Model Framework



Generator

Bojchevski, Aleksandar, et al. "Netgan: Generating graphs via random walks." *International conference on machine learning*. PMLR, 2018.

Random Walk: NetGAN

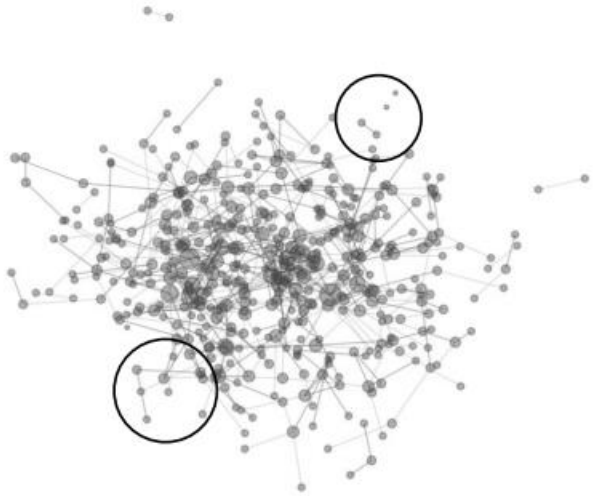


Graph assembly: sample edges with probability proportional to their transition counts

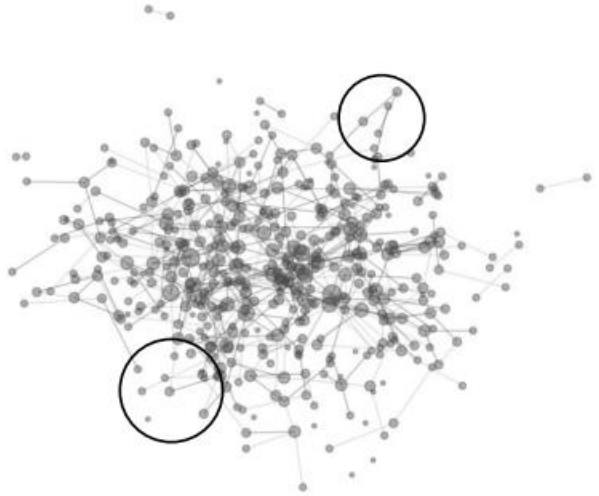
Bojchevski, Aleksandar, et al. "Netgan: Generating graphs via random walks." *International conference on machine learning*. PMLR, 2018.

Random Walk: NetGAN

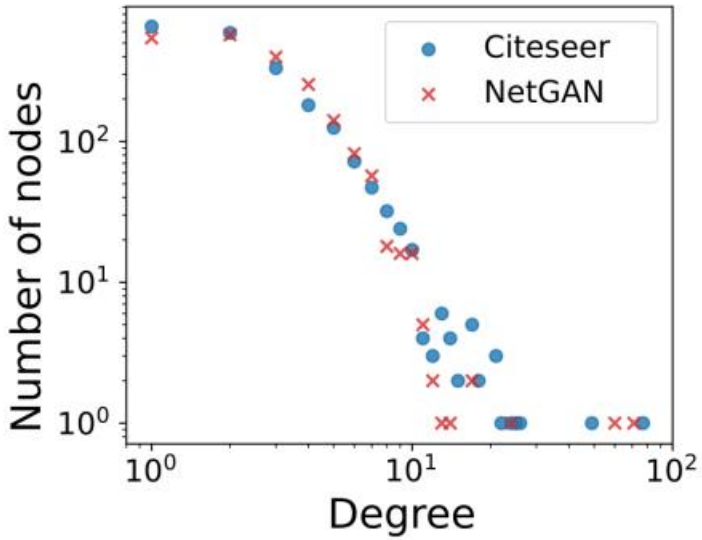
Key point: Generate graphs that have similar structure but are not replicas



Original graph



Graph generated by NetGAN

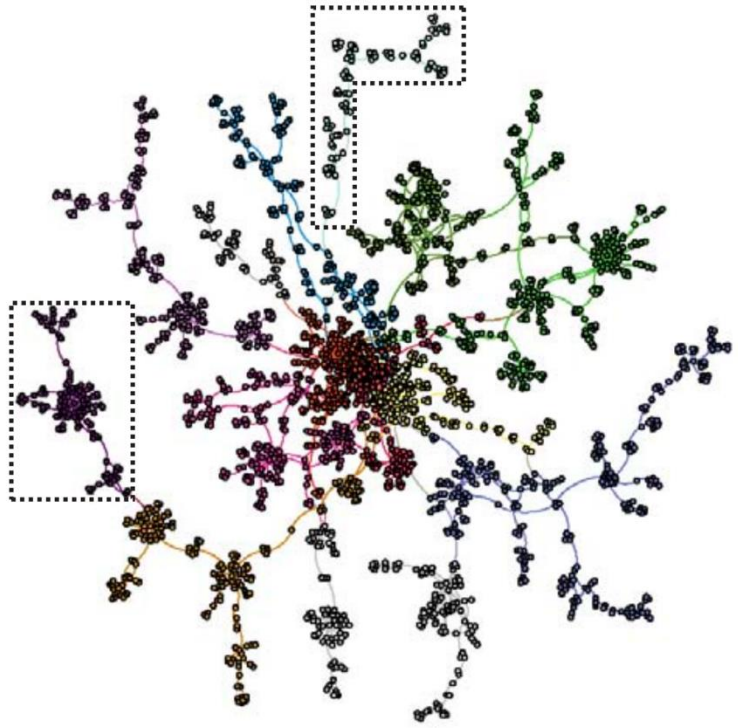


Bojchevski, Aleksandar, et al. "Netgan: Generating graphs via random walks." *International conference on machine learning*. PMLR, 2018.

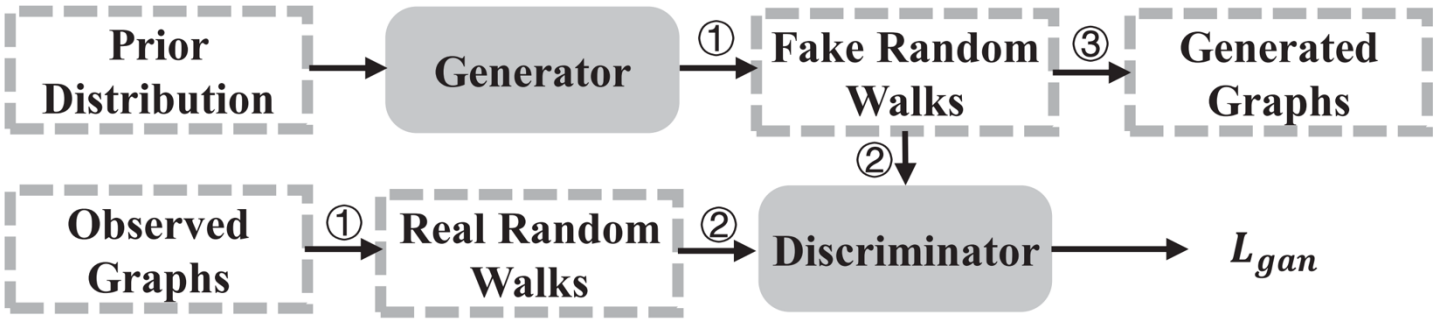
Beyond NetGAN: Community Preserving GAN

◆ Motivation

- **Community Structure,** as the main character of graph data, existing graph generation solutions cannot handle this property.
- Existing autoregressive and random walk-based solutions are not efficient.



real-world community

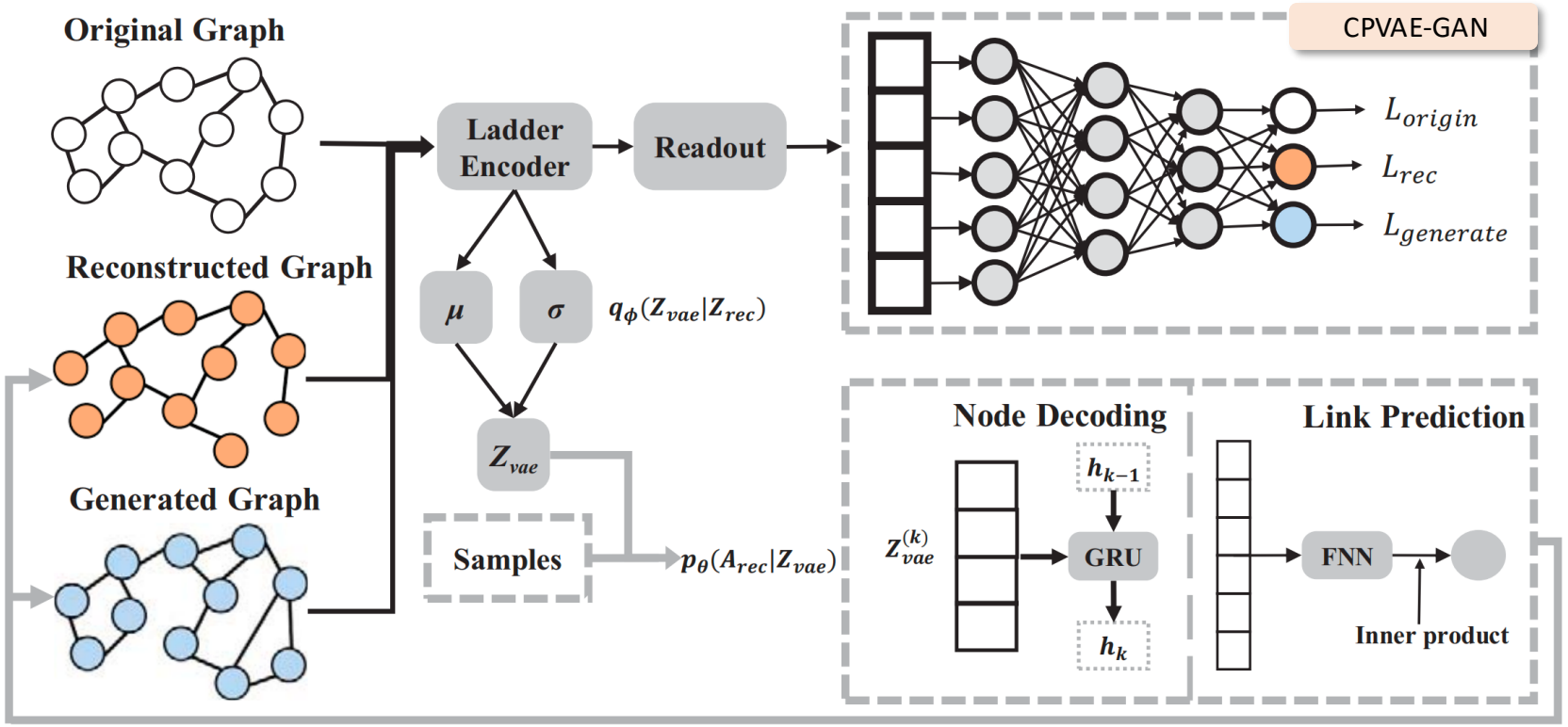


NetGAN is Random Walk-based, with low efficiency

Beyond NetGAN: Community Preserving GAN

◆ Contribution

- Community preserving graph generator: **CPVAE-GAN (CPGAN)**
- Community-preserving graph encoder: **Ladder Encoder**
- deprecate random walk sampling, leveraging autoencoder, which is efficient.



Xiang, S., Cheng, D., Zhang, J., Ma, Z., Wang, X., & Zhang, Y. (2022, May). Efficient learning-based community-preserving graph generation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)* (pp. 1982-1994). IEEE.

Beyond NetGAN: Community-Preserving GAN

◆ Experimental Results

Graph	Citeseer		Pubmed		PPI		3D Point Cloud		Facebook		Google	
	NMI(e-2)	ARI(e-2)	NMI(e-2)	ARI(e-2)	NMI(e-2)	ARI(e-2)	NMI(e-2)	ARI(e-2)	NMI(e-2)	ARI(e-2)	NMI(e-2)	ARI(e-2)
SBM	19.7±0.9	1.9±0.1	4.4±0.2	0.3±0.1	11.3±0.7	1.2±0.1	37.0±1.3	11.4±0.7	14.5±2.0	2.1±0.3	24.4±0.9	1.3±0.4
DCSBM	27.1±0.8	1.7±0.1	18.9±0.2	0.3±0.1	18.6±0.8	1.8±0.3	37.3±1.4	11.5±0.8	17.5±1.5	1.9±0.3	29.4±0.6	5.7±0.5
BTER	27.3±0.7	1.8±0.1	19.1±0.2	0.3±0.1	19.0±0.7	1.7±0.1	38.1±1.2	12.1±0.8	17.9±1.2	2.1±0.2	30.3±0.7	5.8±0.5
MMSB	26.7±0.9	4.4±1.0	OOM	OOM	15.4±0.6	0.8±0.4	7.1±0.4	1.3±0.3	OOM	OOM	OOM	OOM
VGAE	63.0±0.4	29.0±1.5	42.0±0.3	15.0±0.4	50.4±0.6	40.0±1.2	57.0±0.8	8.2±1.1	OOM	OOM	OOM	OOM
Graphite	62.8±0.7	28.2±2.1	43.0±0.5	15.1±0.4	52.3±0.8	33.4±1.9	58.8±0.4	13.2±0.3	OOM	OOM	OOM	OOM
SBMGNN	62.6±0.5	21.5±1.0	39.3±0.5	14.1±0.5	56.9±0.4	31.0±1.6	59.2±0.9	15.9±1.1	OOM	OOM	OOM	OOM
NetGAN	57.9±0.5	20.1±0.3	OOM	OOM	55.2±0.5	30.2±0.3	67.4±0.9	37.8±2.6	OOM	OOM	OOM	OOM
CPGAN	**72.5±0.4**	**44.3±1.5**	**45.8±0.9**	**34.1±1.1**	**57.0±0.7**	**44.2±1.3**	**70.6±0.6**	**39.9±1.4**	**54.7±1.0**	**28.4±1.6**	**38.7±0.5**	**30.8±0.5**

Performance on Community-preserving graph generation

#Nodes	0.1k	1k	10k	100k
MMSB	0.11	0.91	40.3	-
Kronecker	1.39	1.55	**3.25**	**4.73**
GraphRNN-S	1.63	15.4	161	-
VGAE	**0.06**	**0.42**	9.75	-
Graphite	0.07	0.47	10.6	-
SBMGNN	0.08	0.63	12.4	-
NetGAN	0.27	2.80	31.1	-
CondGEN-R	0.18	25.3	-	-
CPGAN	0.35	0.70	6.39	32.9

Comparison on training time

#Nodes	0.1k	1k	10k	100k
E-R	$\mathbf{4.6}e^{-4}$	$9.0e^{-3}$	0.46	10.1
B-A	$1.0e^{-3}$	$1.2e^{-2}$	**0.11**	1.17
Chung-Lu	$7.2e^{-4}$	$2.5e^{-3}$	0.18	2.38
SBM	$6.1e^{-3}$	0.09	2.58	37.1
DCSBM	$6.2e^{-3}$	0.09	2.69	39.3
BTER	$1.28e^{-3}$	$\mathbf{1.9}e^{-3}$	0.16	**0.25**
MMSB	$6.1e^{-3}$	0.09	2.56	-
Kronecker	$8.5e^{-3}$	0.08	1.00	9.69
GraphRNN-S	0.27	4.74	63.6	-
VGAE	$4.2e^{-3}$	0.04	0.38	-
Graphite	$6.1e^{-3}$	0.06	0.64	-
SBMGNN	0.01	0.11	1.18	-
NetGAN	$8.7e^{-3}$	0.09	1.12	-
CondGEN-R	$8.3e^{-3}$	0.15	-	-
CPGAN	$9.1e^{-3}$	0.08	0.95	86.1

Comparison on infernece time

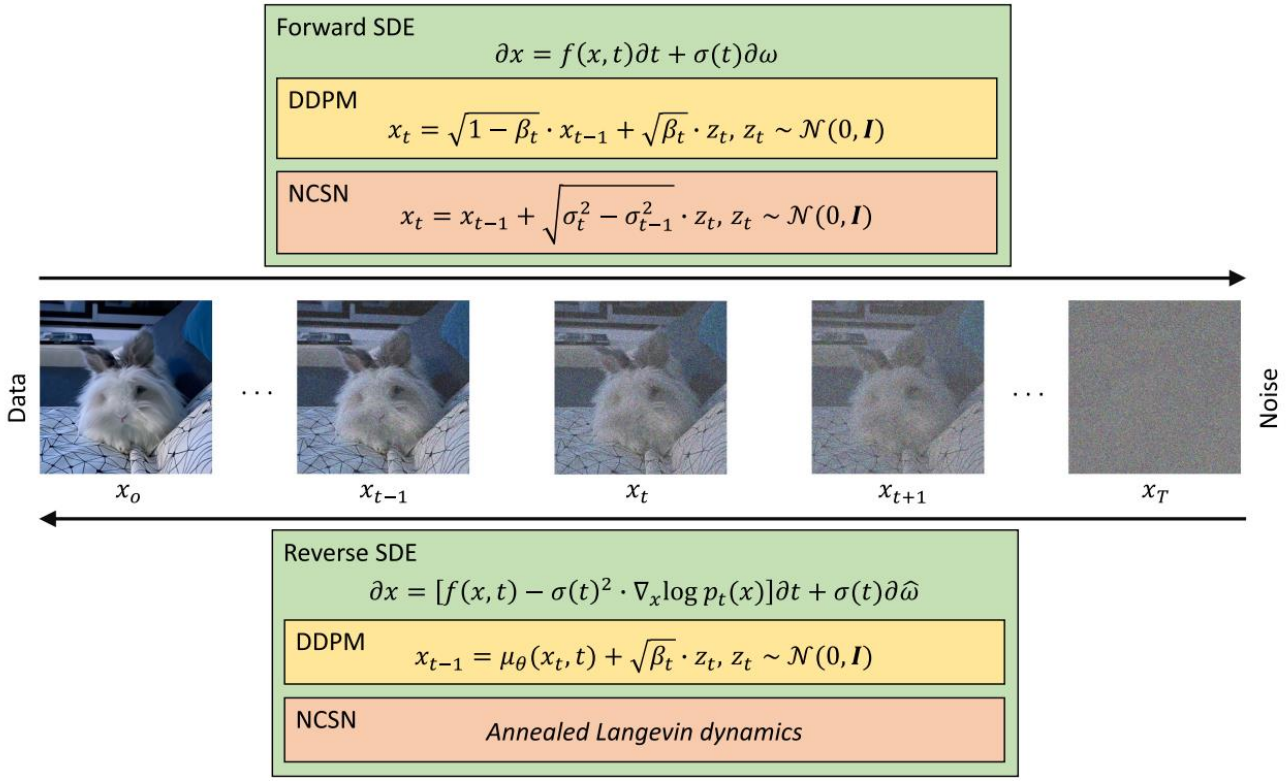
Similarity-Based Graph Generation

- Recursive: Kronecker
- Autoregressive: GraphRNN
- Random Walk: NetGAN
- **Diffusion Model: DiGress**
- Dynamic Graph Generation
- Similarity and Graph Generation

Diffusion Model: DiGress

Diffusion models: Two major processes.

- **Forward process** transforms data into noise.
- **Generative process** learns to transform the noise back into data.

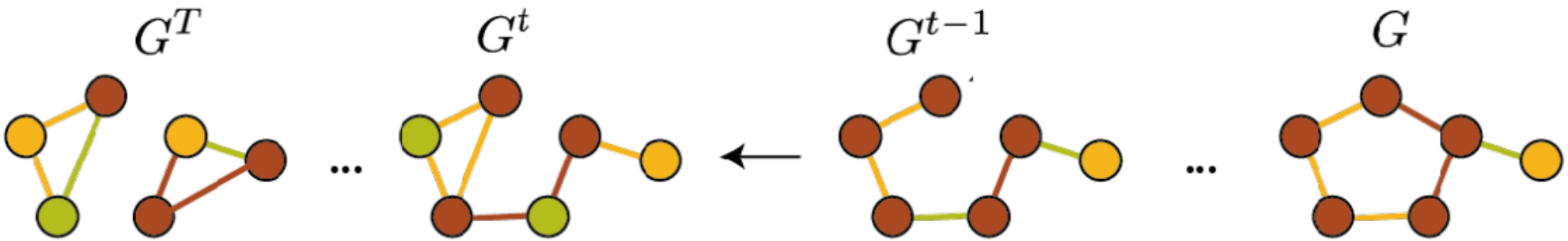


Croitoru, Florinel-Alin, et al. "Diffusion models in vision: A survey." IEEE Transactions on Pattern Analysis and Machine Intelligence 45.9 (2023): 10850-10869.

Diffusion Model: DiGress

Diffusion model for graph generation: Motivation

- Motivation for discrete diffusion: no need to predict continuous values that do not exist in the data + do not break sparsity
- Adding noise = sampling node or edge types from a categorical distribution.
- No edge = one particular edge type.
- The noise is sampled independently on each node and edge.

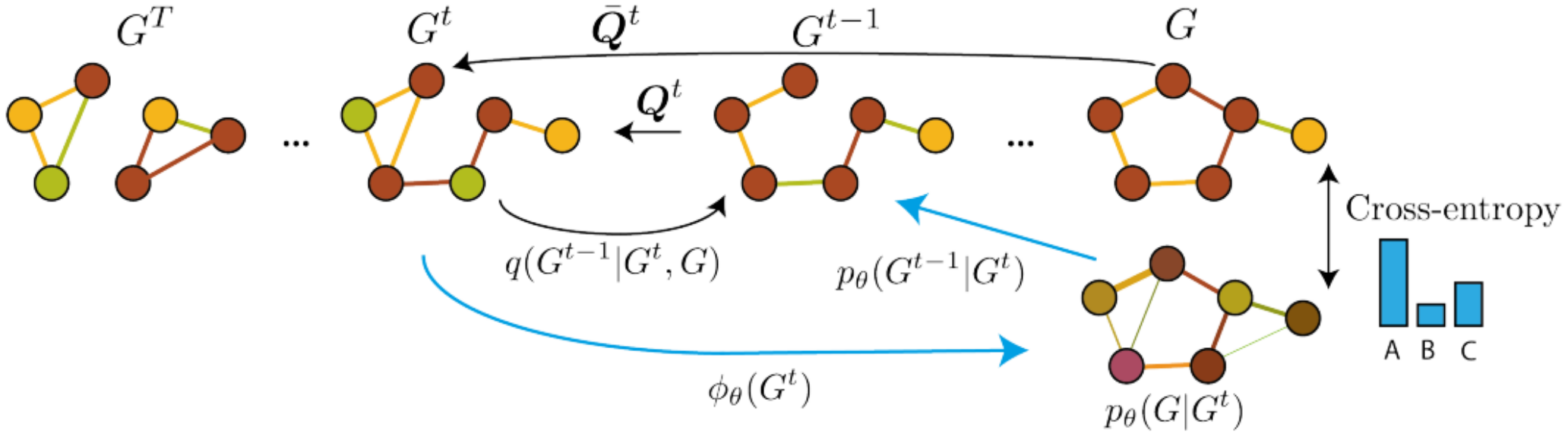


Vignac, Clément, et al. "DiGress: Discrete Denoising diffusion for graph generation." *ICLR*. 2023.

Diffusion Model: DiGress

Diffusion model for graph generation.

- **Forward process** adds noise using Markov transition matrix Q^t.
- **Generative process** learns to transform the noise back into data. A discrete $G^{t-1}$is sampled from the learned categorical distribution.
- Graph generation becomes a sequence of node and edge classification tasks.



Vignac, Clément, et al. "DiGress: Discrete Denoising diffusion for graph generation." *ICLR*. 2023.

Similarity-Based Graph Generation

- Recursive: Kronecker
- Autoregressive: GraphRNN
- Random Walk: NetGAN
- Diffusion Model: DiGress
- **Dynamic Graph Generation**
- Similarity and Graph Generation

Dynamic Graph Generation

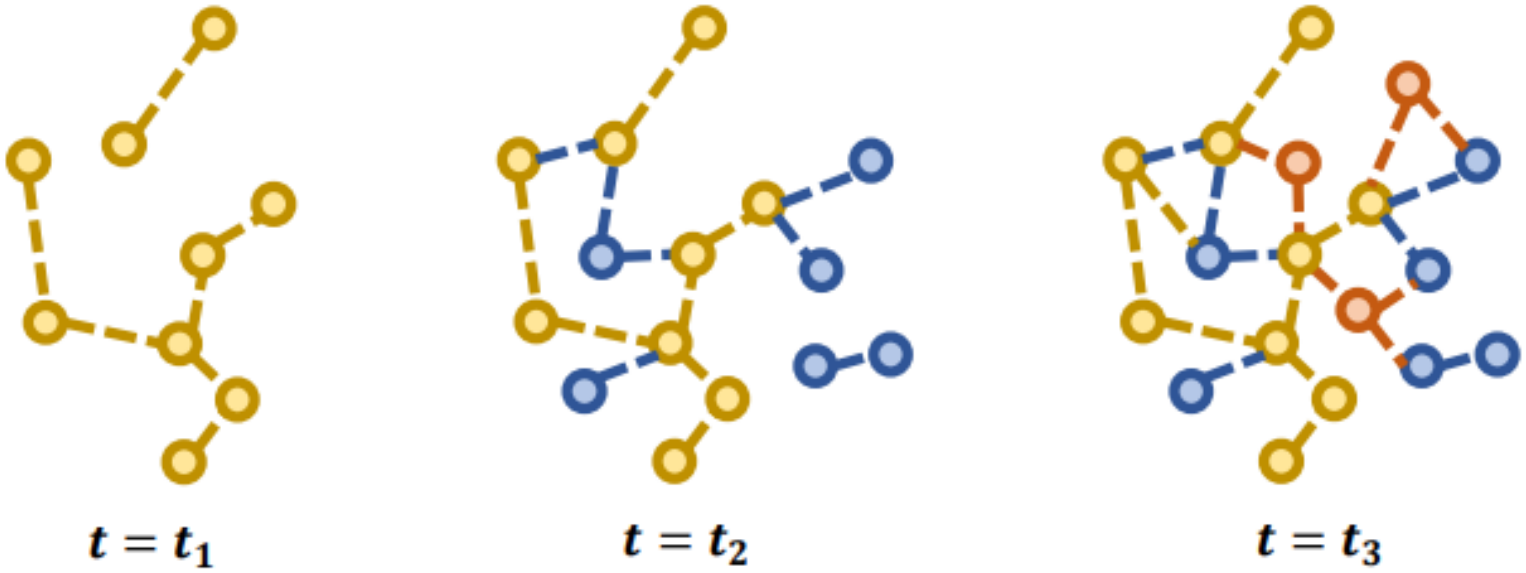
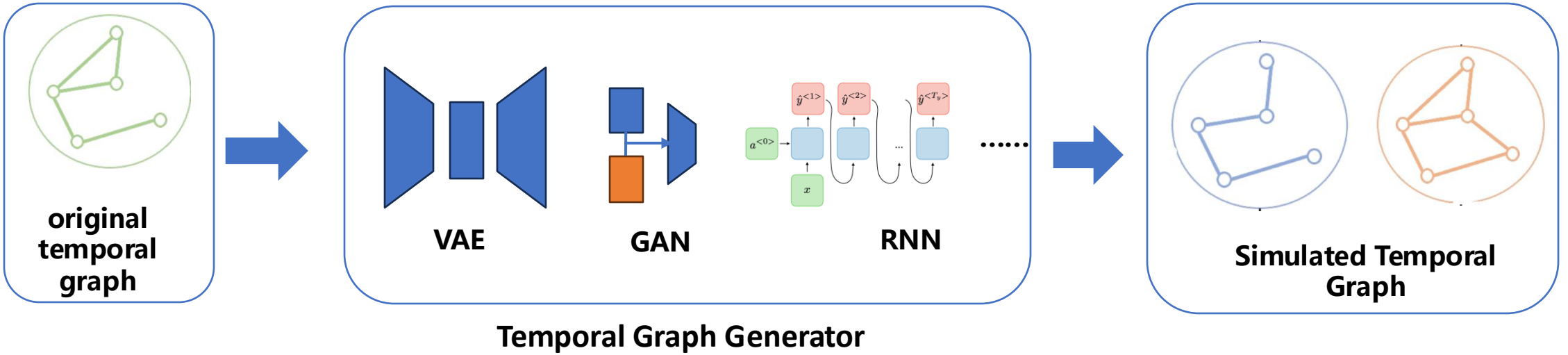


Fig. 1. An example of time-evolving graph.

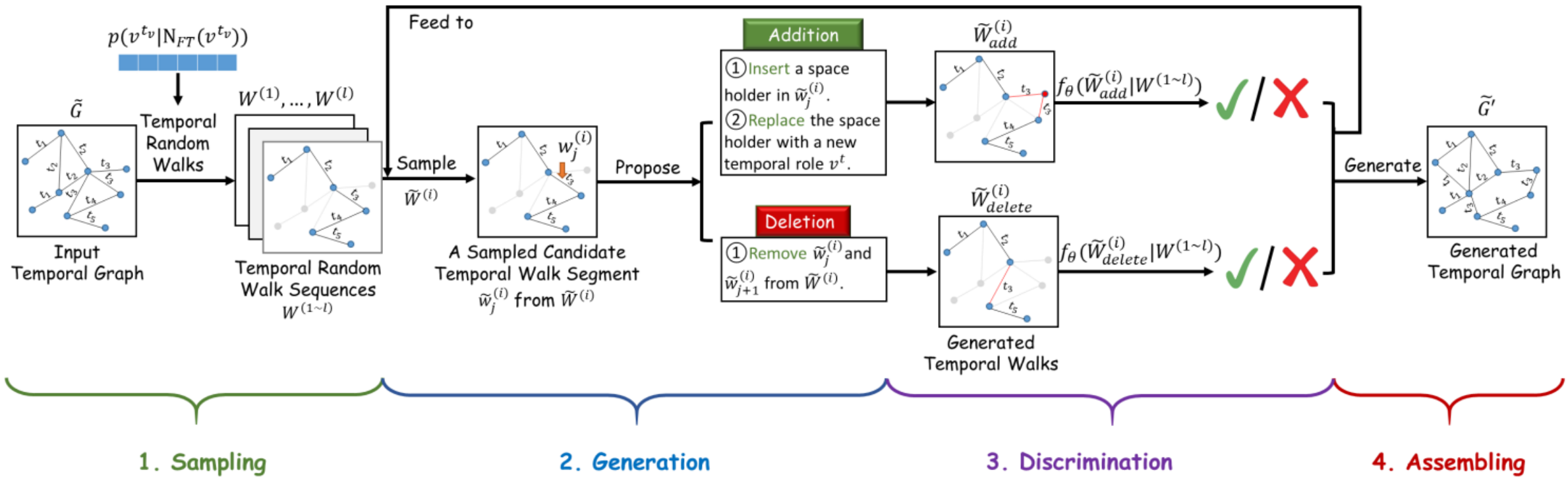
Sheng, Xiang, et al. "Efficient Learning-based Graph Simulation for Temporal Graphs." IEEE ICDE 2025.

Dynamic Graph Generation (Background)



Dynamic Graph Generation

Temporal Random Walk-based Generation (State-of-the-art)

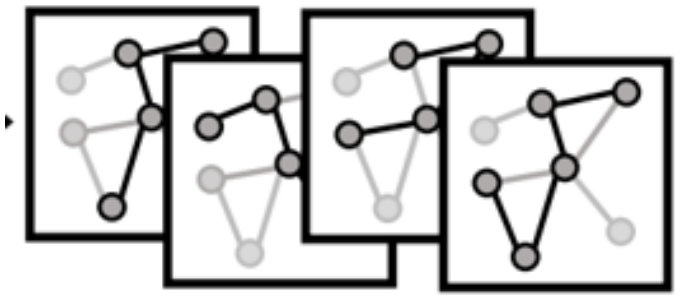


Sheng, Xiang, et al. "Efficient Learning-based Graph Simulation for Temporal Graphs." IEEE ICDE 2025.

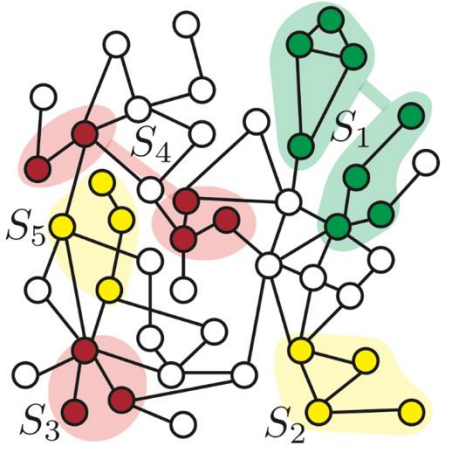
Dynamic Graph Generation

Limitations and Motivation

1. Large amount of sampled random walks.
2. High computation complexity in time level. (e.g., Taggen, TGGAN has $O(N^2T^2)$ complexity.

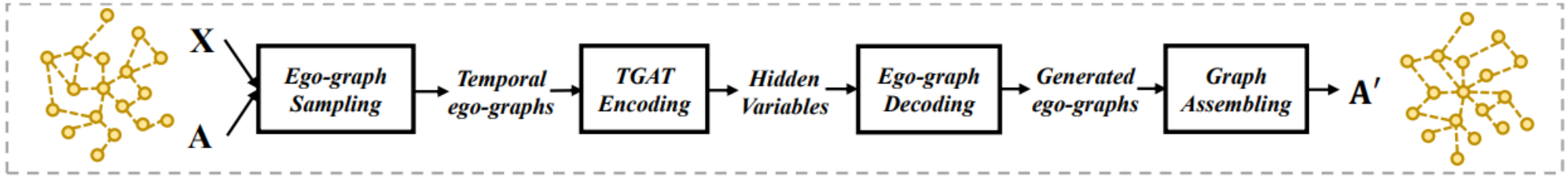


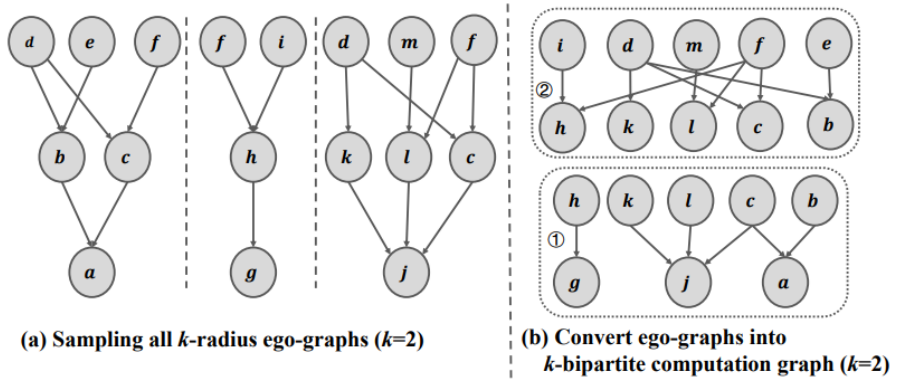
1. Moderate amount of sampled sub-graphs.
2. Acceptable computation complexity in time level. (e.g., our proposed TGAE has $O(N^2T)$ complexity.

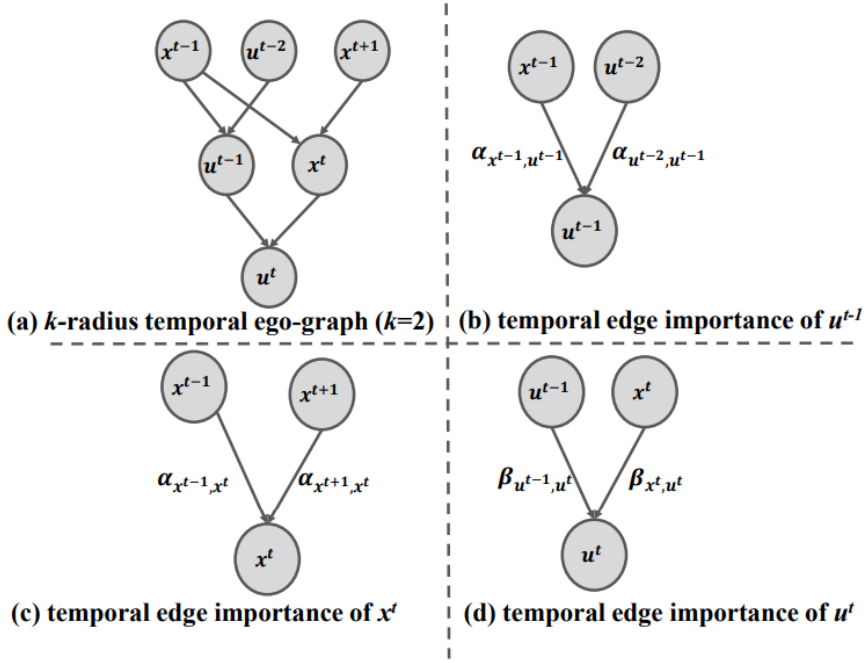


Dynamic Graph Generation

Proposed model: TGAE







Sheng, Xiang, et al. "Efficient Learning-based Graph Simulation for Temporal Graphs." IEEE ICDE 2025.

Dynamic Graph Generation

Comprehensive Comparison on Temporal Graph Simulation

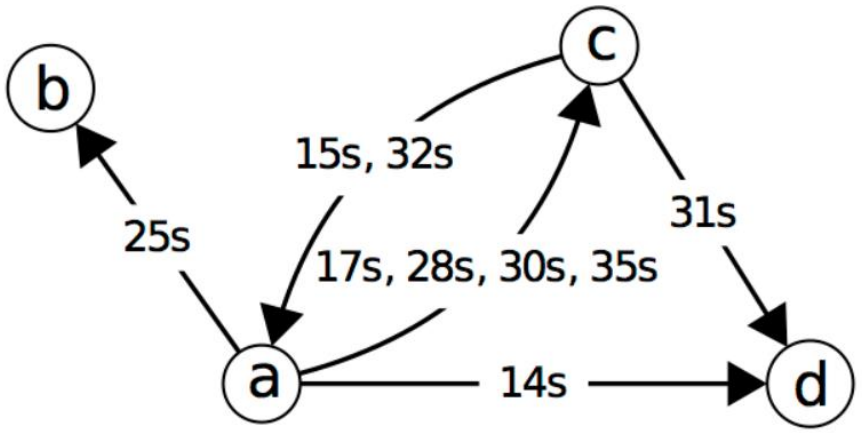
TABLE IV
MEDIAN SCORE $f_{med}(\cdot)$ COMPARISON WITH SEVEN METRICS. (SMALLER METRIC VALUES INDICATE BETTER PERFORMANCE)

Dataset	Metric	TGAE	TIGGER	DYMOND	TGGAN	TagGen	NetGAN	E-R	B-A	VGAE	Graphite	SBMGNN
DBLP	Mean Degree	2.41E-3	3.54E-3	2.98E-3	3.25E-3	**7.46E-4**	4.16E-3	5.52E-3	1.23E-1	1.79E-3	1.79E-3	1.79E-3
	LCC	**2.61E-3**	2.75E-3	2.71E-3	2.77E-3	2.78E-3	3.35E-1	7.27E-1	9.11E-2	5.11E-1	5.40E-1	4.62E-1
	Wedge Count	**4.15E-3**	3.08E-2	2.31E-2	5.38E-1	7.14E-1	5.05E-1	5.07E-1	3.74E-1	1.81E+0	2.15E+0	2.39E+0
	Claw Count	**7.29E-3**	2.64E-2	1.35E-2	2.98E+0	3.02E+0	9.27E-1	8.78E-1	4.52E+0	8.78E+0	1.21E+1	1.42E+1
	Triangle Count	**4.79E-3**	7.85E-2	3.77E-2	5.33E-1	5.44E-1	8.83E-1	9.94E-1	8.24E-1	9.27E+0	9.21E+0	8.66E+0
	PLE	**1.73E-3**	3.34E-2	9.15E-3	1.78E-1	1.79E-1	2.24E-1	1.65E-1	8.45E-2	4.01E-1	4.65E-1	4.25E-1
	N-Components	**3.05E-3**	3.07E-3	3.11E-3	3.39E-3	3.51E-3	2.13E-1	8.36E-1	5.06E-2	5.07E-1	5.49E-1	4.83E-1
MATH	Mean Degree	**2.69E-2**	1.05E-1	OOM	OOM	OOM	2.13E-1	2.29E-1	3.24E-2	2.39E-1	2.44E-1	3.72E-2
	LCC	8.72E-2	9.31E-2	OOM	OOM	OOM	**2.99E-2**	8.83E-1	1.24E-1	5.56E-1	5.30E-1	3.73E-1
	Wedge Count	**1.05E-1**	2.37E-1	OOM	OOM	OOM	2.42E-1	9.27E-1	3.15E-1	6.87E-1	7.50E-1	1.77E+0
	Claw Count	**2.59E-1**	3.75E-1	OOM	OOM	OOM	4.96E-1	9.99E-1	4.86E-1	2.95E+0	3.43E+0	8.13E+0
	Triangle Count	**9.79E-2**	8.78E-1	OOM	OOM	OOM	2.34E+0	1.00E+0	5.84E-1	1.74E+0	1.66E+0	2.24E+0
	PLE	**2.41E-2**	9.36E-1	OOM	OOM	OOM	1.11E+0	2.35E-1	7.81E-2	5.74E-1	5.40E-1	2.49E-1
	N-Components	**3.15E-2**	4.66E-2	OOM	OOM	OOM	3.50E-2	1.00E+0	1.35E-1	5.90E-1	5.61E-1	3.96E-1
UBUNTU	Mean Degree	**9.73E-2**	OOM	OOM	OOM	OOM	OOM	2.32E+1	5.29E-1	OOM	OOM	OOM
	LCC	**1.32E-1**	OOM	OOM	OOM	OOM	OOM	3.71E+0	2.98E+0	OOM	OOM	OOM
	Wedge Count	**3.16E-1**	OOM	OOM	OOM	OOM	OOM	1.45E+1	9.76E-1	OOM	OOM	OOM
	Claw Count	5.60E-1	OOM	OOM	OOM	OOM	OOM	**3.01E-1**	9.96E-1	OOM	OOM	OOM
	Triangle Count	**1.21E-1**	OOM	OOM	OOM	OOM	OOM	5.05E-1	1.00E+0	OOM	OOM	OOM
	PLE	**8.52E-2**	OOM	OOM	OOM	OOM	OOM	7.33E-1	5.31E-1	OOM	OOM	OOM
	N-Components	**2.64E-2**	OOM	OOM	OOM	OOM	OOM	1.00E+0	8.55E-1	OOM	OOM	OOM

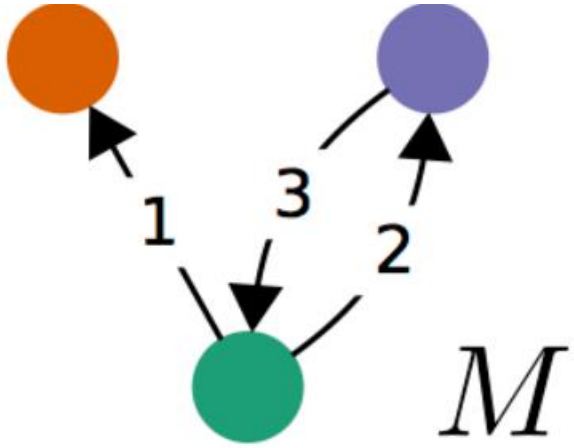
Sheng, Xiang, et al. "Efficient Learning-based Graph Simulation for Temporal Graphs." IEEE ICDE 2025.

Dynamic Graph Generation

Comprehensive Comparison on Temporal Graph Simulation



Dynamic Graph



Key motif structure

TABLE VI

MAXIMUM MEAN DISCREPANCY OF INSTANCE COUNTS OF ALL 2- AND 3-NODE, 3-EDGE δ-TEMPORAL MOTIFS BETWEEN RAW AND GENERATED TEMPORAL NETWORKS (σ REFERS TO THE SIGMA VALUE FOR GAUSSIAN KERNEL)

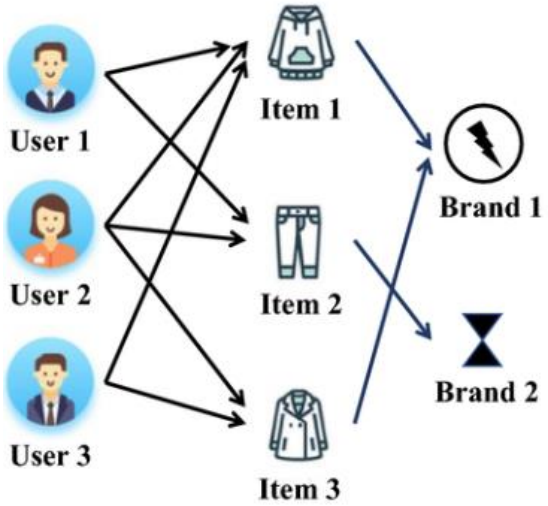
Dataset	TGAE	TIGGER	DYMOND	TGGAN	TagGen	NetGAN	E-R	B-A	VGAE	Graphite	SBMGNN
DBLP	**2.65E-5**	9.68E-4	1.25E-4	2.08E-2	2.31E-2	2.21E-1	6.43E-2	1.08E+0	1.34E+0	1.95E+0	1.99E+0
MSG	**2.27E-5**	2.12E-4	3.77E-5	9.81E-3	1.09E-2	1.85E-2	1.83E-2	1.17E+0	1.98E+0	1.99E+0	1.65E+0
BITCOIN-A	**1.12E-6**	2.76E-5	OOM	OOM	OOM	OOM	1.90E+0	2.00E+0	3.88E-1	5.39E-1	1.08E-1
BITCOIN-O	**5.49E-6**	3.06E-5	OOM	OOM	OOM	OOM	1.80E+0	2.00E+0	1.82E+0	1.98E+0	5.22E-1
EMAIL	**2.12E-2**	7.65E-2	3.27E-2	OOM	OOM	9.78E-2	9.74E-1	1.95E+0	1.95E+0	1.07E+0	1.74E+0
MATH	**7.86E-4**	2.14E-3	OOM	OOM	OOM	5.11E-3	6.59E-3	2.74E-3	2.00E+0	1.89E+0	1.94E+0
UBUNTU	**1.27E-3**	OOM	OOM	OOM	OOM	OOM	1.52E+0	2.00E+0	OOM	OOM	OOM

Sheng, Xiang, et al. "Efficient Learning-based Graph Simulation for Temporal Graphs." IEEE ICDE 2025.

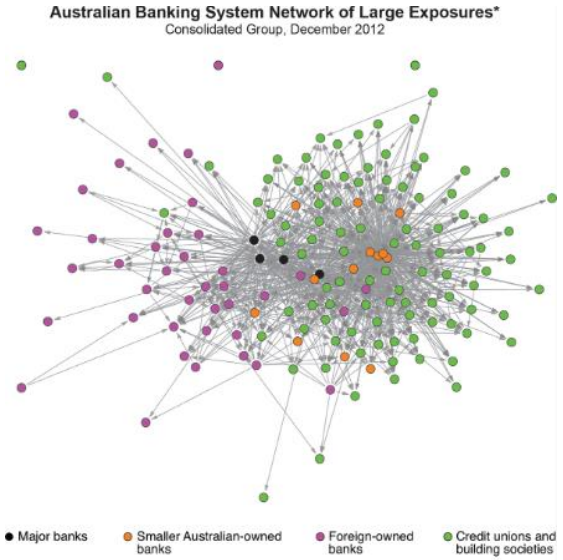
Dynamic Graph Generation

Graph Generation in Data Management (Motivation)

(1) Friend Network



(2) User-item Network



(3) Financial Network

1. A lack of realistic graphs for evaluating the performance of the graph processing system.

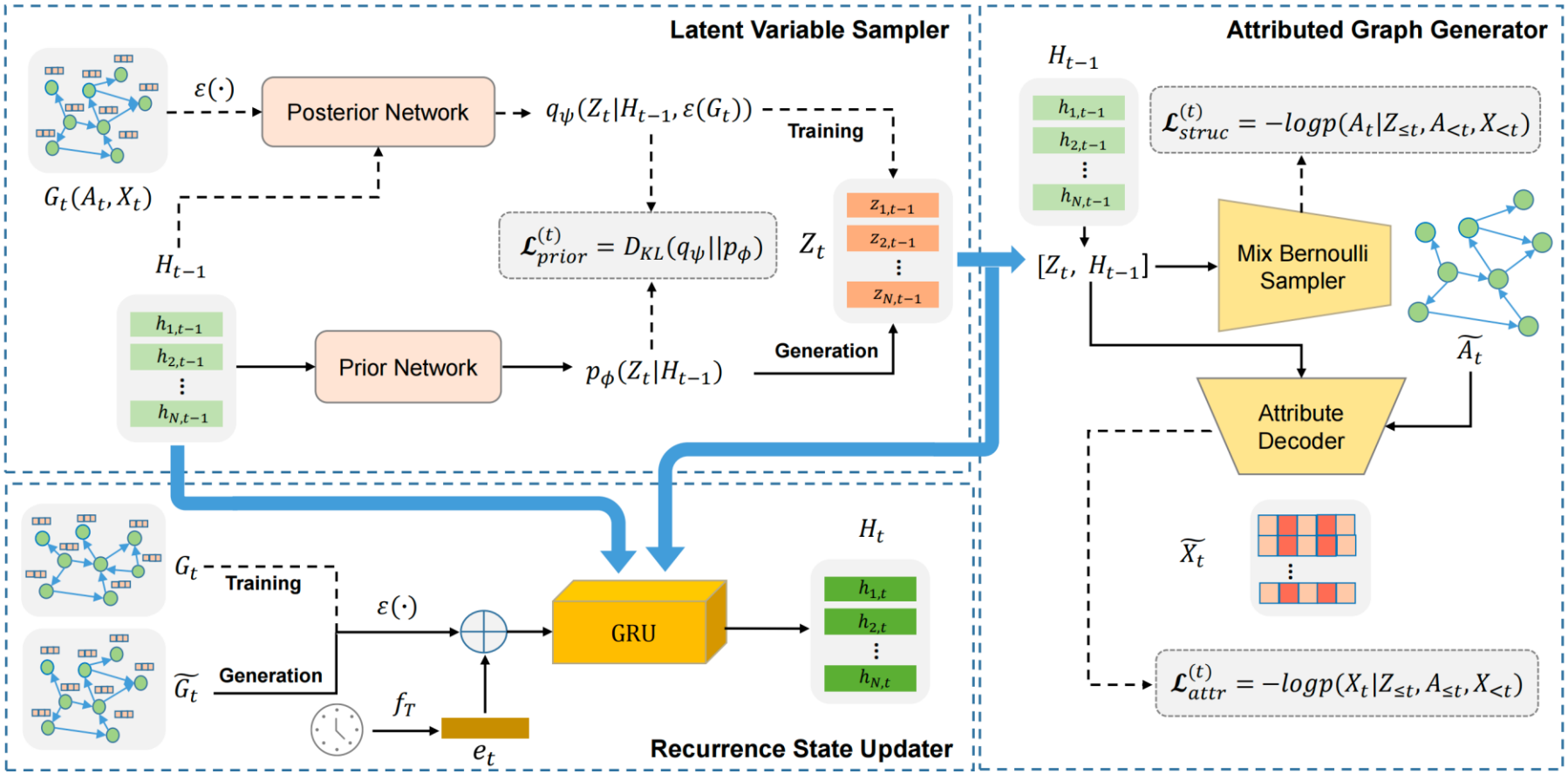
2. Synthetic graphs can be used in many network analysis tasks, such as community detection.

3. The simulated graph anonymizes node entities and their link relationships, preventing the leakage of private data.

Li, F., Wang, X., Cheng, D., Chen, C., Zhang, Y., & Lin, X. (2025). Efficient Dynamic Attributed Graph Generation. In IEEE ICDE 2025.

Dynamic Graph Generation

VRDAG: Variational Recurrent Dynamic Attributed Generator



The overall framework of VRDAG

Li, F., Wang, X., Cheng, D., Chen, C., Zhang, Y., & Lin, X. (2025). Efficient Dynamic Attributed Graph Generation. In IEEE ICDE 2025.

Dynamic Graph Generation

Bi-flow Message Passing Encoder ε

We divide the neighborhood message of v_i into in-flow message ${}_{in}h_{i,t}$ and out-flow message ${}_{out}h_{i,t}$. The bidirectional message passing can be formulated as:

$$ {}_{in}h_{i,t}^{(l)} = f_{in}^{(l)}\left((1+\epsilon_{in}^{(l)})\cdot h_{i,t}^{(l-1)} + \sum_{v_j \in N_{in}(v_i)} h_{j,t}^{(l-1)}\right) $$

$$ {}_{out}h_{i,t}^{(l)} = f_{out}^{(l)}\left((1+\epsilon_{out}^{(l)})\cdot h_{i,t}^{(l-1)} + \sum_{v_j \in N_{out}(v_i)} h_{j,t}^{(l-1)}\right) $$

In each layer, we apply a node aggregator f_{agg} on ${}_{in}h_{i,t}^{(l)}$ and ${}_{out}h_{i,t}^{(l)}$ to get hop-level node message $h_{i,t}^{(l)}$:

$$ h_{i,t}^{(l)} = f_{agg}\left(\left[{}_{in}h_{i,t}^{(l)} || {}_{out}h_{i,t}^{(l)}\right]\right) $$

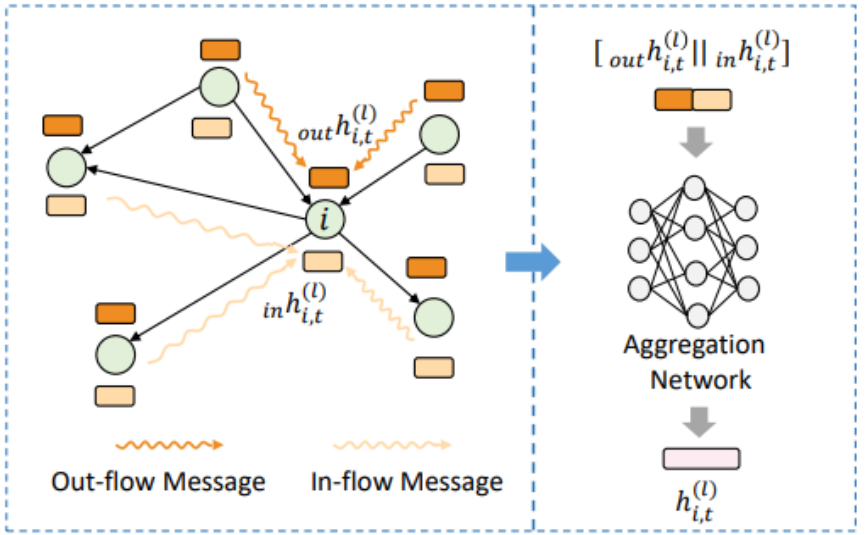


Fig. 2: The bi-flow message passing layer

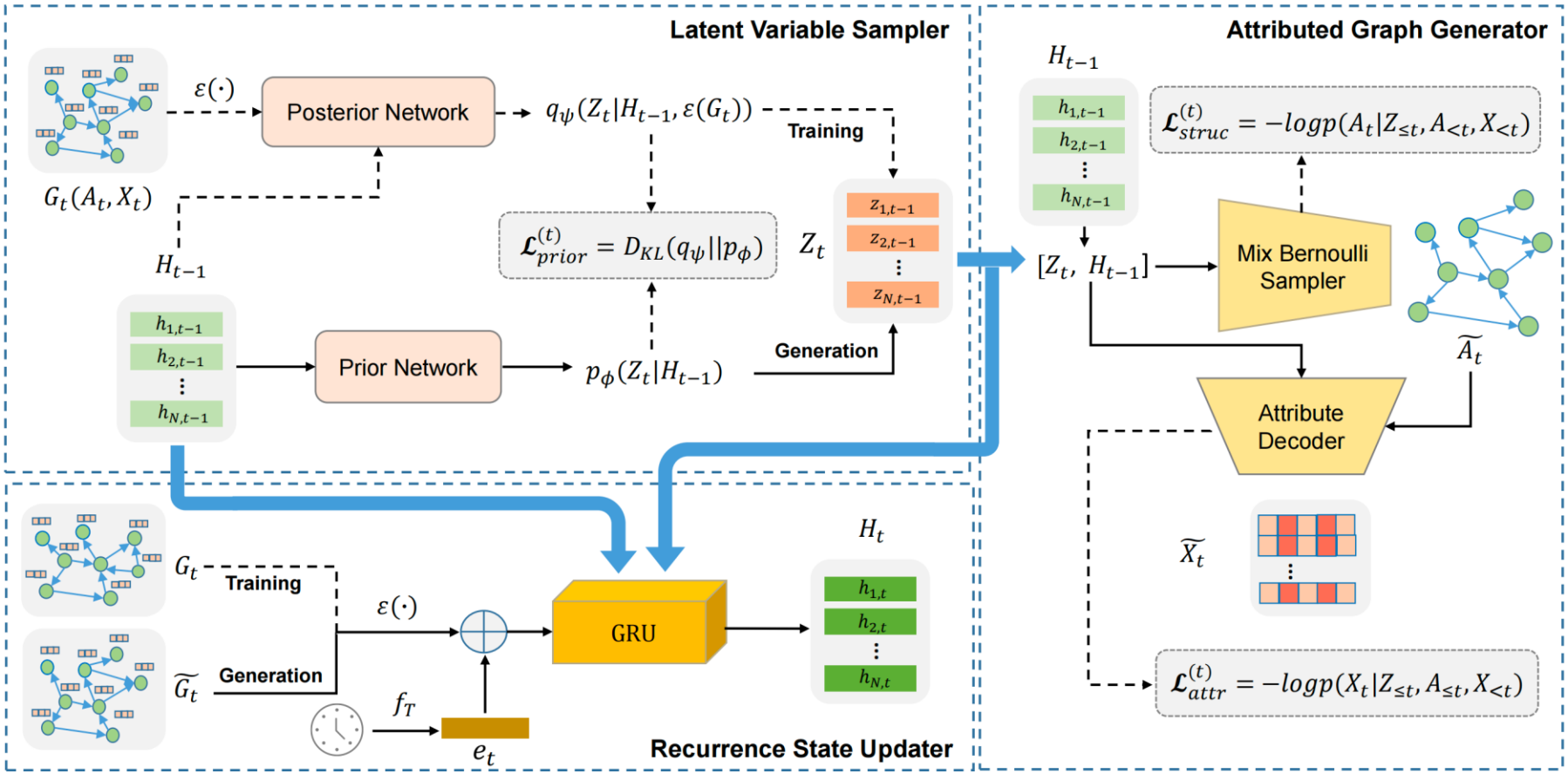
Finally, we use jump-connection technique to integrate hop-level node message from each layer as:

$$ \varepsilon(v_{i,t}) = f_{pool}\left(h_{i,t}^{(1)}, h_{i,t}^{(2)}, ..., h_{i,t}^{(L)}\right) $$

Li, F., Wang, X., Cheng, D., Chen, C., Zhang, Y., & Lin, X. (2025). Efficient Dynamic Attributed Graph Generation. In IEEE ICDE 2025.

Dynamic Graph Generation

VRDAG: Variational Recurrent Dynamic Attributed Generator



The overall framework of VRDAG

Li, F., Wang, X., Cheng, D., Chen, C., Zhang, Y., & Lin, X. (2025). Efficient Dynamic Attributed Graph Generation. In IEEE ICDE 2025.

Similarity-Based Graph Generation

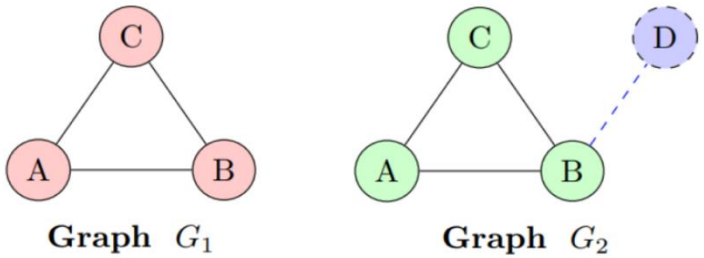
- Recursive: Kronecker
- Autoregressive: GraphRNN
- Random Walk: NetGAN
- Diffusion Model: DiGress
- Dynamic Graph Generation
- **Similarity and Graph Generation**

Similarity and Graph Generation

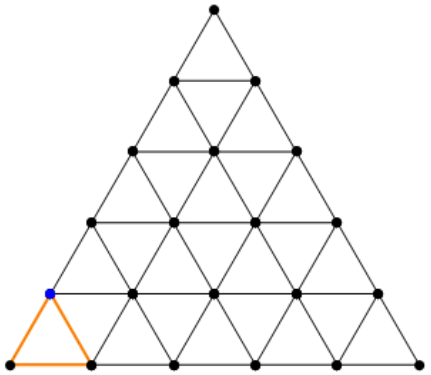
Graph generation aiming to maximize the similarity

Graph generation (simulation) methods are designed to maximize the similarity between generated graphs and observed graphs.

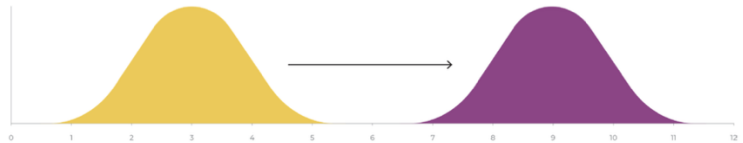
Graph similarities are used as the objectives of graph generation models.



1. Graph Edit Distance(GED)



2. Graph Statistics (e.g., triangle counts)



3. Node Distribution (e.g., Wasserstein Distance)

Xiang, S., Wen, D., Cheng, D., Zhang, Y., Qin, L., Qian, Z. and Lin, X., 2022. General graph generators: experiments, analyses, and improvements. The VLDB Journal, pp.1-29.

Similarity and Graph Generation

How about graph generation for graph similarity

Let's start with a fundamental graph similarity metric: Graph Edit Distance.

Graph Edit Distance aims to determine the minimum number of edit operations required to transform one graph into another, and the sequence of edit operations is called a graph edit path.

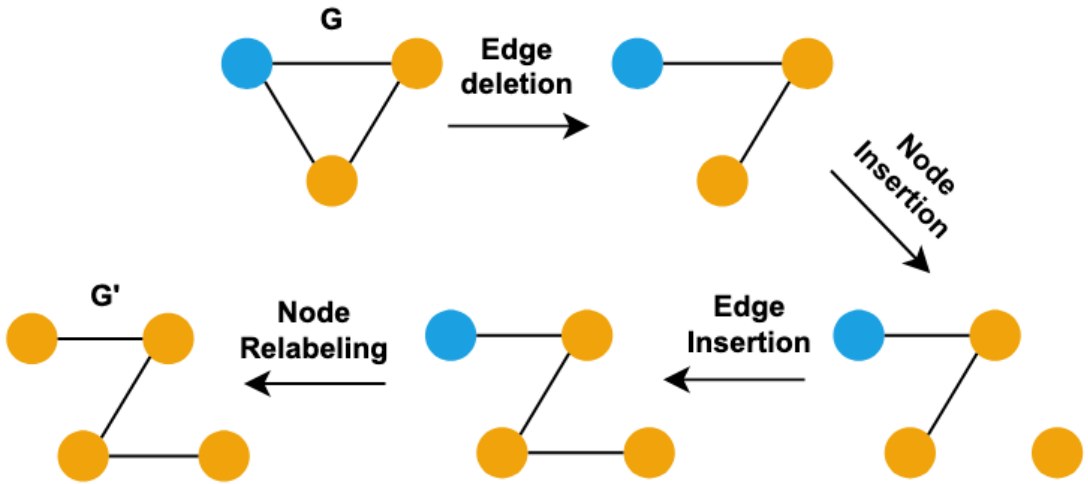


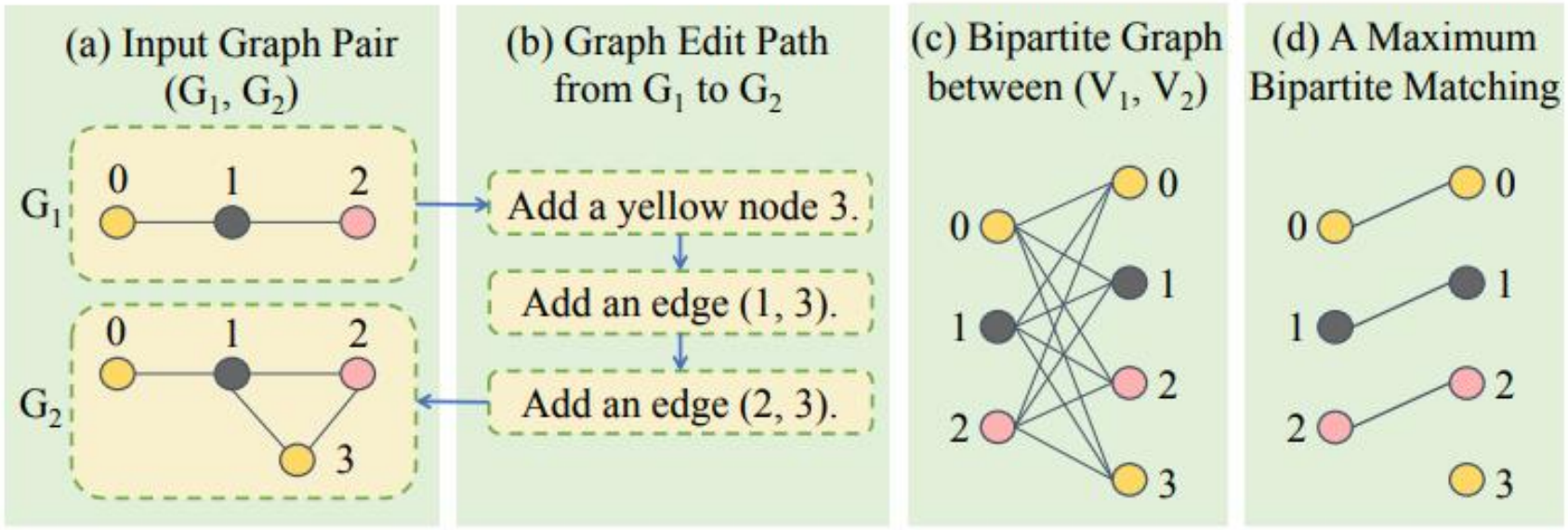
Figure 1: An optimal edit path for transforming G to G'. $\mathbf{GED}(G, G') = 4$.

Huang, Wei, et al. "DiffGED: Computing Graph Edit Distance via Diffusion-based Graph Matching." *arXiv preprint arXiv:2503.18245* (2025).

Similarity and Graph Generation

GEDGNN: Computing Graph Edit Distance via Neural Graph Matching

Graph edit distance can be modelled as maximum bipartite matching.



a, b: An instance of graph edit path.

c, d: Solving GED via bipartite matching.

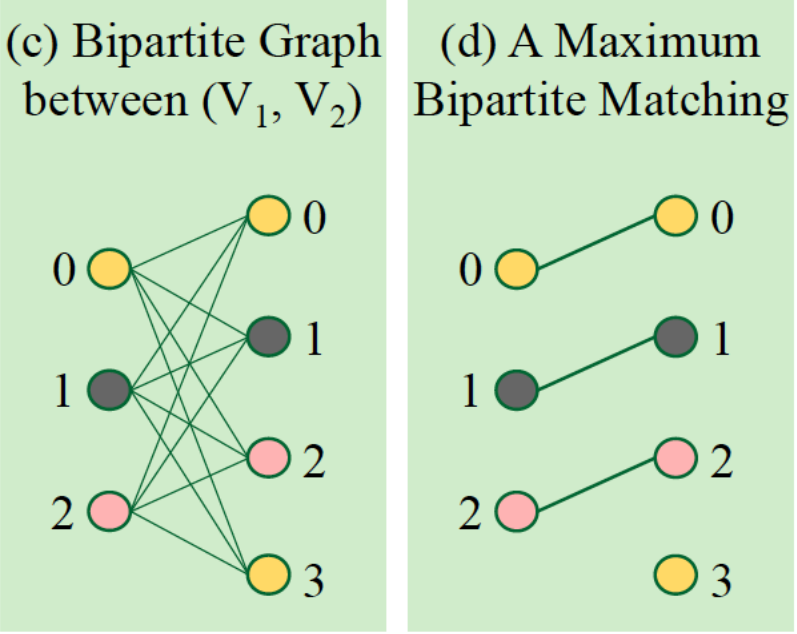
Piao, Chengzhi, et al. "Computing graph edit distance via neural graph matching." Proceedings of the VLDB Endowment 16.8 (2023): 1817-1829.

Similarity and Graph Generation

GEDGNN: Computing Graph Edit Distance via Neural Graph Matching

A Two-step Framework:

- Using GNN to predict a GED and generate a node matching matrix.
- Post-processing the node matching matrix to find a short edit path.



Bipartite graph matching

Bipartite graph generation

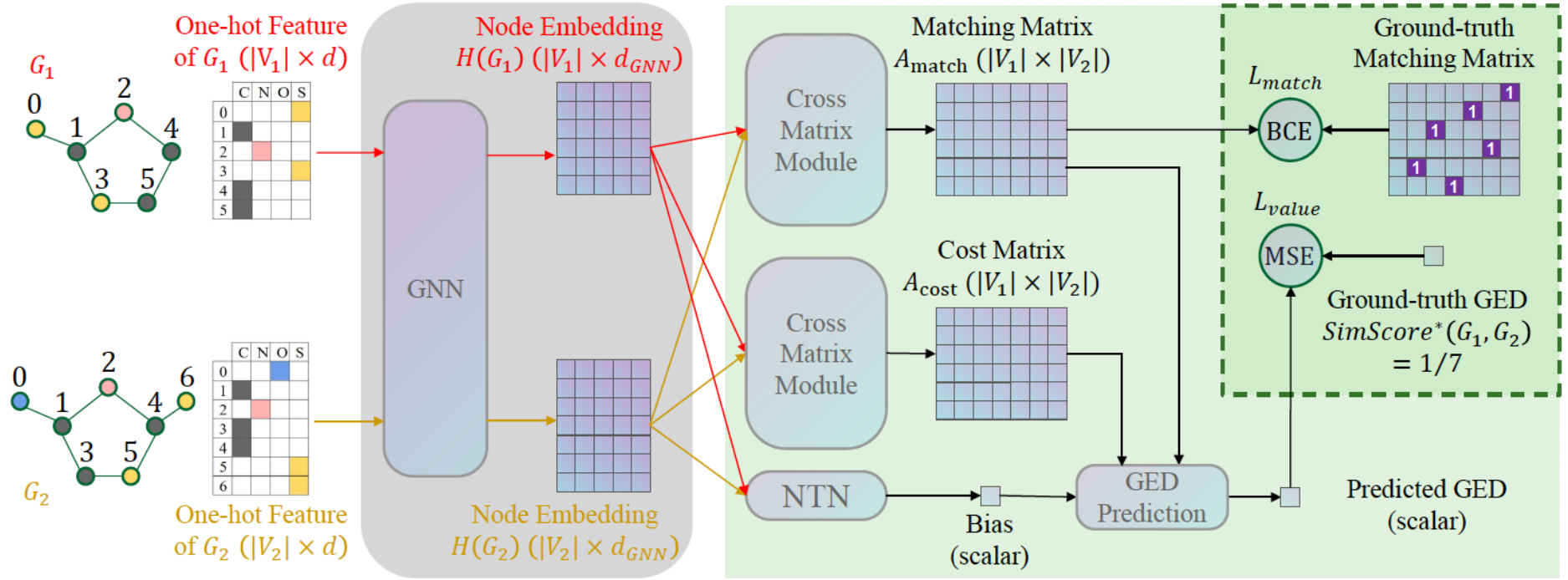
Piao, Chengzhi, et al. "Computing graph edit distance via neural graph matching." Proceedings of the VLDB Endowment 16.8 (2023): 1817-1829.

Similarity and Graph Generation

GEDGNN: Computing Graph Edit Distance via Neural Graph Matching

A Two-step Framework:

- Using GNN to predict a GED and generate a node matching matrix.
- Post-processing the node matching matrix to find a short edit path.



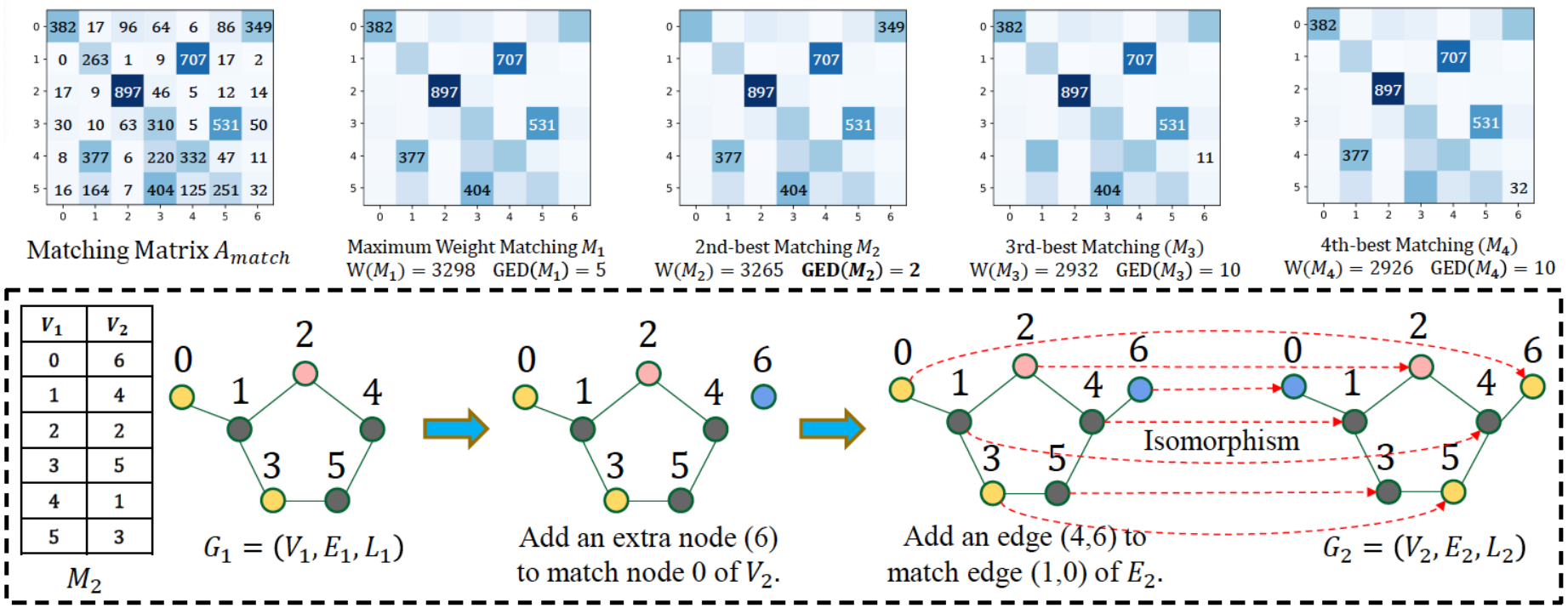
Piao, Chengzhi, et al. "Computing graph edit distance via neural graph matching." Proceedings of the VLDB Endowment 16.8 (2023): 1817-1829.

Similarity and Graph Generation

GEDGNN: Computing Graph Edit Distance via Neural Graph Matching

A Two-step Framework:

- Using GNN to predict a GED and generate a node matching matrix.
- Post-processing the node matching matrix to find a short edit path.



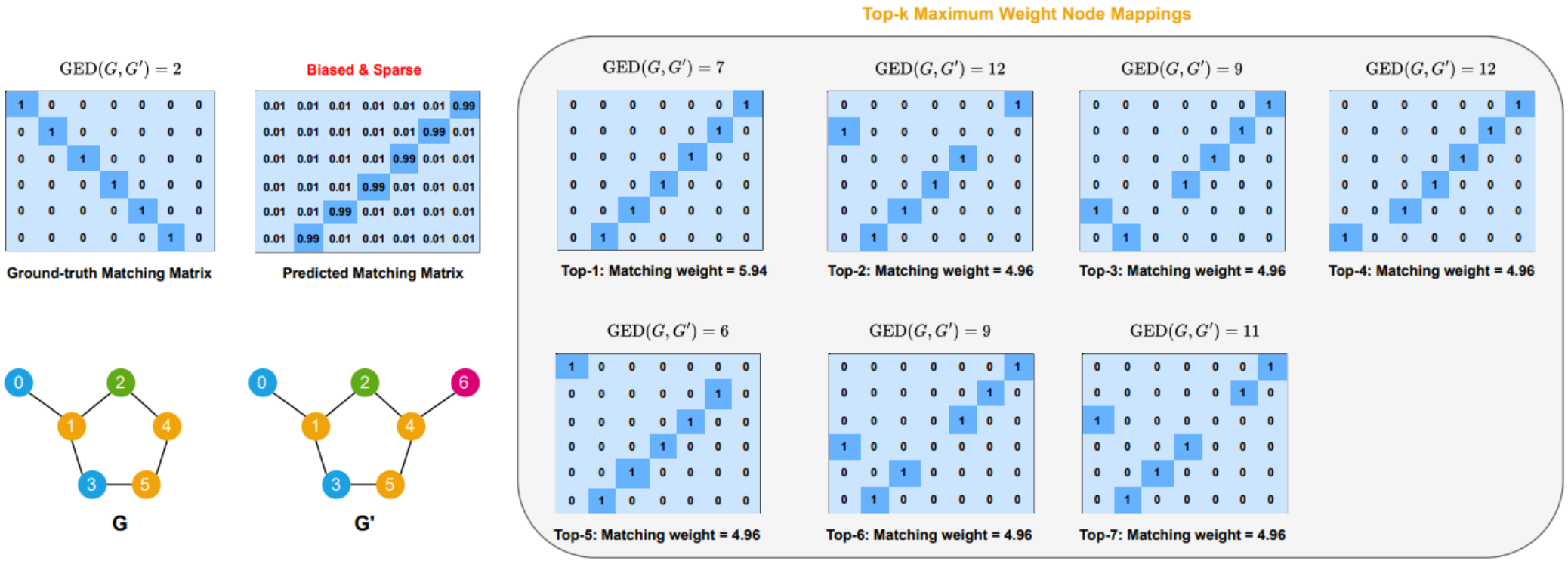
Piao, Chengzhi, et al. "Computing graph edit distance via neural graph matching." Proceedings of the VLDB Endowment 16.8 (2023): 1817-1829.

Similarity and Graph Generation

DiffGED: Computing Graph Edit Distance via Diffusion-based Graph Matching

Can diffusion models be applied on Graph Edit Distance Computation?

Diffusion models for generation of (bipartite) graph matching.

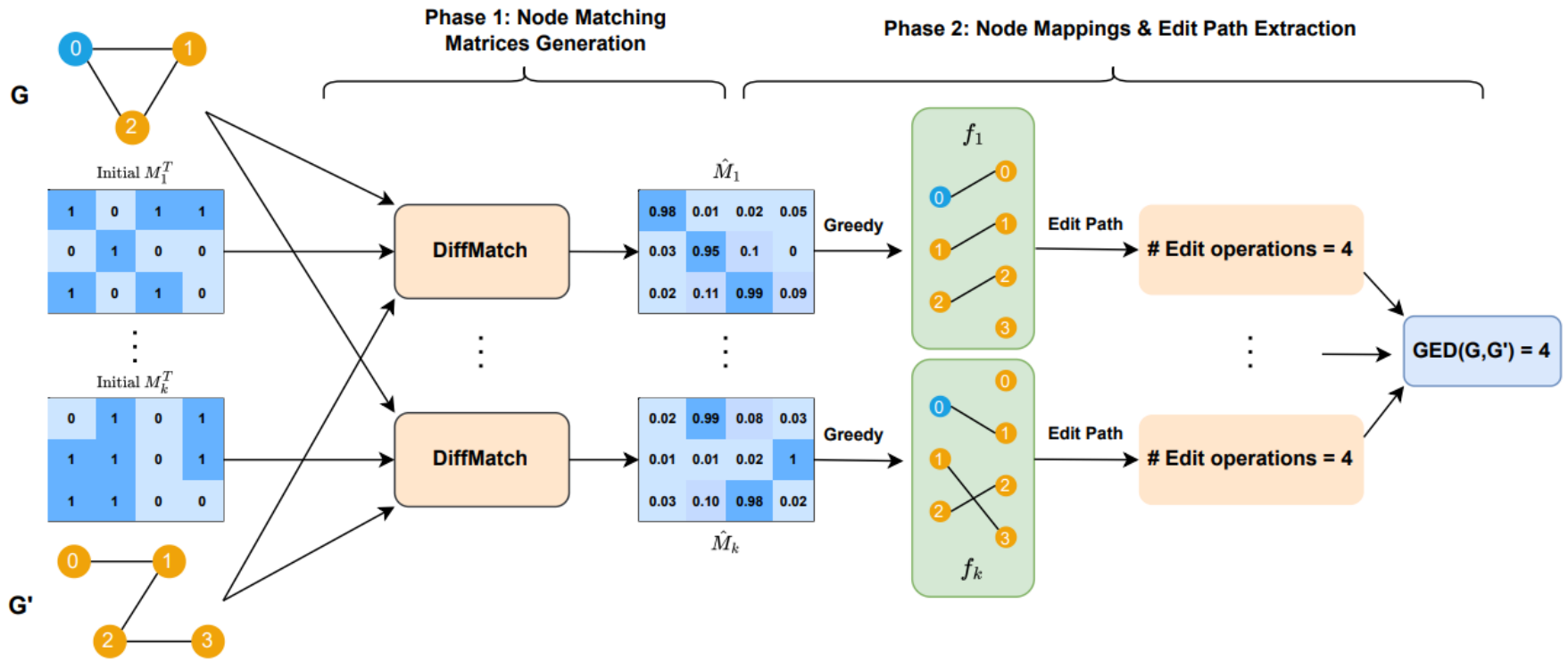


Huang, Wei, et al. "DiffGED: Computing Graph Edit Distance via Diffusion-based Graph Matching." *arXiv preprint arXiv:2503.18245* (2025).

Similarity and Graph Generation

DiffGED: Computing Graph Edit Distance via Diffusion-based Graph Matching

In the first phase, DiffGED first samples k random initial node matching matrices, then DiffMatch will denoise the sampled node matching matrices.
In the second phase, one node mapping will be extracted from each node matching matrix in parallel, and edit paths will be derived from the node mappings.



Huang, Wei, et al. "DiffGED: Computing Graph Edit Distance via Diffusion-based Graph Matching." *arXiv preprint arXiv:2503.18245* (2025).

Similarity and Graph Generation

DiffGED: Computing Graph Edit Distance via Diffusion-based Graph Matching

Experimental results: achieve state-of-the-art performance with nearly 100% accuracy.

Datasets	Models	MAE	Accuracy	ρ	τ	p@10	p@20	Time(s)
AIDS700	Hungarian	8.247	1.1%	0.547	0.431	52.8%	59.9%	0.00011
	VJ	14.085	0.6%	0.372	0.284	41.9%	52%	0.00017
	Noah	3.057	6.6%	0.751	0.629	74.1%	76.9%	0.6158
	GENN-A*	0.632	61.5%	0.903	0.815	85.6%	88%	2.98919
	GEDGNN	1.098	52.5%	0.845	0.752	89.1%	88.3%	0.39448
	MATA*	0.838	58.7%	0.8	0.718	73.6%	77.6%	**0.00487**
	DiffGED (ours)	**0.022**	**98%**	**0.996**	**0.992**	**99.8%**	**99.7%**	0.0763
Linux	Hungarian	5.35	7.4%	0.696	0.605	74.8%	79.6%	0.00009
	VJ	11.123	0.4%	0.594	0.5	72.8%	76%	0.00013
	Noah	1.596	9%	0.9	0.834	92.6%	96%	0.24457
	GENN-A*	0.213	89.4%	0.954	0.905	99.1%	98.1%	0.68176
	GEDGNN	0.094	96.6%	0.979	0.969	98.9%	99.3%	0.12863
	MATA*	0.18	92.3%	0.937	0.893	88.5%	91.8%	**0.00464**
	DiffGED (ours)	**0.0**	**100%**	**1.0**	**1.0**	**100%**	**100%**	0.06982
IMDB	Hungarian	21.673	45.1%	0.778	0.716	83.8%	81.9%	0.0001
	VJ	44.078	26.5%	0.4	0.359	60.1%	62%	0.00038
	Noah	-	-	-	-	-	-	-
	GENN-A*	-	-	-	-	-	-	-
	GEDGNN	2.469	85.5%	0.898	0.879	92.4%	92.1%	0.42428
	MATA*	-	-	-	-	-	-	-
	DiffGED (ours)	**0.937**	**94.6%**	**0.982**	**0.973**	**97.5%**	**98.3%**	**0.15105**

Huang, Wei, et al. "DiffGED: Computing Graph Edit Distance via Diffusion-based Graph Matching." *arXiv preprint arXiv:2503.18245* (2025).

41st IEEE International Conference
on Data Engineering
HONG KONG SAR, CHINA | MAY 19 – 23, 2025

AIGC for Graphs: Current Techniques and Future Trends

Function-driven Graph Generation

Speaker:
Hanchen Wang

Contributors: Hanchen Wang, Dawei Cheng, Ying Zhang and Wenjie Zhang

UTS UNIVERSITY OF TECHNOLOGY SYDNEY
UNSW THE UNIVERSITY OF NEW SOUTH WALES

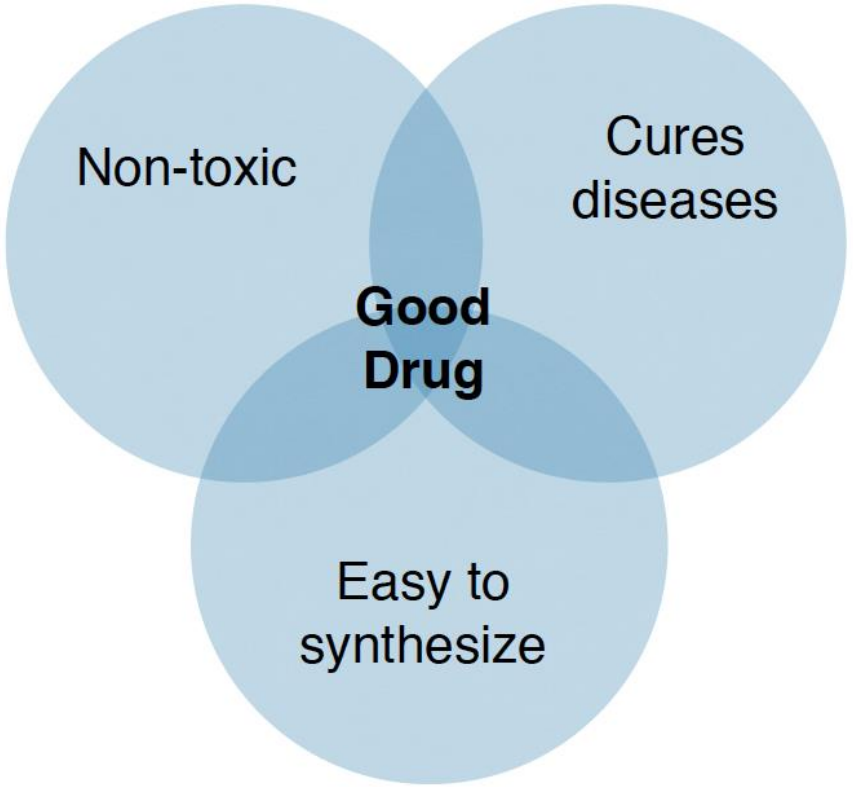
Function-Driven Graph Generation

- **Molecular Graph Generation**
- Protein Structure Generation
- Material Graph Generation

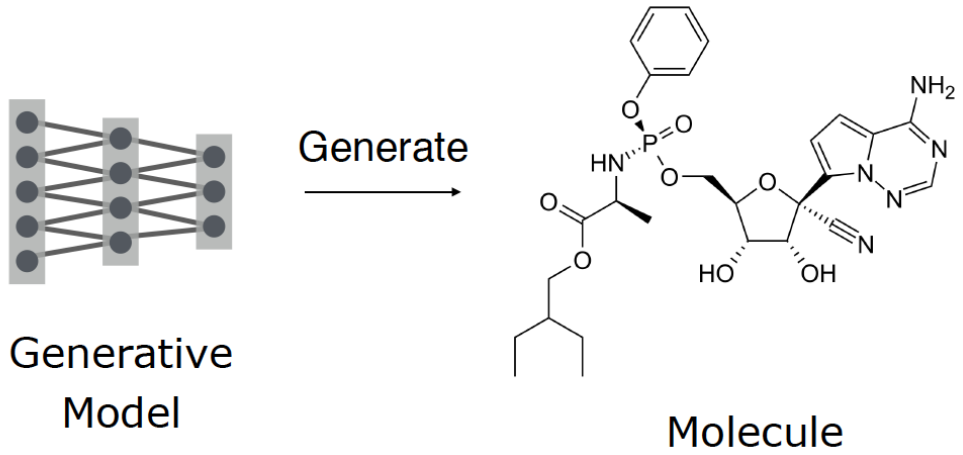
Molecular Graph Generation: applications

Drug Discovery: finding molecules with desired chemical properties.

A good drug needs to satisfy multiple objectives:



- The scale of potential drug-like molecules: $10^{33} \sim 10^{60}$
- The scale of existing chemical database: 10^{6}
- A huge gap!

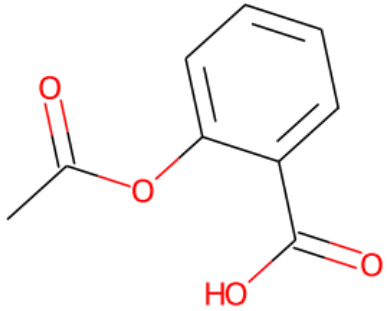
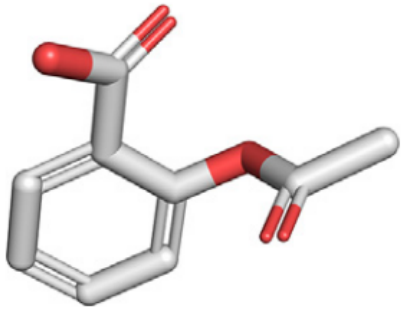


Jin, Wengong, Regina Barzilay, and Tommi Jaakkola. "Multi-objective molecule generation using interpretable substructures." *ICML*. PMLR, 2020.
Zang, Chengxi, and Fei Wang. "Moflow: an invertible flow model for generating molecular graphs." *Proceedings of the 26th ACM SIGKDD*. 2020.

Molecular Graph Generation: representations

Representation of molecular graphs

Three different representation methods of a specific molecule.

Representation Method	Specific Expression
1D	CC(=O)OC1=CC=CC=C1C(=O)O
2D	
3D	

Yang, Nianzu, et al. "Molecule generation for drug design: a graph learning perspective." *Fundamental Research* (2024).

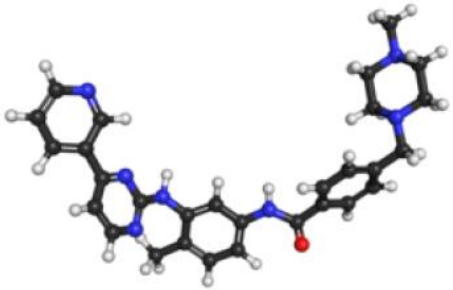
Molecular Graph Generation: representations

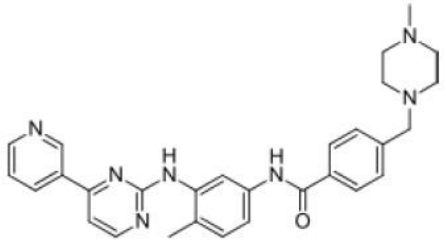
Representation of molecular graphs: SMILES

Chemistry is a language

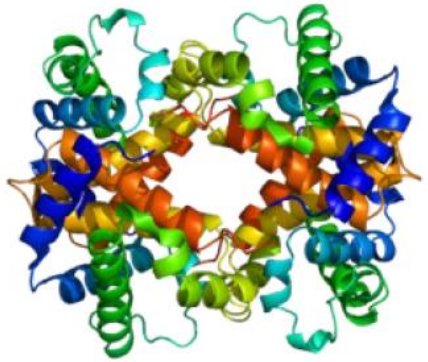
Machine Translation

Chemie ist eine Sprache





Cc1ccc(cc1Nc2nccc(n2)c3cccnc3)NC(=O)c4ccc(cc4)CN5CCN(CC5)C



VLSPADKTNVKAAW
GKVGAHAGEYGAE
ALERMFLSFPTTKTY
FPHFDLSHGSAQV...

actcttctggtccccacagactcag
agagaacccaccatggtgctgtct
cctgccgacaa gaccaacgtca
aggccgcctggggtaaggt ...

Edwards, Carl, Qingyun Wang, and Heng Ji. "Language+ molecules." *ACL: Tutorial Abstracts*. 2024.

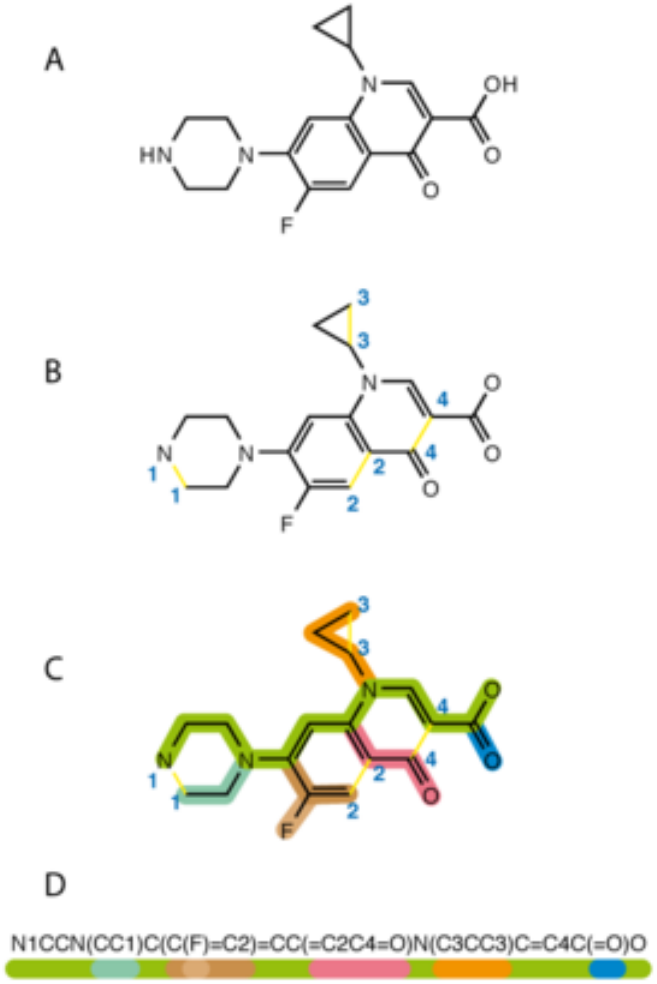
Molecular Graph Generation: representations

Representation of molecular graphs: SMILES

- The best known string representation for molecules is Simplified molecular-input line-entry system, commonly known as SMILES.
- SMILES strings have been used to store molecular structures for decades.
- GPT models have been exposed to SMILES in their training data.

How does SMILES work?

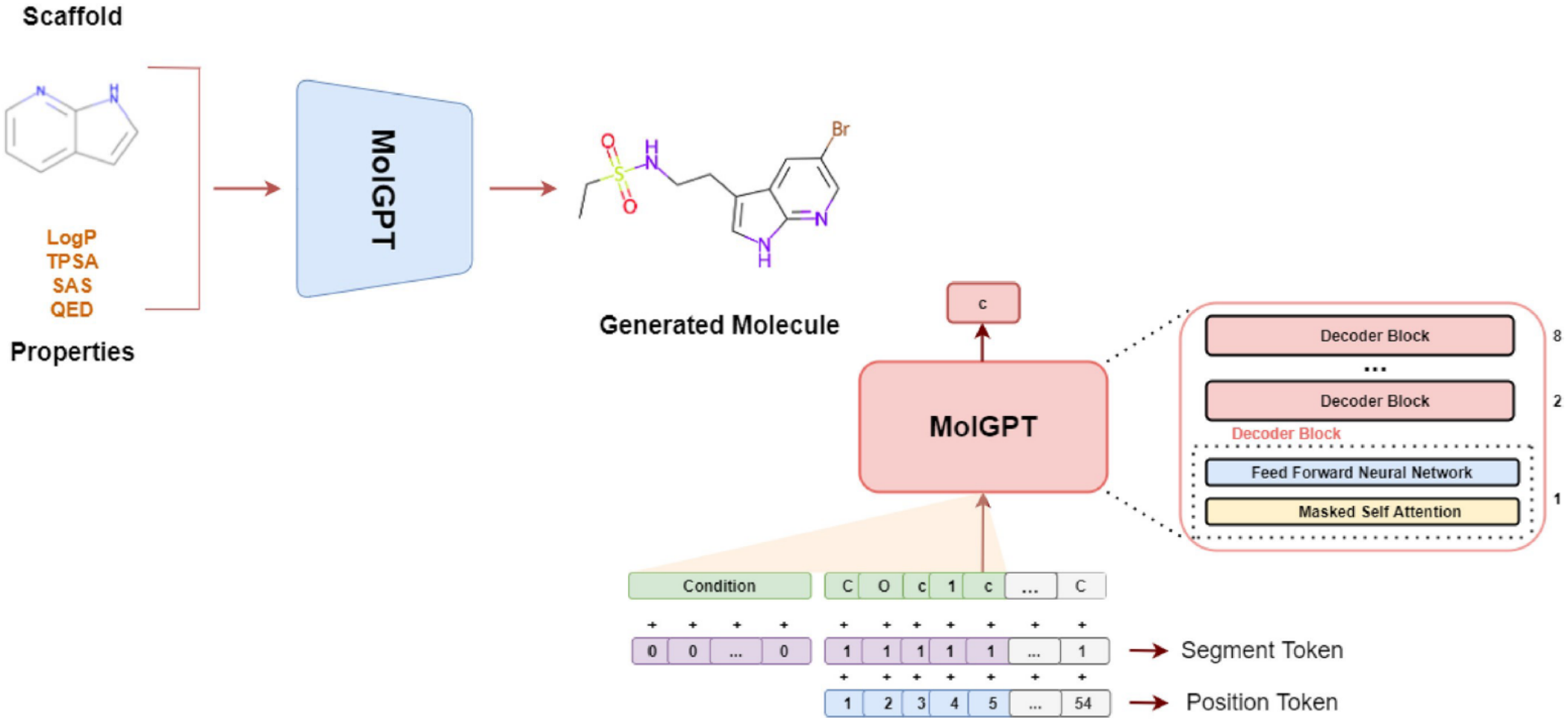
- Atoms are represented as 1-2 characters, such as ‘C’ for carbon, ‘Br’ for bromine, and ‘F’ for fluorine.
- Rings are created through the use of numbers.
- Branches are created with parenthesis.
- Hydrogens are usually implicit.



Edwards, Carl, Qingyun Wang, and Heng Ji. "Language+ molecules." *ACL: Tutorial Abstracts*. 2024.

Molecular Graph Generation: Models

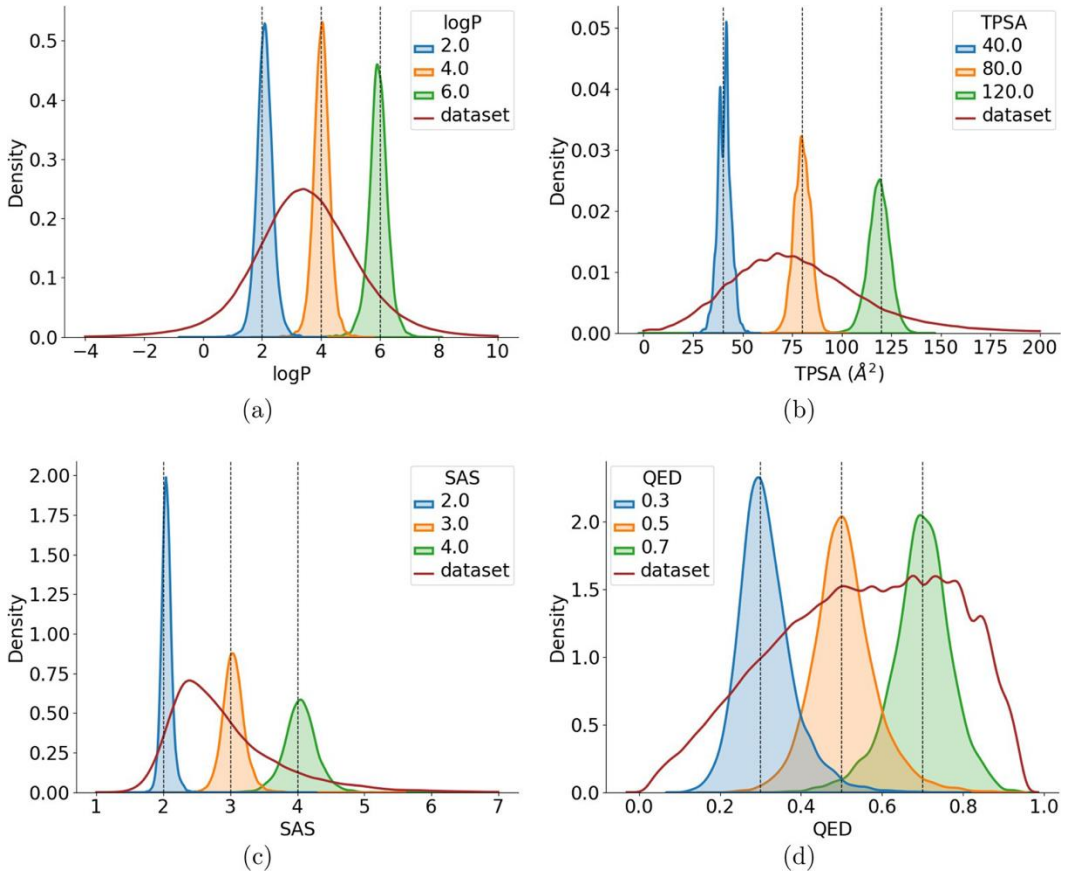
Language models for molecular graph generation



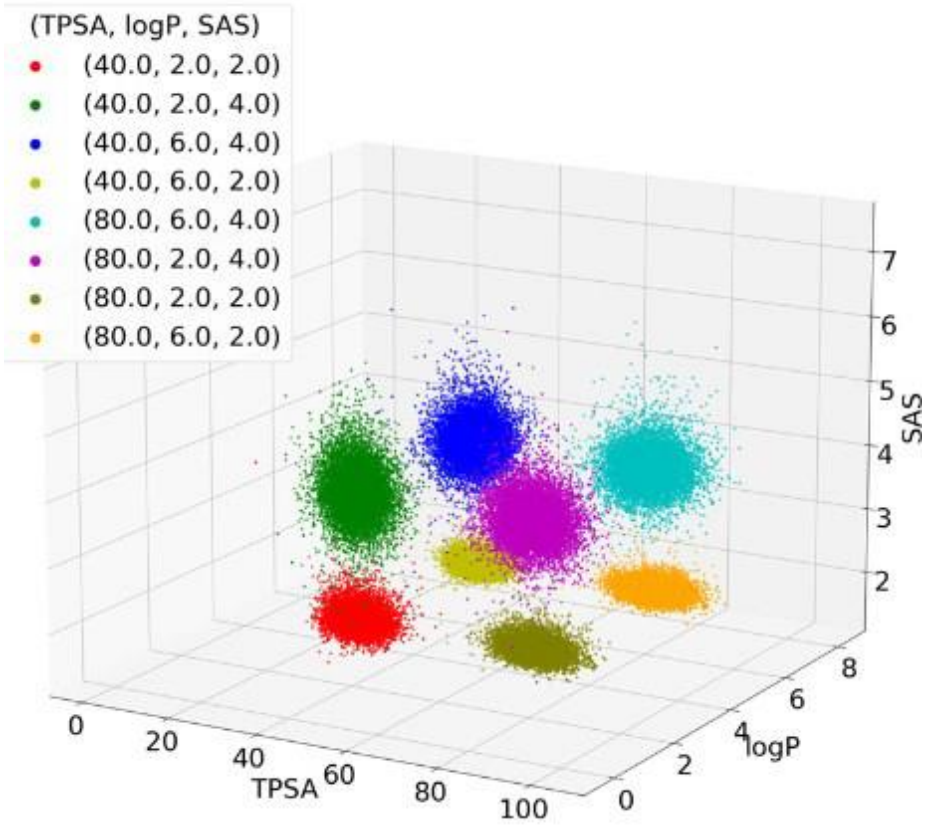
Bagal, Viraj, et al. "MolGPT: molecular generation using a transformer-decoder model." Journal of chemical information and modeling 62.9 (2021): 2064-2076.

Molecular Graph Generation: Models

Language models for molecular graph generation



Conditioning on properties:
(a) logP, (b) TPSA, (c) SAS, and (d) QED

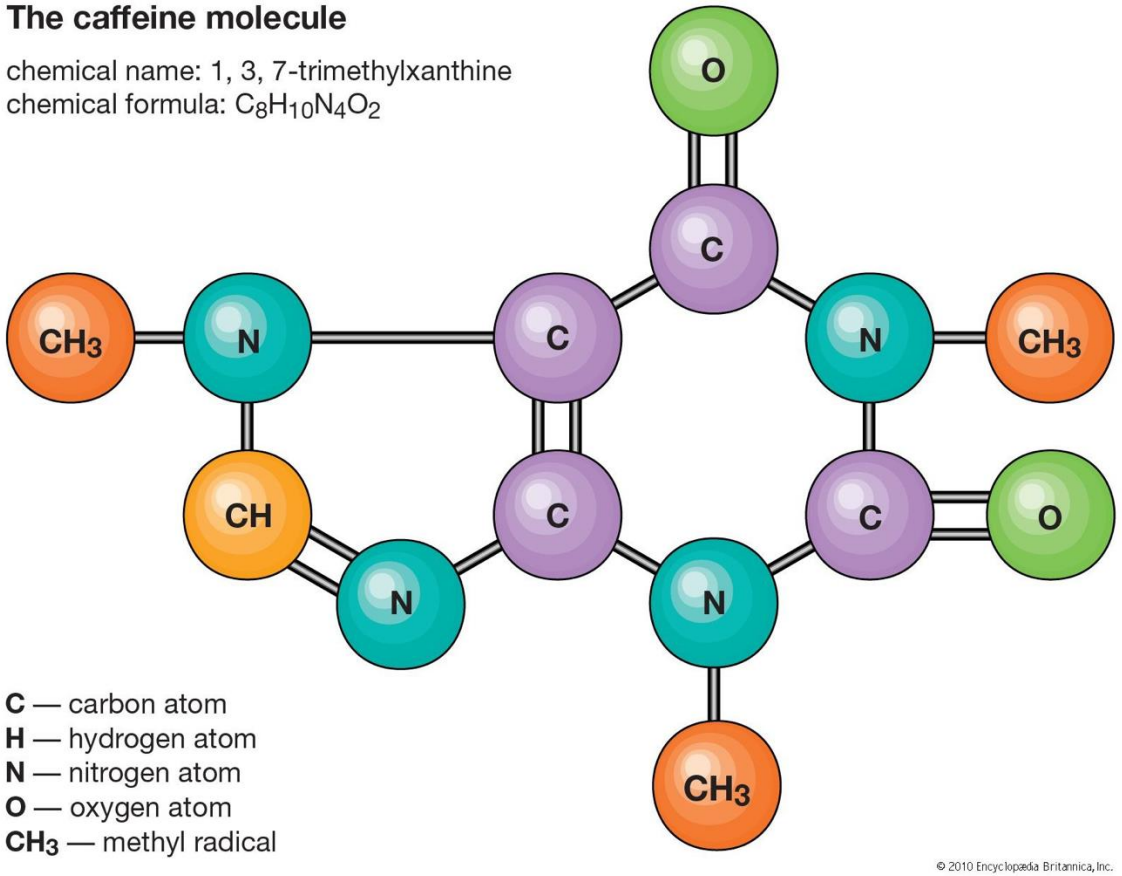


Conditioning on multiple properties at once

Bagal, Viraj, et al. "MolGPT: molecular generation using a transformer-decoder model." Journal of chemical information and modeling 62.9 (2021): 2064-2076.

Molecular Graph Generation: representations

Representation of molecular graphs: graphs



Nodes: Atoms

Edges: Chemical bonds between atoms

Edwards, Carl, Qingyun Wang, and Heng Ji. "Language+ molecules." *ACL: Tutorial Abstracts*. 2024.

Molecular Graph Generation: Models

Goal of Molecule Graph Generation

Generating realistic, novel and unique molecules with desired property.

e.g. drug-likeness, octanol-water partition coefficient

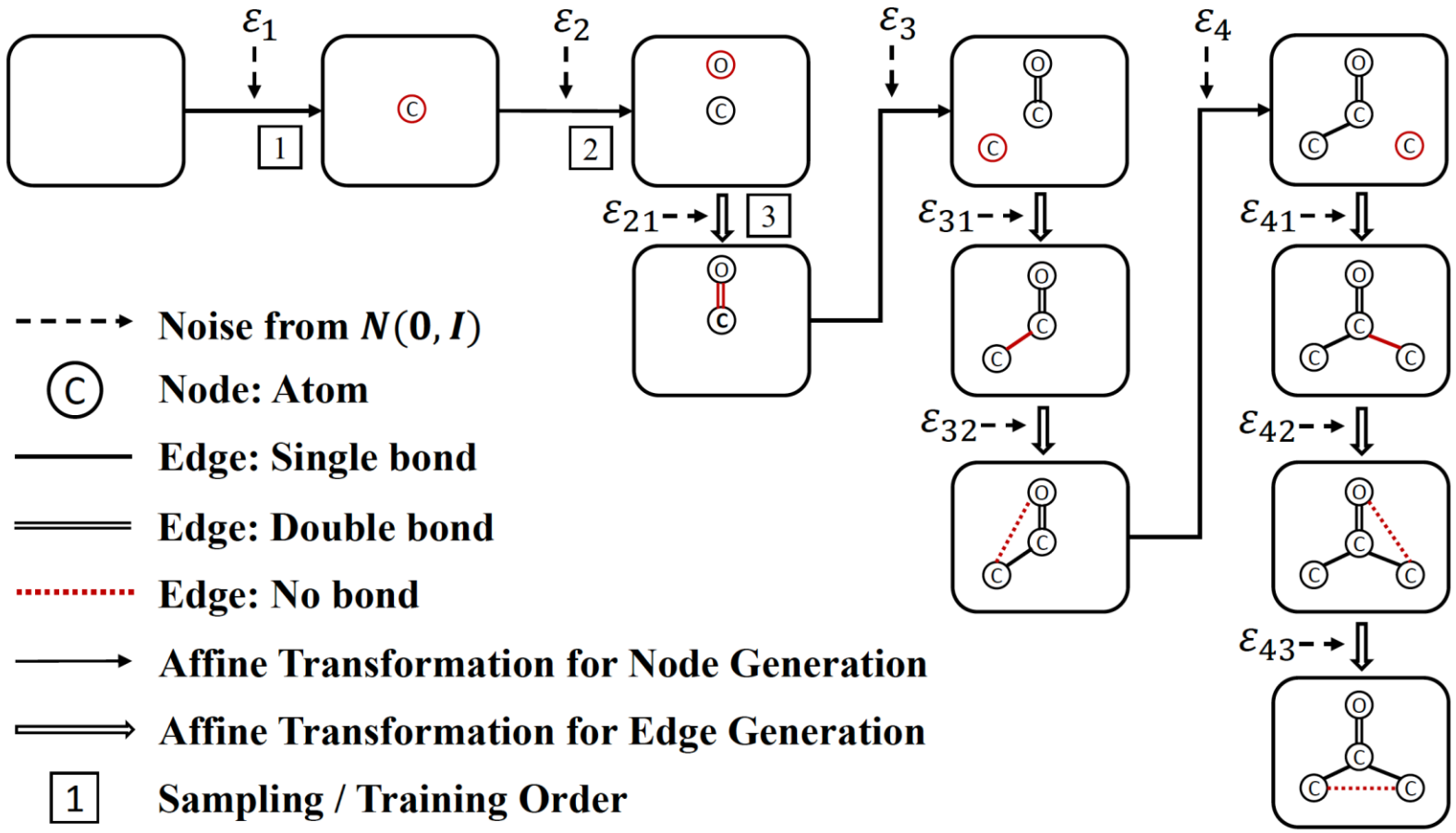
Shi, Chence, et al. "GraphAF: a Flow-based Autoregressive Model for Molecular Graph Generation." ICLR 2020.

Molecular Graph Generation: Models

GraphAF: a Flow-based Autoregressive Model for Molecular Graph Generation

Key Idea

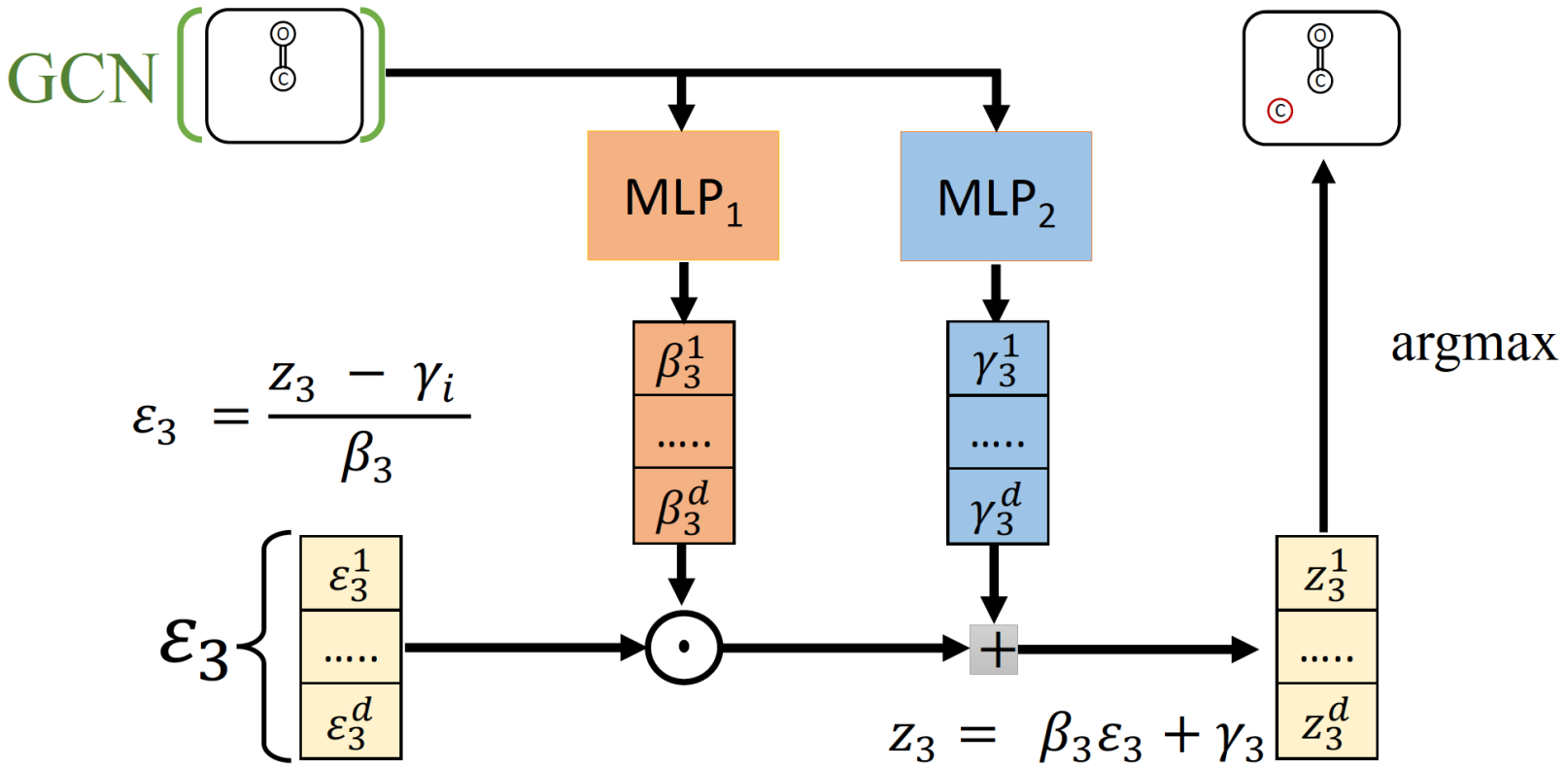
- Decompose molecular graphs into sequences
- Use autoregressive flows to model the sequences



Shi, Chence, et al. "GraphAF: a Flow-based Autoregressive Model for Molecular Graph Generation." ICLR 2020.

Molecular Graph Generation: Models

GraphAF: Model Framework



Shi, Chence, et al. "GraphAF: a Flow-based Autoregressive Model for Molecular Graph Generation." ICLR 2020.

Molecular Graph Generation: Models

GraphAF: Goal-Directed Molecule Generation with RL

For drug discovery, we also want model to be able to optimize the chemical properties of generated molecule.

- **State**: current sub-graph.
- **Policy**: autoregressive flow to generate node/edge based on current subgraph.
- **Reward**: intermediate reward and final reward

Shi, Chence, et al. "GraphAF: a Flow-based Autoregressive Model for Molecular Graph Generation." ICLR 2020.

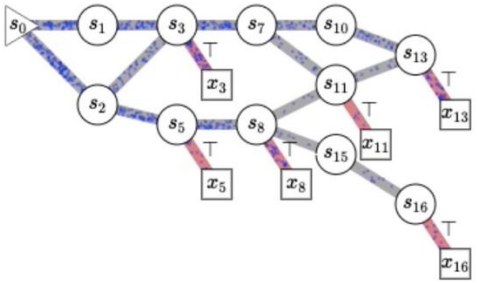
Molecular Graph Generation: Models

RGFN: Synthesizable Molecular Generation Using GFlowNets: Why FlowNet?

Generative Flow Networks (GFlowNets) are a relatively new family of generative models.

Goal: generating **high reward**, **diverse** samples in an **amortized** manner. All crucial in drug discovery!

Shortcomings of the existing methods:
MCMC - lack of amortization,
RL - mean-seeking behaviour; mode collapse.

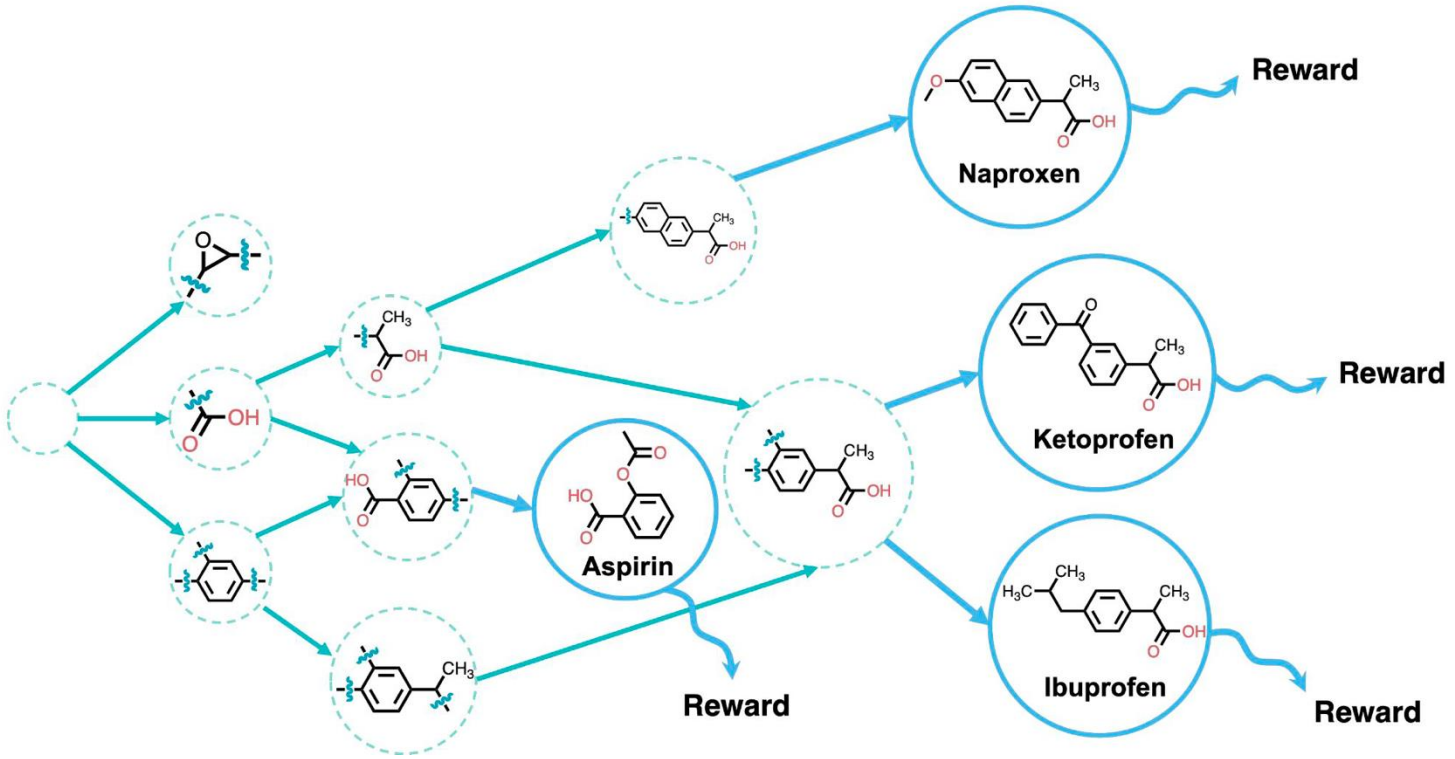


How to do it? On high level: ensure that the probability of generating a sample is proportional to its reward: ***p(x) ~ R(x)***. This can be done by training a sampling policy ***π(x)*** (a machine learning model).

Koziarski, Michał, et al. "Rgfn: Synthesizable molecular generation using gflownets." Advances in Neural Information Processing Systems 37 (2024): 46908-46955.

Molecular Graph Generation: Models

GFlowNet for Molecule Design



Key ingredients of GFlowNets:

State = current molecule
Action space = fragments to add
Reward function = property of interest

How do we ensure molecules are synthesizable?

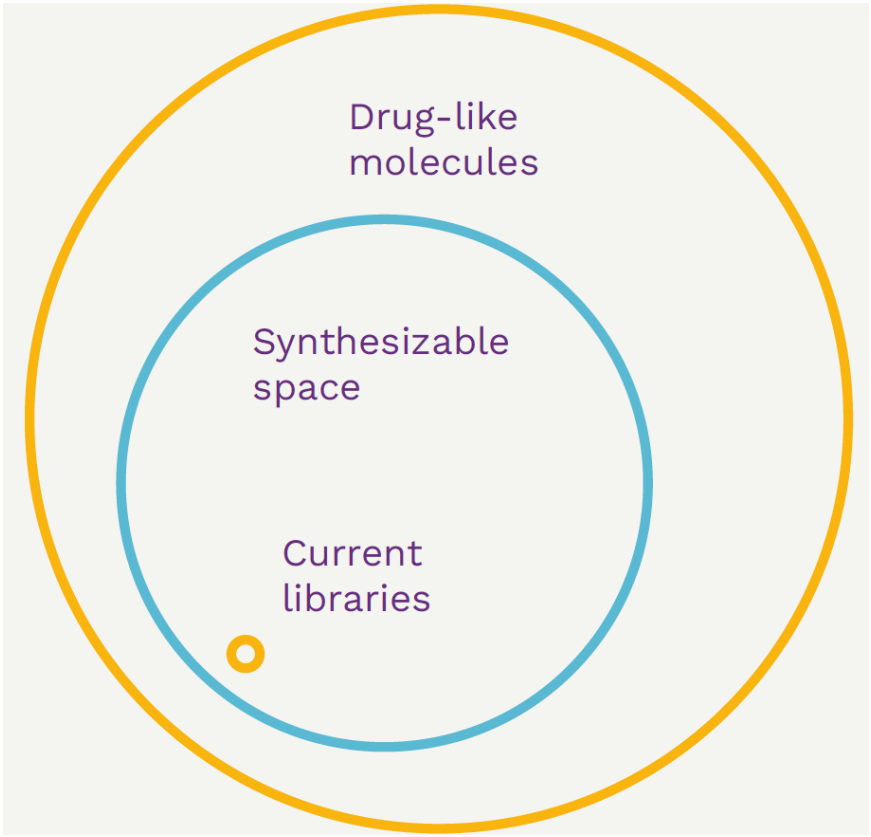
Koziarski, Michał, et al. "Rgfn: Synthesizable molecular generation using gflownets." Advances in Neural Information Processing Systems 37 (2024): 46908-46955.

Molecular Graph Generation: Models

GFlowNet for Molecule Design

The goal: constrain the searchable space to highly synthesizable compounds.

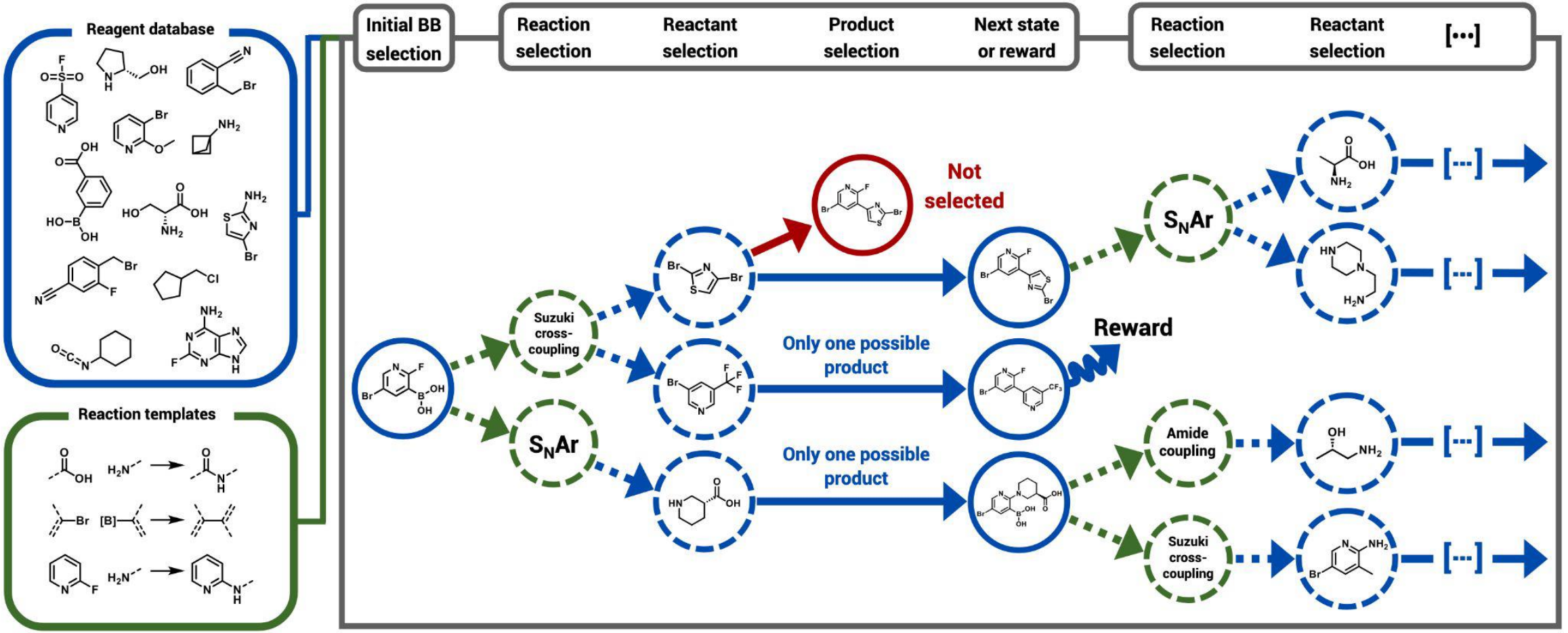
(while increasing the search space size as much as possible!)



Koziarski, Michał, et al. "Rgfn: Synthesizable molecular generation using gflownets." Advances in Neural Information Processing Systems 37 (2024): 46908-46955.

Molecular Graph Generation: Models

RGFN: Synthesizable Molecular Generation Using GFlowNets: Model



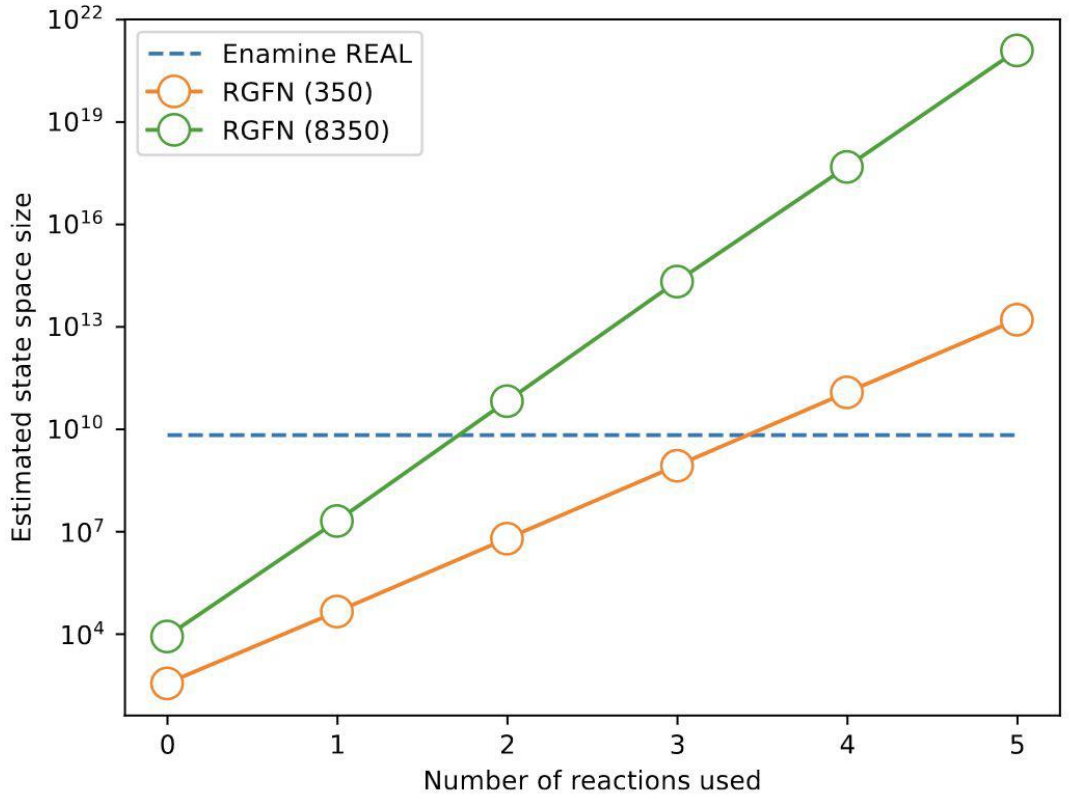
Koziarski, Michał, et al. "Rgfn: Synthesizable molecular generation using gflownets." Advances in Neural Information Processing Systems 37 (2024): 46908-46955.

Molecular Graph Generation: Models

RGFN: Synthesizable Molecular Generation Using GFlowNets: Model

We select a total of **350** affordable reagents (≤$200/g) **and 17** high-yield reactions.

Molecule space generated by this approach with depth 4 is **larger than most compound libraries.**



Koziarski, Michał, et al. "Rgfn: Synthesizable molecular generation using gflownets." Advances in Neural Information Processing Systems 37 (2024): 46908-46955.

Molecular Graph Generation: Models

Molecule Generation with Fragment Retrieval Augmentation

Motivation

Fragment-based drug discovery (FBDD) has been considered as an effective approach to explore the chemical space.

- Generative models have been adopted in the field of FBDD to accelerate the process.
- Many fragment-based molecule generation methods show limited exploration as they only reassemble or slightly modify the given fragments.
- FBDD + RAG → **Fragment Retrieval-Augmented Generation (*f*-RAG)**. f-RAG augments the pre-trained molecular language model SAFE-GPT with two types of retrieved fragments: **hard fragments and soft fragments**.

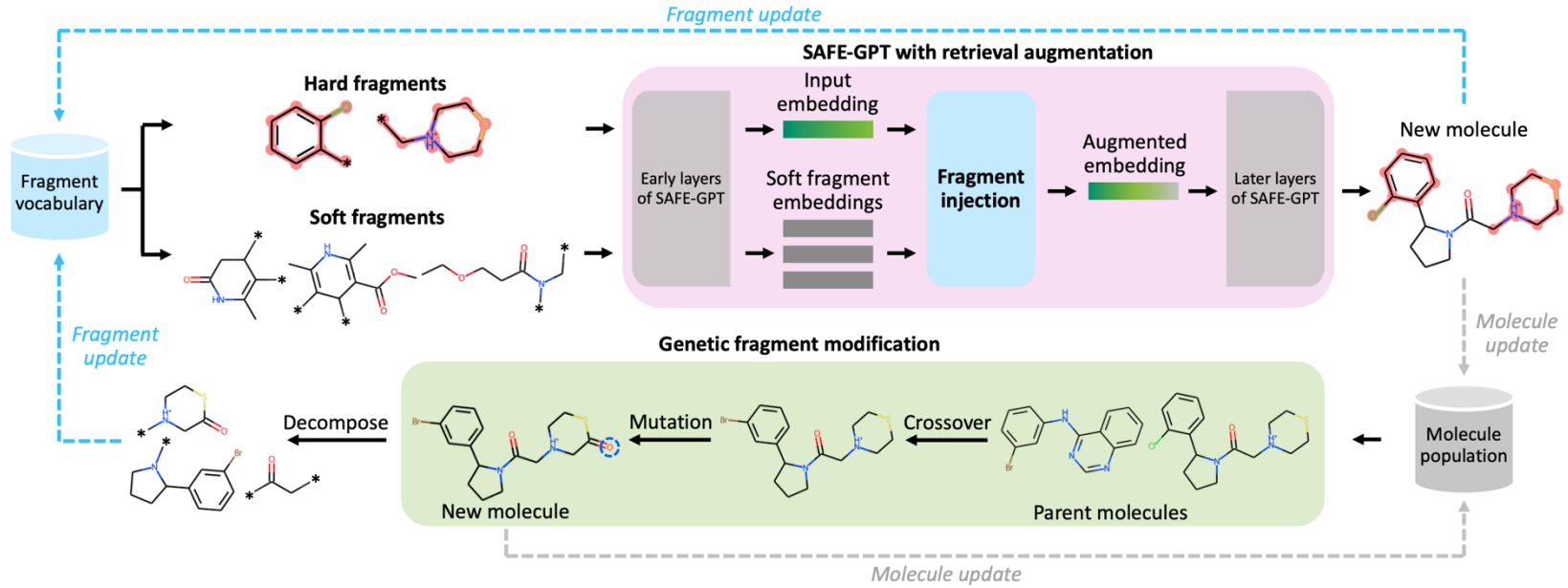
Lee, Seul, et al. "Molecule generation with fragment retrieval augmentation." Advances in Neural Information Processing Systems 37 (2024): 132463-132490.

Molecular Graph Generation: Models

Molecule Generation with Fragment Retrieval Augmentation

Construct a fragment vocabulary.

- Decompose known molecules from the existing library into fragments and scoring the fragments.



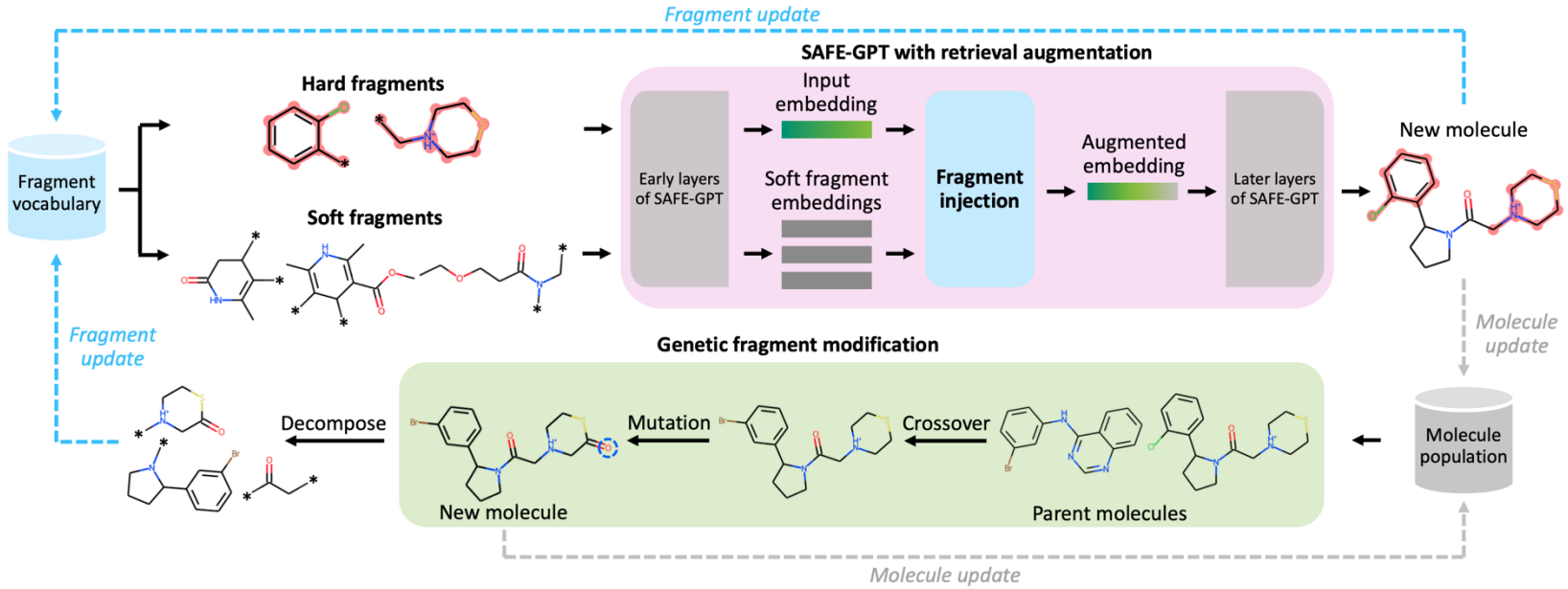
Lee, Seul, et al. "Molecule generation with fragment retrieval augmentation." Advances in Neural Information Processing Systems 37 (2024): 132463-132490.

Molecular Graph Generation: Models

Molecule Generation with Fragment Retrieval Augmentation

f-RAG retrieves fragments that will be explicitly included in the new molecule (i.e., **hard fragments**).

- Hard fragments serve as the input context to the molecular language model that predicts the remaining fragments.



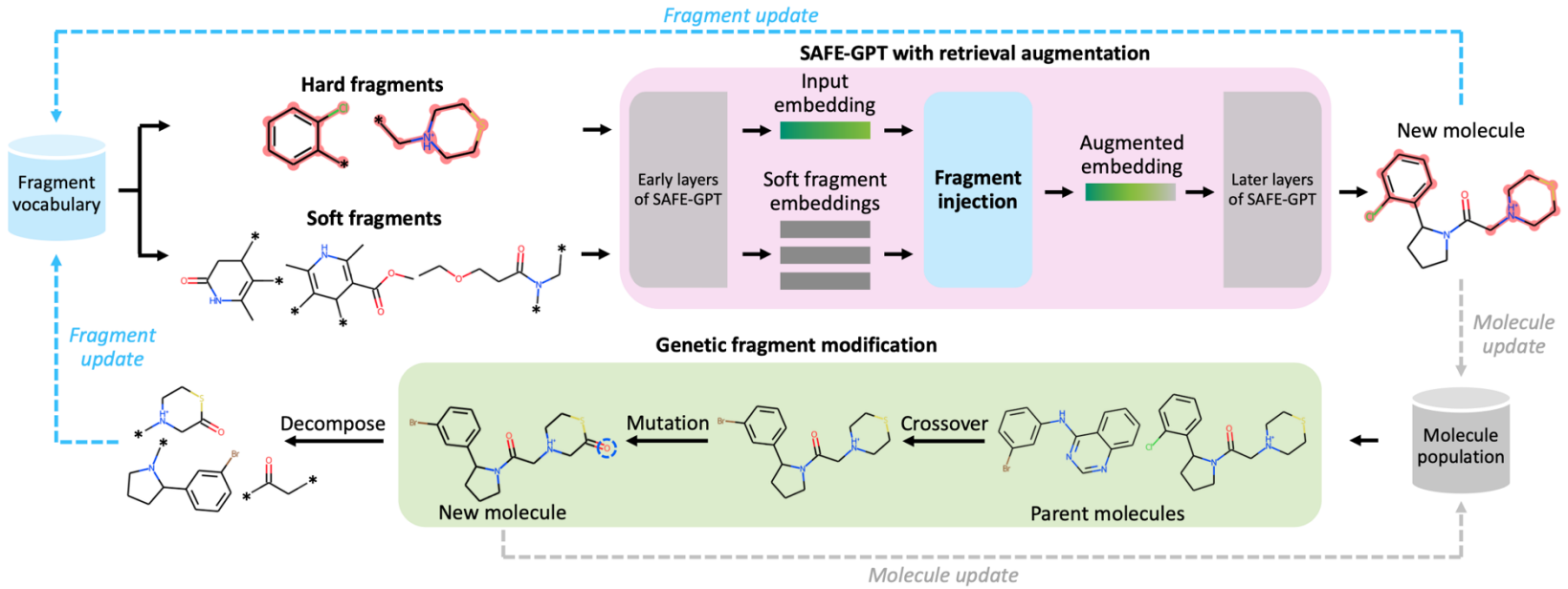
Lee, Seul, et al. "Molecule generation with fragment retrieval augmentation." Advances in Neural Information Processing Systems 37 (2024): 132463-132490.

Molecular Graph Generation: Models

Molecule Generation with Fragment Retrieval Augmentation

f-RAG retrieves fragments that will not be part of the generated molecule but provide guidance (i.e., **soft fragments**).

- The soft fragment embeddings are fused with the hard fragment embeddings through a lightweight **fragment injection module** in the middle of SAFE-GPT.

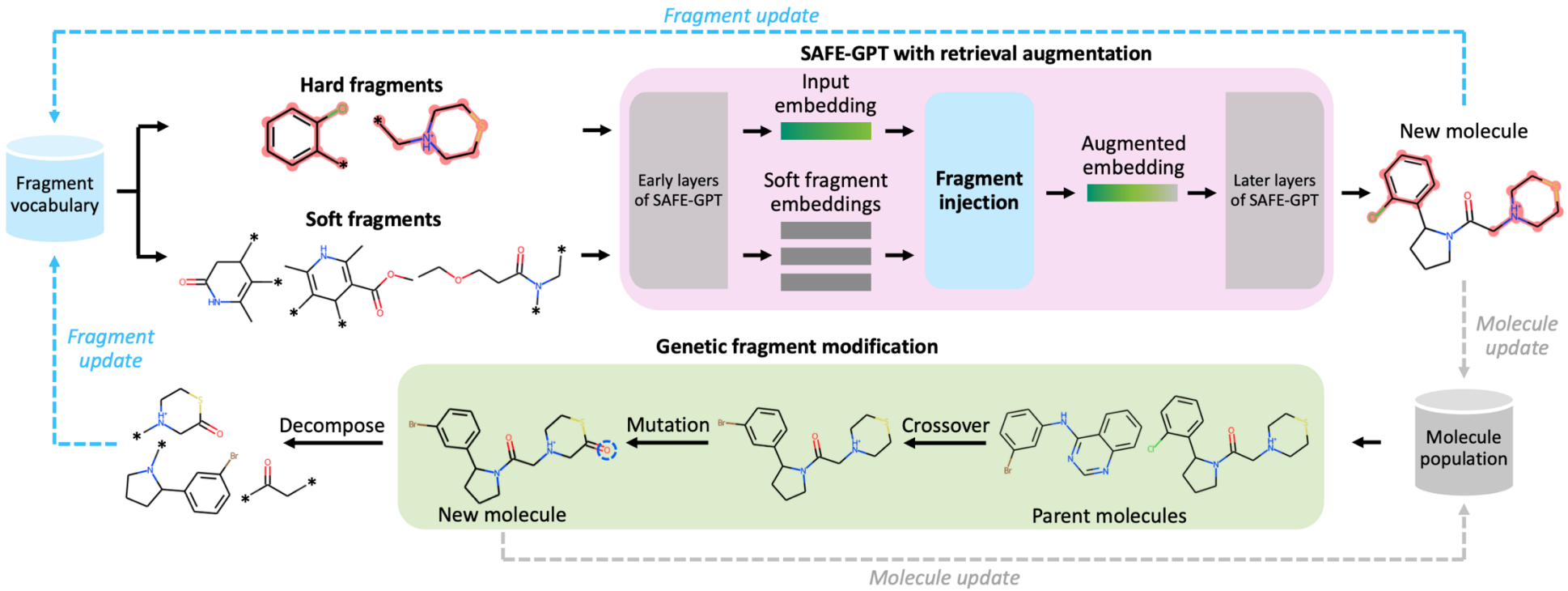


Lee, Seul, et al. "Molecule generation with fragment retrieval augmentation." Advances in Neural Information Processing Systems 37 (2024): 132463-132490.

Molecular Graph Generation: Models

Molecule Generation with Fragment Retrieval Augmentation

f-RAG updates the fragment vocabulary with generated fragments via an iterative refinement process which is further enhanced with **post-hoc genetic fragment modification**.



Lee, Seul, et al. "Molecule generation with fragment retrieval augmentation." Advances in Neural Information Processing Systems 37 (2024): 132463-132490.

Molecular Graph Generation: Models

Molecule Generation with Fragment Retrieval Augmentation: Experiment

f-RAG outperformed the previous methods in the PMO goal-directed hit generation benchmark.
f-RAG achieved improved trade-offs between optimization performance, diversity, novelty, and synthesizability.

Oracle	f-RAG (ours)	Genetic GFN	Mol GA	REINVENT	Graph GA
albuterol_similarity	**0.977** ± 0.002	0.949 ± 0.010	0.896 ± 0.035	0.882 ± 0.006	0.838 ± 0.016
amlodipine_mpo	0.749 ± 0.019	**0.761** ± 0.019	0.688 ± 0.039	0.635 ± 0.035	0.661 ± 0.020
celecoxib_rediscovery	0.778 ± 0.007	**0.802** ± 0.029	0.567 ± 0.083	0.713 ± 0.067	0.630 ± 0.097
deco_hop	**0.936** ± 0.011	0.733 ± 0.109	0.649 ± 0.025	0.666 ± 0.044	0.619 ± 0.004
drd2	**0.992** ± 0.000	0.974 ± 0.006	0.936 ± 0.016	0.945 ± 0.007	0.964 ± 0.012
fexofenadine_mpo	**0.856** ± 0.016	**0.856** ± 0.039	0.825 ± 0.019	0.784 ± 0.006	0.760 ± 0.011
gsk3b	**0.969** ± 0.003	0.881 ± 0.042	0.843 ± 0.039	0.865 ± 0.043	0.788 ± 0.070
isomers_c7h8n2o2	0.955 ± 0.008	**0.969** ± 0.003	0.878 ± 0.026	0.852 ± 0.036	0.862 ± 0.065
isomers_c9h10n2o2pf2cl	0.850 ± 0.005	**0.897** ± 0.007	0.865 ± 0.012	0.642 ± 0.054	0.719 ± 0.047
jnk3	**0.904** ± 0.004	0.764 ± 0.069	0.702 ± 0.123	0.783 ± 0.023	0.553 ± 0.136
median1	0.340 ± 0.007	**0.379** ± 0.010	0.257 ± 0.009	0.356 ± 0.009	0.294 ± 0.021
median2	**0.323** ± 0.005	0.294 ± 0.007	0.301 ± 0.021	0.276 ± 0.008	0.273 ± 0.009
mestranol_similarity	0.671 ± 0.021	**0.708** ± 0.057	0.591 ± 0.053	0.618 ± 0.048	0.579 ± 0.022
osimertinib_mpo	**0.866** ± 0.009	0.860 ± 0.008	0.844 ± 0.015	0.837 ± 0.009	0.831 ± 0.005
perindopril_mpo	**0.681** ± 0.017	0.595 ± 0.014	0.547 ± 0.022	0.537 ± 0.016	0.538 ± 0.009
qed	0.939 ± 0.001	**0.942** ± 0.000	0.941 ± 0.001	0.941 ± 0.000	0.940 ± 0.000
ranolazine_mpo	**0.820** ± 0.016	0.819 ± 0.018	0.804 ± 0.011	0.760 ± 0.009	0.728 ± 0.012
scaffold_hop	0.576 ± 0.014	**0.615** ± 0.100	0.527 ± 0.025	0.560 ± 0.019	0.517 ± 0.007
sitagliptin_mpo	0.601 ± 0.011	**0.634** ± 0.039	0.582 ± 0.040	0.021 ± 0.003	0.433 ± 0.075
thiothixene_rediscovery	**0.584** ± 0.009	0.583 ± 0.034	0.519 ± 0.041	0.534 ± 0.013	0.479 ± 0.025
troglitazone_rediscovery	0.448 ± 0.017	**0.511** ± 0.054	0.427 ± 0.031	0.441 ± 0.032	0.390 ± 0.016
valsartan_smarts	**0.627** ± 0.058	0.135 ± 0.271	0.000 ± 0.000	0.178 ± 0.358	0.000 ± 0.000
zaleplon_mpo	0.486 ± 0.004	**0.552** ± 0.033	0.519 ± 0.029	0.358 ± 0.062	0.346 ± 0.032
Sum	**16.928**	16.213	14.708	14.196	13.751

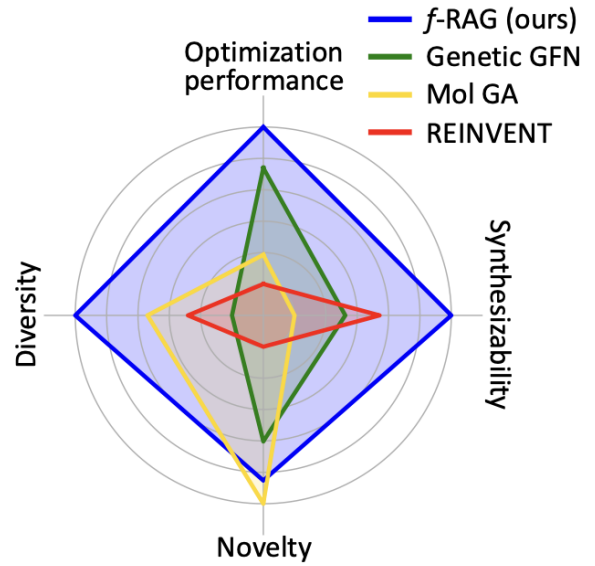


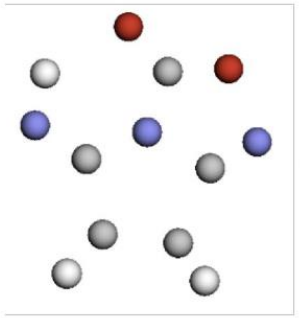
Figure 1: **A radar plot of target properties.** f-RAG strikes better balance among optimization performance, diversity, novelty, and synthesizability than the state-of-the-art techniques on the PMO benchmark [10].

Lee, Seul, et al. "Molecule generation with fragment retrieval augmentation." Advances in Neural Information Processing Systems 37 (2024): 132463-132490.

Molecular Graph Generation: 3D Generation

Background

3D point clouds



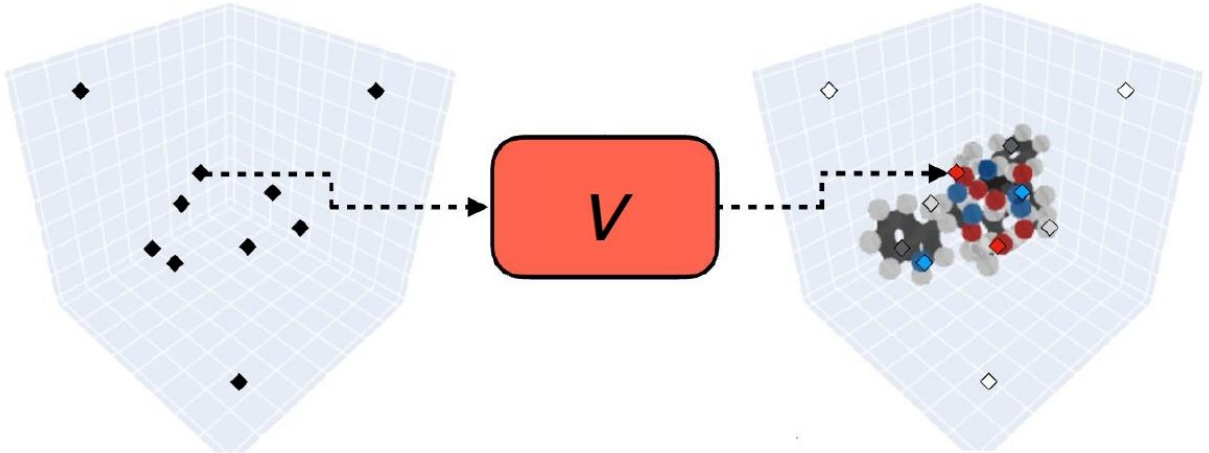
SE(3)-equivariant GNNs

+: Built-in equivariance

-: Not scalable (O(atoms^2))
Limited expressivity

Molecules as Images

Computer vision models + data augmentation



$x \in \mathbb{R}^3$

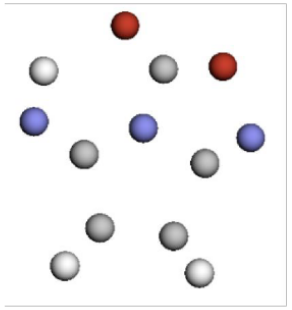
$(v_c, v_H, v_N, v_0)(x) \in \mathbb{R}^n$

Kirchmeyer, Matthieu, Pedro O. O Pinheiro, and Saeed Saremi. "Score-based 3D molecule generation with neural fields." NeurIPS (2024): 10646-10678.

Molecular Graph Generation: 3D Generation

Background

3D point clouds



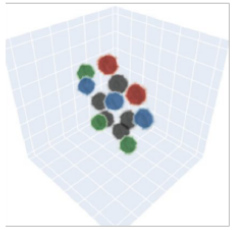
SE(3)-equivariant GNNs

+: Built-in equivariance

-: Not scalable (O(atoms^2))
Limited expressivity

Molecules as Images

Voxels

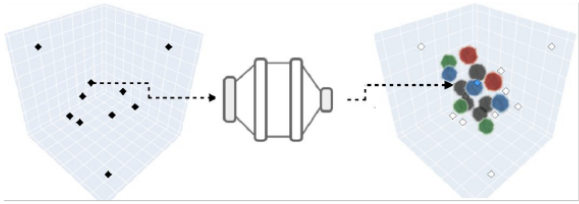


3D UNet

+: Expressive, state-of-the-art

-: Not scalable (O(grid^3))

Fields



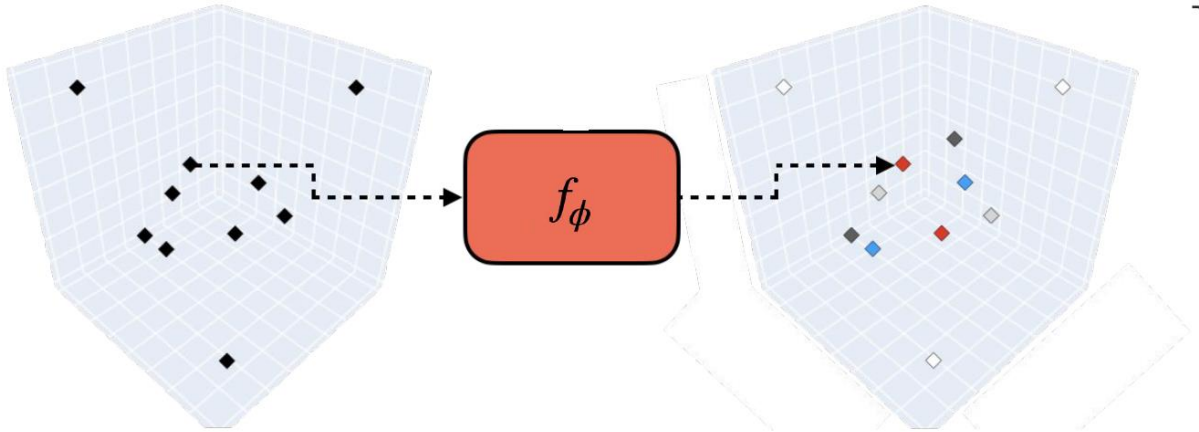
Neural Fields

+: Scalable, Expressive, Fast

Kirchmeyer, Matthieu, Pedro O. O Pinheiro, and Saeed Saremi. "Score-based 3D molecule generation with neural fields." NeurIPS (2024): 10646-10678.

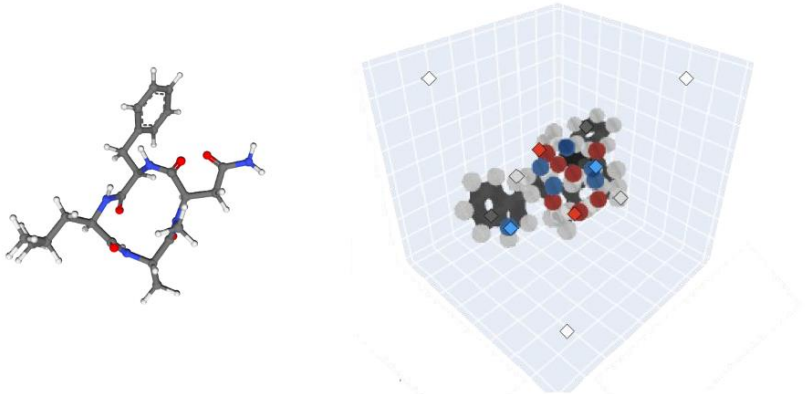
Molecular Graph Generation: 3D Generation

What is Neural Field



$x \in \mathbb{R}^3$

$f_\phi(x) = (f_c, f_H, f_N, f_O)(x) \in \mathbb{R}^n$



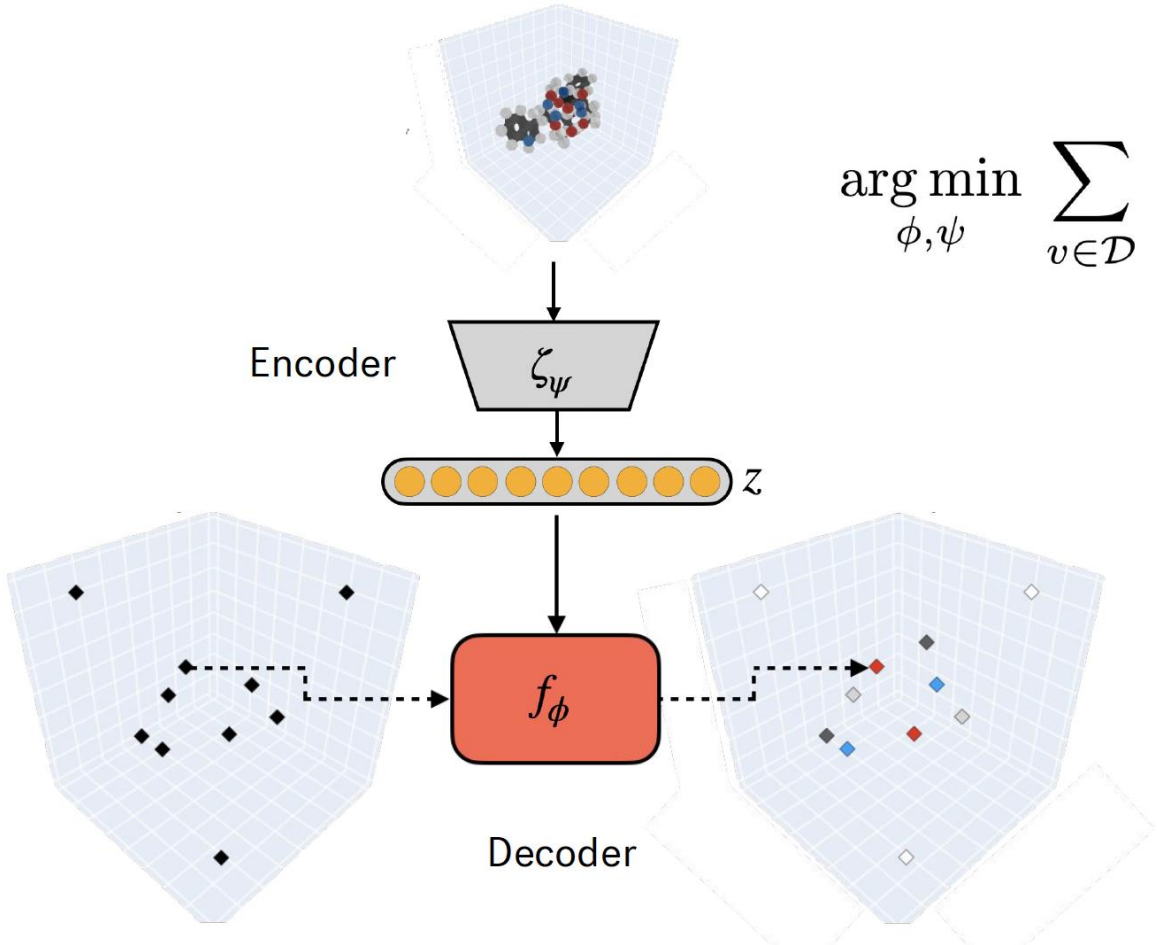
$v(x) = (v_c, v_H, v_N, v_O)(x) \in \mathbb{R}^n$

$$\arg\min_{\phi} \int \|f_\phi(x) - v(x)\|_2^2 \mathrm{d}x$$

Kirchmeyer, Matthieu, Pedro O. O Pinheiro, and Saeed Saremi. "Score-based 3D molecule generation with neural fields." NeurIPS (2024): 10646-10678.

Molecular Graph Generation: 3D Generation

Neural Field based Auto-encoder



$$\arg\min_{\phi,\psi} \sum_{v \in \mathcal{D}} \int \| f_\phi(x, \zeta_\psi(v)) - v(x) \|_2^2 \, \mathrm{d}x$$

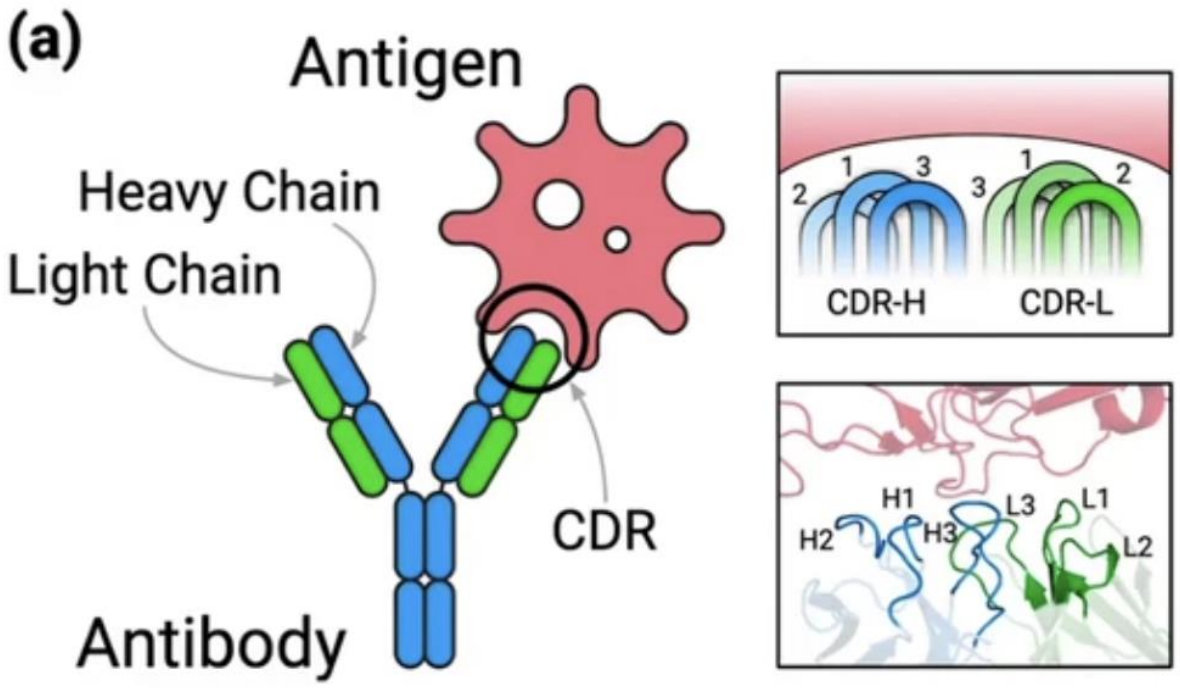
Kirchmeyer, Matthieu, Pedro O. O Pinheiro, and Saeed Saremi. "Score-based 3D molecule generation with neural fields." NeurIPS (2024): 10646-10678.

Function-Driven Graph Generation

- Molecular Graph Generation
- **Protein Structure Generation**
- Material Graph Generation

Protein Generation

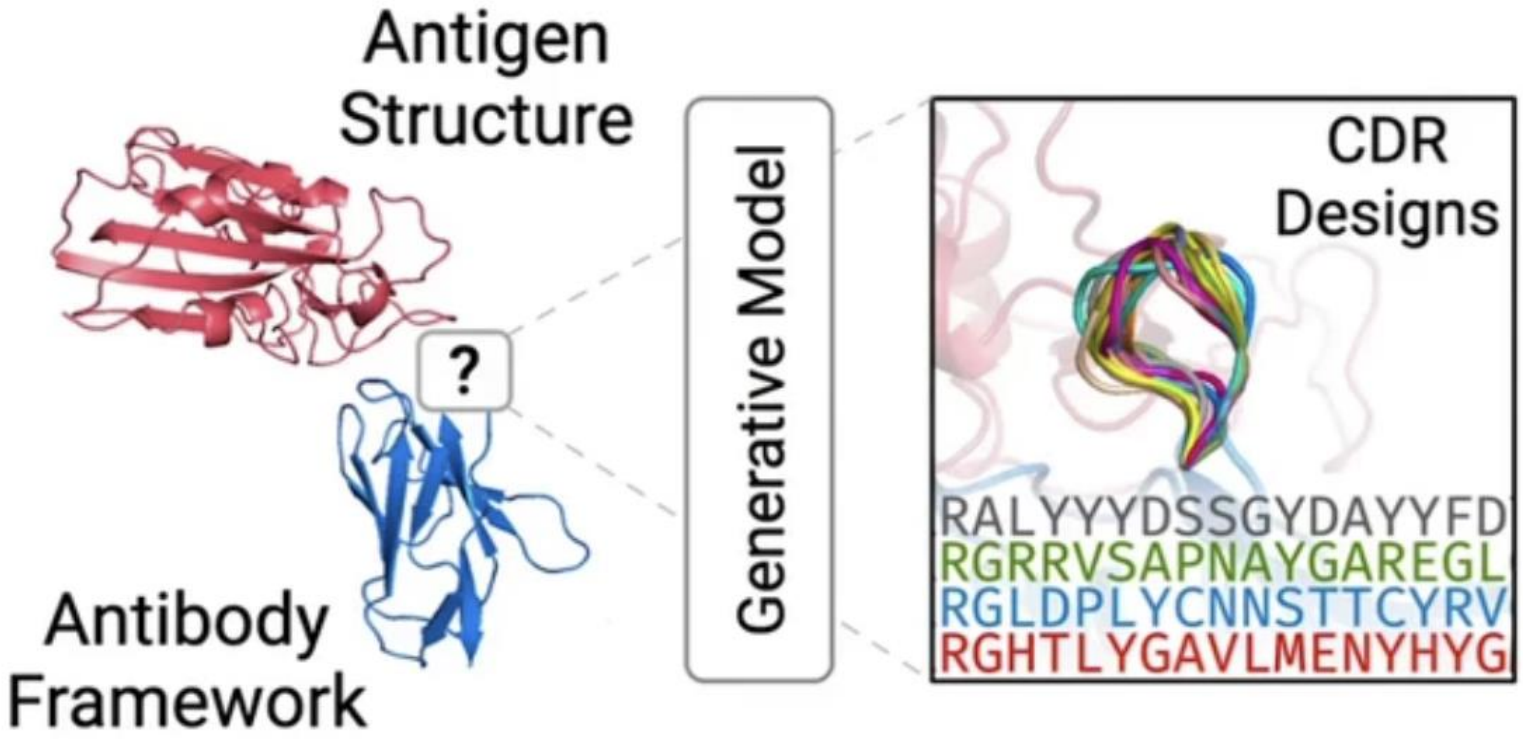
Antibody generation



Luo, Shitong, et al. "Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures." NeurIPS (2022): 9754-9767.

Protein Generation

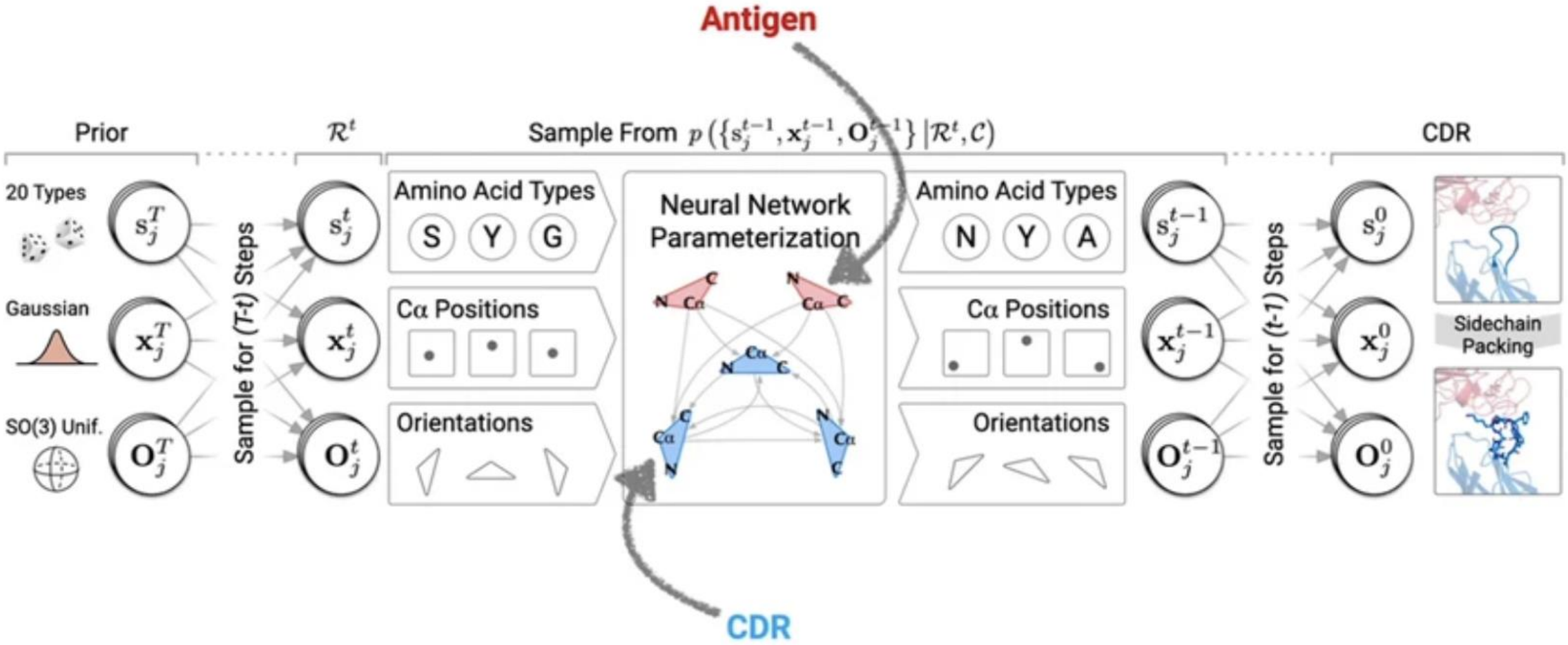
Antibody generation



Luo, Shitong, et al. "Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures." NeurIPS (2022): 9754-9767.

Protein Generation

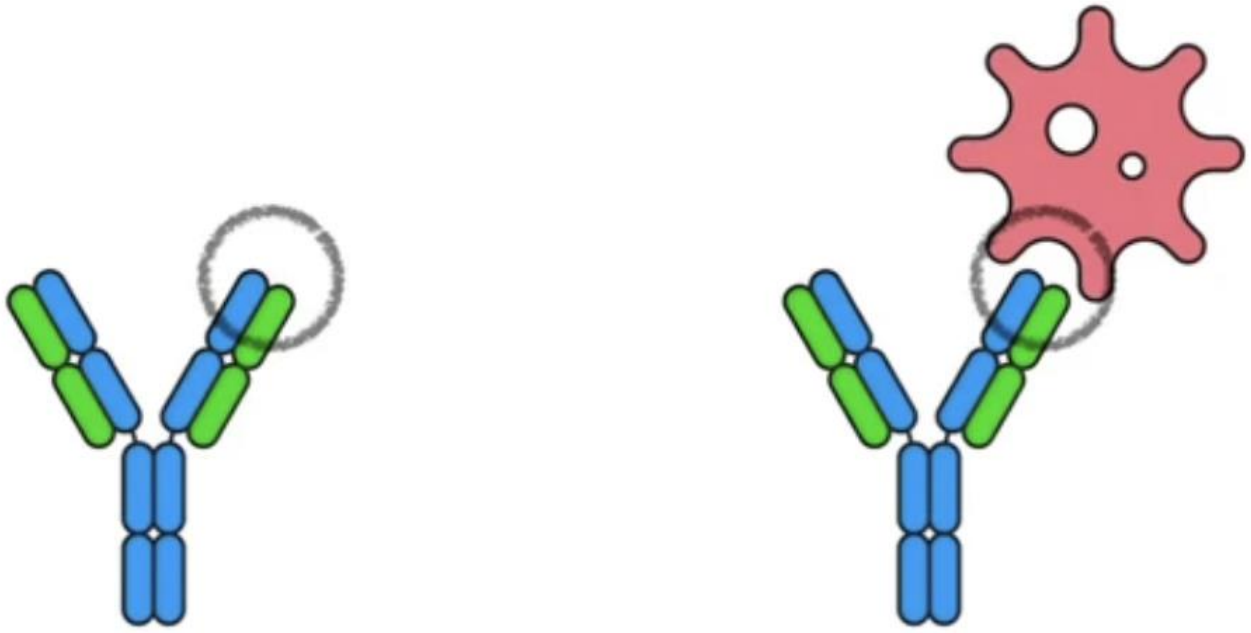
Antibody generation



Luo, Shitong, et al. "Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures." NeurIPS (2022): 9754-9767.

Protein Generation

Antibody generation



Previous Work on
Antibody Sequence-Structure
Co-design

Our Work
(Explicitly conditional on antigen structure)

Luo, Shitong, et al. "Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures." NeurIPS (2022): 9754-9767.

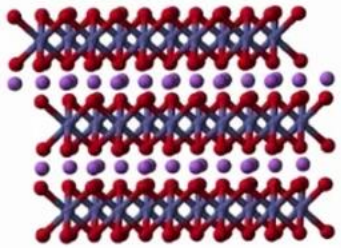
Function-Driven Graph Generation

- Molecular Graph Generation
- Protein Structure Generation
- **Material Graph Generation**

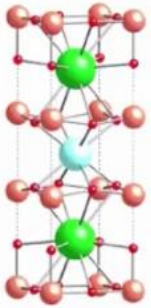
Material Graph Generation

What are Material Graphs

Materials are infinite periodic arrangements of atoms in 3D

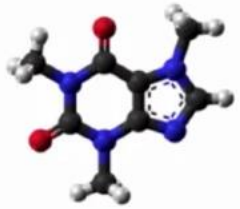


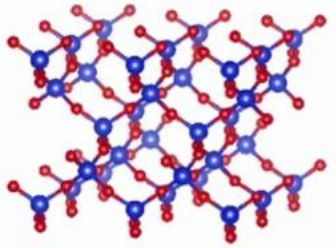
$LiCoO_2$
Cathode material for Li-ion battery
2019 Nobel Prize in Chemistry



$YBa_2Cu_3O_7$
First high-T superconductor
1987 Nobel Prize in Physics

Small molecules	Materials
• Non-periodic, finite	• Periodic, infinite
• 5 – 10 elements	• All 94 naturally occurring elements
• Simple 2D graph	• Graph difficult to define
• Relatively simple valency rules	• No general valency rules





3D material structures must be directly generated, rather than relying on intermediate graphs.

Xie, Tian, et al. "Crystal Diffusion Variational Autoencoder for Periodic Material Generation." *International Conference on Learning Representations 2022*.

Material Graph Generation

Why Generate Materials?

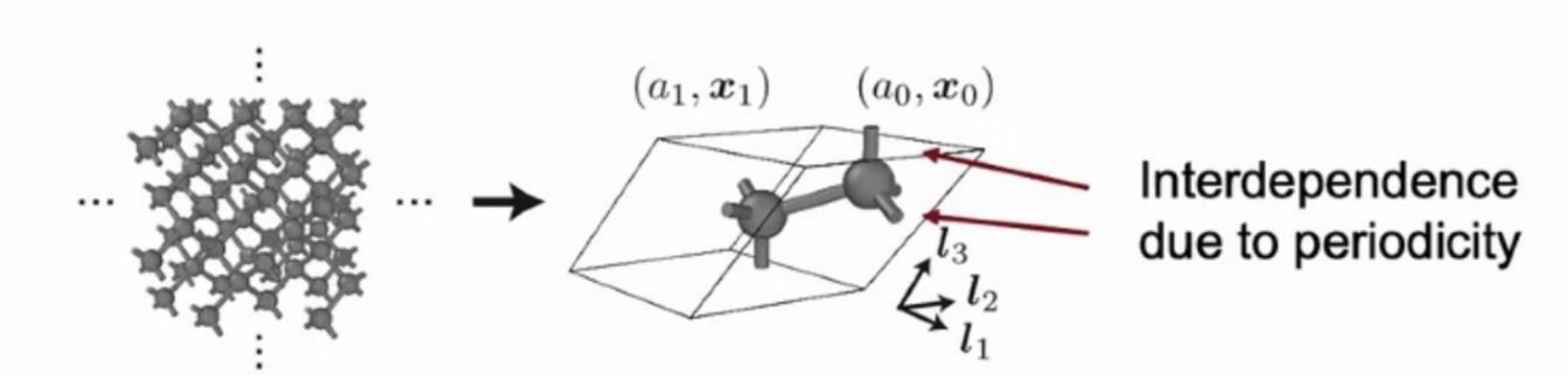
There are only **~200k** unique materials that are experimentally known (in contrast, **ZINC** includes close to a billion drug-like molecules).

Today's material discovery is centred on these **~200k known materials**. Moving beyond them could offer exciting new opportunities for multiple domains in materials science.

Xie, Tian, et al. "Crystal Diffusion Variational Autoencoder for Periodic Material Generation." *International Conference on Learning Representations 2022.*

Material Graph Generation

Representation of Periodic Materials



Key Components:

The unit cell (smallest repeating unit) of a material *M* can be fully defined by three lists:

Atom types: $A=(a_1,\ldots,a_N)\in A^N$

Atom coordinates: $X = (x_1, \ldots, x_N) \in \mathbb{R}^{N\times 3}$

Periodic lattice: $L = (l_1, l_2, l_3) \in \mathbb{R}^{3\times 3}$

Infinite Periodic Structure:

$$\{(zi', r_i') \mid zi' = zi,\ ri' = ri + k_1 l_1 + k_2 l_2 + k_3 l_3,\ k_1, k_2, k_3 \in \mathbb{Z}\}$$

This represents the repetition of the unit cell across all integer translations of the lattice vectors.

Interdependence Due to Periodicity:

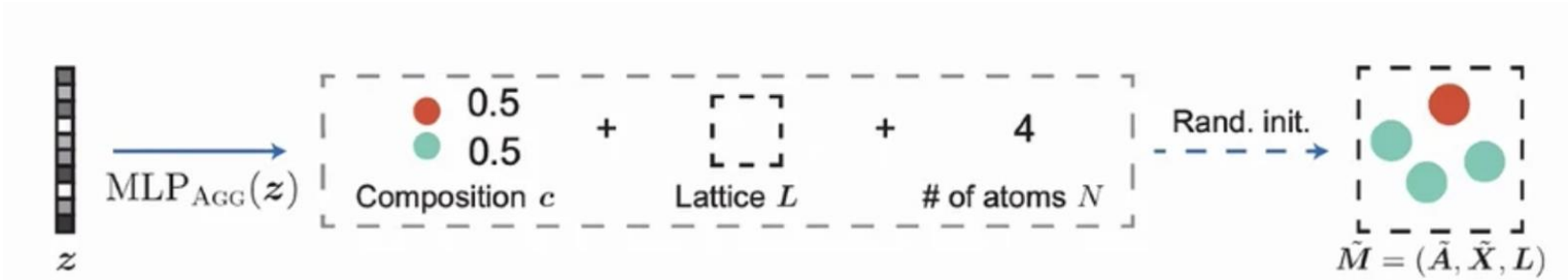
The periodic lattice L and atomic coordinates X are interdependent, as the lattice defines how atoms repeat in 3D space.

Goal: Jointly generate $M = (A, X, L)$ that corresponds to a **stable material**.

Xie, Tian, et al. "Crystal Diffusion Variational Autoencoder for Periodic Material Generation." *International Conference on Learning Representations 2022*.

Material Graph Generation

Generate a close random structure



Use **$\mathrm{MLP}_{\mathrm{AGG}}(z)$** (a neural network) to predict three aggregated properties for material generation:

Composition c: Sparse probability distribution over 100 element.

Lattice L: Rotation-invariant representation of the periodic lattice.

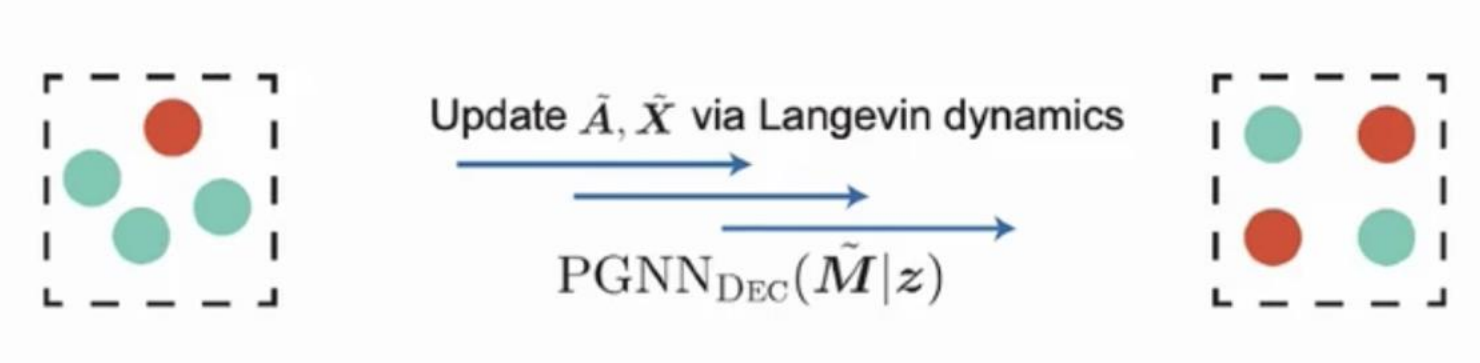
Number of atoms N: Probability distribution over possible atom counts.

Motivation: Use these easy-to-predict properties to simplify the task.

Xie, Tian, et al. "Crystal Diffusion Variational Autoencoder for Periodic Material Generation." *International Conference on Learning Representations 2022*.

Material Graph Generation

Denoise the random structure



Gradually deform $\tilde{M}$ into a stable material structure $M = (A, X, L)$ by iteratively:

- Adjusting atom coordinates.
- Updating atom types.

Physics-Guided Design: The GNN's architecture inherently preserves physical constraints (e.g., lattice periodicity, bond lengths).

Efficiency: Focuses updates on critical regions of instability.

Xie, Tian, et al. "Crystal Diffusion Variational Autoencoder for Periodic Material Generation." *International Conference on Learning Representations 2022*.

Material Graph Generation

Generate novel realistic materials

Task: Sample from latent space to generate 10,000 materials

Evaluation metrics:

Validity: Generated materials satisfy struc./comp. requirements

COV: How many test materials are covered with a similar one

Property statistics: Similarity of property distributions

Result: Significantly outperforming all baselines

Method	Data	Validity (%)[3] ↑		COV (%) ↑		Property Statistics ↓		
		Struc.	Comp.	R.	P.	ρ	E	# elem.
FTCP[4]	Perov-5	0.24	54.24	0.00	0.00	10.27	156.0	0.6297
	Carbon-24	0.08	–	0.00	0.00	5.206	19.05	–
	MP-20	1.55	48.37	4.72	0.09	23.71	160.9	0.7363
Cond-DFC-VAE	Perov-5	73.60	82.95	73.92	10.13	2.268	4.111	0.8373
G-SchNet	Perov-5	99.92	**98.79**	0.18	0.23	1.625	4.746	**0.03684**
	Carbon-24	99.94	–	0.00	0.00	0.9427	1.320	–
	MP-20	99.65	75.96	38.33	**99.57**	3.034	42.09	0.6411
P-G-SchNet	Perov-5	79.63	**99.13**	0.37	0.25	0.2755	1.388	0.4552
	Carbon-24	48.39	–	0.00	0.00	1.533	134.7	–
	MP-20	77.51	76.40	41.93	**99.74**	4.04	2.448	**0.6234**
CDVAE	Perov-5	**100.0**	**98.59**	**99.45**	**98.46**	**0.1258**	**0.0264**	0.0628
	Carbon-24	**100.0**	–	**99.80**	**83.08**	**0.1407**	**0.2850**	–
	MP-20	**100.0**	**86.70**	**99.15**	**99.49**	**0.6875**	**0.2778**	1.432

Xie, Tian, et al. "Crystal Diffusion Variational Autoencoder for Periodic Material Generation." *International Conference on Learning Representations 2022*.

AIGC for Graphs: Current Techniques and Future Trends

Current Trend and Future Direction

Speaker:
Hanchen Wang

Contributors: Hanchen Wang, Dawei Cheng, Ying Zhang and Wenjie Zhang

Large Language Model and Graph Generation

Current Trend

Demystifying the Power of Large Language Models in Graph Structure Generation

Yu Wang[1] **Ryan Rossi**[2] **Namyong Park** **Nesreen Ahmed**[3]
Danai Koutra[4] **Franck Dernoncourt**[2] **Tyler Derr**[5]
[1] University of Oregon [2] Adobe Research [3] Cisco AI Research
[4] University of Michigan [5] Vanderbilt University

OSDA AGENT: LEVERAGING LARGE LANGUAGE MODELS FOR DE NOVO DESIGN OF ORGANIC STRUCTURE DIRECTING AGENTS

Zhaolin Hu[1,2] , **Yixiao Zhou**[2,3], **Zhongan Wang**[2], **Xin Li** [4*],**Weimin Yang**[4]
Hehe Fan[2], **Yi Yang**[1,2*]
[1]The State Key Laboratory of Brain-Machine Intelligence, Zhejiang University, China
[2]CCAI, Zhejiang University, China [3]Shanghai Innovation Institute, China
[4]The State Key Laboratory of Green Chemical Engineering and Industrial Catalysis, Sinopec Shanghai Research Institute of Petrochemical Technology, China
{12321165,12421181,zhonganwang,hehefan,yangyics}@zju.edu.cn
{lixin.sshy,yangwm.sshy}@sinopec.com

FRAGMENT AND GEOMETRY AWARE TOKENIZATION OF MOLECULES FOR STRUCTURE-BASED DRUG DESIGN USING LANGUAGE MODELS

Cong Fu[1*], **Xiner Li**[1*], **Blake Olson**[1], **Heng Ji**[2], **Shuiwang Ji**[1]
[1] Texas A&M University, College Station, TX, USA
[2] University of Illinois Urbana-Champaign, Champaign, IL, USA
{congfu, lxe, blakeolson, sji}@tamu.edu
{hengji}@illinois.edu

NEXT-MOL: 3D DIFFUSION MEETS 1D LANGUAGE MODELING FOR 3D MOLECULE GENERATION

Zhiyuan Liu[1*], **Yanchen Luo**[2*], **Han Huang**[3], **Enzhi Zhang**[4], **Sihang Li**[2],
Junfeng Fang[2], **Yaorui Shi**[2], **Xiang Wang**[2†], **Kenji Kawaguchi**[1], **Tat-Seng Chua**[1]
[1] National University of Singapore, [2] University of Science and Technology of China,
[3] Chinese University of Hong Kong, [4] Hokkaido University
zhiyuan@nus.edu.sg, luoyanchen@mail.ustc.edu.cn

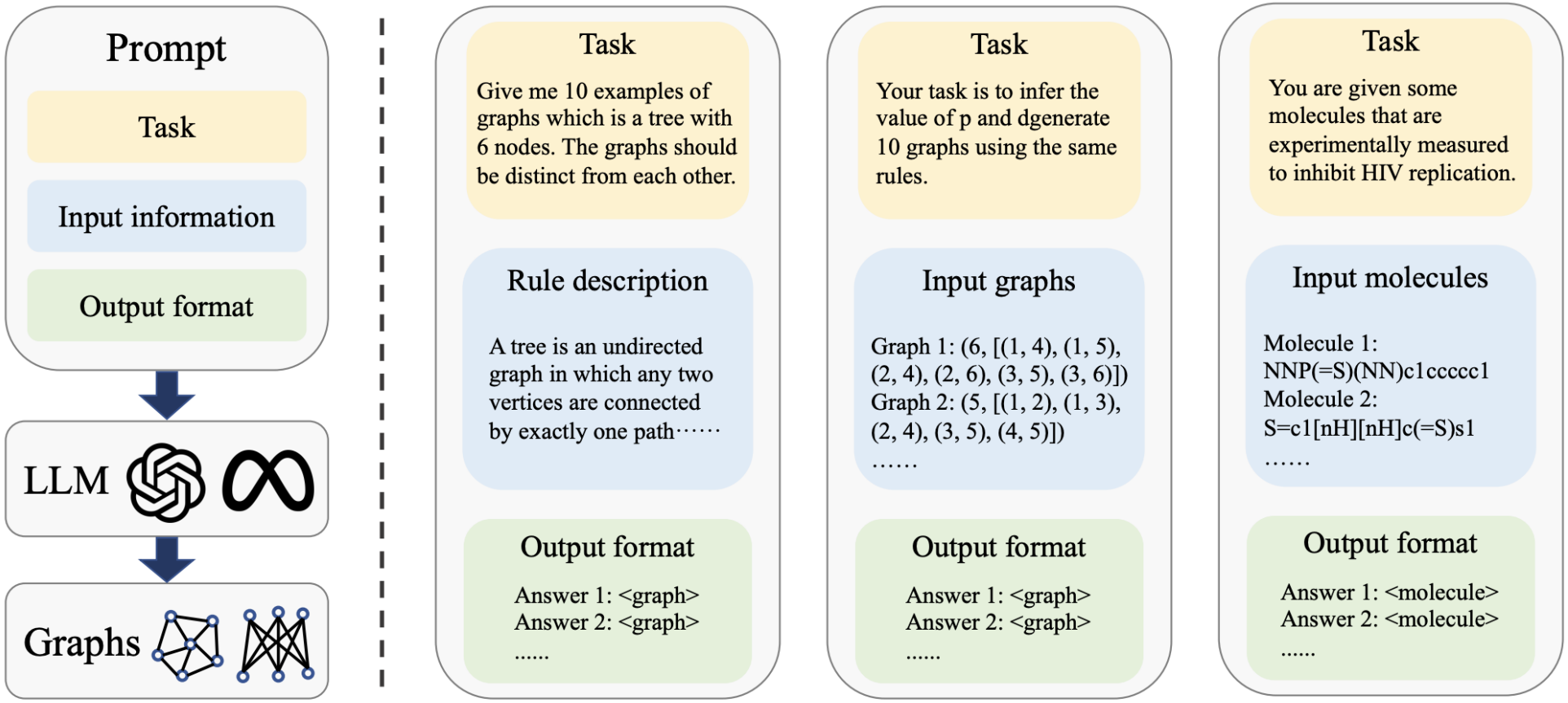
EXPLORING THE POTENTIAL OF LARGE LANGUAGE MODELS IN GRAPH GENERATION

A PREPRINT

Yang Yao[1], **Xin Wang**[1,*], **Zeyang Zhang**[1], **Yijian Qin**[1], **Ziwei Zhang**[1], **Xu Chu**[1], **Yuekui Yang**[1], **Wenwu Zhu**[1,*], **and Hong Mei**[2]
[1]Tsinghua University
[2]Peking University
[*]Corresponding authors.

Large Language Model and Graph Generation

LLM for graph generation (Based on Prompt Engineering)



(a) General (b) Rule-based (c) Distribution-based (d) Property-based

Yao, Yang, et al. "Exploring the potential of large language models in graph generation." *arXiv preprint arXiv:2403.14358* (2024).

Large Language Model and Graph Generation

Rule-based graph generation

Prompt Design:

- **Zero-shot**: The prompt contains the relevant information about the rules, as well as a specification of the output format. The model is then asked to generate graphs using the given rules.
- **Few-shot**: In addition to the zero-shot prompt, the model is given several graph examples that follow the given rules. The edges of the graphs are sorted by the node ID to facilitate the model understanding.

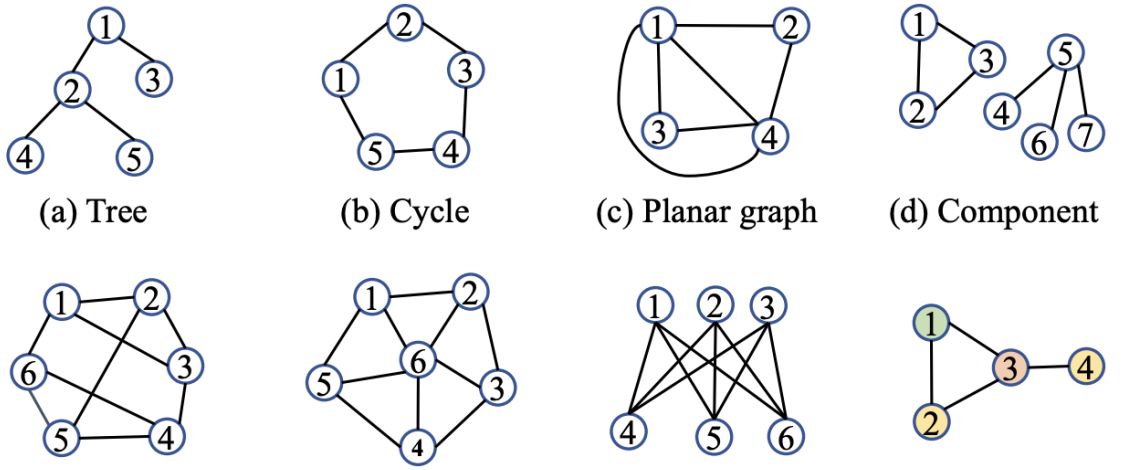


Figure 2: An illustration of graphs with regard to different rules.

Yao, Yang, et al. "Exploring the potential of large language models in graph generation." *arXiv preprint arXiv:2403.14358* (2024).

Large Language Model and Graph Generation

Distribution-based graph generation

Prompt Design:

In prompt, we introduce the graph generation task and the **target distribution**, and provide a **set of graphs sampled from the target distribution** as the input.
Then, the model is asked to infer the value of p (i.e., whether the graph follows the given distribution) and generate new graphs.

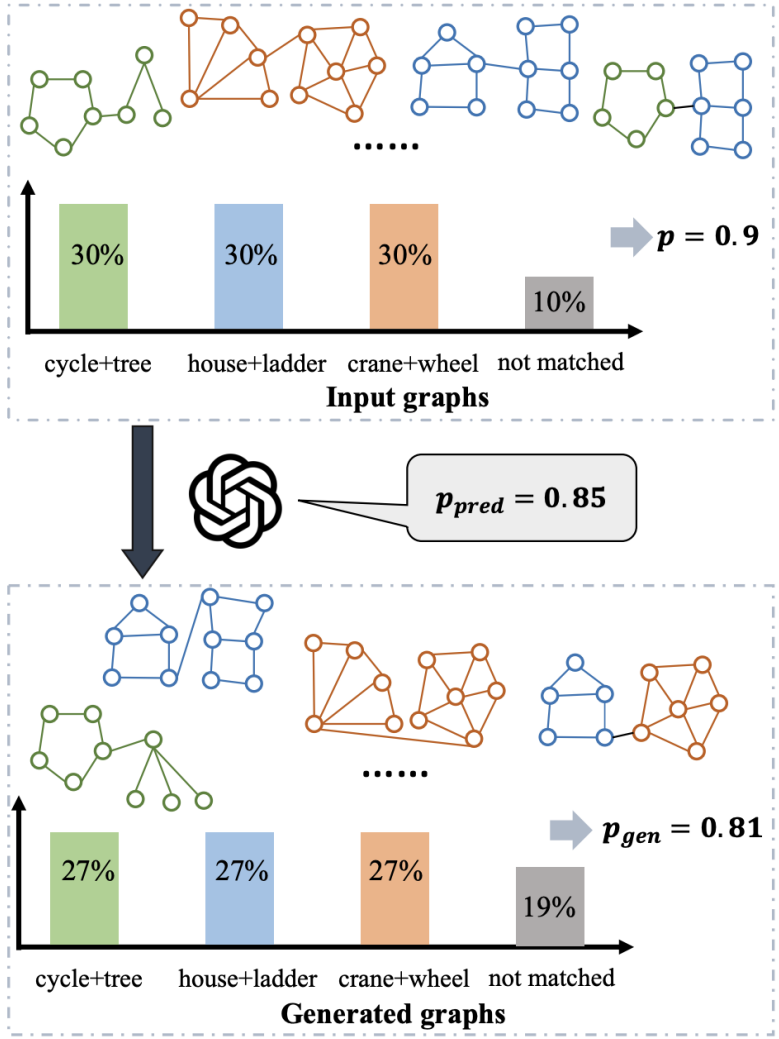


Figure 3: An illustration of distribution-based graph generation.

Yao, Yang, et al. "Exploring the potential of large language models in graph generation." *arXiv preprint arXiv:2403.14358* (2024).

Large Language Model and Graph Generation

Property-based graph generation

Prompt Design:

LLM is given a **description of the desired property** and a collection of molecules that have the property. Then, the model is asked to generate new molecules with **the same property**.

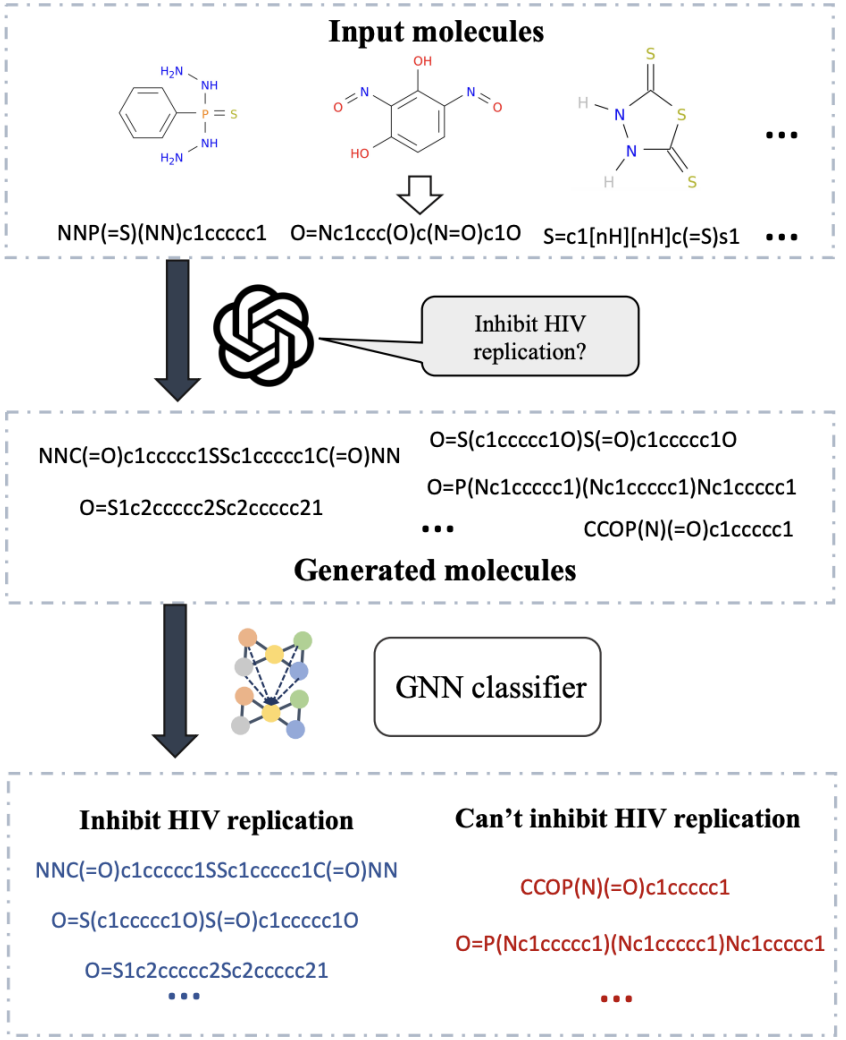


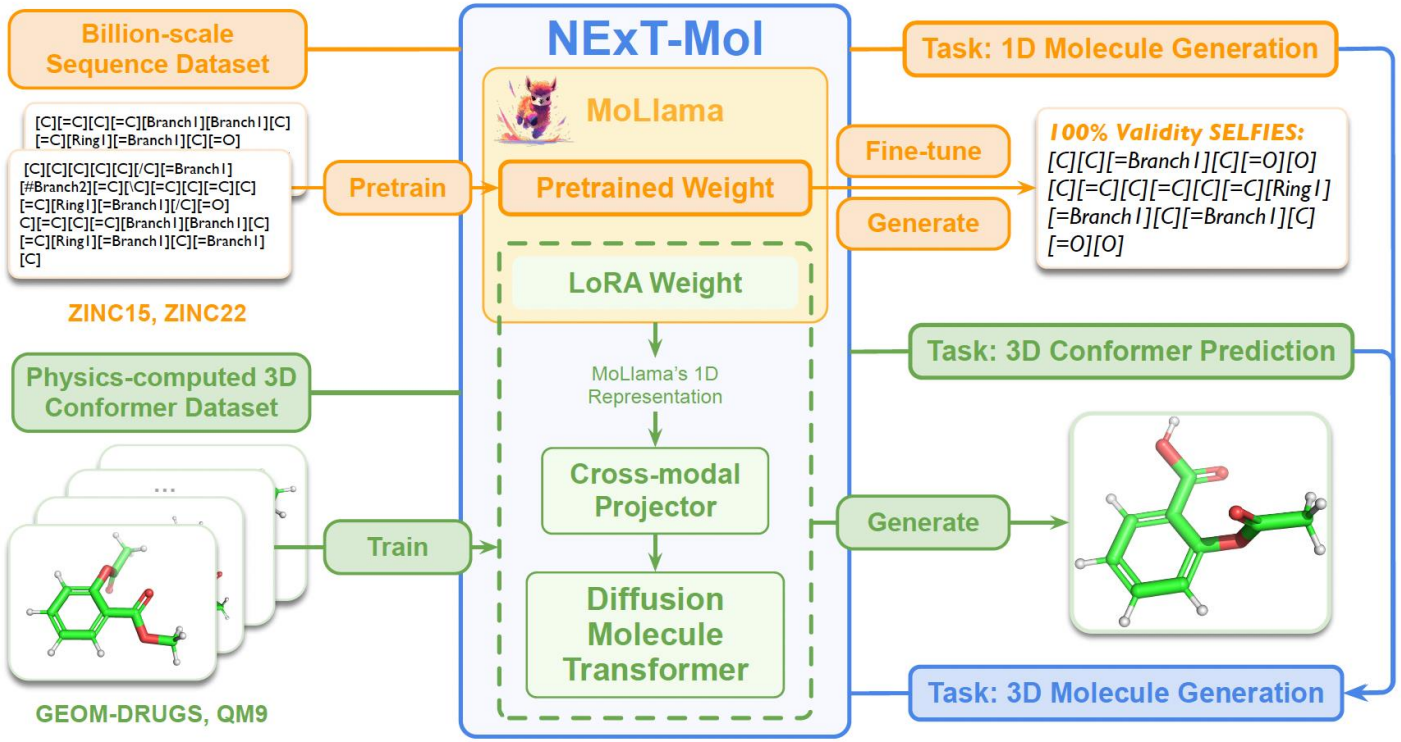
Figure 4: An illustration of property-based graph generation.

Yao, Yang, et al. "Exploring the potential of large language models in graph generation." *arXiv preprint arXiv:2403.14358* (2024).

Large Language Model and Graph Generation

Fine-tuning on LLMs

(1) MoLlama, a large LM for generating 1D molecule sequences
(2) DMT, a diffusion model to predict 3D conformers from the 1D sequences
(3)NExT-Mol leverages transfer learning to enhance DMT's 3D prediction with MoLlama's 1D representations.



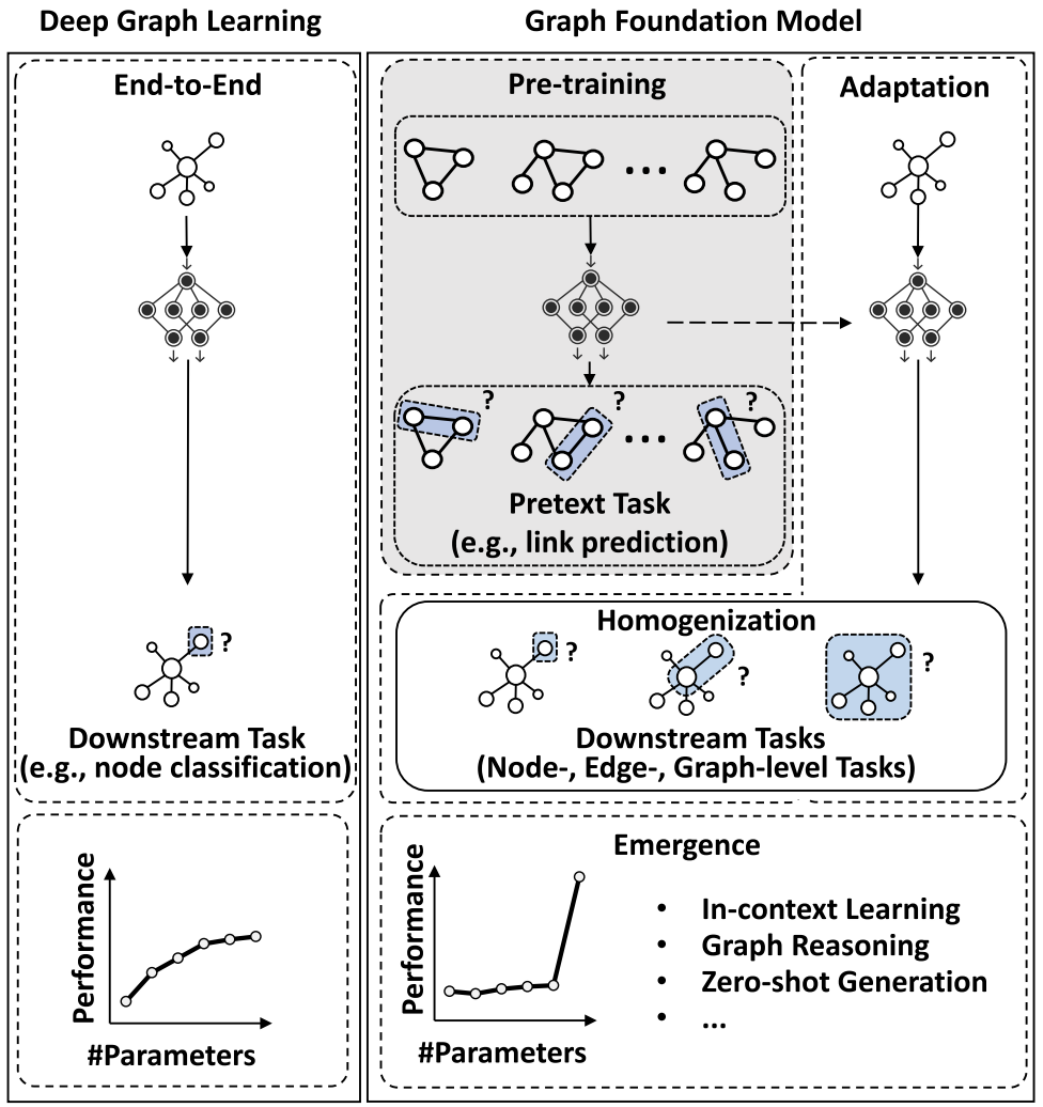
Liu, Zhiyuan, et al. "NExT-Mol: 3D Diffusion Meets 1D Language Modeling for 3D Molecule Generation." ICLR 2025

Future Direction and Opportunity

Graph Foundation Models?

A **graph foundation model** is a large-scale, pre-trained artificial intelligence model designed to understand, analyze, and generate graph-structured data.

Using a graph foundation model to generate graphs with specific distribution and property is still a wide-open direction.



Liu, Jiawei, et al. "Graph foundation models: Concepts, opportunities and challenges." IEEE Transactions on Pattern Analysis and Machine Intelligence (2025).

Future Direction and Opportunity

- **Efficient Generation**: reduce the computational cost for graph generation.
- **Interpretable Generation**: generate graphs with explicit constraints.
- **Integration with Physical Law**: generate graphs that follow chemical and biomedical law.
- **Cross-Modal Generation**: Build models that convert between graph and non-graph data.

Q & A

Thank you!

Hanchen Wang
University of Technology Sydney
hanchen.wang@uts.edu.au

Homepage: https://hanchen-wang.com/