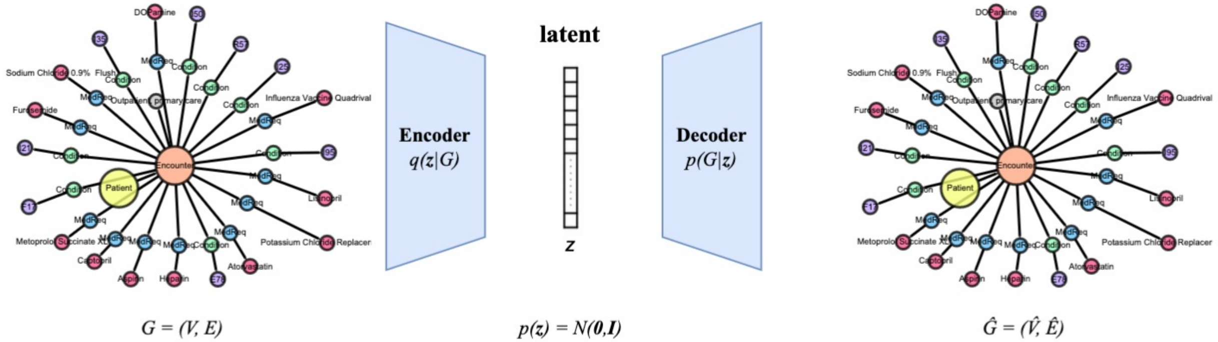
# Graph Generative models and some applications


latent
Encoder
$q(z|G)$
z
Decoder
$p(G|\mathbf{z})$
$G = (V, E)$
$p(\mathbf{z}) = N(\mathbf{0},\mathbf{I})$
$\hat{G} = (\hat{V}, \hat{E})$

Michalis Vazirgiannis

Ecole Polytechnique, IPP, France

http://www.lix.polytechnique.fr/dascim

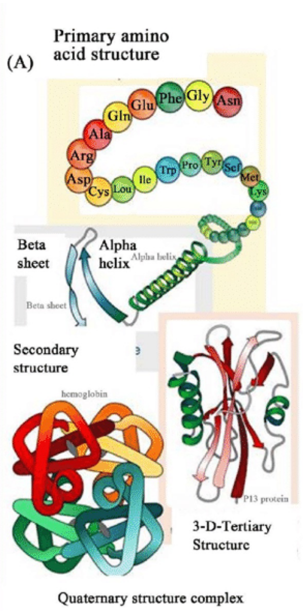
June 2024

# GNNs and Graph Generative models and applications

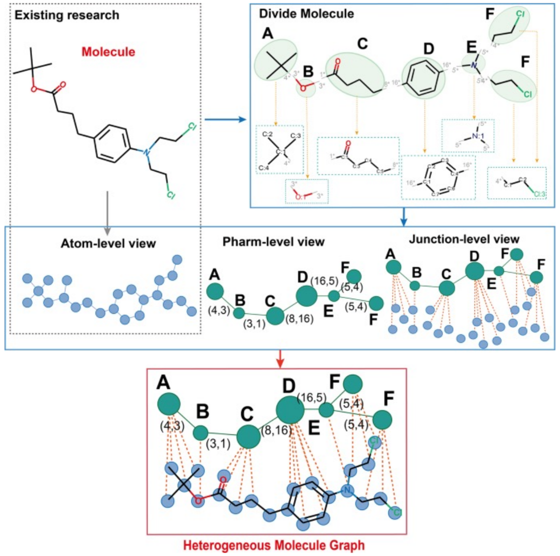
- **Graph Generative models**
- Generative models for Medical Graphs
- Large generative models
- Graph / LLMs
- Multi modality for molecule/protein generation
- Conclusions

# Graphs are ubiquitous

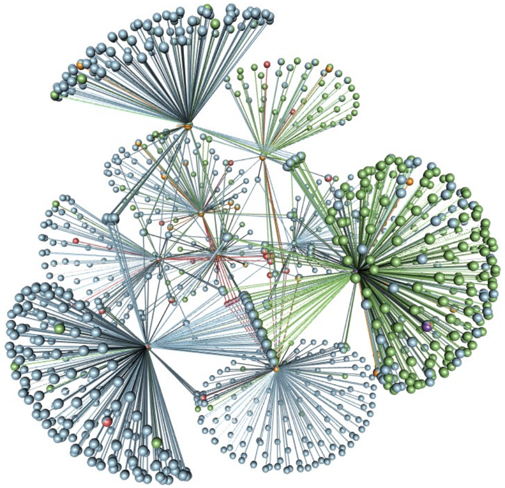
- Chemistry – Bio/Pharma
  - Space of molecules: $10^{60}$
  - New proteins, molecules generation



**Guided Folding of Life's Proteins in Integrate Cells with Holographic Memory and GM-Biophysical Steering,** Dirk K F Meijer, Hans J. H. Geesink, 2018, Open Journal of Biophysics 8(03):117-154 DOI:10.4236/ojbiphy.2018.83010,



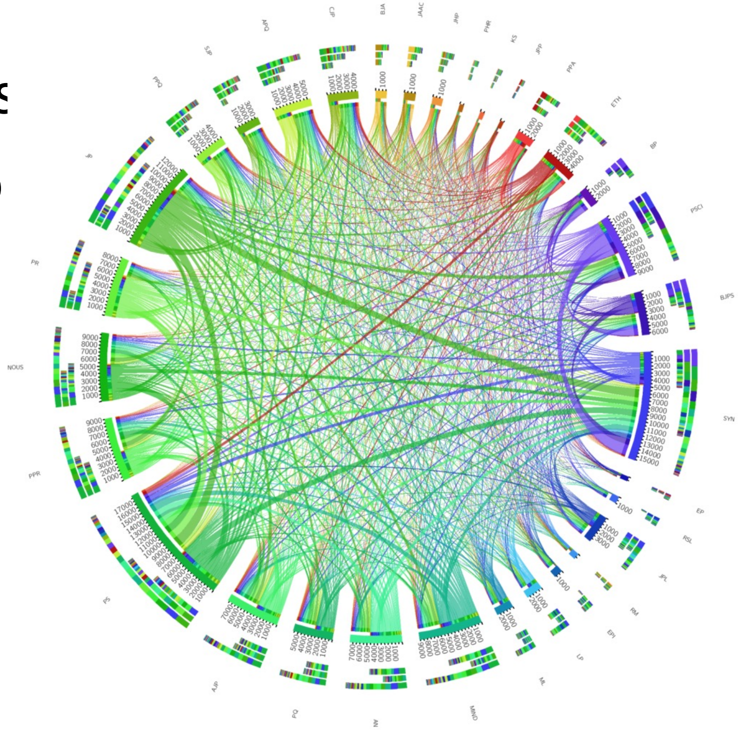
Jiang, Y., Jin, S., Jin, X. *et al.* **Pharmacophoric-constrained heterogeneous graph transformer model for molecular property prediction**. *Commun Chem* **6**, 60 (2023). https://doi.org/10.1038/s42004-023-00857-x



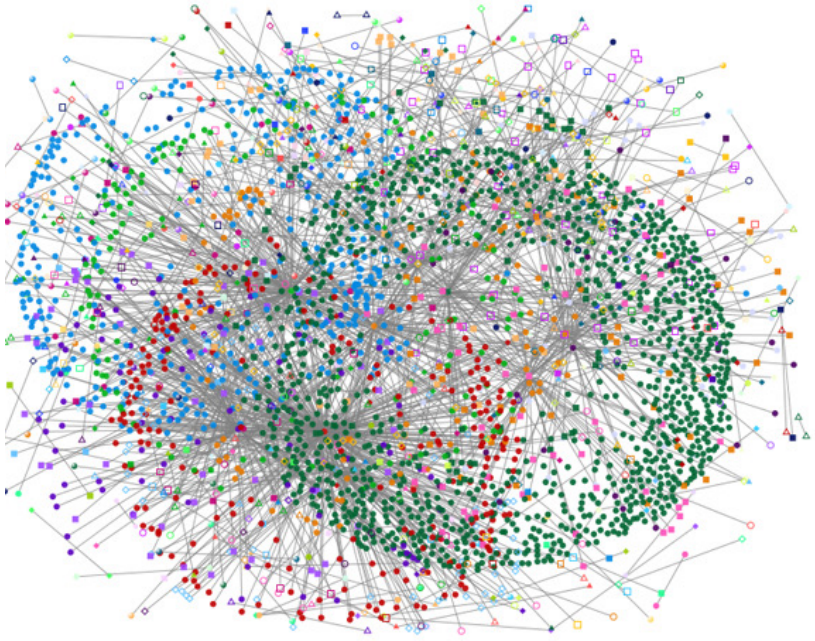
**Multiplex Human HIV-1 protein-protein interaction network**
**https://commons.wikimedia.org/wiki/File:Multiplex_Human_HIV-1_protein-protein_interaction_network_%28edge-colored_visualization%29.png**

# Graphs are ubiquitous

- Social Networks
- Internet/Teleco
- Citation graphs



http://tar.weatherson.org/2017/05/04/citation-graphs-and-methodology/

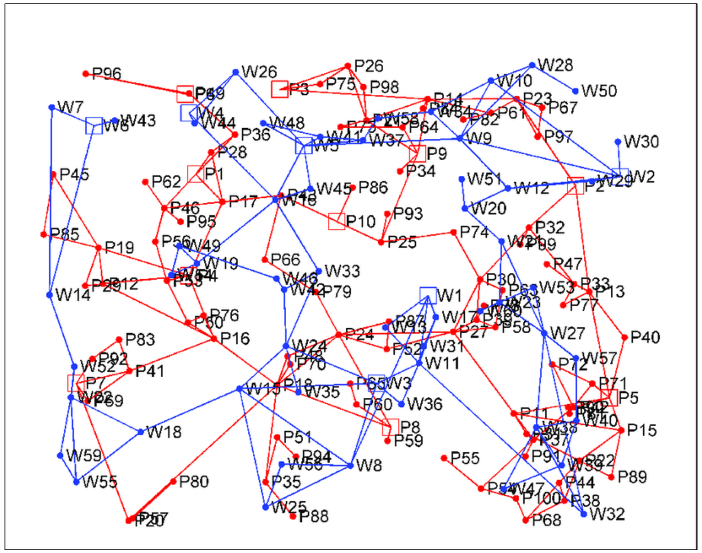


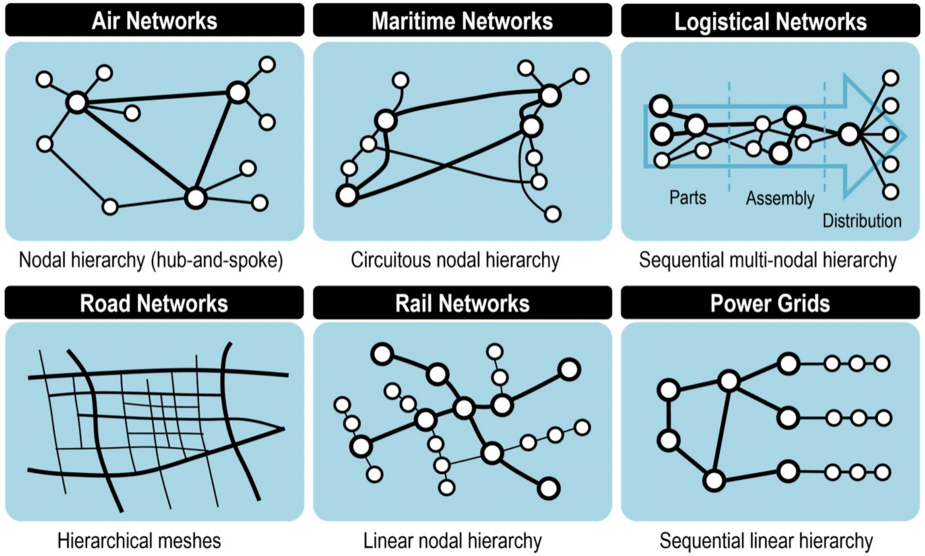
https://threatpost.com/researchers-graph-social-networks-spot-spammers-061711/75346/

# Graphs are ubiquitous

power/water distribution networks

Transport/road networks



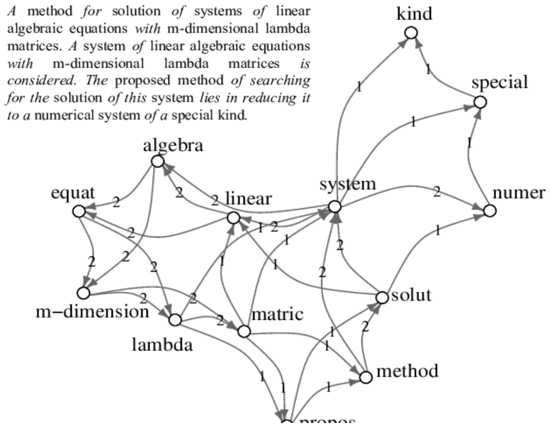


doi.org/10.4324/9780429346323

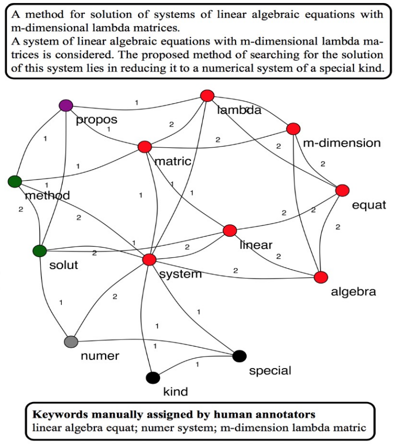
https://doi.org/10.1371/journal.pone.0195727.g005

# Graphs in NLP

- Graph of Words
  - Information retrieval[1]
  - Keyword extraction [2]
  - Event detection [4]
  - Summarization
  - Document classification [3][1]

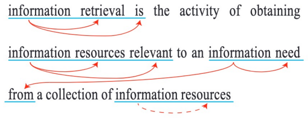


*A* method *for* solution *of* systems *of* linear algebraic equations *with* m-dimensional lambda matrices. *A* system *of* linear algebraic equations *with* m-dimensional lambda matrices *is considered. The* proposed method *of searching for the* solution *of this* system *lies in reducing it to a* numerical system *of a* special kind.



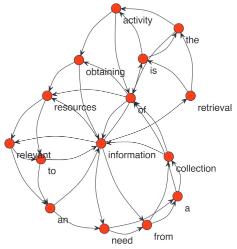
A method for solution of systems of linear algebraic equations with m-dimensional lambda matrices.
A system of linear algebraic equations with m-dimensional lambda matrices is considered. The proposed method of searching for the solution of this system lies in reducing it to a numerical system of a special kind.

**Keywords manually assigned by human annotators**
linear algebra equat; numer system; m-dimension lambda matric

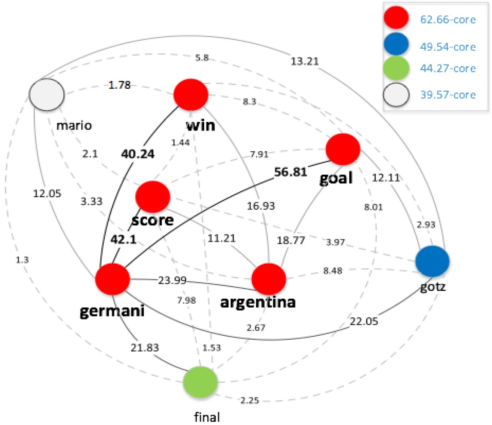


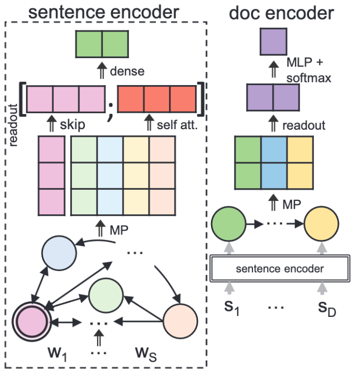
information retrieval is the activity of obtaining information resources relevant to an information need from a collection of information resources

Bag of words: ((activity,1), (collection,1) (information,4), (relevant,1), (resources, 2), (retrieval, 1)..)



Captures: frequency, order and distance, …





[1] Graph-of-word and TW-IDF: new approach to ad hoc IR, F.Rousseau, Michalis Vazirgiannis - **CIKM '13**: https://doi.org/10.1145/2505515.2505671, Best paper mention award

[2]Main Core Retention on Graph-of-words for Single-Document Keyword Extraction, F. Rousseau,M. Vazirgiannis. **ECIR2015**

[3]Text Categorization as a Graph Classification Problem, F Rousseau, E Kiagias, M Vazirgiannis, **ACL 2015**

[4] Degeneracy-based real-time sub-event detection in twitter stream, P Meladianos, et. al. **AAAI - ICWSM 2015**

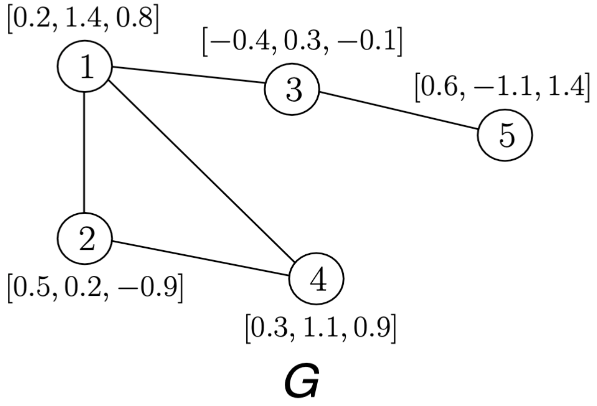
[5]Message Passing Attention Networks for Document Understanding, G. Nikolentzos, A. Tixier, M.Vazirgiannis , **AAAI2020**, https://doi.org/10.1609/aaai.v34i05.6376

# Why Graph ML is important & different than sequential ML

- handles complex and rich data structures (graphs, networks, trees, and hypergraphs) not easily represented by vectors or matrices
- capture relational information and dependencies among nodes, and capitalise on the graph structure and properties to enhance the learning process – capture *longer term dependencies*
- can *leverage GNNs to learn powerful/expressive graph representations* useful for downstream tasks: node classification, link prediction, graph generation, and graph matching.
- Graph pretrained models tend to have fewer parameters than traditional DL models, especially those based on transformers.

# Graph Machine learning tasks

An attributed graph is a graph with attributes on vertices. Each vertex $v \in V$ is annotated with a feature vector $h_v$



$h_1, \ldots, h_5 \in \mathbb{R}^3$

$h_1 = [0.2, 1.4, 0.8]^\top \quad h_3 = [-0.4, 0.3, -0.1]^\top$

- *Node classification*: given a graph with labels on some nodes, provide a high quality labelling for the rest of nodes
- *Graph clustering*: given a graph, group its vertices into clusters in such a way that there are *many edges within each cluster* and relatively few between the clusters (community detection)
- *Link Prediction*: given a pair of vertices, predict if they should be linked with an edge
- ***Graph classification***: given a set of graphs with known class labels for some of them, decide to which class the rest of the graphs belong.
- ***Graph Regression…***

# Abundance of GNN methods

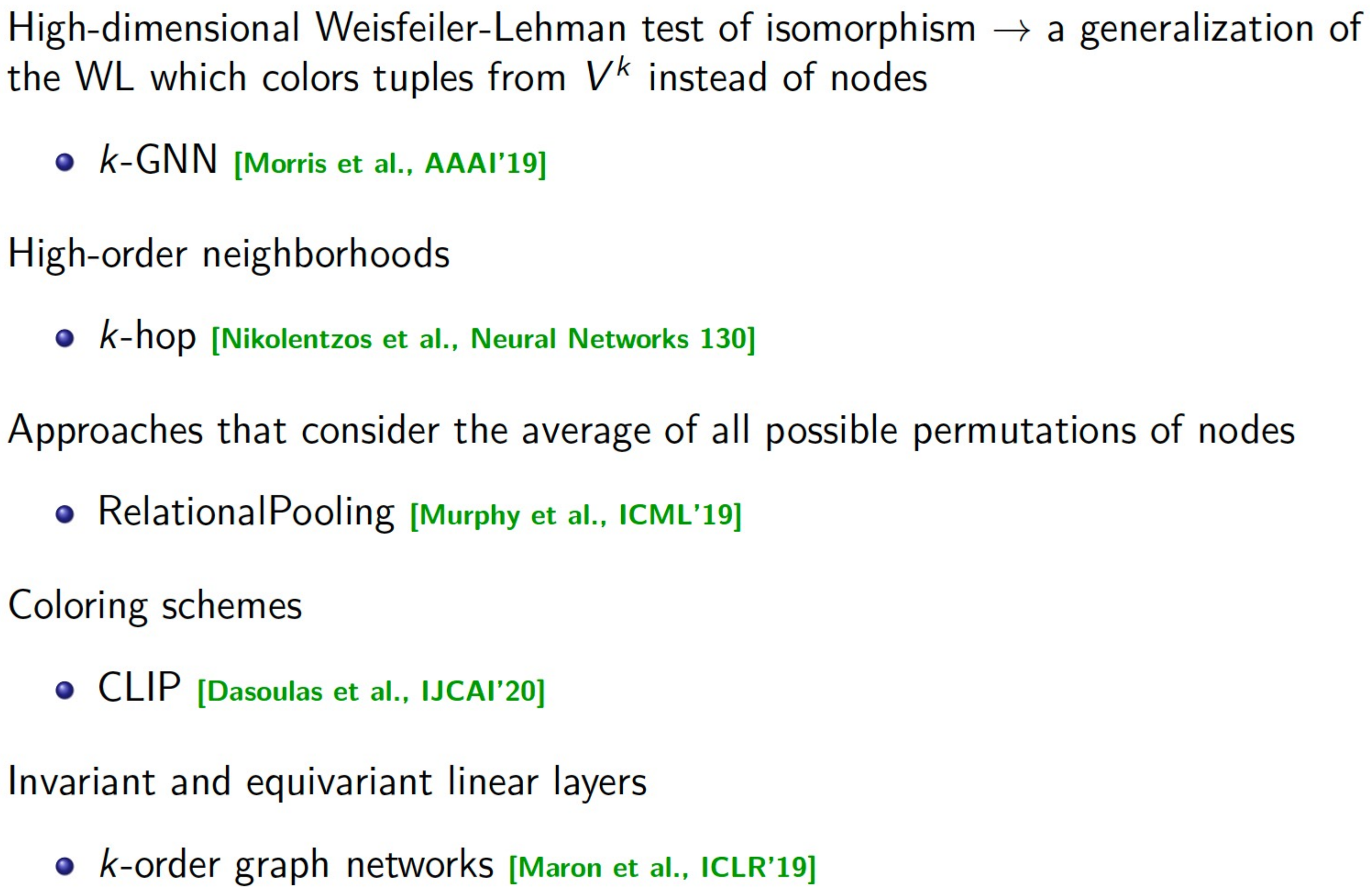
Graph Convolutional Network (GCN)

[Kipf and Welling, ICLR'17]

Graph attention networks (GAT)

[Veličković et al., ICLR'18]

[Hamilton et al., NIPS'17]



High-dimensional Weisfeiler-Lehman test of isomorphism → a generalization of the WL which colors tuples from $V^k$ instead of nodes

- $k$-GNN [Morris et al., AAAI'19]

High-order neighborhoods

- $k$-hop [Nikolentzos et al., Neural Networks 130]

Approaches that consider the average of all possible permutations of nodes

- RelationalPooling [Murphy et al., ICML'19]

Coloring schemes

- CLIP [Dasoulas et al., IJCAI'20]

Invariant and equivariant linear layers

- $k$-order graph networks [Maron et al., ICLR'19]

# Need for Graph Generators

- Graph generator models can produce graphs with given properties or typology for various applications
- Modeling and studying networks in biology, engineering, and social sciences.
  - simulate the evolution of social networks,
  - the structure of protein-protein interactions,
  - topology of power grids.
- Discovering new graph structures and properties.
  - generate novel chemical and molecular structures,
  - design new materials,
  - explore the space of possible graphs with certain characteristics.
- Completing and enhancing existing graphs.
  - fill in missing nodes and edges,
  - add new features and attributes,
  - improve the quality and diversity of graph data.

# Graph Generators – heuristic based models

- Erdős–Rényi Random Graph Model
- Barabasi Albert graph generator
- Kronecker graphs – pattern recursion
- Stochastic Block models
- ....

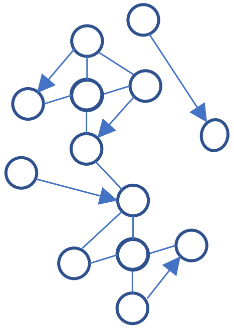
# Need for **Deep** Graph Generators

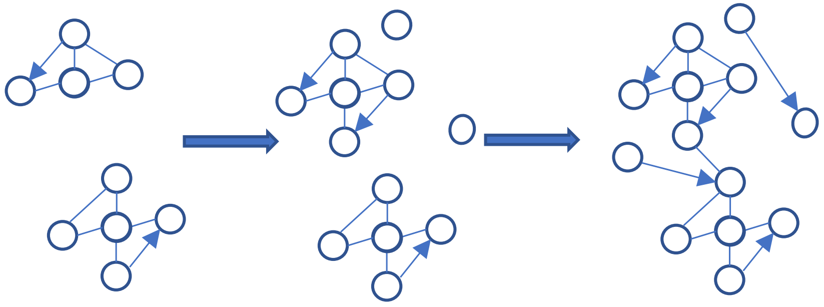
- Traditional graph generation models (i.e. *Erdős-Rényi, Barabási-Albert model, Kronecker graphs, Stochastic block models*) based on *assumptions / heuristics* oversimplifying the underlying distributions of graphs.
- Deep models for graph-structured data enable effective complex graph generation
  - *can learn the generative model directly from observed data*, without relying on hand-engineered processes or pre-defined statistical properties.
  - *capture the complex joint probability of all nodes and edges in the graph*, and generate realistic graphs that match the structural characteristics of the target distribution.
  - incorporate various advanced methods: (i.e. attention mechanisms, reinforcement learning) to enhance the quality and diversity of the generated graphs.

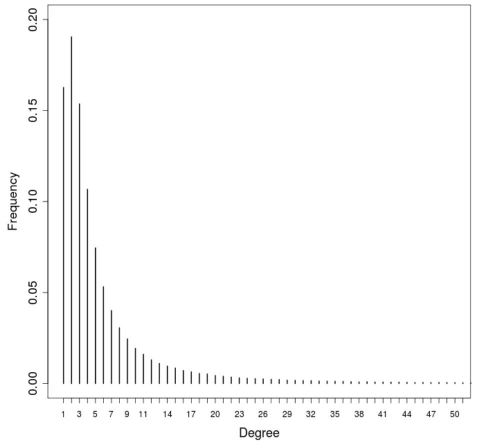
# Graph Generative Models

Graph generation challenging task:

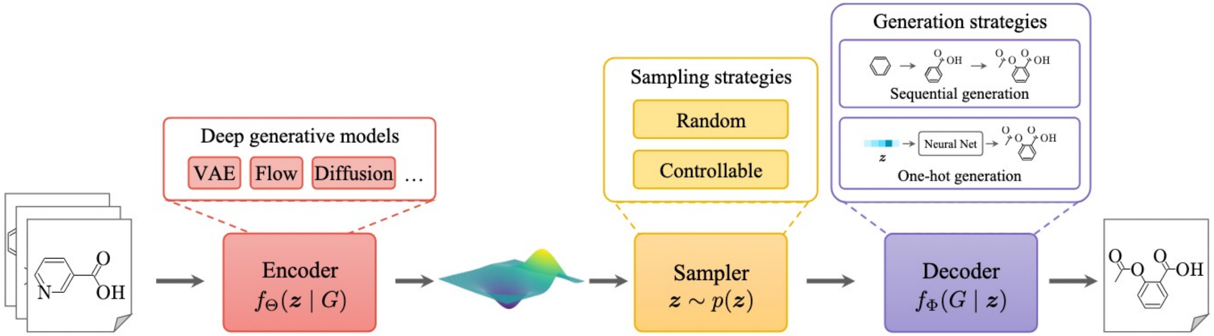
- higher-order (and non symmetric) relationships
- Sparsity and no deterministic order in processing nodes,
- long-tailed distribution of relationships: some are frequent others very rare in real-life graphs.
- dynamic and temporal: change over time, different states with time.







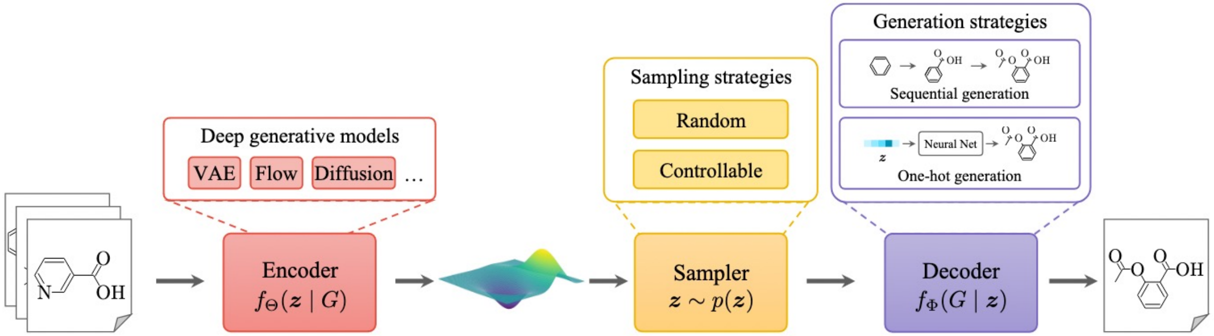
# Overview of deep graph generation



A Survey on Deep Graph Generation: Methods and Applications, Yanqiao Zhu et al, LOG 2023

- *encoder* maps observed graphs into a stochastic distribution;
- *sampler* draws latent representations from that distribution;
- *decoder* receives latent codes and produces graphs

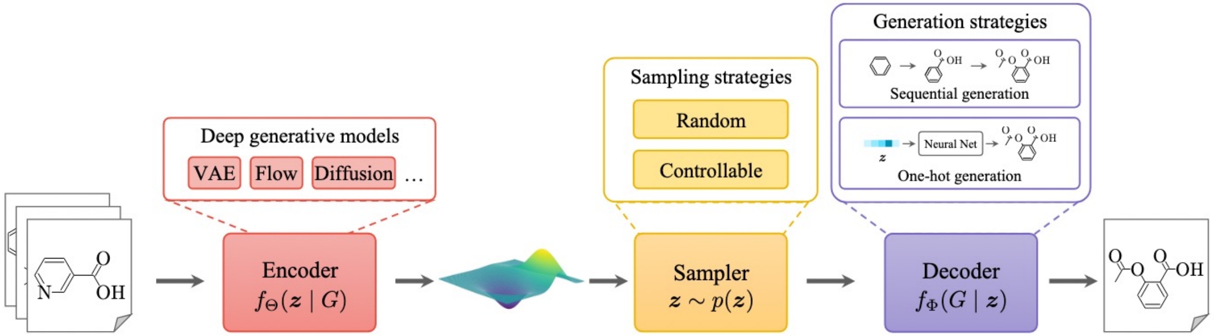
# Overview of deep graph generation - Encoder



A Survey on Deep Graph Generation: Methods and Applications, Yanqiao Zhu et al, LOG 2023

- Encoder - The encoding function $f_\Theta(z \mid G)$ represent discrete graph objects as dense, continuous vectors.
- employ probabilistic generative models (e.g., variational graph neural networks) as the encoder.
- encoder function $f_\Theta$ outputs the parameters of a stochastic distribution following a prior distribution $p(z)$.

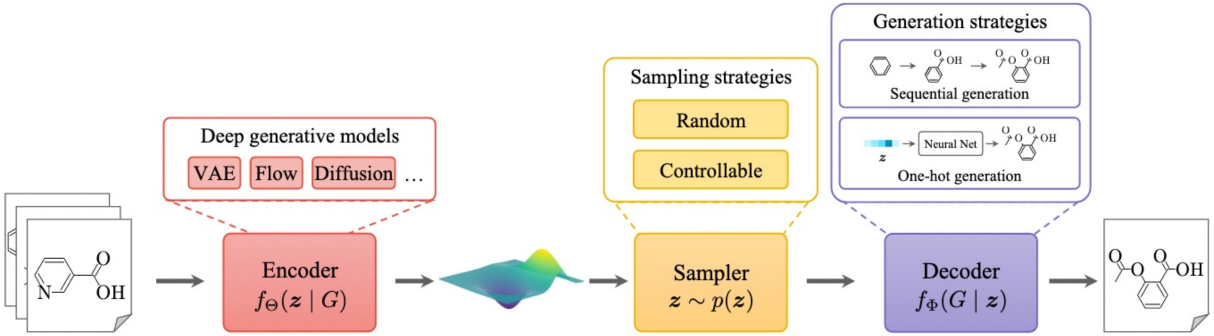
# Overview of deep graph generation - Sampler



A Survey on Deep Graph Generation: Methods and Applications, Yanqiao Zhu et al, LOG 2023

- sample latent representations from learned distribution $z \sim p(z)$.
- two sampling strategies: random sampling and controllable sampling.
- Random: randomly sampling latent codes from the learned distribution.
- controllable: sample latent code in an attempt to generate new graphs with desired properties.
  - In practice, controllable sampling usually depends on different types of deep generative models and requires an additional optimization term beyond random generation.

# Overview of deep graph generation - Decoder

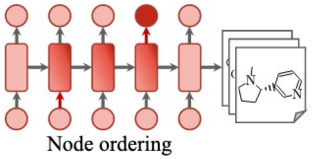


A Survey on Deep Graph Generation: Methods and Applications, Yanqiao Zhu et al, LOG 2023

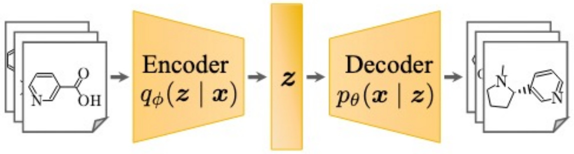
- The decoder receives the latent representations sampled from the learned distribution and generates graph structures.
- decoder is more complicated due to the discrete, non-Euclidean nature of graph objects.
- Decoder types:
  - sequential generation: generating graphs in consecutive steps, one node/edge at a time.
  - one-shot generation - generating node/edge feature matrices in single step.
- not all methods include all components – i.e. (GANs) do not include a specific encoder

# Graph generative models for deep graph generation

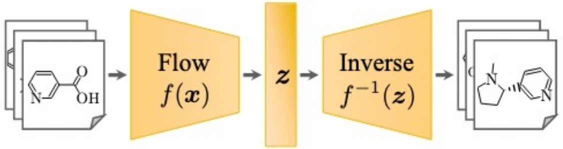
- **auto-regressive models**



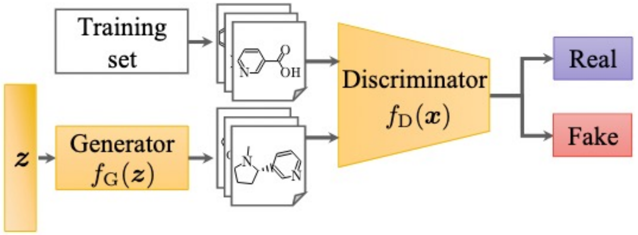
- variational autoencoders



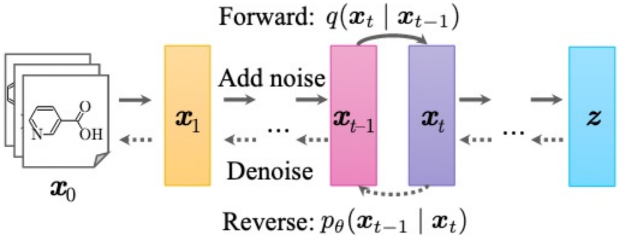
- normalizing flows



- generative adversarial networks



- diffusion models


Forward: $q(x_t \mid x_{t-1})$
Add noise
Denoise
Reverse: $p_\theta(x_{t-1} \mid x_t)$
$x_0$
$x_1$
$x_{t-1}$
$x_t$
$z$
...
O
OH
N

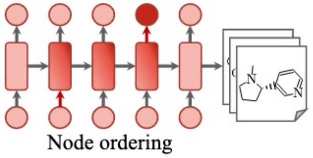
A Survey on Deep Graph Generation: Methods and Applications, Yanqiao Zhu et al, LOG 2023

# Deep Graph Generators -Auto-Regressive models

- AR models: Likelihood of a joint distribution over N random variables (nodes/edges) - chain rule of probability.
- generation process: determines the next step action (add node/edge/stop)given the current subgraph. The general formulation of AR models is as follows:

$$p(G^{\pi}) = \prod_{i=1}^{N} p(G_i^{\pi} \mid G_1^{\pi}, G_2^{\pi}, \cdots, G_{i-1}^{\pi}) = \prod_{i=1}^{N} p(G_i^{\pi} \mid G_{<i}^{\pi})$$

- where $G_{<i}^{\pi} = \{G_1^{\pi}, G_2^{\pi}, \cdots, G_{i-1}^{\pi}\}$ random variables in the previous N steps.
- constraint:  AR - sequential generation, requires pre-specified ordering of nodes in the graph.
- Many efforts [GraphRNN, 2018], [DeepGMG2018][Bacciu et al.2020], [Goyal et al.2020] ,  MolecularRNN [2021]…



# Deep Generative Models of Graphs - DeepGMG [Li et al., 2018]

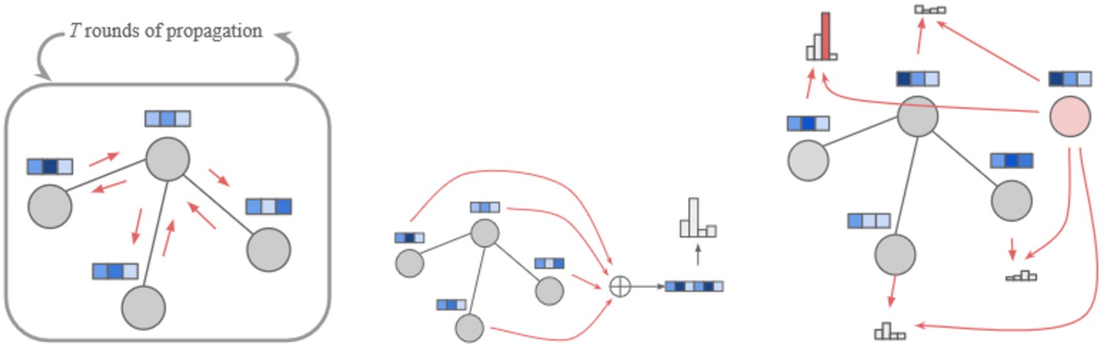
➔ **Autoregressive model** for graph generation: No prior structural assumption

➔ Generation process based on **sequential decisions**

- ◆ Generate **one node at a time**
- ◆ Connect node to graph at current state by creating edges (**one by one**)

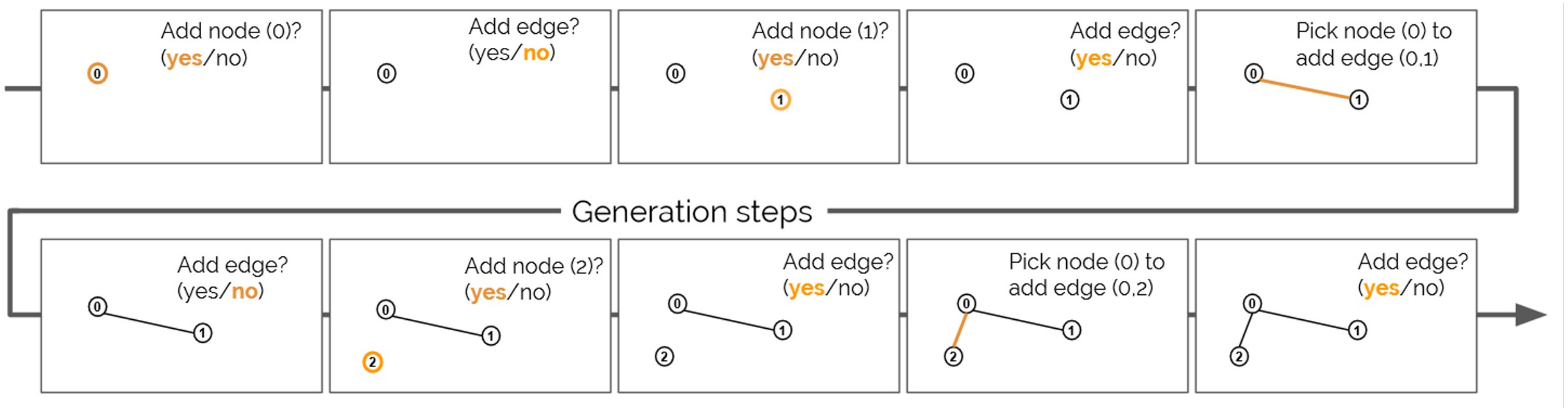
➔ Probability of new event depends on **history** of graph derivation

➔ Graphs are modeled by GNN



**Learning Deep Generative Models of Graphs**
Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, Peter Battaglia, https://arxiv.org/abs/1803.03324

# DeepGMG - Method



1. Check whether to add **a new node of a particular type** or **terminate** ← $f_{\text{add\_node}}$
2. If a node type is chosen, add that type node
3. Check if **any further edges** are needed to connect the new node to the existing graph ← $f_{\text{add\_edge}}$
4. If yes, **select a node** in the graph and **add an edge** connecting the new node to the selected node ← $f_{\text{nodes}}$
5. Return to (3)
6. Repeat **until** the model decides **not to add another edge**
7. Return to (1) to **add subsequent nodes**.

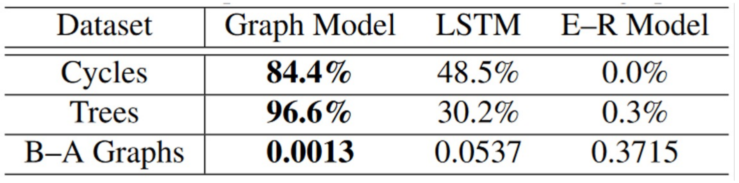
# Evaluation for DeepGMG

**Synthetic Graphs with Certain Topological Properties**

- Cycles, Trees and Barabasi–Albert graphs (power-law degree distribution)
- Report portion of valid samples that satisfy the given characteristic
- Report the KL-divergence between the degree distributions of samples and data for B-A graphs

**Molecule Generation**

- **ChEMBL molecule** database (20 most heavy atoms)
- RDKit: Convert SMILES string representations to Graph representation of molecules
- Node/Edge ordering
  - Fixed ordering: **canonical from SMILES**
  - **Uniform random ordering** by permutation
- Report **Negative Log-Likelihood (NLL)**
- Report potion of well-formatted (**valid**) samples and unique **novel** samples not seen in training set
- Report estimated **marginal likelihood** on small molecules (intractable on large molecules)



| Dataset | Graph Model | LSTM | E–R Model |
|---|---|---|---|
| Cycles | **84.4%** | 48.5% | 0.0% |
| Trees | **96.6%** | 30.2% | 0.3% |
| B–A Graphs | **0.0013** | 0.0537 | 0.3715 |

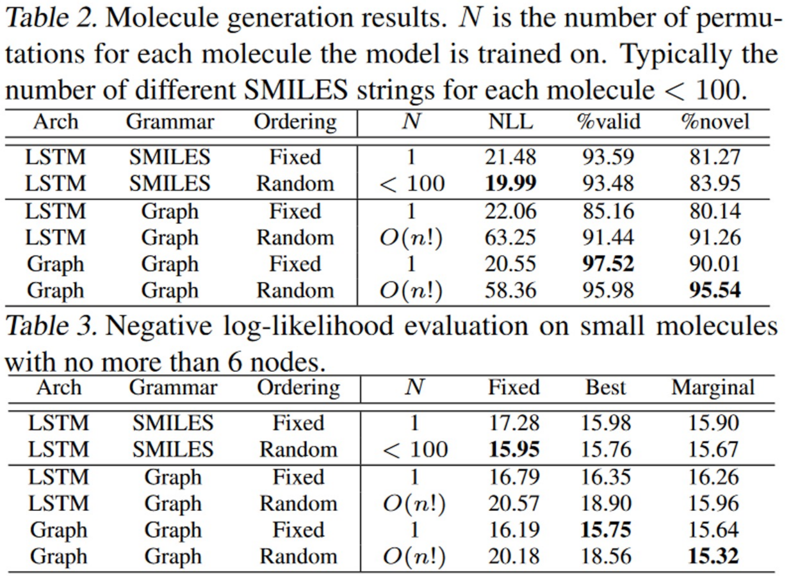


*Table 2.* Molecule generation results. $N$ is the number of permutations for each molecule the model is trained on. Typically the number of different SMILES strings for each molecule $< 100$.

| Arch | Grammar | Ordering | $N$ | NLL | %valid | %novel |
|---|---|---|---|---|---|---|
| LSTM | SMILES | Fixed | 1 | 21.48 | 93.59 | 81.27 |
| LSTM | SMILES | Random | $< 100$ | **19.99** | 93.48 | 83.95 |
| LSTM | Graph | Fixed | 1 | 22.06 | 85.16 | 80.14 |
| LSTM | Graph | Random | $O(n!)$ | 63.25 | 91.44 | 91.26 |
| Graph | Graph | Fixed | 1 | 20.55 | **97.52** | 90.01 |
| Graph | Graph | Random | $O(n!)$ | 58.36 | 95.98 | **95.54** |

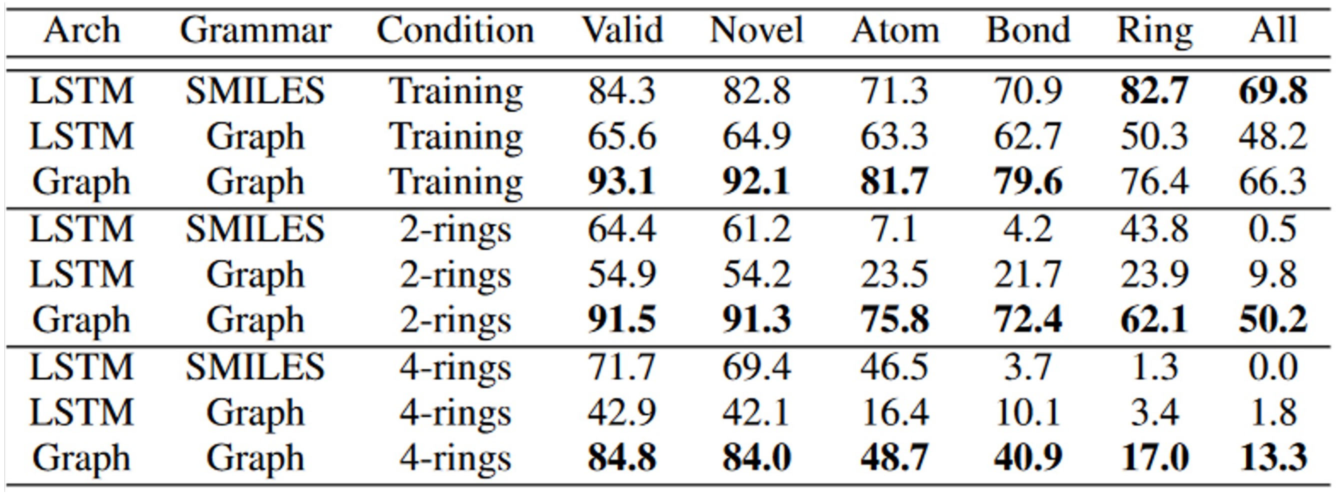
*Table 3.* Negative log-likelihood evaluation on small molecules with no more than 6 nodes.

| Arch | Grammar | Ordering | $N$ | Fixed | Best | Marginal |
|---|---|---|---|---|---|---|
| LSTM | SMILES | Fixed | 1 | 17.28 | 15.98 | 15.90 |
| LSTM | SMILES | Random | $< 100$ | **15.95** | 15.76 | 15.67 |
| LSTM | Graph | Fixed | 1 | 16.79 | 16.35 | 16.26 |
| LSTM | Graph | Random | $O(n!)$ | 20.57 | 18.90 | 15.96 |
| Graph | Graph | Fixed | 1 | 16.19 | **15.75** | 15.64 |
| Graph | Graph | Random | $O(n!)$ | 20.18 | 18.56 | **15.32** |

# Evaluation for DeepGMG

Conditional Graph Generation

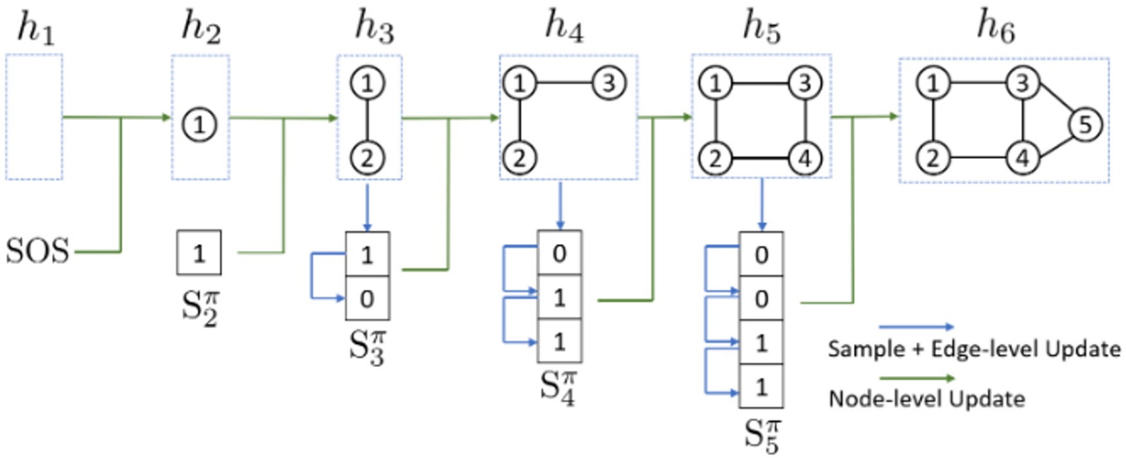
- ➔ A subset of previous ChEMBL database with contains molecules of 0, 1 and 3 aromatic rings
- ➔ Report potion of well-formatted (**valid**) samples and unique **novel** samples not seen in training set
- ➔ Report portion of samples that have the number of atoms, bonds, rings, and all three that **match the given condition**



| Arch | Grammar | Condition | Valid | Novel | Atom | Bond | Ring | All |
|---|---|---|---|---|---|---|---|---|
| LSTM | SMILES | Training | 84.3 | 82.8 | 71.3 | 70.9 | **82.7** | **69.8** |
| LSTM | Graph | Training | 65.6 | 64.9 | 63.3 | 62.7 | 50.3 | 48.2 |
| Graph | Graph | Training | **93.1** | **92.1** | **81.7** | **79.6** | 76.4 | 66.3 |
| LSTM | SMILES | 2-rings | 64.4 | 61.2 | 7.1 | 4.2 | 43.8 | 0.5 |
| LSTM | Graph | 2-rings | 54.9 | 54.2 | 23.5 | 21.7 | 23.9 | 9.8 |
| Graph | Graph | 2-rings | **91.5** | **91.3** | **75.8** | **72.4** | **62.1** | **50.2** |
| LSTM | SMILES | 4-rings | 71.7 | 69.4 | 46.5 | 3.7 | 1.3 | 0.0 |
| LSTM | Graph | 4-rings | 42.9 | 42.1 | 16.4 | 10.1 | 3.4 | 1.8 |
| Graph | Graph | 4-rings | **84.8** | **84.0** | **48.7** | **40.9** | **17.0** | **13.3** |

# GraphRNN [You et al., 2018]

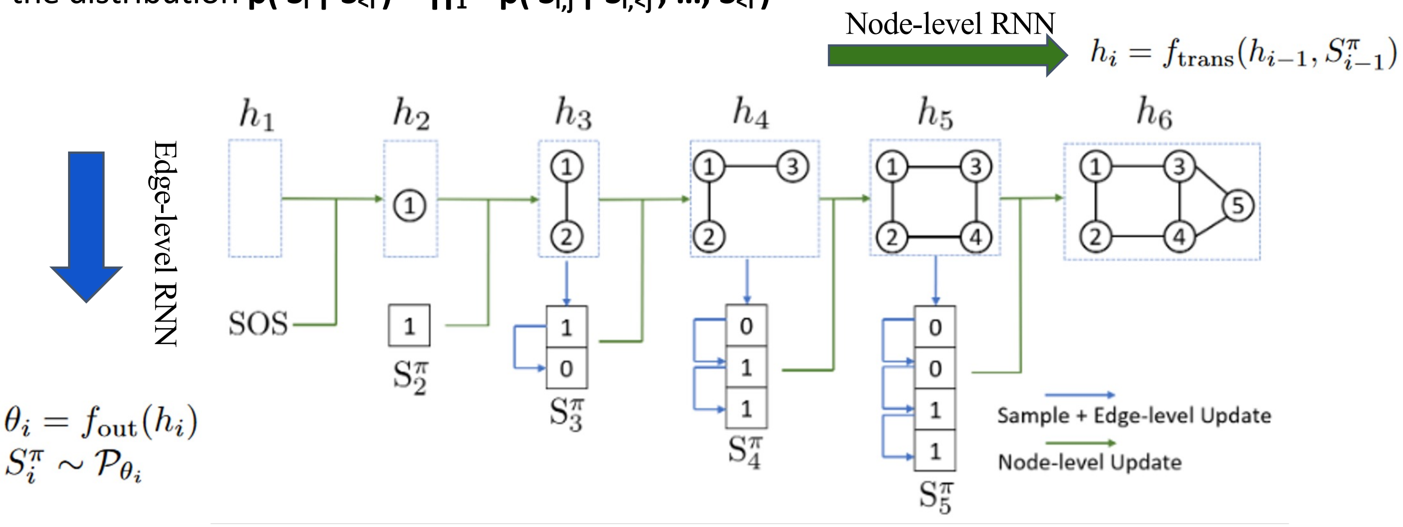
- **Autoregressive model** for graph generation
  - Key insight: Graph **G** with node permutation $\boldsymbol{\pi}$ can be uniquely mapped into a sequence of node and edge additions $\mathbf{S}^{\boldsymbol{\pi}}$
  - Model the generation process with two RNNs
    - Node-level: generate a state for a new node
    - Edge-level: generate edges for the new node based its hidden state



GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models
Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, Jure Leskovec

# GraphRNN - Method

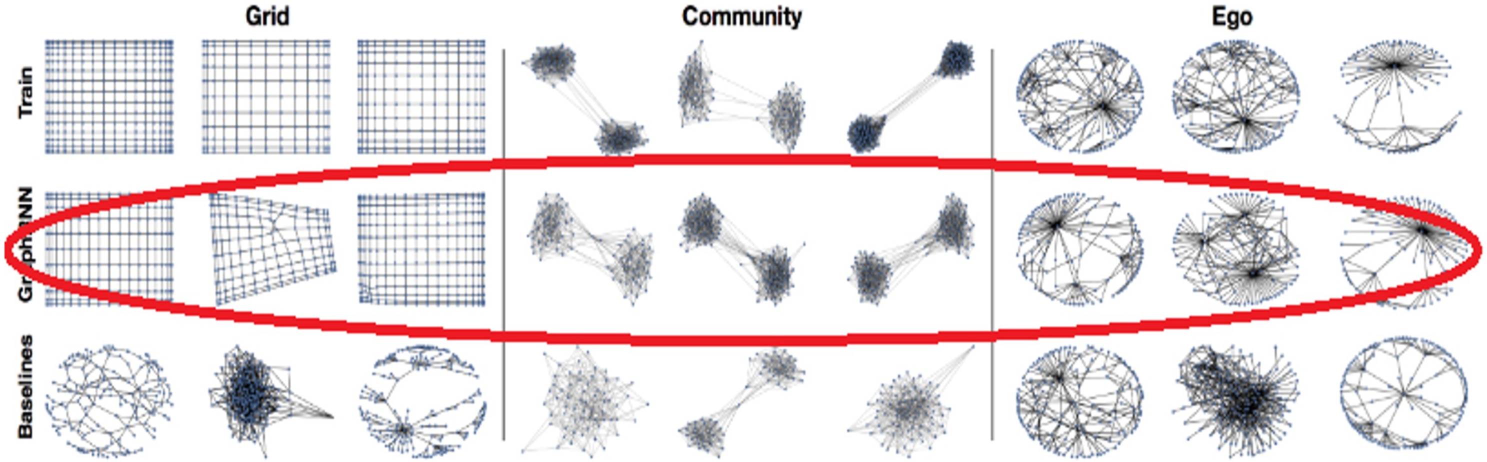
- ➔ Omitting symbol of node permutation **π**, Graph **G~p(G)** can be represented as a sequence of adjacency vectors **($S_1$ , …, $S_n$)**
- ➔ $S_i$ is a **(i-1)** dimensional *vector represents edges between **i** and previous nodes*: **{0, 1}$^{i-1}$**
- ➔ **p(G)** is related to **p(S)** with **p(S) = $\prod_1^{n+1}$ p( $S_i$ | $S_1$, …, $S_{i-1}$ )**. This product can be modelled by a RNN (node-level) with state transition possible modelled by another RNN (edge-level), which models the distribution **p( $S_i$ | $S_{<i}$ ) = $\prod_1^{i-1}$ p( $S_{i,j}$ | $S_{i,<j}$ , …, $S_{<i}$ )**


Node-level RNN
$h_i = f_{\text{trans}}(h_{i-1}, S^\pi_{i-1})$
Edge-level RNN
$h_1$
$h_2$
$h_3$
$h_4$
$h_5$
$h_6$
1
2
3
4
5
SOS
1
$S^\pi_2$
1
0
$S^\pi_3$
0
1
1
$S^\pi_4$
0
0
1
1
$S^\pi_5$
$\theta_i = f_{\text{out}}(h_i)$
$S^\pi_i \sim \mathcal{P}_{\theta_i}$
Sample + Edge-level Update
Node-level Update

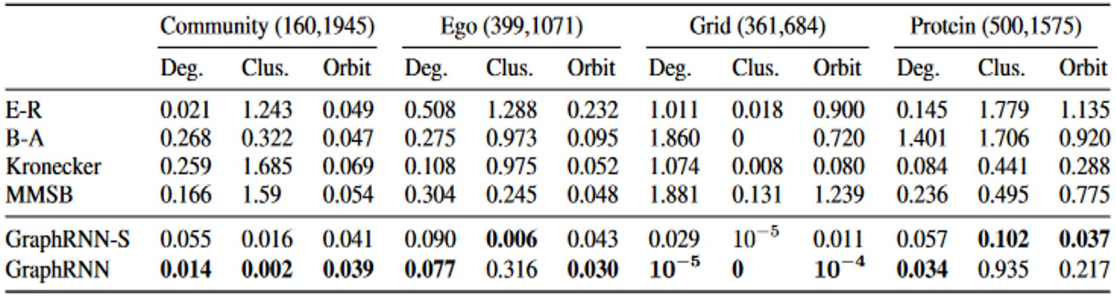
# Evaluation for GraphRNN

Visual comparison

- ➔ First row: Training set
- ➔ Third row: Kronecker graph, Mixed-Membership Stochastic Block model and Barabasi–Albert graph

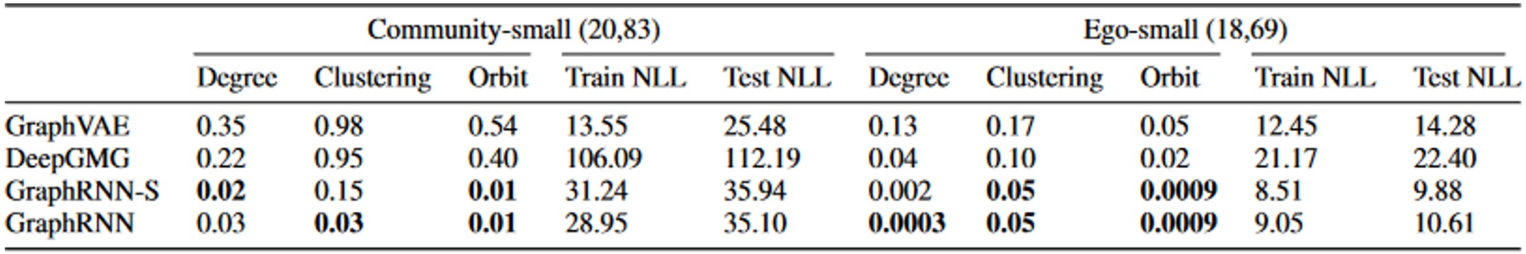


# Evaluation for GraphRNN



| | 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 |

$$\mathrm{MMD}^2(p\|q) = \mathbb{E}_{x,y\sim p}[k(x,y)] + \mathbb{E}_{x,y\sim q}[k(x,y)] \\ - 2\mathbb{E}_{x\sim p,y\sim q}[k(x,y)].$$



| | Community-small (20,83) | | | | | Ego-small (18,69) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Degree | Clustering | Orbit | Train NLL | Test NLL | Degree | Clustering | Orbit | Train NLL | Test NLL |
| GraphVAE | 0.35 | 0.98 | 0.54 | 13.55 | 25.48 | 0.13 | 0.17 | 0.05 | 12.45 | 14.28 |
| DeepGMG | 0.22 | 0.95 | 0.40 | 106.09 | 112.19 | 0.04 | 0.10 | 0.02 | 21.17 | 22.40 |
| GraphRNN-S | **0.02** | 0.15 | **0.01** | 31.24 | 35.94 | 0.002 | **0.05** | **0.0009** | 8.51 | 9.88 |
| GraphRNN | 0.03 | **0.03** | **0.01** | 28.95 | 35.10 | **0.0003** | **0.05** | **0.0009** | 9.05 | 10.61 |

**Maximum Mean Discrepancy (MMD)**

➔ Compare all moments of the empirical distributions using an exponential kernel with Wasserstein distance

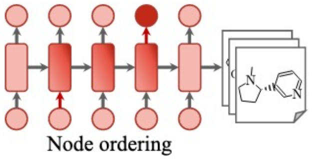
**80% decrease** of **MMD** over traditional baselines: E-R, B-A, Kronecker, MMSB

**90% decrease** of **MMD** over deep learning baselines

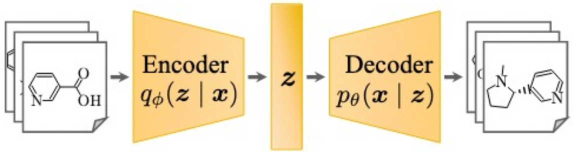
**22% smaller** average **NLL** gap compared to deep learning baselines

# Graph generative models for deep graph generation

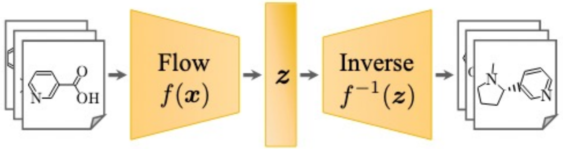
- auto-regressive models



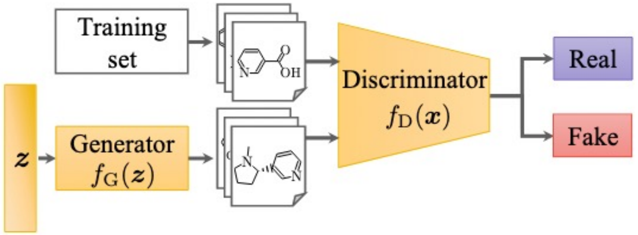
- **variational autoencoders**



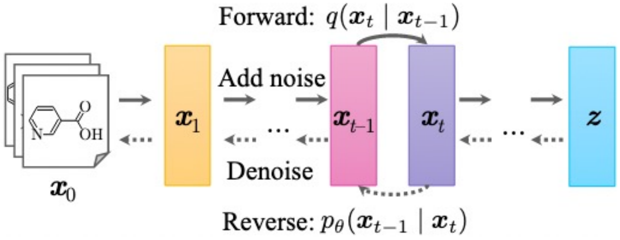
- normalizing flows



- generative adversarial networks

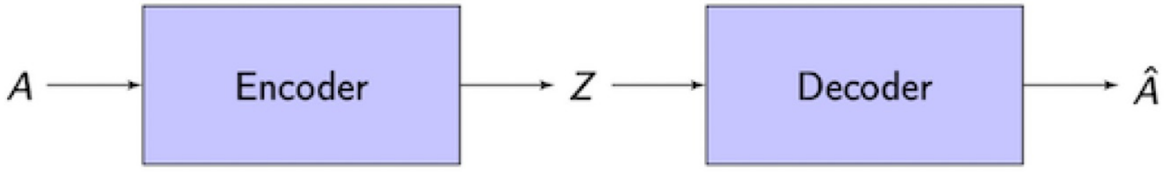


- diffusion models

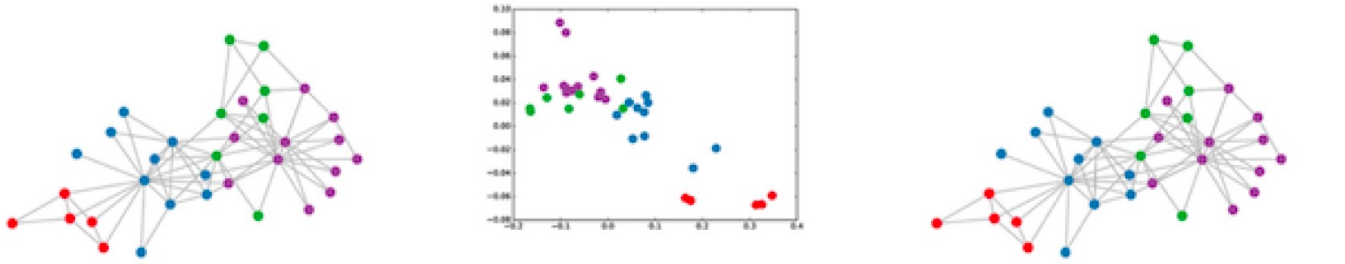

Forward: $q(x_t \mid x_{t-1})$
Add noise
$x_0$
$x_1$
$x_{t-1}$
$x_t$
$z$
Denoise
Reverse: $p_\theta(x_{t-1} \mid x_t)$

A Survey on Deep Graph Generation: Methods and Applications, Yanqiao Zhu et al, LOG 2023

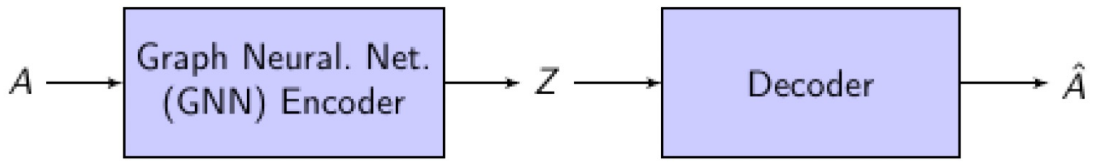
# Graph Auto encoders



$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & \dots & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 0.523 & -1.012 \\ 2.127 & 0.316 \\ 0.912 & 0.127 \\ \dots & \dots \\ -1.210 & 0.026 \end{pmatrix} \qquad \hat{A} = \begin{pmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 0 & \dots & 0 \end{pmatrix}$$



# Graph Auto encoders



A

Graph Neural. Net. (GNN) Encoder

Z

Decoder

$\hat{A}$

**Graph AE:**

1. **encoder:** $\boxed{Z = \mathrm{GNN}(A, X)}$
2. **decoder:** $\boxed{\hat{A} = \sigma(ZZ^\top)}$
   i.e., for all node pairs $(i, j)$, we have $\hat{A}_{ij} = \sigma(z_i^\top z_j)$

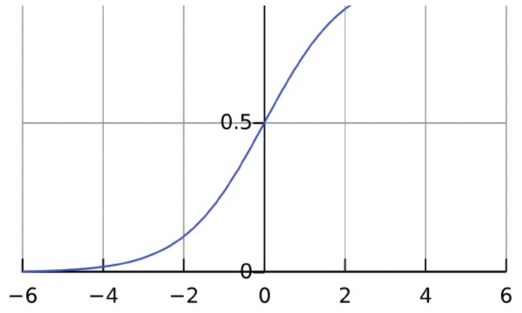
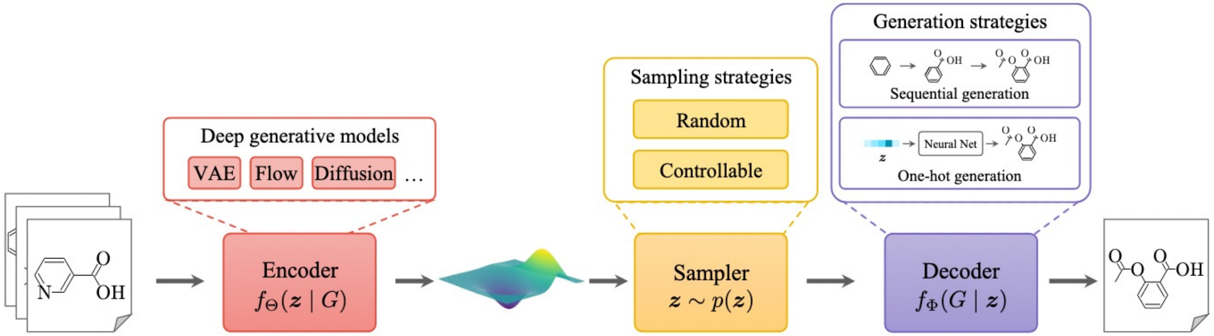


Figure: Sigmoid activation: $\sigma(x) = \frac{1}{1+e^{-x}}$

**Reconstruction Loss**[1]**:** capturing the similarity between A and $\hat{A}$

- e.g., **cross-entropy loss**: $-\sum_{i=1}^{n}\sum_{j=1}^{n}(A_{ij}\log(\hat{A}_{ij}) + (1 - A_{ij})\log(1 - \hat{A}_{ij}))$
- or **MSE** loss: $\sum_{i=1}^{n}\sum_{j=1}^{n}(A_{ij} - \hat{A}_{ij})^2$

# Graph Variational Auto Encoders



A Survey on Deep Graph Generation: Methods and Applications, Yanqiao Zhu et al, LOG 2023

- VAE[1] estimates the distributions of graphs $p(G)$ by maximizing the Evidence Lower Bound (ELBO):

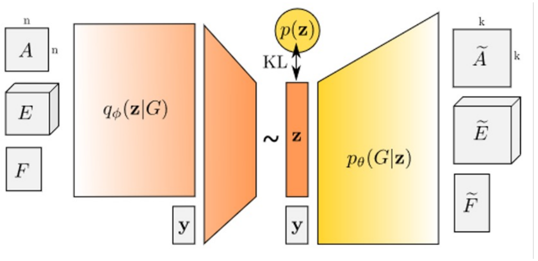
$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{z\sim q_\phi(\boldsymbol{z}|G)} \log(p_\theta(G \mid \boldsymbol{z})) - D_{\text{KL}}(q_\phi(\boldsymbol{z} \mid G) \,\|\, p_\theta(\boldsymbol{z})))$$

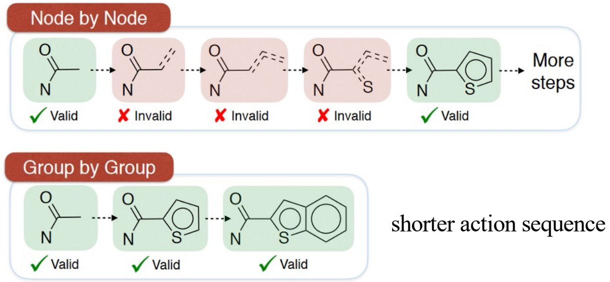
- reconstruction loss between the input $G$ and the reconstructed graph, distance among decoder $q_\phi(z/G)$ and the prior distribution $p_\theta(z)$ – usually Gaussian.
- The encoder $p(z/G)$ and decoder $q(G/z)$ are typically GNNs (i.e. GCN, GAT…).

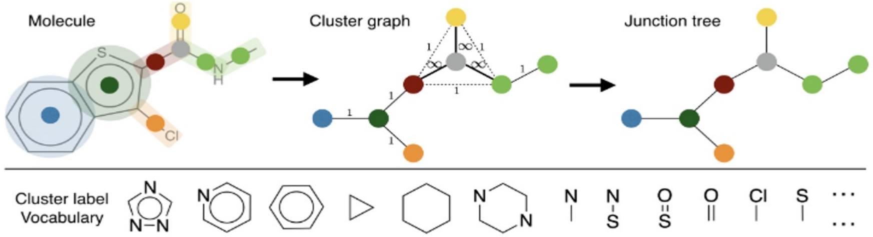
[1] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In ICLR, 2014.

# Junction Tree Variational AutoEncoder – JTVAE [Jin et al., 2018]

- ➔ GraphVAE [Simonovsky and Komodakis, 2017]
  - ◆ Generate a probabilistic fully-connected graph
    - ● Model node/edge with Bernoulli distribution
    - ● Model node/edge features using multinomial distribution
  - ◆ **Loss: Similarity (KL divergence)** between $q_\theta(z|G)$ and prior distribution (normal) $p(z)$ + **similarity (likelihood)** between generated graph $\tilde{G}$ and input graph $G$ .
- ➔ Junction Tree VAE: **molecule generation** leveraging chemical domain knowledge
  - ◆ Instead of generating graph **node by node**, generate **group (structure) by group (structure)**: **functional groups** by tree decomposition of molecular graphs
  - ◆ Less than 800 groups given 250K molecules



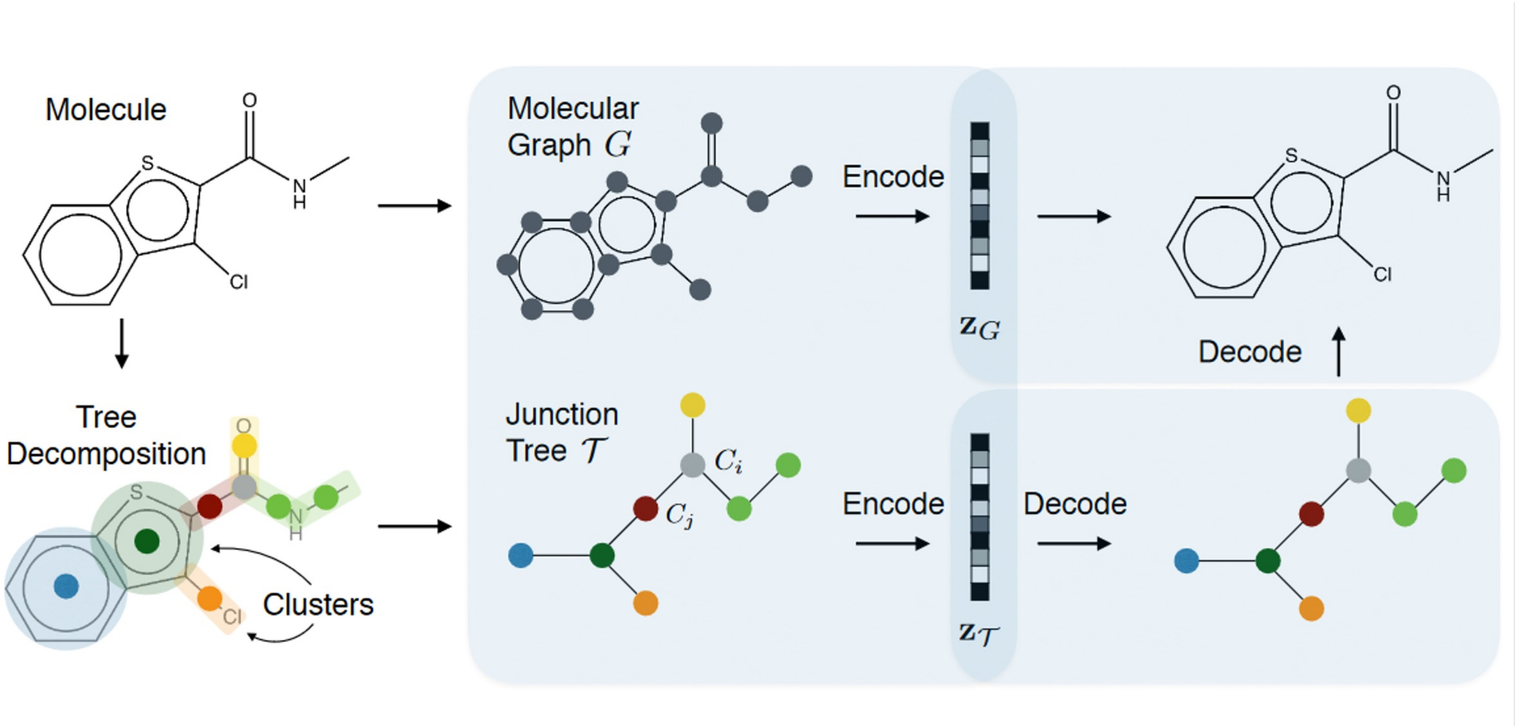




Motivations:

- ● **Not** every VAE- generated graph is **chemically valid**
- ● Long action sequence (intermediate states) are **hard to validate** and **difficult to train** (DeepGMG, [Li et al., 2018])

# JTVAE - Method

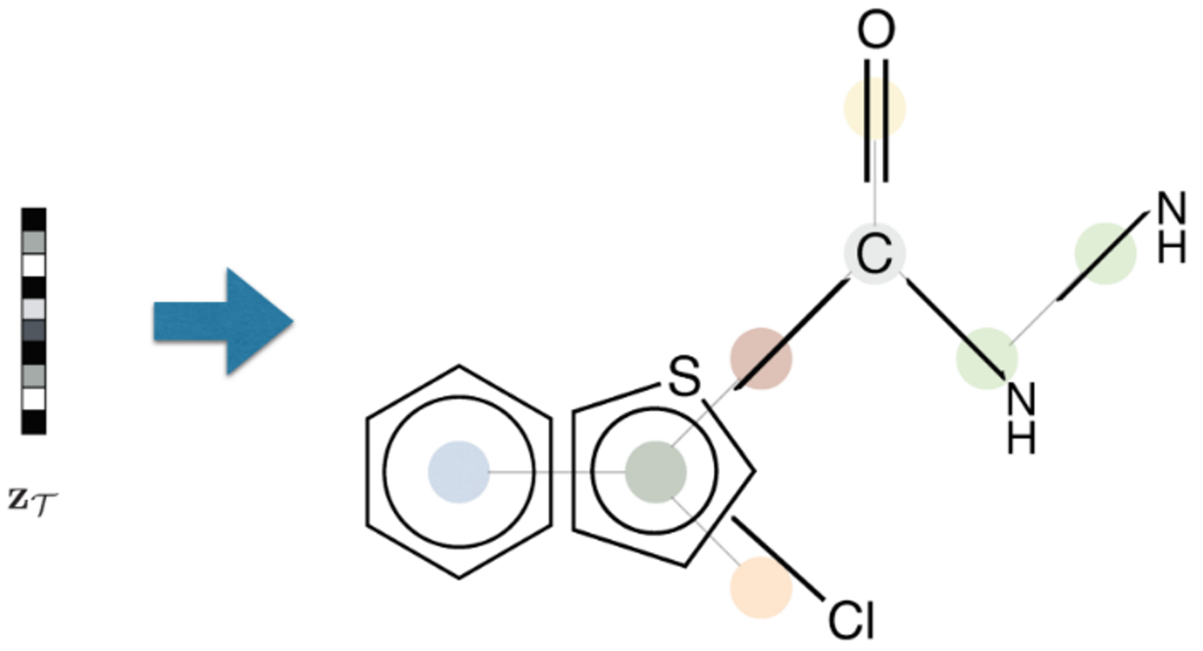


Graph & Tree **encoder**: **graph message passing networks** [Dai et al., 2016]

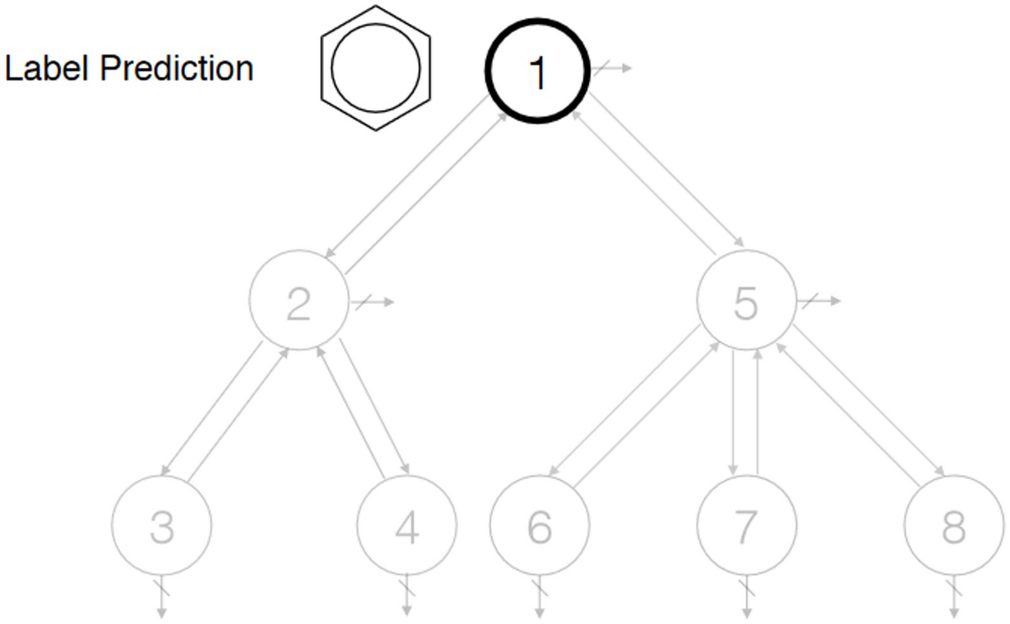
- Graph encoder output: average pooling
- Tree encoder output: embedding of the root node

**Decoder?**

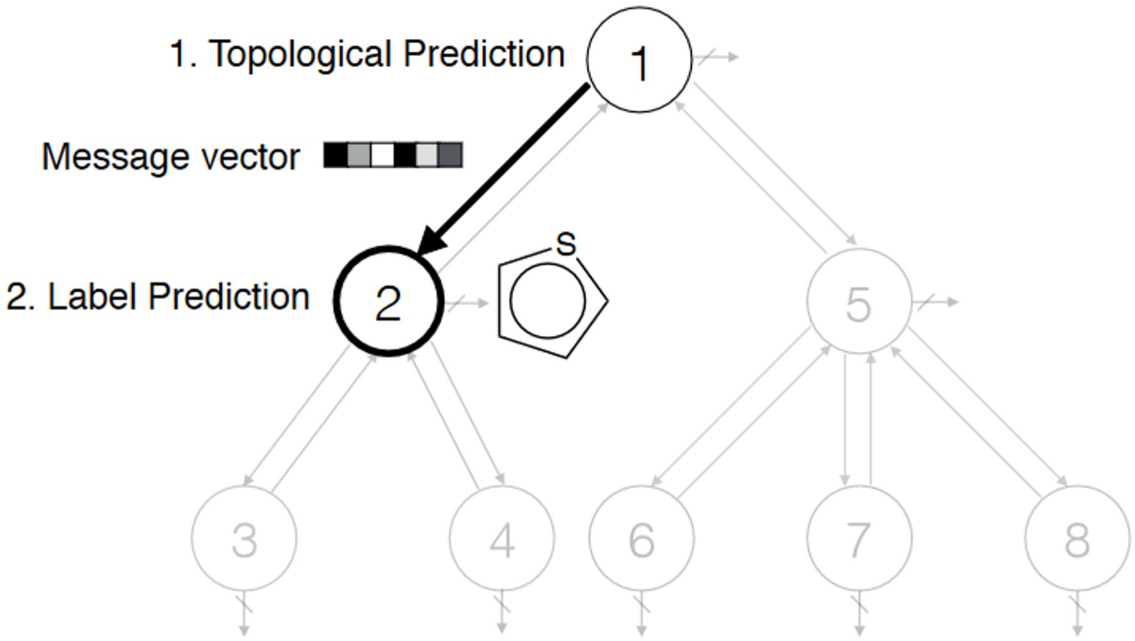
# JTVAE - Tree Decoding



# JTVAE - Tree Decoding



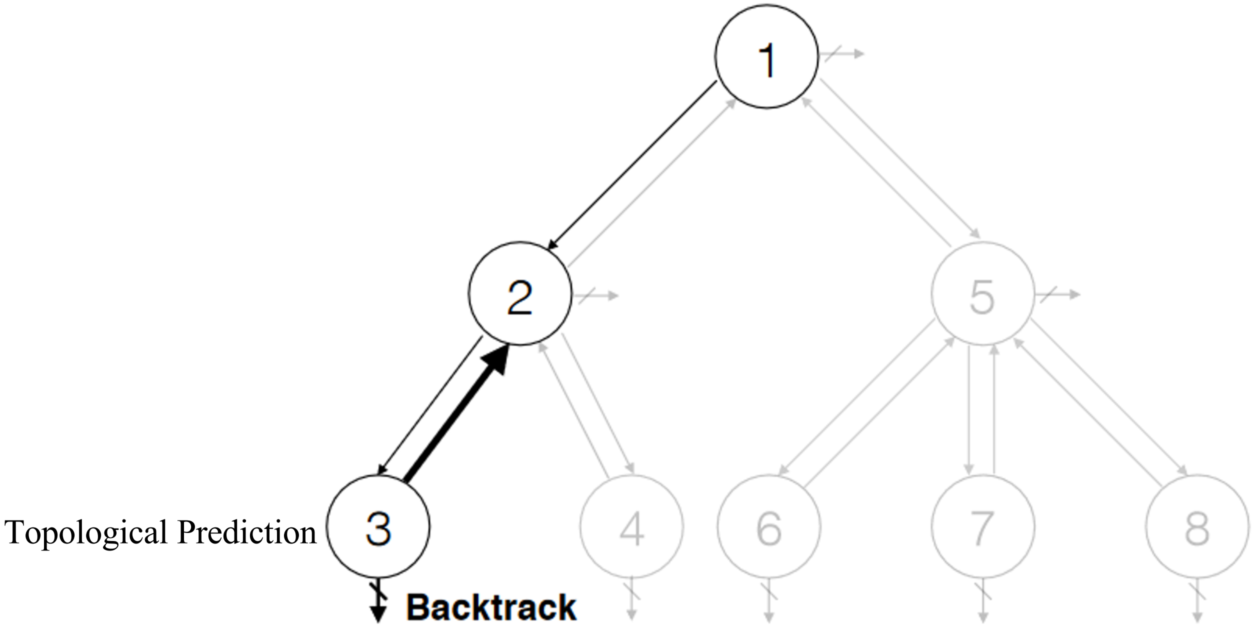
# JTVAE - Tree Decoding



**Topological Prediction:** Whether to **add a child node** or **backtrack**?

**Label Prediction:** What is the **label** of the new node?

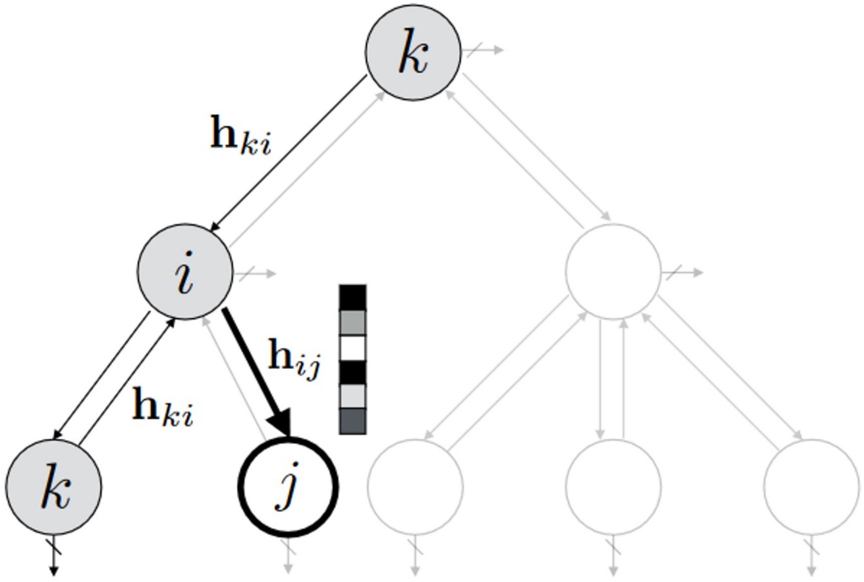
# JTVAE - Tree Decoding

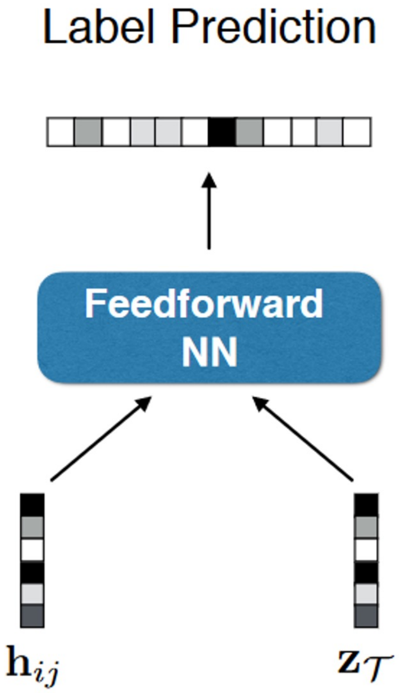


**Topological Prediction:** Whether to **add a child node** or **backtrack**?

**Label Prediction:** What is the **label** of the new node?

# JTVAE - Tree Decoding





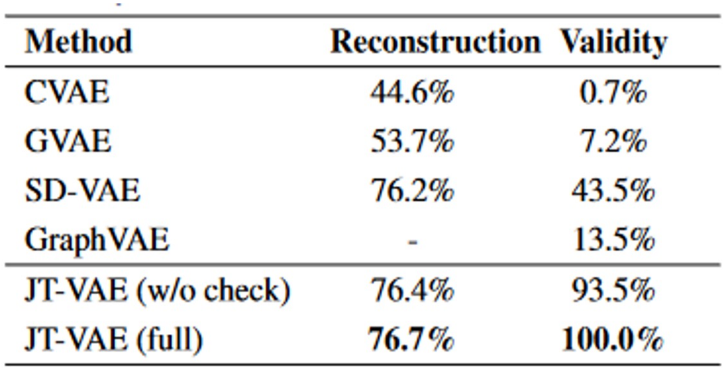
$$\mathbf{h}_{ij} = \mathrm{GRU}(\mathbf{x}_i, \{\mathbf{h}_{ki}\}_{k \in N_t(i) \backslash j})$$

Encodes the entire subtree of current state

Functional group features

# Evaluation for JTVAE

- ➔ **Molecule Reconstruction on ZINC [Sterling and Irwin, 2015]**
  - ◆ Reconstruct input molecules from latent representations
  - ◆ 100 montecarlo trials [Kusner et al., 2017] per molecule
  - ◆ **Report portion** of decoded molecules **identical to input**
- ➔ **Molecule Validity**
  - ◆ Decode valid molecules when sampling from prior distribution
  - ◆ 100 monte-carlo trials per latent **z** sampled from prior distribution
  - ◆ **Report portion** of decoded molecules **chemically valid (RDKit)**

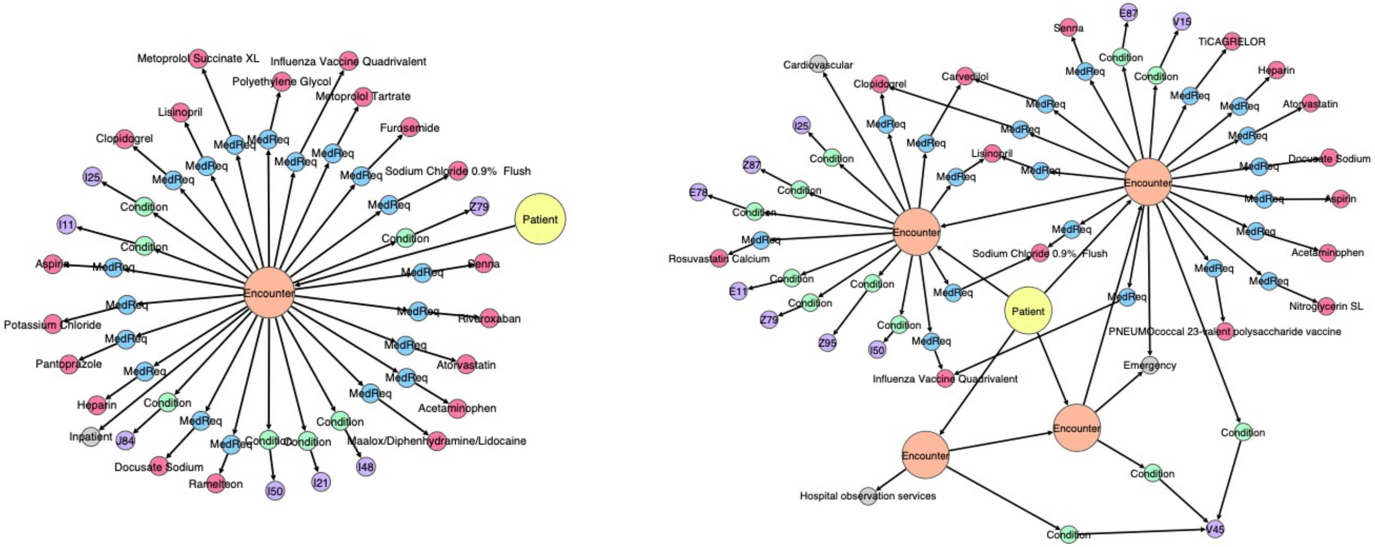


| **Method** | **Reconstruction** | **Validity** |
|---|---|---|
| CVAE | 44.6% | 0.7% |
| GVAE | 53.7% | 7.2% |
| SD-VAE | 76.2% | 43.5% |
| GraphVAE | - | 13.5% |
| JT-VAE (w/o check) | 76.4% | 93.5% |
| JT-VAE (full) | **76.7%** | **100.0%** |

# GNNs and Graph Generative models for biomedical applications

- Graph Generative models
- **Generative models for Medical Graphs**
- Large generative models
- Graph / LLMs
- Multi modality for molecule generation
- Conclusions

# GVAE for Generating Synthetic Patient Trajectories



Synthetic electronic health records generated with variational graph autoencoders,
G. Nikolentzos, M. Vazirgiannis, C. Xypolopoulos, M. Lingman, E. G. Brandt NATURE DIGITAL MEDICINE 2023,
https://www.nature.com/articles/s41746-023-00822-x

# Health data - constraints

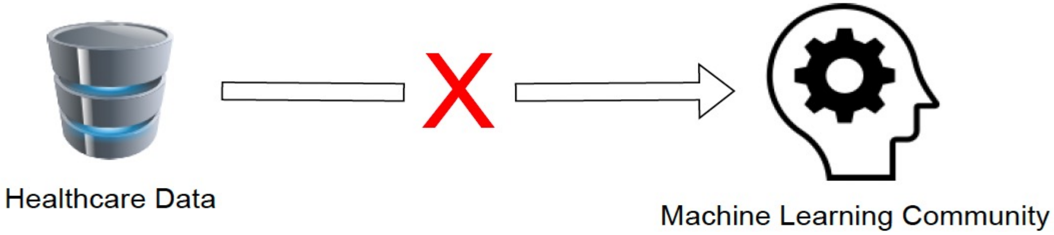
In many fields, open access data sets lead to significant progress, e. g.,

- Computer vision → Imagenet
- Natural language processing → Wordnet

However, in the case of healthcare data:

- Lack of high-quality healthcare data
- Strict regulations for data access:
  - The use of patient data leads to privacy concerns
  - Regulations like HIPAA prohibit the unauthorized use and disclosure of protected health information
  - Patient data cannot be freely shared

The above impedes machine learning research in healthcare!



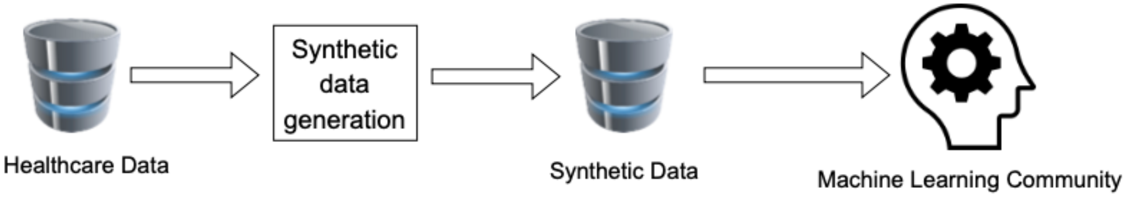
Healthcare Data

X

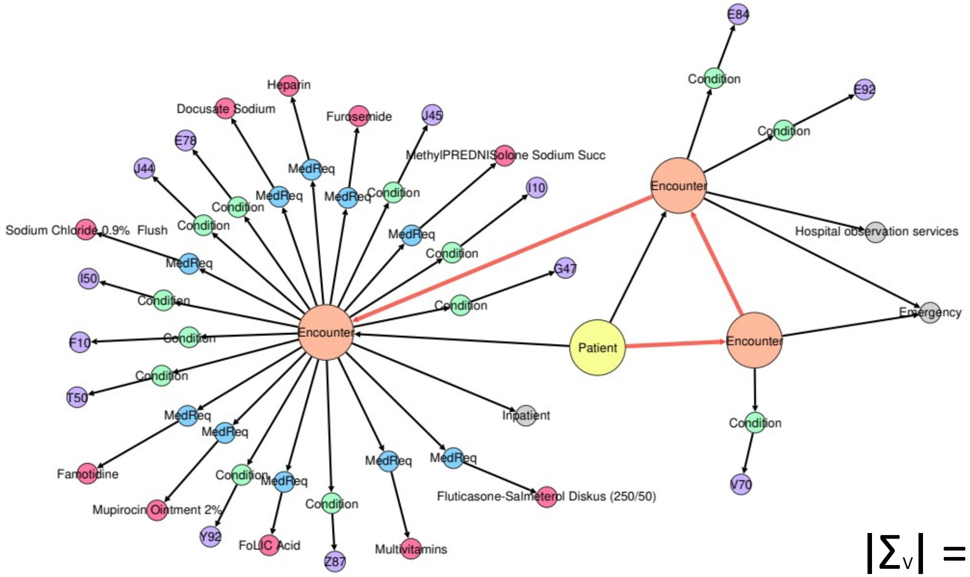
Machine Learning Community

# Synthetic Data Generation

- Approaches that generate synthetic data could help the research community deal with those challenges
- Neural networks are recently used to generate synthetic data
  ↪ Real data is fed into the model which learns to produce synthetic data that very closely resembles the real dataset
- The neural network model is designed to produce synthetic data that does not violate data privacy regulations
- Once the synthetic dataset has been created
  - it can be used to train machine learning models (develop analytics, data augmentation, increase robustness of models)
  - it can be shared across research teams and institutions facilitating reproducibility of different models and collaboration



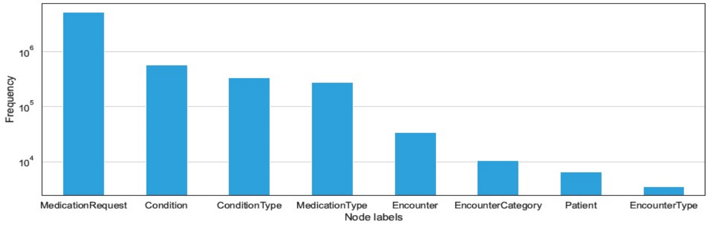
# Shaarpeec graph model for patient trajectories

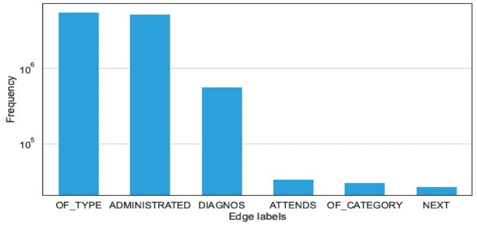


@SHAARPEC graph data model → patient trajectories represented as graphs
https://shaarpec.com/

$|\Sigma_V| = 13980$ node labels

$|\Sigma_E| = \ 6$ edge labels.



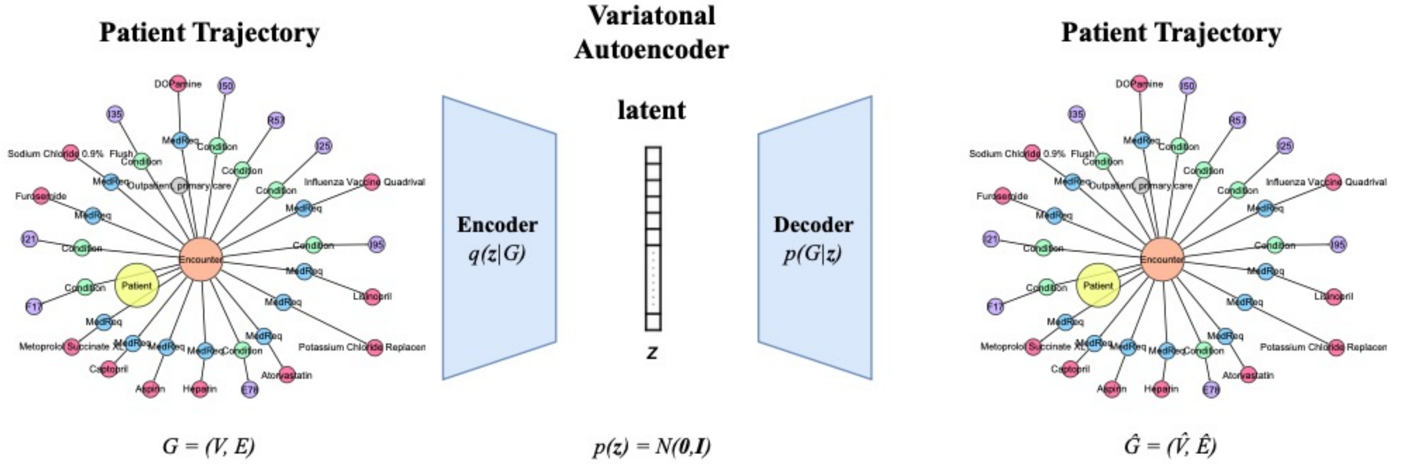


# GVAE architecture

We use a variational autoencoder:

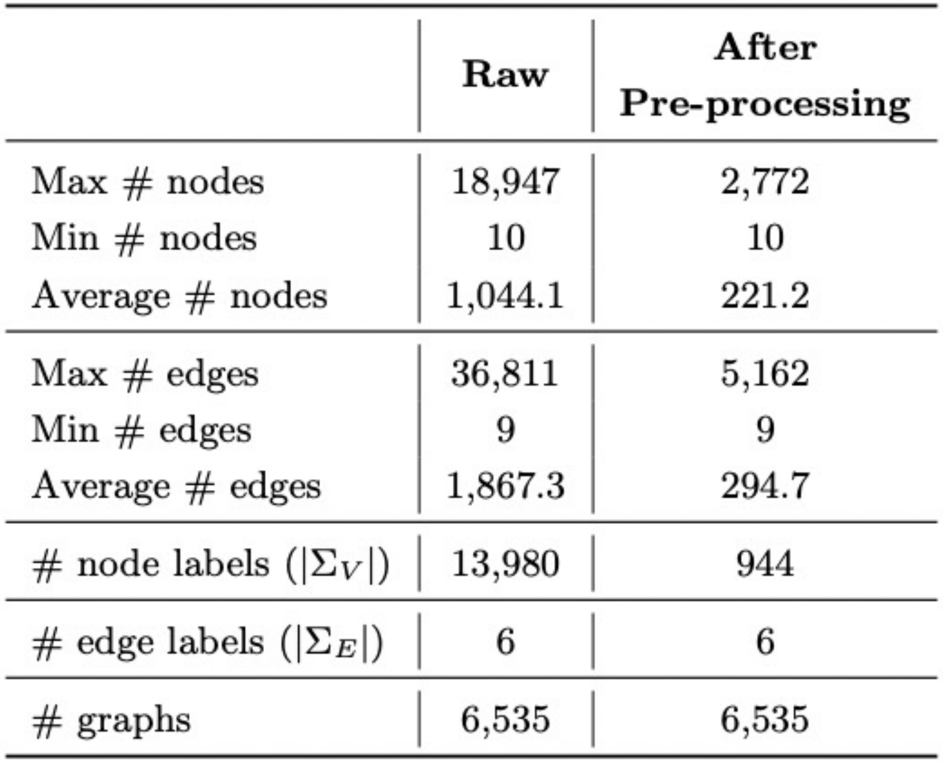
- The encoder maps input DAGs into d-dimensional gaussian distributions

- The decoder reconstructs the input DAGS given vectors sampled from the gaussian distributions



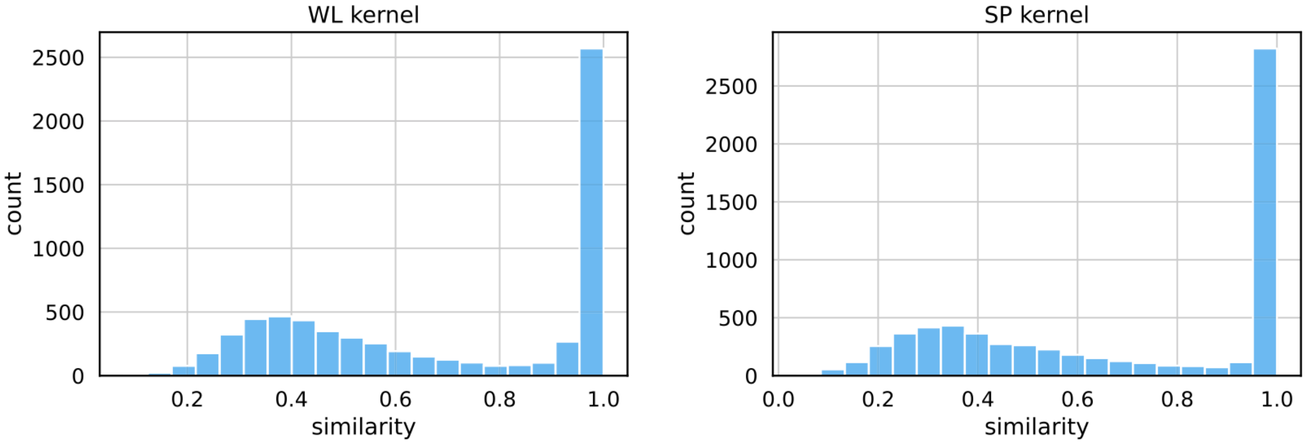
# MIMIC dataset

Statistics on the patient trajectories calculated from the *atrial fibrillation* cohort from the MIMIC-IV database.



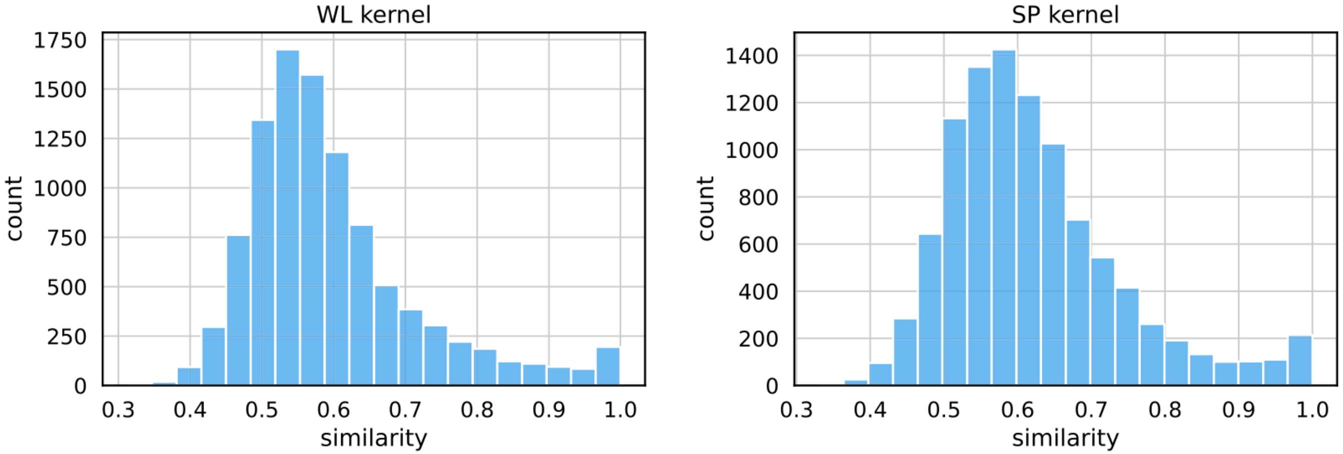
| | **Raw** | **After Pre-processing** |
|---|---|---|
| Max # nodes | 18,947 | 2,772 |
| Min # nodes | 10 | 10 |
| Average # nodes | 1,044.1 | 221.2 |
| Max # edges | 36,811 | 5,162 |
| Min # edges | 9 | 9 |
| Average # edges | 1,867.3 | 294.7 |
| # node labels ($\lvert\Sigma_V\rvert$) | 13,980 | 944 |
| # edge labels ($\lvert\Sigma_E\rvert$) | 6 | 6 |
| # graphs | 6,535 | 6,535 |

# Quality of graph reconstruction



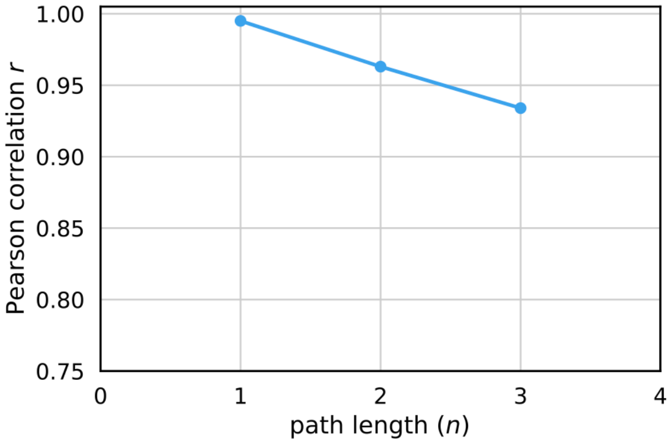
- Real graph input to encoder - decoder produces a reconstructed version
- Histogram of similarities between input graphs and reconstructed graphs using the Weisfeiler Lehman subtree (WL) kernel and the shortest path (SP) kernel.

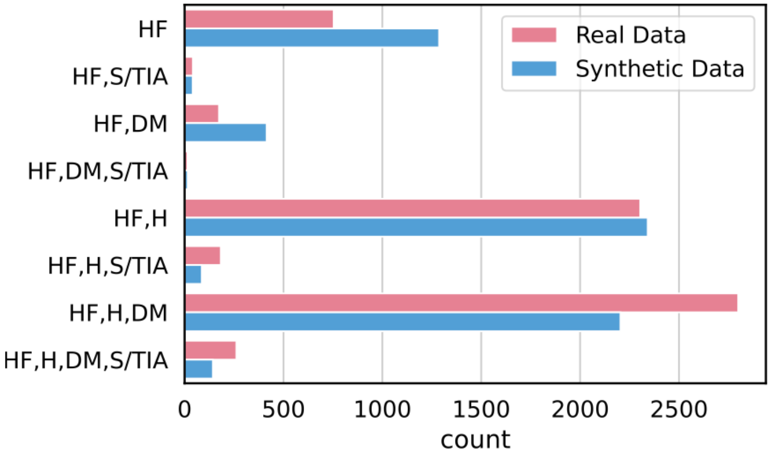
# Graph generation – similarity to real graphs



- Generated graph: Sample from the learned distribution and feed the decoder
- Similarity histogram between input and generated graphs using the Weisfeiler-Lehman subtree (WL) kernel and the shortest path (SP) kernel.
- lower similarity => privacy

# Graph generation – similarity to real graphs

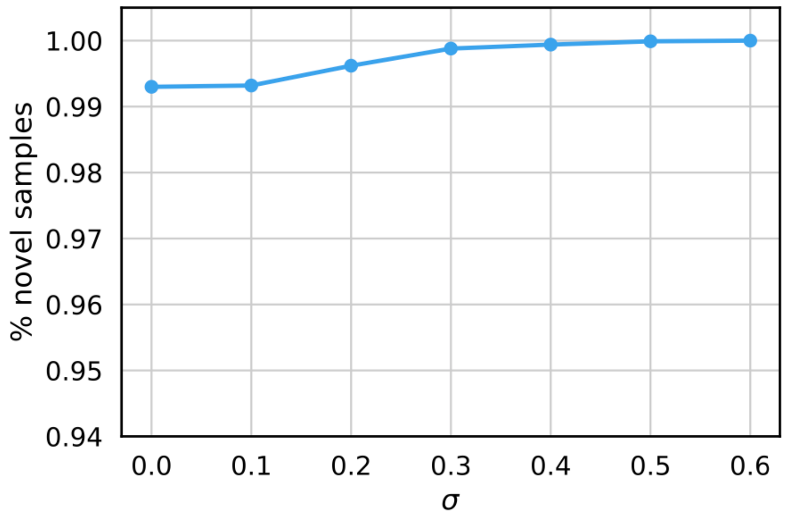




- Pearson's r between *frequency of different structures in training trajectories and generated trajectories.*
- 1-paths (node labels), 2-paths, and 3-paths

- Comorbidities for real and synthetic atrial fibrillation cohorts.
- comorbidities: HF heart failure, S/TIA stroke/TIA, DM diabetes mellitus, H hypertension.

# Privacy concerns



- % novel trajectories vs standard deviation of the Gaussian noise.
- mean Gaussian = **0** vector.
- noise added to Encounters.
- σ ∈ {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6}, 20000 trajectories generated, compared to real

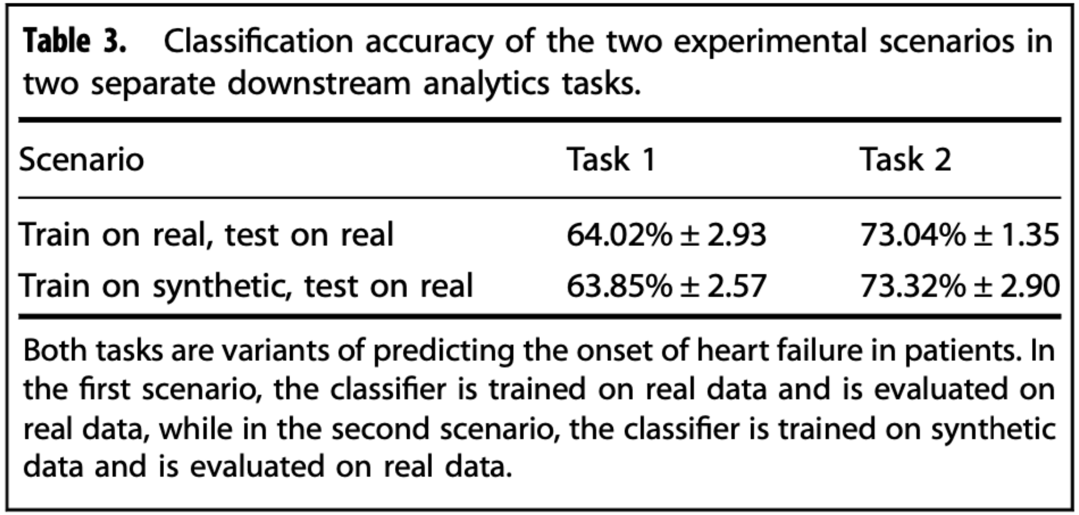


**Table 3.** Classification accuracy of the two experimental scenarios in two separate downstream analytics tasks.

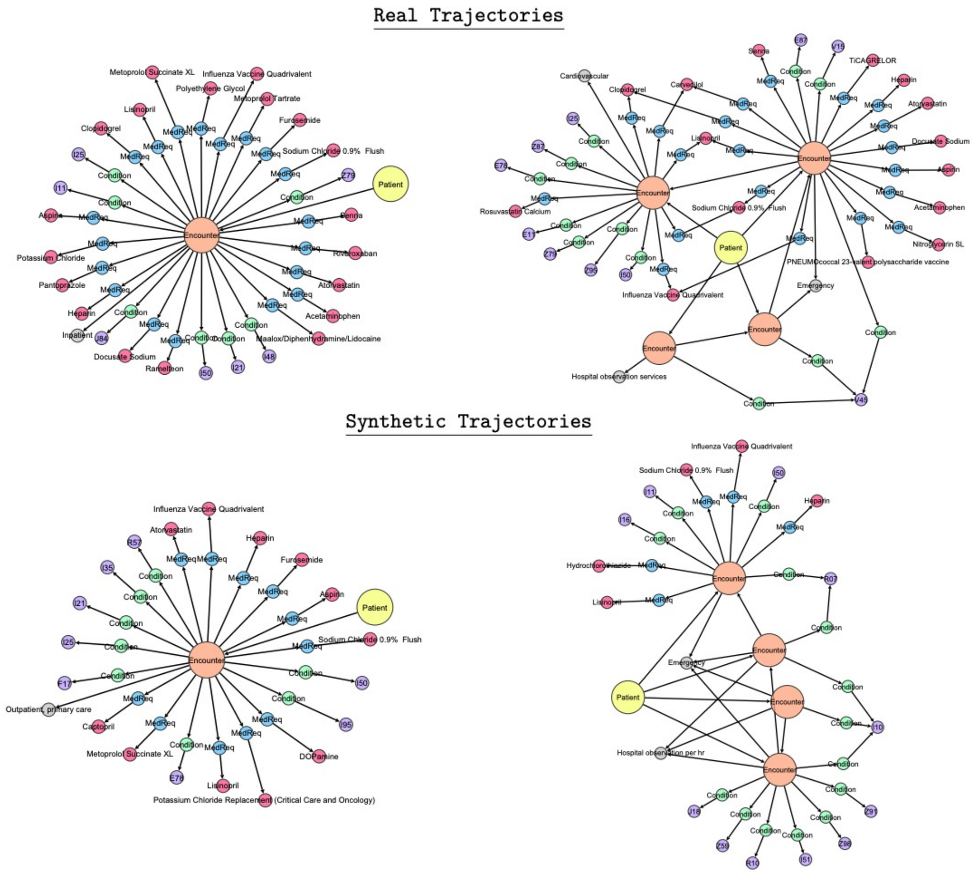
| Scenario | Task 1 | Task 2 |
|---|---|---|
| Train on real, test on real | 64.02% ± 2.93 | 73.04% ± 1.35 |
| Train on synthetic, test on real | 63.85% ± 2.57 | 73.32% ± 2.90 |

Both tasks are variants of predicting the onset of heart failure in patients. In the first scenario, the classifier is trained on real data and is evaluated on real data, while in the second scenario, the classifier is trained on synthetic data and is evaluated on real data.

- Real data are barely distinguishable by the synthetic ones.

# GVAEs for medical graph generation - Conclusions

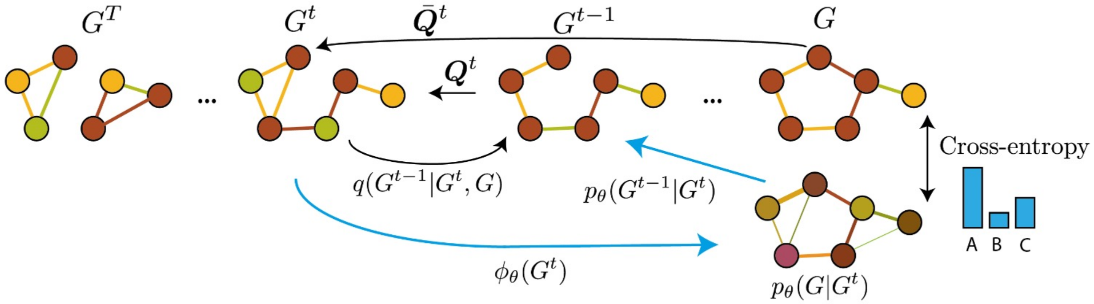
- the model generates novel synthetic patient trajectories,
- sufficiently different to preserve patient privacy,
- yet retains the characteristics of the real-world data. Thus data exchange and ML is feasible
- most significant feature: capacity to learn long-range correlations between trajectory nodes.


Real Trajectories
Synthetic Trajectories
Patient
Encounter
Condition
MedReq
Influenza Vaccine Quadrivalent
Metoprolol Succinate XL
Sodium Chloride 0.9% Flush
Hospital observation services
Emergency
Outpatient primary care
Hospital observation per hr
Potassium Chloride Replacement (Critical Care and Oncology)
PNEUMOcoccal 23-valent polysaccharide vaccine
Maalox/Diphenhydramine/Lidocaine

# GNNs and Graph Generative models for biomedical applications

- Graph Generative models
- Generative models for Medical Graphs
- **Large generative models**
- Graph & LLMs
- Multi modality for molecule generation
- Conclusions

# Digress [1]

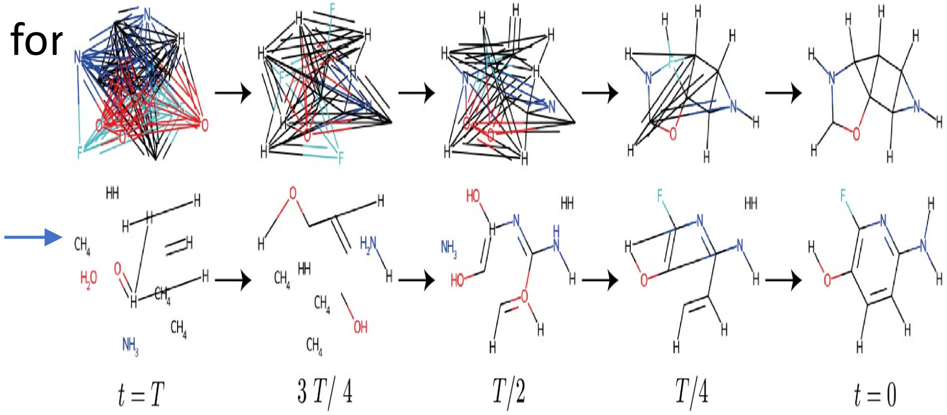


- Diffusion process:
  - Inspired by statistical physics – uncertainty of particles position
  - *Noising model*: $q$ progressively corrupts a data point $x$ to create a sequence of increasingly noisy data points $(z_1, \ldots, z_T)$
    - Markovian structure $q(z^1, \ldots, z^T |x) = q(z^1|x)\prod_{t=2}^{T} q(z^t|z^{t-1})$
  - *Denoising* $\phi_\theta$ is trained to invert the noising process by predicting $z^{t-1}$ from $z^t$.
  - *Inference*:
    - generate new samples: noise is sampled from a prior distribution
    - inverted by iterative application of the denoising network. ,

[1]DIGRESS: DISCRETE DENOISING DIFFUSION FOR GRAPH GENERATION Clement Vignac et al, ICLR23, https://arxiv.org/pdf/2209.14734

# Digress - Discrete diffusion

- Continuous diffusion: Gaussian is poor noise model for graphs: destroys sparsity as well as graph theoretic notions such as connectivity.

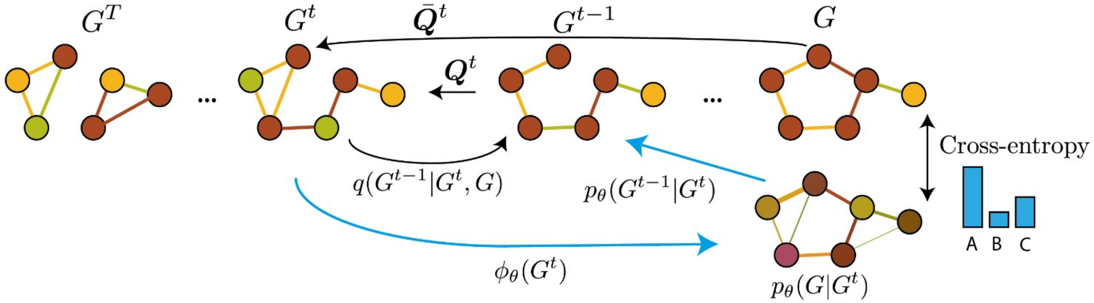


- Discrete diffusion more appropriate to graph generation tasks.
  - Recent works have considered the discrete diffusion problem for text, image and audio data [Hoogeboom et al., 2021; Johnson et al., 2021; Yang et al., 2022]
- data $x \in R^d$ (one-hot encoding, $d$: classes), noise represented by transition matrices $(Q^1, ..., Q^T)$ such that
  - $[Q^t]_{ij}$ : probability state $i$ => $j$: $q(z^t|z^{t-1}) = z^{t-1}Q^t$
  - as process is Markovian: $\bar{Q}^t = Q^1 ... Q^t$ the noisy state $z^t$ can be built from $x$: $q(z^t|x) = x\bar{Q}^t$
  - posterior distribution $q(z^{t-1}|z^t, x)$ - closed-form Bayes rule $q(z^{t-1}|z^t, x) \propto \boldsymbol{z}^t\ (\boldsymbol{Q}^t)' \odot \boldsymbol{x}\ \bar{\boldsymbol{Q}}^{t-1}$

______________________-

[1]DIGRESS: DISCRETE DENOISING DIFFUSION FOR GRAPH GENERATION Clement Vignac et al, ICLR23, https://arxiv.org/pdf/2209.14734

# Overview of Digress



- *Discrete noising process* for node/edge ($X^t/E^t$) labels, Q noising matrix.

$$q(G^t|G^{t-1}) = (\boldsymbol{X}^{t-1}\boldsymbol{Q}^t_X, \mathbf{E}^{t-1}\boldsymbol{Q}^t_E) \quad \text{and} \quad q(G^t|G) = (\boldsymbol{X}\bar{\boldsymbol{Q}}^t_X, \mathbf{E}\bar{\boldsymbol{Q}}^t_E)$$

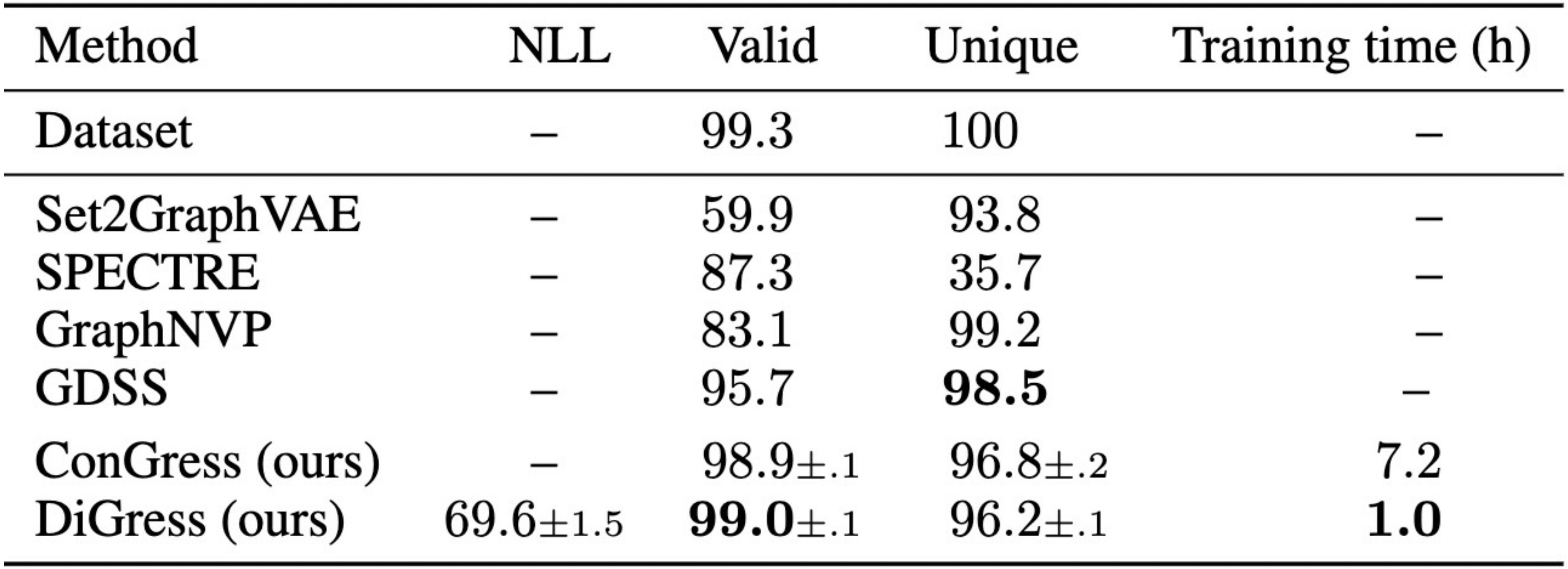
- *Discrete noising process* for node/edge ($X^t/E^t$) labels, $Q$ noising matrix.
- DiGress *denoising* neural network $\phi^\theta$ parametrized by $\theta$.
  - Input: noisy graph $G^t = (X^t, E^t)$: aims to predict the "clean" graph G,
  - To train $\phi^\theta$ , optimize cross-entropy loss: between the predicted probabilities $\hat{p}^G = (\hat{p}^X, \hat{p}^E)$ for each node and edge and the true graph G:

$$l(\hat{p}^G, G) = \sum_{1\leq i\leq n} \text{cross-entropy}(x_i, \hat{p}^X_i) + \lambda \sum_{1\leq i,j\leq n} \text{cross-entropy}(e_{ij}, \hat{p}^E_{ij})$$

______________________________-

[1]DIGRESS: DISCRETE DENOISING DIFFUSION FOR GRAPH GENERATION Clement Vignac et al, ICLR23, https://arxiv.org/pdf/2209.14734

# Digress - Experiments

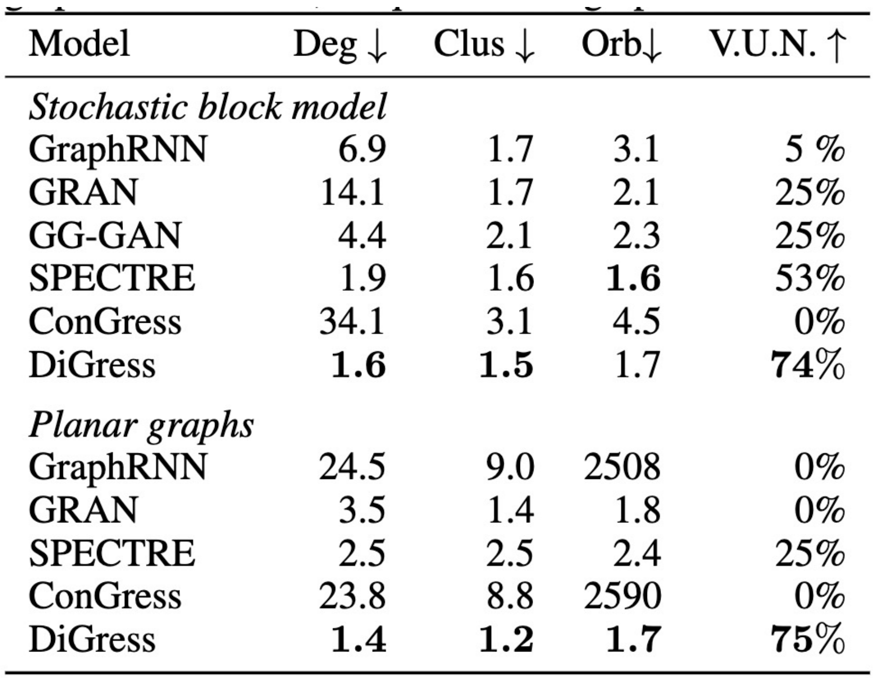


| Method | NLL | Valid | Unique | Training time (h) |
|---|---|---|---|---|
| Dataset | – | 99.3 | 100 | – |
| Set2GraphVAE | – | 59.9 | 93.8 | – |
| SPECTRE | – | 87.3 | 35.7 | – |
| GraphNVP | – | 83.1 | 99.2 | – |
| GDSS | – | 95.7 | **98.5** | – |
| ConGress (ours) | – | 98.9±.1 | 96.8±.2 | 7.2 |
| DiGress (ours) | 69.6±1.5 | **99.0**±.1 | 96.2±.1 | **1.0** |

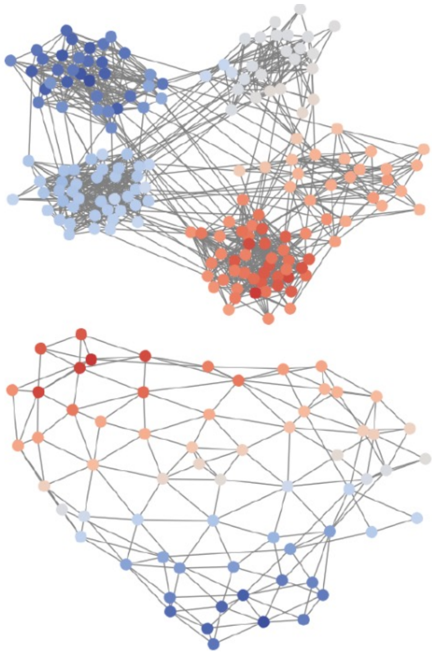
- Molecule generation - QM9.
- Training time: time to reach 99% validity.
- DiGress achieves similar results to the continuous model but faster to train.

# Digress - Experiments

Unconditional generation on SBM and planar graphs. VUN: valid, unique & novel graphs.



| Model | Deg ↓ | Clus ↓ | Orb↓ | V.U.N. ↑ |
|---|---|---|---|---|
| *Stochastic block model* | | | | |
| GraphRNN | 6.9 | 1.7 | 3.1 | 5 % |
| GRAN | 14.1 | 1.7 | 2.1 | 25% |
| GG-GAN | 4.4 | 2.1 | 2.3 | 25% |
| SPECTRE | 1.9 | 1.6 | **1.6** | 53% |
| ConGress | 34.1 | 3.1 | 4.5 | 0% |
| DiGress | **1.6** | **1.5** | 1.7 | **74%** |
| *Planar graphs* | | | | |
| GraphRNN | 24.5 | 9.0 | 2508 | 0% |
| GRAN | 3.5 | 1.4 | 1.8 | 0% |
| SPECTRE | 2.5 | 2.5 | 2.4 | 25% |
| ConGress | 23.8 | 8.8 | 2590 | 0% |
| DiGress | **1.4** | **1.2** | **1.7** | **75%** |



- Martinkus et al. (2022): i. 200 graphs drawn from stochastic block model ii. 200 planar graphs.
- We evaluate ability
  - correctly model various properties of these graphs,
  - generated graphs are statistically distinguishable from the SBM model or if they are planar and connected.

# Digress - Experiments

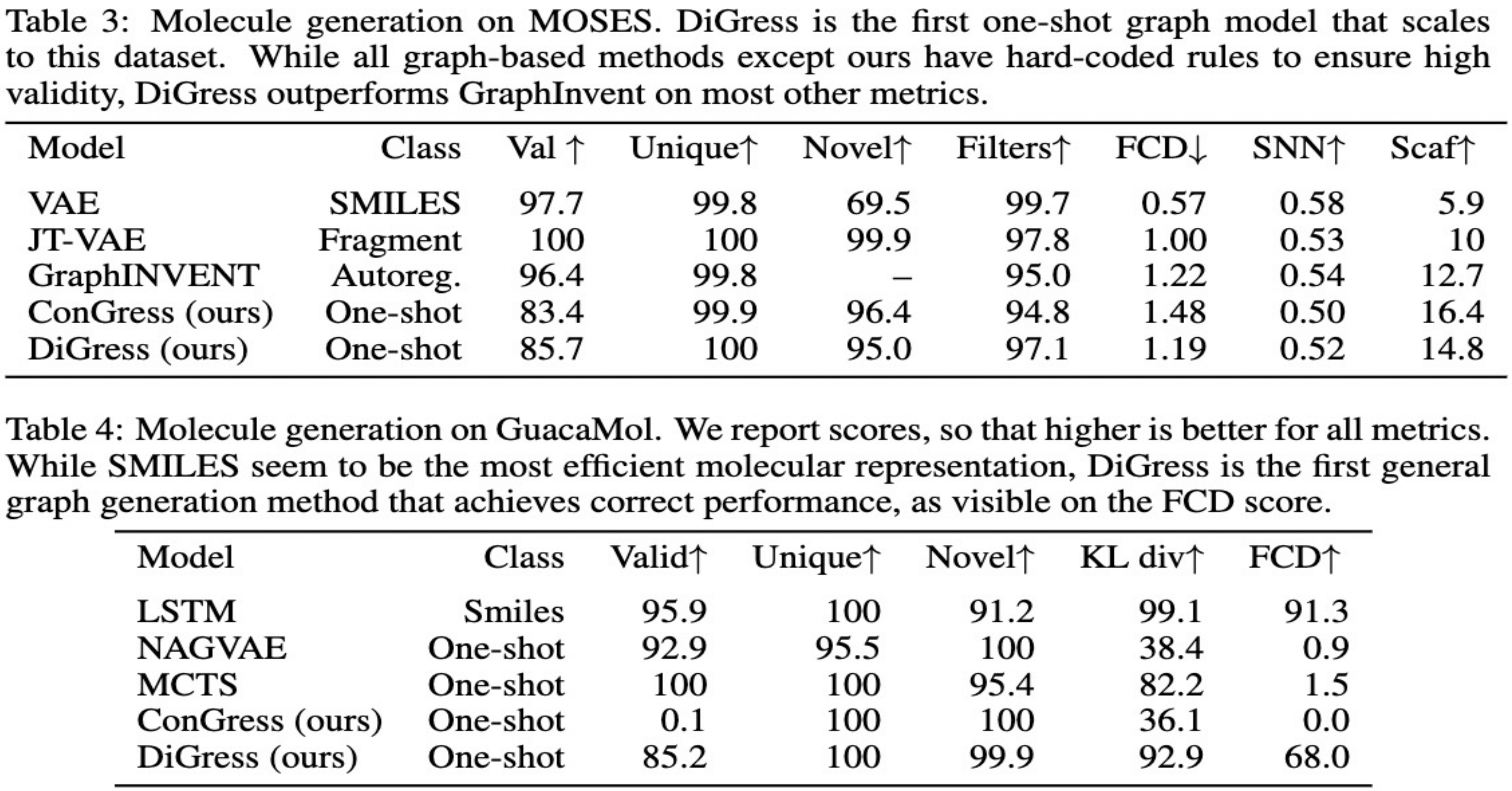


Table 3: Molecule generation on MOSES. DiGress is the first one-shot graph model that scales to this dataset. While all graph-based methods except ours have hard-coded rules to ensure high validity, DiGress outperforms GraphInvent on most other metrics.

| Model | Class | Val ↑ | Unique↑ | Novel↑ | Filters↑ | FCD↓ | SNN↑ | Scaf↑ |
|---|---|---|---|---|---|---|---|---|
| VAE | SMILES | 97.7 | 99.8 | 69.5 | 99.7 | 0.57 | 0.58 | 5.9 |
| JT-VAE | Fragment | 100 | 100 | 99.9 | 97.8 | 1.00 | 0.53 | 10 |
| GraphINVENT | Autoreg. | 96.4 | 99.8 | – | 95.0 | 1.22 | 0.54 | 12.7 |
| ConGress (ours) | One-shot | 83.4 | 99.9 | 96.4 | 94.8 | 1.48 | 0.50 | 16.4 |
| DiGress (ours) | One-shot | 85.7 | 100 | 95.0 | 97.1 | 1.19 | 0.52 | 14.8 |

Table 4: Molecule generation on GuacaMol. We report scores, so that higher is better for all metrics. While SMILES seem to be the most efficient molecular representation, DiGress is the first general graph generation method that achieves correct performance, as visible on the FCD score.

| Model | Class | Valid↑ | Unique↑ | Novel↑ | KL div↑ | FCD↑ |
|---|---|---|---|---|---|---|
| LSTM | Smiles | 95.9 | 100 | 91.2 | 99.1 | 91.3 |
| NAGVAE | One-shot | 92.9 | 95.5 | 100 | 38.4 | 0.9 |
| MCTS | One-shot | 100 | 100 | 95.4 | 82.2 | 1.5 |
| ConGress (ours) | One-shot | 0.1 | 100 | 100 | 36.1 | 0.0 |
| DiGress (ours) | One-shot | 85.2 | 100 | 99.9 | 92.9 | 68.0 |

# Neural Graph Generator (NGG)

- a novel graph generative model which leverages latent diffusion for conditional graph generation.
- represents a significant shift from traditional graph generation methods, focusing on *prompting with a vector that includes a set of diverse properties of the graph.*
- introduce a large-scale dataset of synthetic graphs that covers several different types of graphs on which our model was trained. This dataset can be used for pre-training any graph generative model in the future.
- extensively evaluate our model across various graph generation tasks, demonstrating its effectiveness in capturing specific graph properties, generalizing to larger graphs, and generating graphs from subsets of properties.
- release the pre-trained autoencoder, the pre-trained latent diffusion model, and the synthetic dataset of 1M graphs to be useful for both practitioners and the scientific community.

1. Neural Graph Generator: Feature-Conditioned Graph Generation using Latent Diffusion Models, Evdaimon et. al, https://arxiv.org/pdf/2403.01535.pdf

# Neural Graph Generator (NGG)

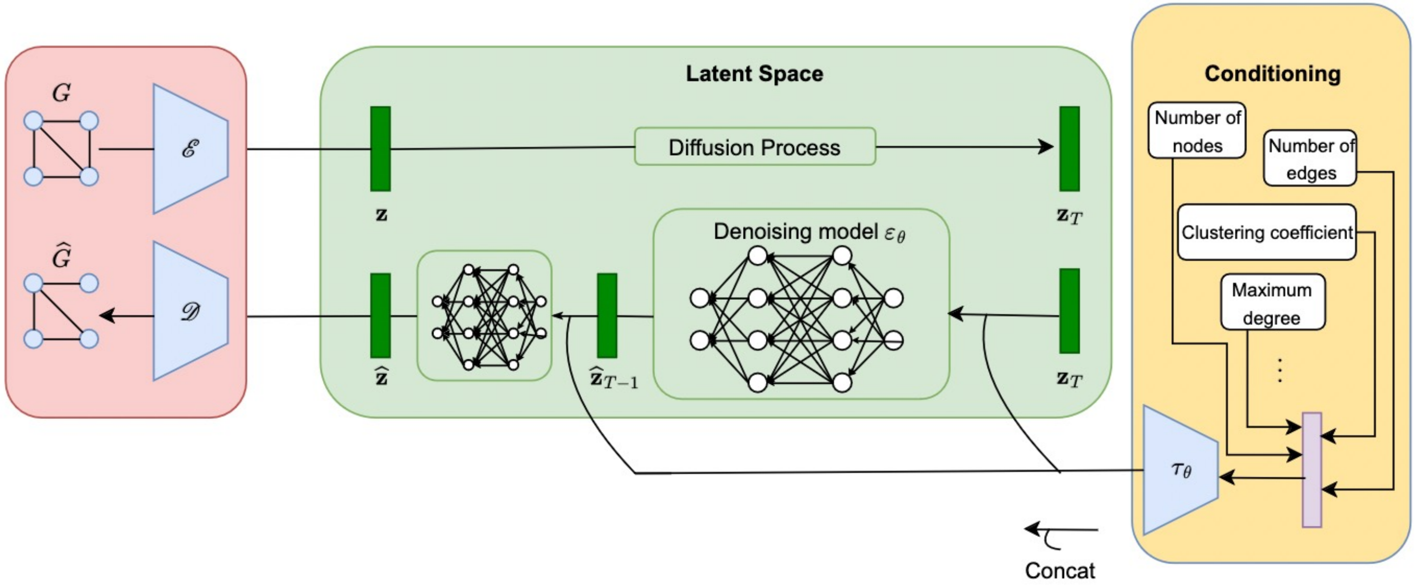
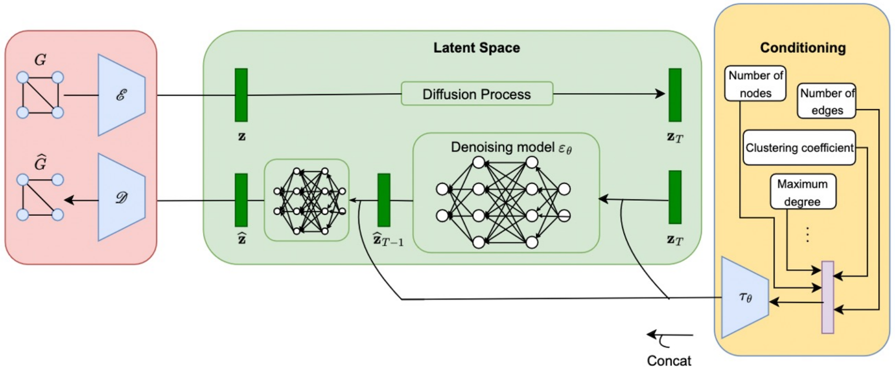


Figure 1: Overview of the proposed architecture. The variational graph autoencoder is responsible for generating a compressed latent representation for each graph. Those representations are fed to the diffusion model which operates in that latent space. The denoising process is conditioned on a vector that contains the graph's properties. The output of the diffusion model is passed on to the decoder which generates a graph.

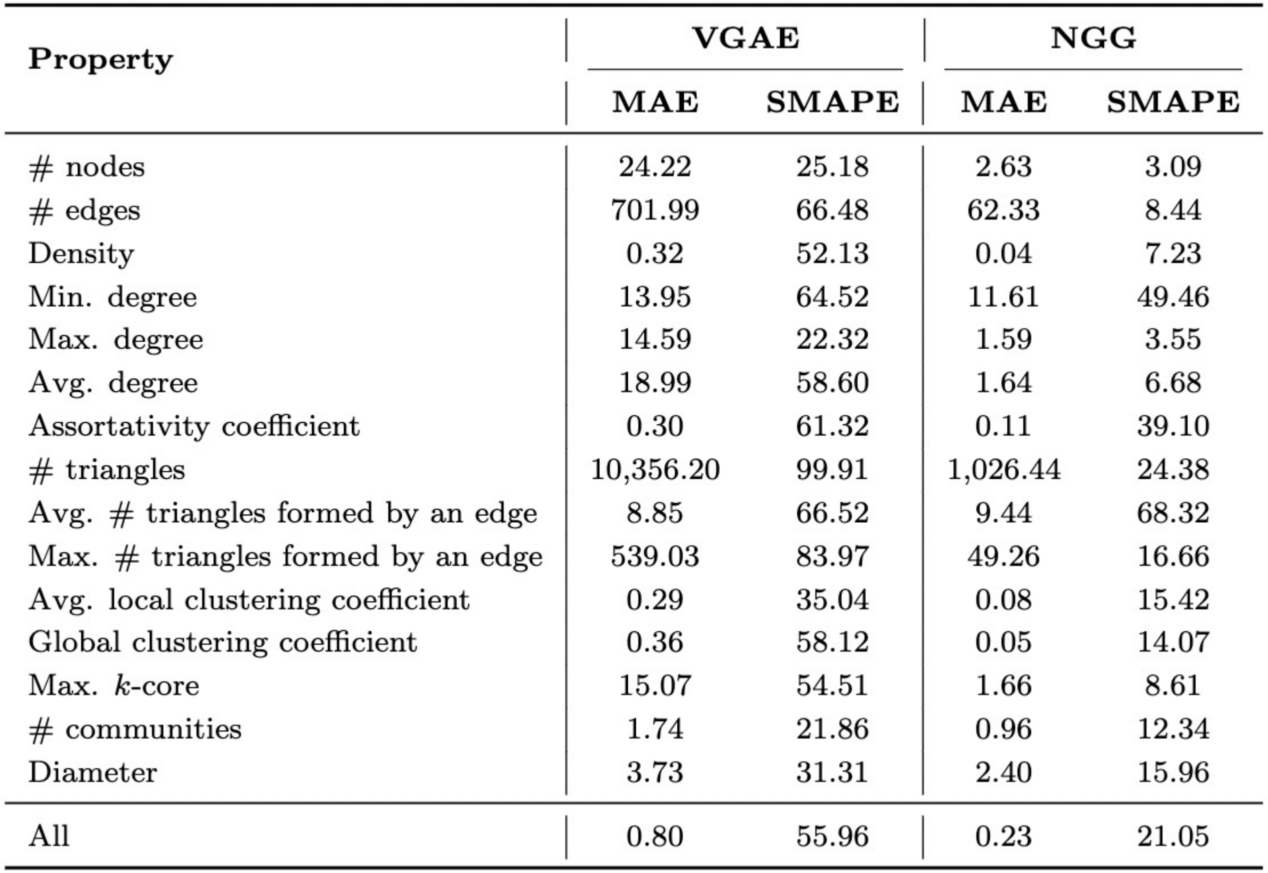
# Neural Graph Generator (NGG) - conditioning

- Dataset: 1M synthetic graphs (<=100 nodes).
- use different types of graph generators.
- 17 families of graphs: (1) paths (2) cycles (3) wheels (4) stars (5) ladders; (6)lollipops (7) Erdos-Renyi random graphs; (8) Newman–Watts–Strogatz small-world graphs; (9) Watts–Strogatz small-world graphs, (10) random d-degree regular graphs (11) Barabasi–Albert graphs; (12) dual Barabasi–Albert graphs (13) extended Barabasi–Albert graphs (14) graphs generated using the Holme and Kim algorithm (15) random lobsters (16) stochastic block model graphs and (17) random partition graphs.
- The generated graphs are devoid of self-loops, isolated nodes, and multigraphs are also excluded.
- NGG model: *3.6M parameters*



# Neural Graph Generator (NGG) – exp results

Within distribution performance

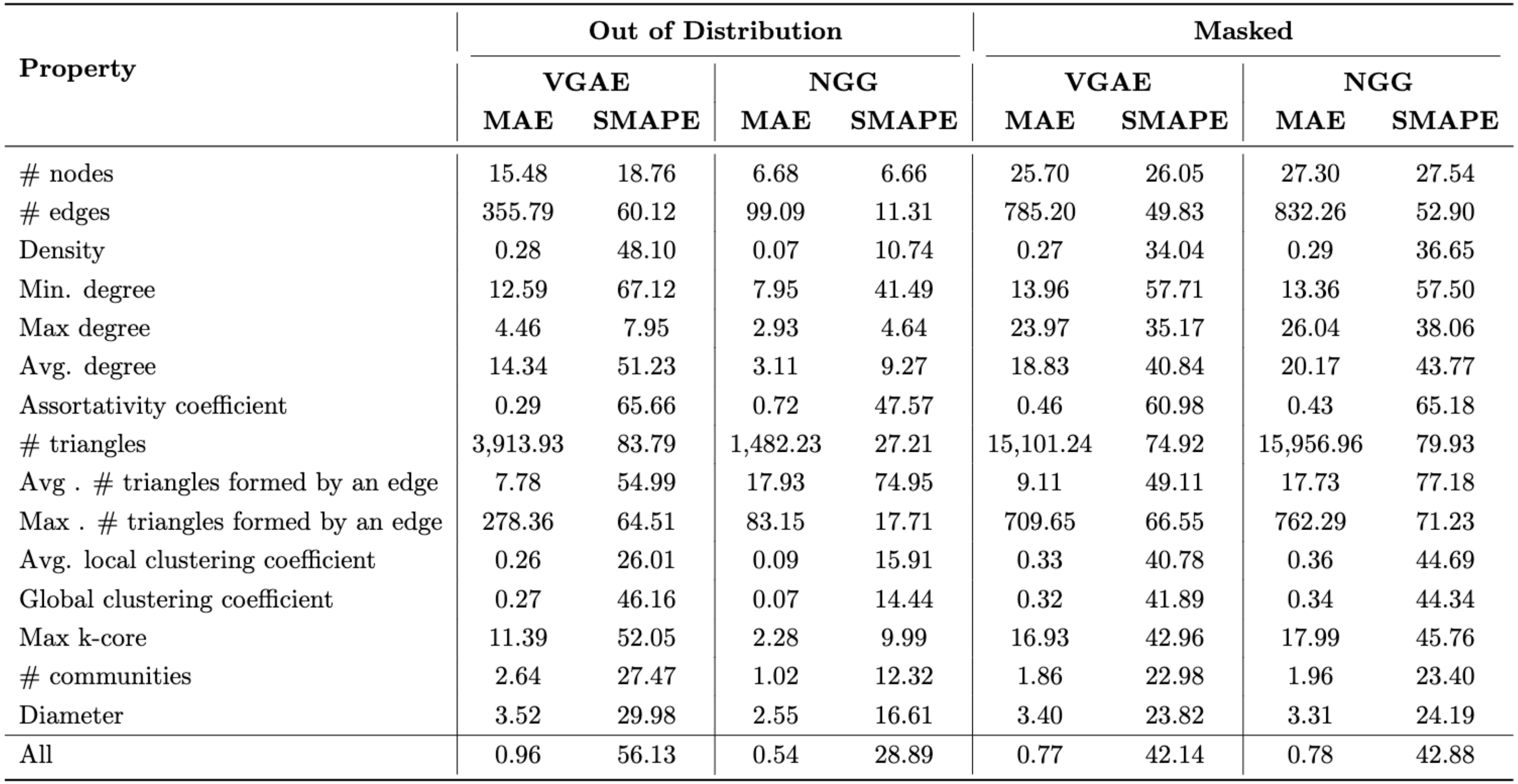


| Property | VGAE | | NGG | |
|---|---|---|---|---|
| | MAE | SMAPE | MAE | SMAPE |
| # nodes | 24.22 | 25.18 | 2.63 | 3.09 |
| # edges | 701.99 | 66.48 | 62.33 | 8.44 |
| Density | 0.32 | 52.13 | 0.04 | 7.23 |
| Min. degree | 13.95 | 64.52 | 11.61 | 49.46 |
| Max. degree | 14.59 | 22.32 | 1.59 | 3.55 |
| Avg. degree | 18.99 | 58.60 | 1.64 | 6.68 |
| Assortativity coefficient | 0.30 | 61.32 | 0.11 | 39.10 |
| # triangles | 10,356.20 | 99.91 | 1,026.44 | 24.38 |
| Avg. # triangles formed by an edge | 8.85 | 66.52 | 9.44 | 68.32 |
| Max. # triangles formed by an edge | 539.03 | 83.97 | 49.26 | 16.66 |
| Avg. local clustering coefficient | 0.29 | 35.04 | 0.08 | 15.42 |
| Global clustering coefficient | 0.36 | 58.12 | 0.05 | 14.07 |
| Max. *k*-core | 15.07 | 54.51 | 1.66 | 8.61 |
| # communities | 1.74 | 21.86 | 0.96 | 12.34 |
| Diameter | 3.73 | 31.31 | 2.40 | 15.96 |
| All | 0.80 | 55.96 | 0.23 | 21.05 |

# Neural Graph Generator (NGG) – exp results

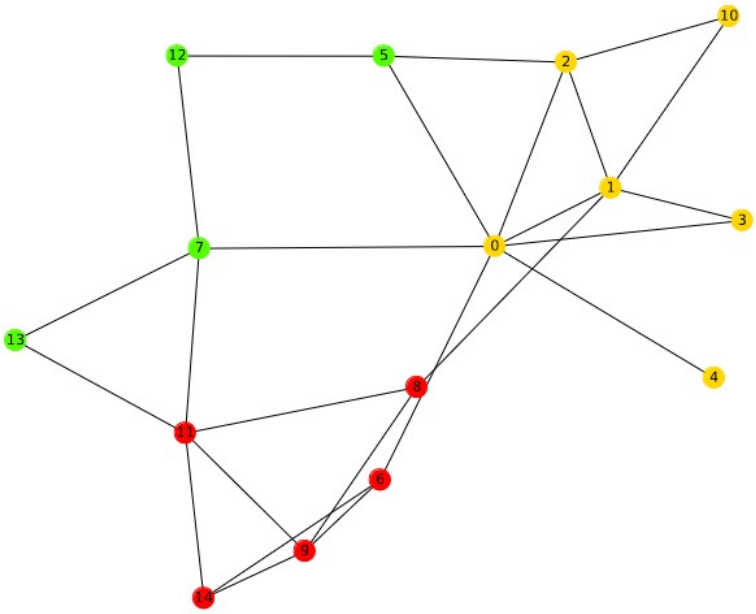
Performance Comparison of NGG and Baseline Model :

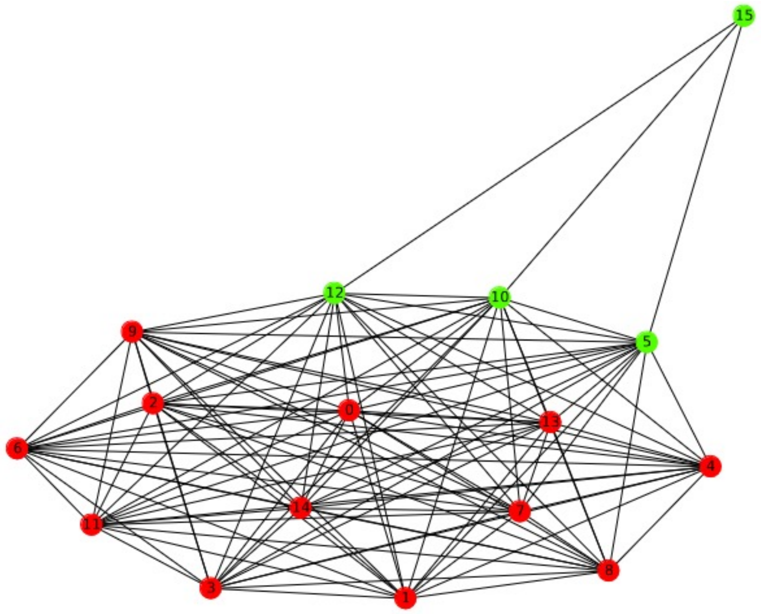
- *Out-of-Distribution Performance* (trained on graphs with up to 50 nodes and evaluated on larger graphs)
- *Within-Distribution Performance with Masking* Applied to some Condition Vector Elements.



| Property | Out of Distribution | | | | Masked | | | |
|---|---|---|---|---|---|---|---|---|
| | VGAE | | NGG | | VGAE | | NGG | |
| | MAE | SMAPE | MAE | SMAPE | MAE | SMAPE | MAE | SMAPE |
| # nodes | 15.48 | 18.76 | 6.68 | 6.66 | 25.70 | 26.05 | 27.30 | 27.54 |
| # edges | 355.79 | 60.12 | 99.09 | 11.31 | 785.20 | 49.83 | 832.26 | 52.90 |
| Density | 0.28 | 48.10 | 0.07 | 10.74 | 0.27 | 34.04 | 0.29 | 36.65 |
| Min. degree | 12.59 | 67.12 | 7.95 | 41.49 | 13.96 | 57.71 | 13.36 | 57.50 |
| Max degree | 4.46 | 7.95 | 2.93 | 4.64 | 23.97 | 35.17 | 26.04 | 38.06 |
| Avg. degree | 14.34 | 51.23 | 3.11 | 9.27 | 18.83 | 40.84 | 20.17 | 43.77 |
| Assortativity coefficient | 0.29 | 65.66 | 0.72 | 47.57 | 0.46 | 60.98 | 0.43 | 65.18 |
| # triangles | 3,913.93 | 83.79 | 1,482.23 | 27.21 | 15,101.24 | 74.92 | 15,956.96 | 79.93 |
| Avg . # triangles formed by an edge | 7.78 | 54.99 | 17.93 | 74.95 | 9.11 | 49.11 | 17.73 | 77.18 |
| Max . # triangles formed by an edge | 278.36 | 64.51 | 83.15 | 17.71 | 709.65 | 66.55 | 762.29 | 71.23 |
| Avg. local clustering coefficient | 0.26 | 26.01 | 0.09 | 15.91 | 0.33 | 40.78 | 0.36 | 44.69 |
| Global clustering coefficient | 0.27 | 46.16 | 0.07 | 14.44 | 0.32 | 41.89 | 0.34 | 44.34 |
| Max k-core | 11.39 | 52.05 | 2.28 | 9.99 | 16.93 | 42.96 | 17.99 | 45.76 |
| # communities | 2.64 | 27.47 | 1.02 | 12.32 | 1.86 | 22.98 | 1.96 | 23.40 |
| Diameter | 3.52 | 29.98 | 2.55 | 16.61 | 3.40 | 23.82 | 3.31 | 24.19 |
| All | 0.96 | 56.13 | 0.54 | 28.89 | 0.77 | 42.14 | 0.78 | 42.88 |

# Neural Graph Generator (NGG) – exp results





$$\mathbf{c}_1 = \begin{pmatrix} 15 & 34 & 0.32 & 2 & 8 & 4.5 & -0.046 & 17 & 1.5 & 8 & 0.4 & 0.35 & 4 & 4 & 4 \end{pmatrix}^\top$$

$$\hat{\mathbf{c}}_1 = \begin{pmatrix} 15 & 25 & 0.23 & 1 & 7 & 3.3 & -0.380 & 8 & 0.96 & 3 & 0.43 & 0.32 & 2 & 3 & 4 \end{pmatrix}^\top$$

$$\mathbf{c}_2 = \begin{pmatrix} 15 & 94 & 0.89 & 10 & 14 & 12.5 & -0.149 & 329 & 10.5 & 80 & 0.9 & 0.9 & 10 & 2 & 2 \end{pmatrix}^\top$$

$$\hat{\mathbf{c}}_2 = \begin{pmatrix} 16 & 108 & 0.9 & 3 & 15 & 13.5 & -0.148 & 458 & 12.7 & 93 & 0.97 & 0.97 & 14 & 2 & 2 \end{pmatrix}^\top$$

# GNNs and Graph Generative models for biomedical applications

- Graph Generative models
- Generative models for Medical Graphs
- Large generative models
- **Graph / LLMs**
- Multi modality for molecule generation
- Conclusions

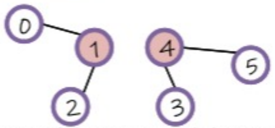
# Graphs & LLMs [1]

- Despite not being explicitly designed for graph-structured data, LLMs are increasingly leveraged for graph machine learning tasks.
- Are LLMs capable of mapping textual descriptions of graphs and structures to grounded conceptual spaces and solving graph algorithm problems explicitly with natural language?
- NLGraph[1] constructs 29,370 problems, 8 graph reasoning tasks - varying complexity
    - simple tasks: *connectivity, shortest path*
    - complex problems: *maximum flow, simulating graph neural networks*.
- Conclusion:
    - LLMs do possess preliminary graph reasoning abilities.
    - The benefit of advanced prompting methods diminishes with complex problems.
    - Few shot learning does not help on complex graph reasoning problems.

1) Wang, Heng, et al. "Can language models solve graph problems in natural language?." Advances in Neural Information Processing Systems 36 (2024).

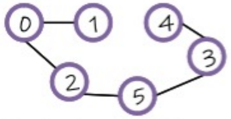
# NLGraph Benchmark - eight tasks, varying complexity

**1. Connectivity**



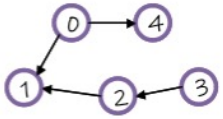
Determine if there is a path between two nodes in the graph. Note that (i,j) means that node i and node j are connected with an undirected edge.
Graph: (0,1) (1,2) (3,4) (4,5)
**Q**: Is there a path between node 1 and node 4?

**2. Cycle**



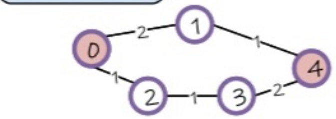
In an undirected graph, (i,j) means that node i and node j are connected with an undirected edge.
The nodes are numbered from 0 to 5, and the edges are: (3,4) (3,5) (1,0) (2,5) (2,0)
**Q**: Is there a cycle in this graph?

**3. Topological Sort**



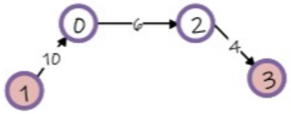
In a directed graph with 5 nodes numbered from 0 to 4:
node 0 should be visited before node 4, ...
**Q**: Can all the nodes be visited? Give the solution.

**4. Shortest Path**



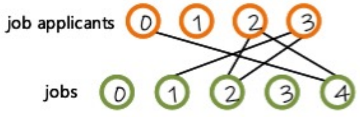
In an undirected graph, the nodes are numbered from 0 to 4, and the edges are:
an edge between node 0 and node 1 with weight 2, ...
**Q**: Give the shortest path from node 0 to node 4.

**5. Maximum Flow**



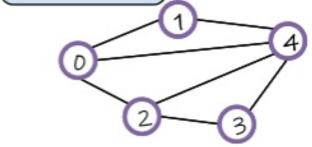
In a directed graph, the nodes are numbered from 0 to 3, and the edges are:
an edge from node 1 to node 0 with capacity 10,
an edge from node 0 to node 2 with capacity 6,
an edge from node 2 to node 3 with capacity 4.
**Q**: What is the maximum flow from node 1 to node 3?

**6. Bipartite Graph Matching**



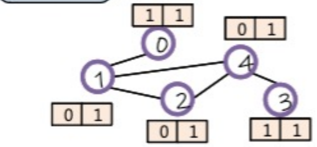
There are 4 job applicants numbered from 0 to 3, and 5 jobs numbered from 0 to 4. Each applicant is interested in some of the jobs. Each job can only accept one applicant and a job applicant can be appointed for only one job.
Applicant 0 is interested in job 4, ...
**Q**: Find an assignment of jobs to applicants in such that the maximum number of applicants find the job they are interested in.

**7. Hamilton Path**



In an undirected graph, (i,j) means that node i and node j are connected with an undirected edge.
The nodes are numbered from 0 to 4, and the edges are: (4,2) (0,4) (4,3) (0,1) (0,2) (4,1) (2,3)
**Q**: Is there a path in this graph that visits every node exactly once? If yes, give the path. Note that in a path, adjacent nodes must be connected with edges.

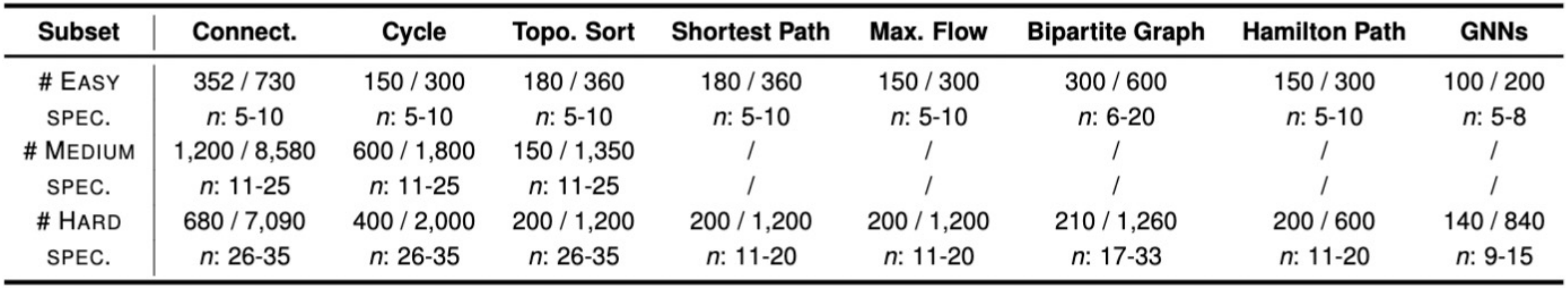
**8. GNN**



In an undirected graph, the nodes are numbered from 0 to 4, and every node has an embedding. (i,j) means that node i and node j are connected with an undirected edge.
Embeddings: node 0: [1,1], ···
The edges are: (0,1) ...
In a simple graph convolution layer, each node's embedding is updated by the sum of its neighbors' embeddings.
**Q**: What's the embedding of each node after one layer of simple graph convolution layer?

# NLGraph Benchmark

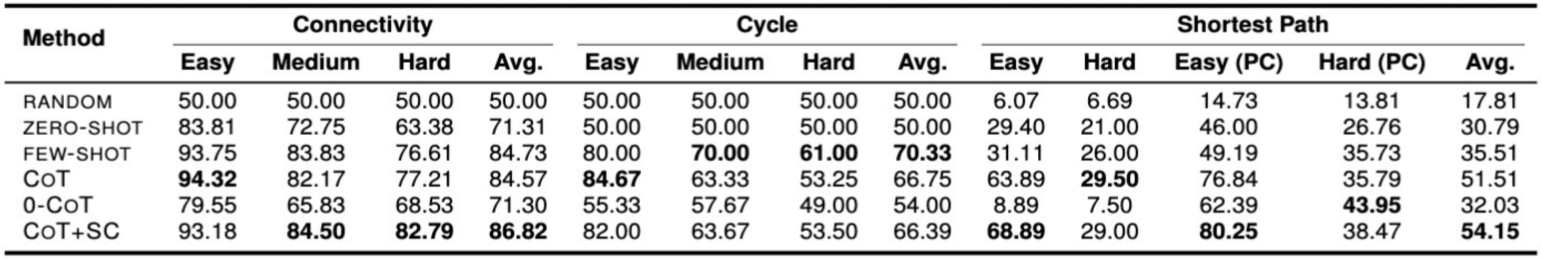
- random graph generator to generate graphs and structures while controlling for the network size, graph sparsity, and more.
- adopting generated graphs as bases to synthetically generate problems for eight graph-based reasoning tasks with varying algorithmic difficulties



| Subset | Connect. | Cycle | Topo. Sort | Shortest Path | Max. Flow | Bipartite Graph | Hamilton Path | GNNs |
|---|---|---|---|---|---|---|---|---|
| # EASY | 352 / 730 | 150 / 300 | 180 / 360 | 180 / 360 | 150 / 300 | 300 / 600 | 150 / 300 | 100 / 200 |
| SPEC. | *n*: 5-10 | *n*: 5-10 | *n*: 5-10 | *n*: 5-10 | *n*: 5-10 | *n*: 6-20 | *n*: 5-10 | *n*: 5-8 |
| # MEDIUM | 1,200 / 8,580 | 600 / 1,800 | 150 / 1,350 | / | / | / | / | / |
| SPEC. | *n*: 11-25 | *n*: 11-25 | *n*: 11-25 | / | / | / | / | / |
| # HARD | 680 / 7,090 | 400 / 2,000 | 200 / 1,200 | 200 / 1,200 | 200 / 1,200 | 210 / 1,260 | 200 / 600 | 140 / 840 |
| SPEC. | *n*: 26-35 | *n*: 26-35 | *n*: 26-35 | *n*: 11-20 | *n*: 11-20 | *n*: 17-33 | *n*: 11-20 | *n*: 9-15 |

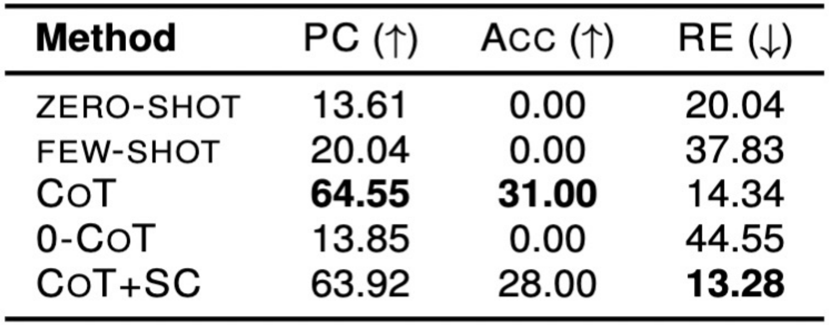
- Statistics of the NLGraph benchmark. A / B indicates that there are A and B problems in the standard and extended set of NLGraph. SPEC. denotes difficulty specifications.

# Results



| Method | Connectivity | | | | Cycle | | | | Shortest Path | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Easy | Medium | Hard | Avg. | Easy | Medium | Hard | Avg. | Easy | Hard | Easy (PC) | Hard (PC) | Avg. |
| RANDOM | 50.00 | 50.00 | 50.00 | 50.00 | 50.00 | 50.00 | 50.00 | 50.00 | 6.07 | 6.69 | 14.73 | 13.81 | 17.81 |
| ZERO-SHOT | 83.81 | 72.75 | 63.38 | 71.31 | 50.00 | 50.00 | 50.00 | 50.00 | 29.40 | 21.00 | 46.00 | 26.76 | 30.79 |
| FEW-SHOT | 93.75 | 83.83 | 76.61 | 84.73 | 80.00 | **70.00** | **61.00** | **70.33** | 31.11 | 26.00 | 49.19 | 35.73 | 35.51 |
| CoT | **94.32** | 82.17 | 77.21 | 84.57 | **84.67** | 63.33 | 53.25 | 66.75 | 63.89 | **29.50** | 76.84 | 35.79 | 51.51 |
| 0-CoT | 79.55 | 65.83 | 68.53 | 71.30 | 55.33 | 57.67 | 49.00 | 54.00 | 8.89 | 7.50 | 62.39 | **43.95** | 32.03 |
| CoT+SC | 93.18 | **84.50** | **82.79** | **86.82** | 82.00 | 63.67 | 53.50 | 66.39 | **68.89** | 29.00 | **80.25** | 38.47 | **54.15** |

Table: Model performance on the connectivity, cycle, and shortest path tasks. PC denotes partial credit. Large language models with CoT or CoT+SC prompting greatly outperforms the random baseline by 37.33% to 57.82%, indicating that LLMs have preliminary graph reasoning abilities.



| Method | PC (↑) | Acc (↑) | RE (↓) |
|---|---|---|---|
| ZERO-SHOT | 13.61 | 0.00 | 20.04 |
| FEW-SHOT | 20.04 | 0.00 | 37.83 |
| CoT | **64.55** | **31.00** | 14.34 |
| 0-CoT | 13.85 | 0.00 | 44.55 |
| CoT+SC | 63.92 | 28.00 | **13.28** |

Table: Model performance on the task of simulating graph neural networks. PC and RE are two partial credit metrics. Chain-of-thought prompting significantly improves the model performance across all metrics.

* CoT+SC: Construct multiple chains of thought, evaluate each one, and ultimately select the most effective and coherent chain.

# Results

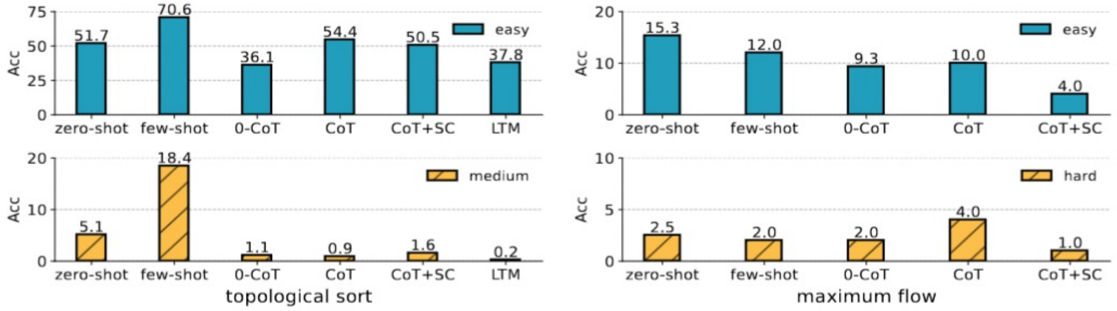


Figure: (left) Model performance on the topological sort task. CoT, LTM, and self-consistency are mostly ineffective on this problem. (right) Model performance on the maximum flow task. FEW-SHOT prompting outperforms CoT+SC prompting on both easy and hard subsets, suggesting that LLMs fall short of generating valid intermediate steps to solve the more complex graph reasoning problem. Together these results demonstrate that advanced prompting is ineffective for advanced graph reasoning.

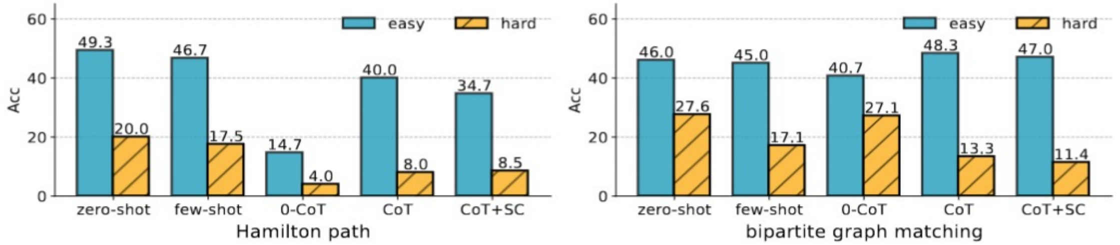


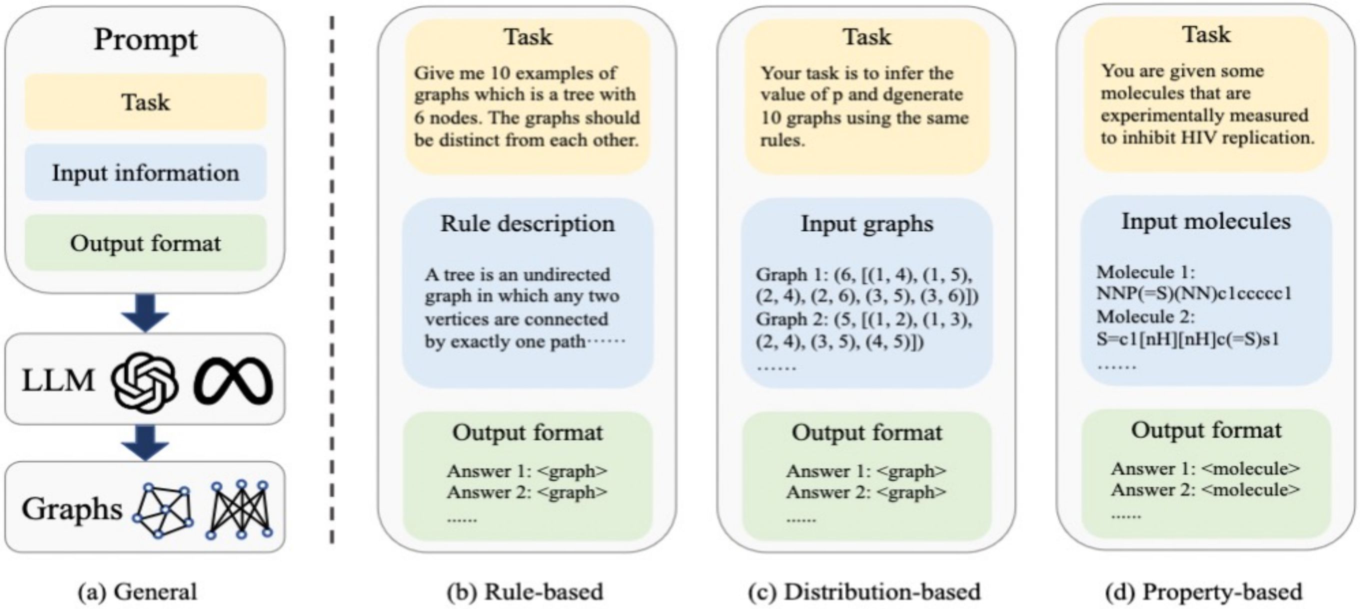
Figure: (left) Model performance on the Hamilton path task. ZERO-SHOT prompting consistently outperforms all other prompting techniques. (right) Model performance on the bipartite graph matching task. The effect of in-context learning and advanced prompting is also mostly marginal in this complex graph reasoning problem. Together these results demonstrate that in-context learning can be counterproductive in advanced graph reasoning problems.

# LLMs for graph generation

- Graph generation requires the LLM to generate graphs with given properties
- valuable real-world applications such as drug discovery ⇒ more challenging than graph reasoning.
- first work[1] towards this direction investigated the questions regarding:
    - LLMs’ understanding of different graph structure rules.
    - Their ability to capture structural type distributions.
    - Their utilization of domain knowledge for property-based graph generation.
- Conclusion:
    - LLMs exhibit *preliminary* abilities in graph generation tasks.
    - LLMs show potential in generating molecules with specific properties.
    - Popular prompting methods do not consistently enhance performance.

[1] Yao, Yang, et al. "Exploring the Potential of Large Language Models in Graph Generation." arXiv preprint arXiv:2403.14358 (2024).

# Overview



Task

Give me 10 examples of graphs which is a tree with 6 nodes. The graphs should be distinct from each other.

Rule description

A tree is an undirected graph in which any two vertices are connected by exactly one path······

Output format

Answer 1: <graph>
Answer 2: <graph>
......

Task

Your task is to infer the value of p and dgenerate 10 graphs using the same rules.

Input graphs

Graph 1: (6, [(1, 4), (1, 5), (2, 4), (2, 6), (3, 5), (3, 6)])
Graph 2: (5, [(1, 2), (1, 3), (2, 4), (3, 5), (4, 5)])
......

Output format

Answer 1: <graph>
Answer 2: <graph>
......

Task

You are given some molecules that are experimentally measured to inhibit HIV replication.

Input molecules

Molecule 1:
NNP(=S)(NN)c1ccccc1
Molecule 2:
S=c1[nH][nH]c(=S)s1
......

Output format

Answer 1: <molecule>
Answer 2: <molecule>
......

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

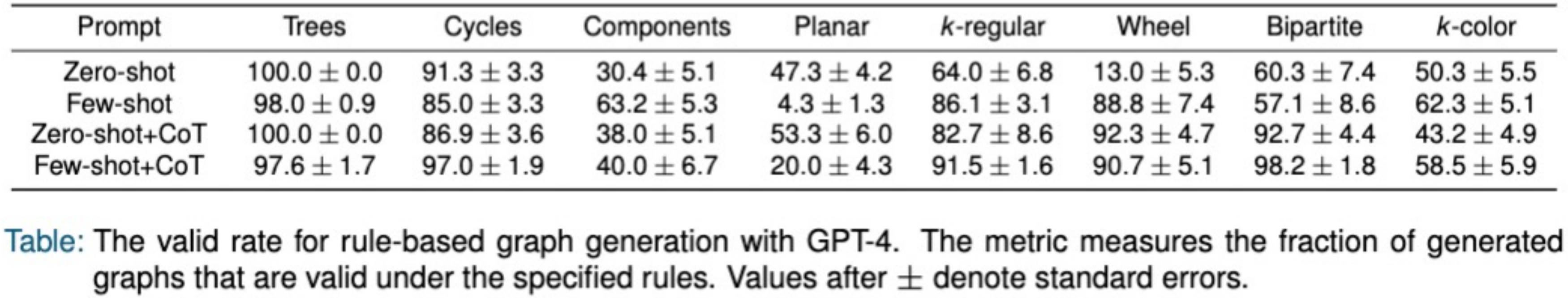
Figure: LLM4GraphGen designs a prompt tailored to each graph generation task, which is subsequently used as the input to the LLM to generate the desired graphs. Each prompt encompasses both the task description and the required output format. In the case of rule-based generation, the prompt contains the description of the rule. For distribution-based generation, a collection of graphs is provided to facilitate the LLM's learning of the underlying distribution. For property-based generation, a collection of molecules is included to enable the LLM to understand molecular properties.

1) Yao, Yang, et al. "Exploring the Potential of Large Language Models in Graph Generation." arXiv preprint arXiv:2403.14358 (2024).

# Rule-based generation

*Can LLMs understand the rules of different types of graph structures?*

- Task: generating graphs of basic structure types, given rules describing the desired structures, e.g., trees, cycles, wheel graphs, etc.
- Observations:
  - GPT-4 has reasonably good abilities for rule-based graph generation.
  - Providing examples has inconsistent impact
  - CoT prompt has diverse impacts on different evaluation metrics for graph generation.
  - As the graph size increases, the performance of LLM in graph generation decreases for most rules, except for simple cases such as cycles.



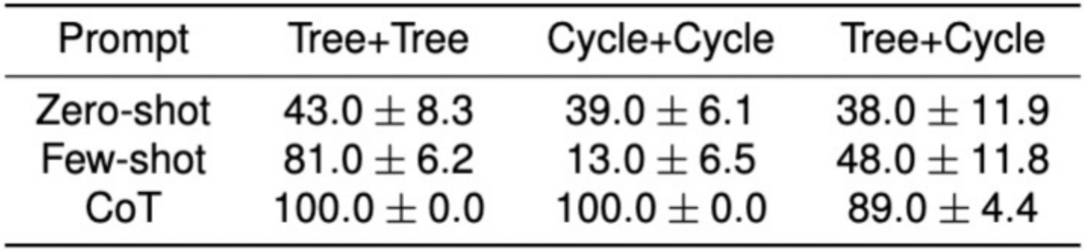
| Prompt | Trees | Cycles | Components | Planar | *k*-regular | Wheel | Bipartite | *k*-color |
|---|---|---|---|---|---|---|---|---|
| Zero-shot | $100.0 \pm 0.0$ | $91.3 \pm 3.3$ | $30.4 \pm 5.1$ | $47.3 \pm 4.2$ | $64.0 \pm 6.8$ | $13.0 \pm 5.3$ | $60.3 \pm 7.4$ | $50.3 \pm 5.5$ |
| Few-shot | $98.0 \pm 0.9$ | $85.0 \pm 3.3$ | $63.2 \pm 5.3$ | $4.3 \pm 1.3$ | $86.1 \pm 3.1$ | $88.8 \pm 7.4$ | $57.1 \pm 8.6$ | $62.3 \pm 5.1$ |
| Zero-shot+CoT | $100.0 \pm 0.0$ | $86.9 \pm 3.6$ | $38.0 \pm 5.1$ | $53.3 \pm 6.0$ | $82.7 \pm 8.6$ | $92.3 \pm 4.7$ | $92.7 \pm 4.4$ | $43.2 \pm 4.9$ |
| Few-shot+CoT | $97.6 \pm 1.7$ | $97.0 \pm 1.9$ | $40.0 \pm 6.7$ | $20.0 \pm 4.3$ | $91.5 \pm 1.6$ | $90.7 \pm 5.1$ | $98.2 \pm 1.8$ | $58.5 \pm 5.9$ |

Table: The valid rate for rule-based graph generation with GPT-4. The metric measures the fraction of generated graphs that are valid under the specified rules. Values after ± denote standard errors.

# Distribution-based generation

*Can LLMs understand the distribution of different types of graph structures?*

- Task: generating graphs following a structural type distribution p, given a set of example graphs with the same distribution.
  - Trees or cycles: Exploring the distribution of graphs.
  - Union of components: Exploring the distribution of subgraph combinations within a graph.
  - Motif graph: Exploring the distribution of subgraph combinations within the graph for more complex situations.
- Observations:
  - LLMs perorm well for simple distributions, but perform poorly in complex situations.
  - Detailed examples and CoT are helpful for distribution-based graph generation.



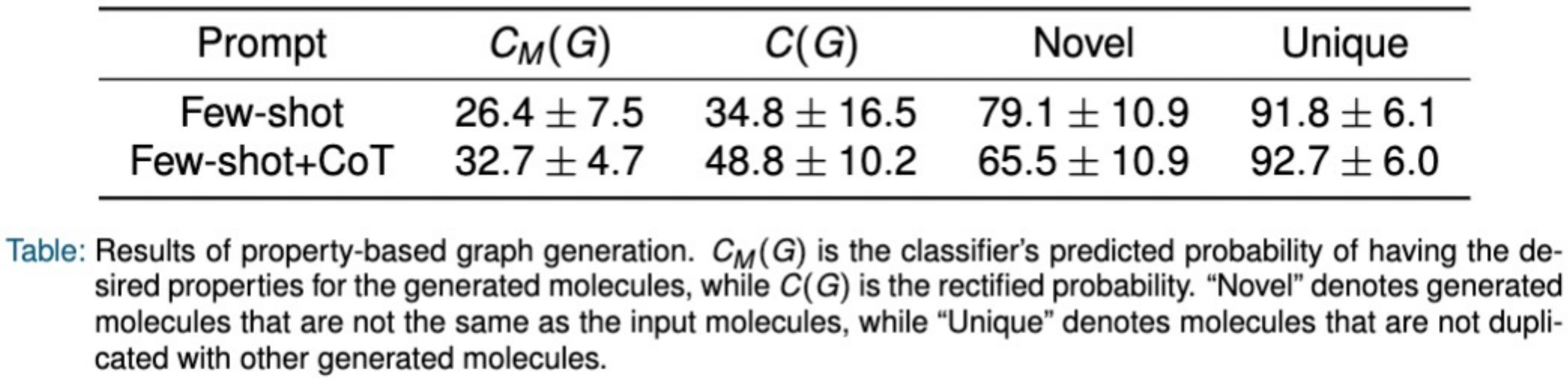
| Prompt | Tree+Tree | Cycle+Cycle | Tree+Cycle |
|---|---|---|---|
| Zero-shot | $43.0 \pm 8.3$ | $39.0 \pm 6.1$ | $38.0 \pm 11.9$ |
| Few-shot | $81.0 \pm 6.2$ | $13.0 \pm 6.5$ | $48.0 \pm 11.8$ |
| CoT | $100.0 \pm 0.0$ | $100.0 \pm 0.0$ | $89.0 \pm 4.4$ |

Table: The valid rate of two-component graph generation.

# Property-based generation

*Can LLMs understand domain knowledge of graph generation?*

- Task: generating molecule structures with specific properties, given example molecules (SMILES format).
- Observations: LLMs show preliminary abilities in generating molecules with certain properties.



| Prompt | $C_M(G)$ | $C(G)$ | Novel | Unique |
|---|---|---|---|---|
| Few-shot | $26.4 \pm 7.5$ | $34.8 \pm 16.5$ | $79.1 \pm 10.9$ | $91.8 \pm 6.1$ |
| Few-shot+CoT | $32.7 \pm 4.7$ | $48.8 \pm 10.2$ | $65.5 \pm 10.9$ | $92.7 \pm 6.0$ |

Table: Results of property-based graph generation. $C_M(G)$ is the classifier's predicted probability of having the desired properties for the generated molecules, while $C(G)$ is the rectified probability. "Novel" denotes generated molecules that are not the same as the input molecules, while "Unique" denotes molecules that are not duplicated with other generated molecules.

# GNNs and Graph Generative models for biomedical applications

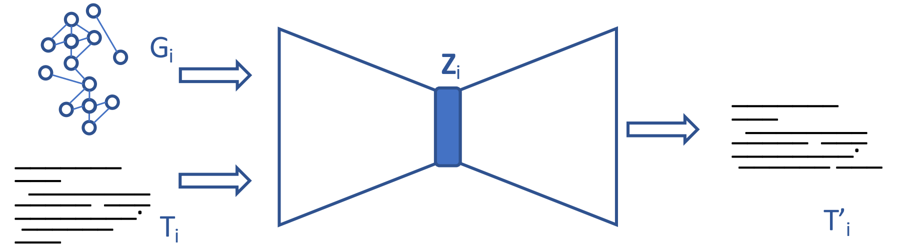
- Graph Generative models
- Generative models for Medical Graphs
- Large generative models
- Graph / LLMs
- **Multi modality for molecule generation**
- Conclusions

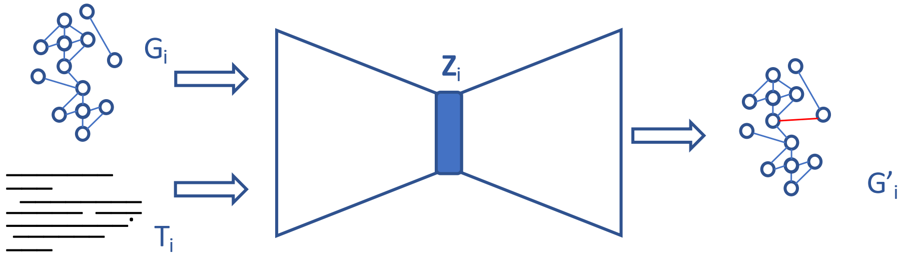
# Multimodal graph pretrained models

- Modality $m_i$ in {text, image, sound, …}
- Pretrain {$m_i$}+graph => graph/ $m_i$

Recent efforts:

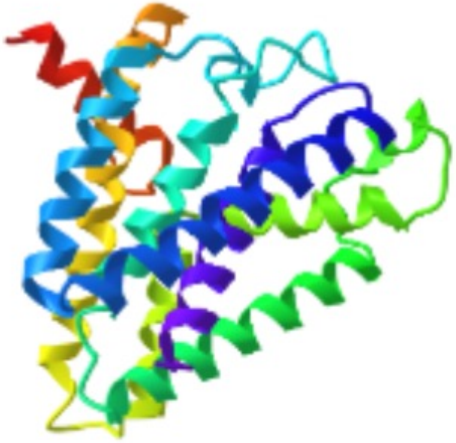
- Multimodal learning with graphs [Ektefaie et al, 2023] – *guidelines and applications*
- Investigating Pretrained Language Models for Graph-to-Text Generation[Ribeiro et al, 2021]
- Structural Information Preserving for Graph-to-Text Generation[Song et al, 2021] – *linearised graph; transformer encoder decoder – no graph encoding/generation*
- ProtNLM: Model-based Natural Language Protein [Gane et al, 2022] – *does merely classification*





# Prot2Text: Multimodal Protein's Function Generation with GNNs and Transformer

- Understanding proteins' function is a central challenge in the field of biological sciences
- essential for: drug discovery, enabling identify and target specific proteins that play critical roles in disease pathways.
- Traditionally, proteins' functions prediction is assigning predefined labels based on their characteristics [Kulmanov and Hoehndorf 2019].
- To address this limitation, *we propose a novel method that given an unknown protein produces a free text predicting its function*

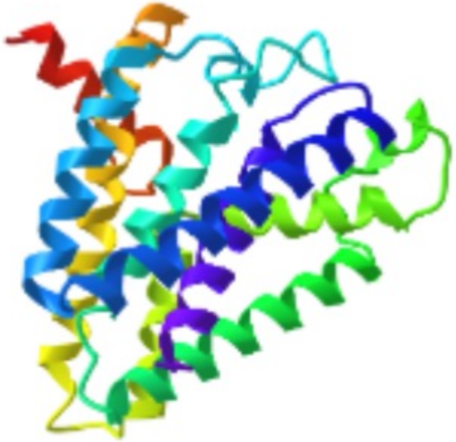


**Prot2Text: Multimodal Protein's Function Generation with GNNs and Transformers, H. Abdine, M. Chatzianastasis, C. Bouyioukos, M. Vazirgiannis, AAAI 2024**

# Prot2Text: Multimodal Protein's Function Generation with GNNs and Transformer

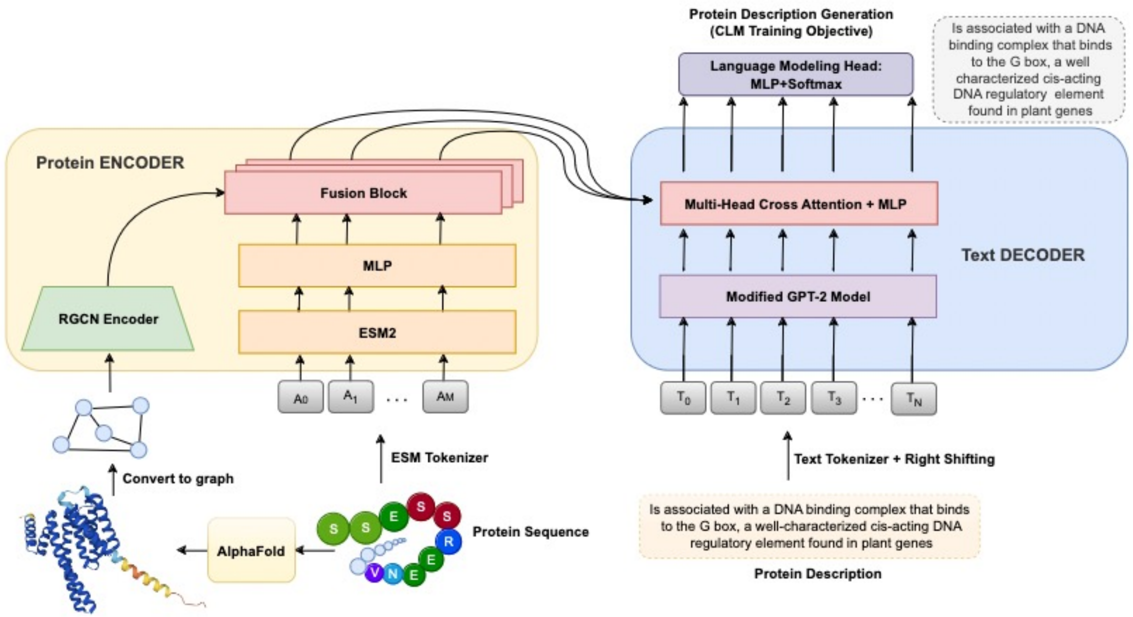
**Contributions**

- a novel *multimodal framework*, that generate detailed and accurate descriptions of proteins' functions in free text
- first time - integrate GNNs and Large Language Models (LLMs), to encompass both structural (protein's 3D structure) and sequential (amino acid's sequence) information
- propose various baselines for protein text generation
- demonstrate *integration of both graph and sequence protein information leads to better generation capabilities*.
- release a comprehensive multimodal protein dataset - 256, 690 protein structures, sequences, and textual function descriptions.



# Prot2Text: Multimodal Protein’s Function Generation with GNNs and Transformer

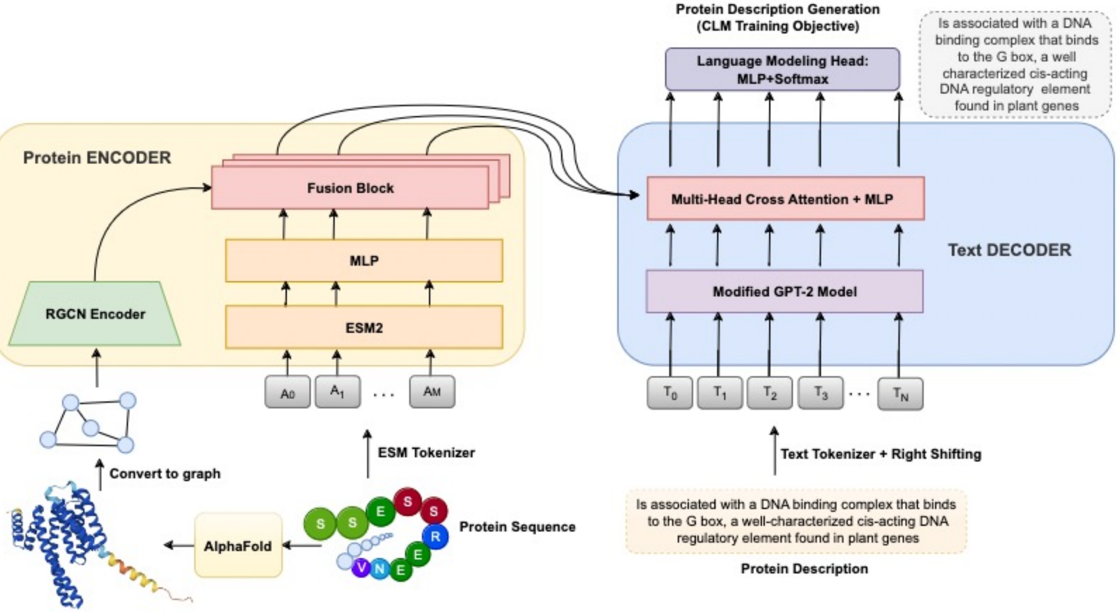
- Encoder- Decoder framework forms the backbone of the model
- encoder component
  - Relational graph convolution network (RGCN) [Schlichtkrullet al. 2018] to process the protein graphs,
  - ESM protein language model (Lin et al. 2023a)) to encode the protein’s sequence.
- cross-attention mechanism facilitates exchange of relevant information between the graph-encoded and the sequence-encoded vectors => a fused representation synthesizing structural and textual aspects.
- Decoder component: pre-trained GPT-2 generates detailed and accurate protein descriptions from fused protein representation.



Protein Description Generation (CLM Training Objective)
Language Modeling Head: MLP+Softmax
Is associated with a DNA binding complex that binds to the G box, a well characterized cis-acting DNA regulatory element found in plant genes
Protein ENCODER
Fusion Block
MLP
ESM2
RGCN Encoder
Multi-Head Cross Attention + MLP
Text DECODER
Modified GPT-2 Model
A0
A1
...
AM
T0
T1
T2
T3
...
TN
ESM Tokenizer
Convert to graph
AlphaFold
S
S
E
S
S
R
E
E
N
V
Protein Sequence
Text Tokenizer + Right Shifting
Is associated with a DNA binding complex that binds to the G box, a well-characterized cis-acting DNA regulatory element found in plant genes
Protein Description

# Prot2Text: Multimodal Protein's Function Generation with GNNs and Transformer

- **Graph Construction** - obtaining the 3D proteins' from AlphaFold
- Protein graph $G = (V, E, R)$, where
  - $V = [N] := \{1, \dots, N\}$ is the set of nodes/amino-acids of the proteins,
  - $E \subseteq V \times V$ is the set of edges/interactions between the nodes
  - $R$: set of different edge interactions.
  - Each node $u$ is associated with a feature vector $x_u \in R^d$, with attributes: *local structural features, physico-chemical properties of amino-acids.*


Protein Description Generation
(CLM Training Objective)
Is associated with a DNA binding complex that binds to the G box, a well characterized cis-acting DNA regulatory element found in plant genes
Language Modeling Head: MLP+Softmax
Protein ENCODER
Fusion Block
MLP
ESM2
RGCN Encoder
A0
A1
AM
Multi-Head Cross Attention + MLP
Text DECODER
Modified GPT-2 Model
T0
T1
T2
T3
TN
Convert to graph
ESM Tokenizer
Text Tokenizer + Right Shifting
AlphaFold
Protein Sequence
S S E S S R E E N V
Is associated with a DNA binding complex that binds to the G box, a well-characterized cis-acting DNA regulatory element found in plant genes
Protein Description

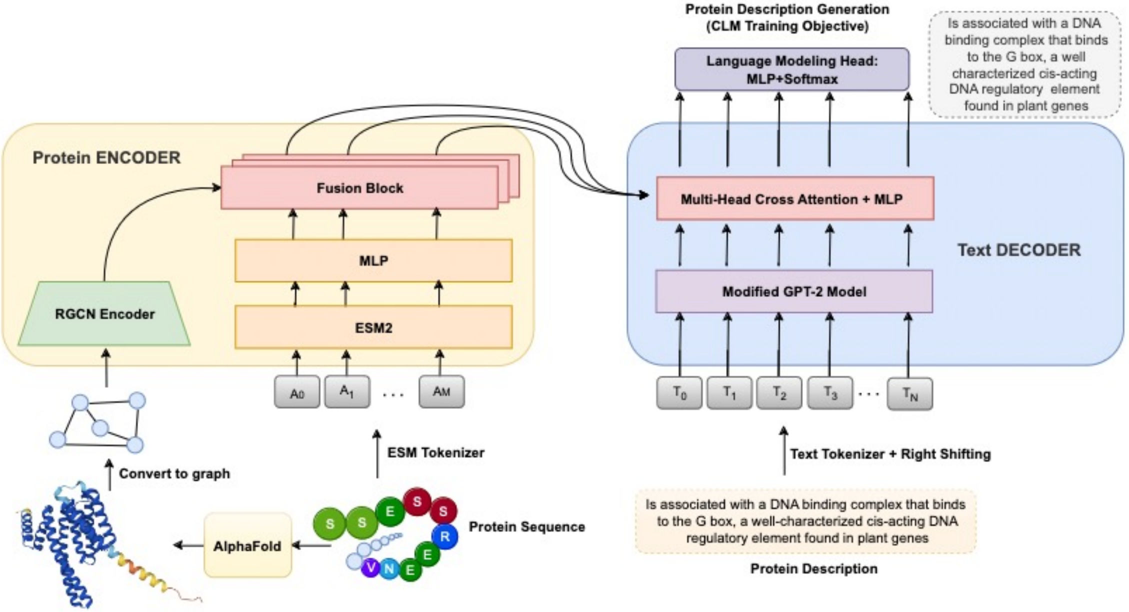
# Prot2Text: Multimodal Protein's Function Generation with GNNs and Transformer

- **Graph Construction** – 3D graphs from AlphaFold
- Protein graph $G = (V, E, R)$, where
  - $V = [N] := \{1, \ldots, N\}$ is the set of nodes/amino-acids of the proteins,
  - $E \subseteq V \times V$ is the set of edges/interactions between the nodes
  - $R$: set of different edge interactions.
  - Each node $u$ is associated with a feature vector $x_u \in R^d$, with attributes: local structural features, physico-chemical properties of amino-acids.
- **Graph Encoding**. employ a RGCN: effectively treats edge types in the message-passing mechanism.
- In layer k of the GNN, update node representations

$$\boldsymbol{x}_i^k = \sigma\left(\boldsymbol{W}_{\text{root}}^k \cdot \boldsymbol{x}_i^{k-1} + \sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_r(i)} \frac{1}{|\mathcal{N}_r(i)|}\boldsymbol{W}_r^k \cdot \boldsymbol{x}_j^{k-1}\right)$$

- Final graph representation:

$$\boldsymbol{h}_G = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i^K$$

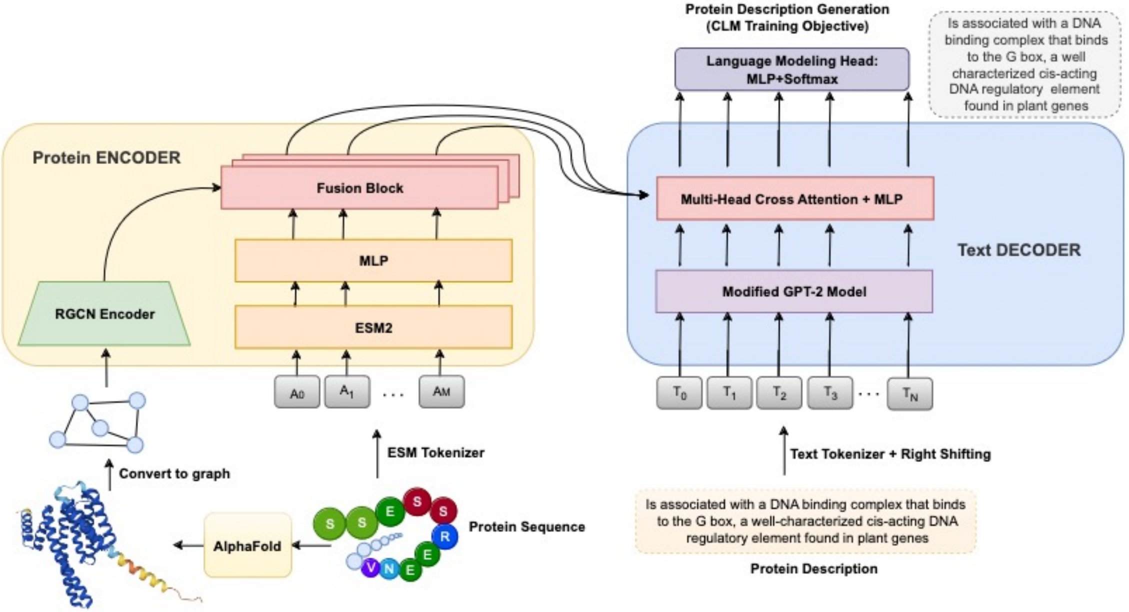

Protein Description Generation
(CLM Training Objective)
Is associated with a DNA binding complex that binds to the G box, a well characterized cis-acting DNA regulatory element found in plant genes
Language Modeling Head: MLP+Softmax
Protein ENCODER
Fusion Block
Multi-Head Cross Attention + MLP
MLP
Text DECODER
Modified GPT-2 Model
RGCN Encoder
ESM2
A0
A1
AM
T0
T1
T2
T3
TN
Convert to graph
ESM Tokenizer
Text Tokenizer + Right Shifting
AlphaFold
Protein Sequence
S
S
E
S
S
R
E
E
N
V
Is associated with a DNA binding complex that binds to the G box, a well-characterized cis-acting DNA regulatory element found in plant genes
Protein Description

# Prot2Text: Multimodal Protein's Function Generation with GNNs and Transformer

- **Sequence Encoding**. used ESM2-35M (Lin et al. 2023a) as our base model. $\boldsymbol{H}_S^0 = \boldsymbol{ESM}(\boldsymbol{P_S})\boldsymbol{W}_p$
- transforms the individual amino-acid representations, derived from the ESM embedding dimension, into the graph embedding dimension $d_{out}$.
- **Multimodal Fusion**.
- obtain the final protein encoding, with a fusion block combining the two representations:

$$\boldsymbol{H}_S^{k+1} = \left(\boldsymbol{H}_S^k + \mathbf{1}_n \boldsymbol{h}_G \boldsymbol{W}_V^k\right) \boldsymbol{W}_O^k$$

- **Text Generation** transformer decoder architecture for generating protein descriptions.
- initialize decoder components (text embedding matrix, self-attention, and language modelling head), with the pre-trained weights of GPT-2.
- forward the protein representation obtained from the protein encoder as input to the multihead *cross-attention module* within the transformer decoder.
- enabled to *effectively incorporate context from the protein representation*, to generate coherent and meaningful protein descriptions.



Protein Description Generation
(CLM Training Objective)
Is associated with a DNA binding complex that binds to the G box, a well characterized cis-acting DNA regulatory element found in plant genes
Language Modeling Head: MLP+Softmax
Protein ENCODER
Fusion Block
MLP
ESM2
RGCN Encoder
Multi-Head Cross Attention + MLP
Text DECODER
Modified GPT-2 Model
A0 A1 ... AM
T0 T1 T2 T3 ... TN
ESM Tokenizer
Convert to graph
Text Tokenizer + Right Shifting
AlphaFold
Protein Sequence
S S E S S R E E N V
Is associated with a DNA binding complex that binds to the G box, a well-characterized cis-acting DNA regulatory element found in plant genes
Protein Description

# Prot2Text: Multimodal Protein’s Function Generation with GNNs and Transformer

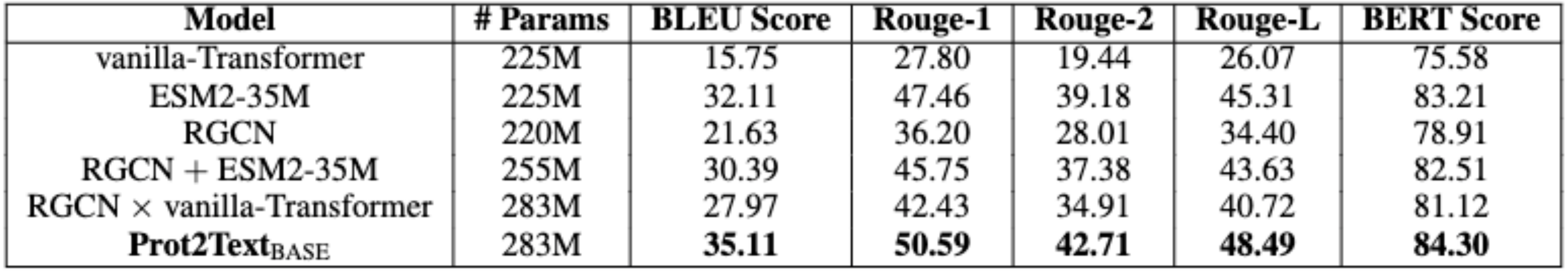
**Dataset.**

- build a *multimodal dataset with 256, 690 proteins.*
- For each protein: sequence, the AlphaFold accession ID and the textual description.
- To build this dataset, we used the SwissProt dababase (Bairoch and Apweiler 1996)
- Apply CD-HIT clustering algorithm (Li and Godzik 2006) to create a train/validation/test scheme (248.215/ 4. 172/4, 023 proteins respectively).
- maximum similarity threshold between the (train, validation test)
  sets used in the CD-HIT algorithm is 40%.

**Metrics**

- evaluate the performance of the model in the text generation task.
- BLEU Score (Papineni et al.2002): similarity between generated and reference text based onn-grams.
- Rouge-* (Lin 2004): uni/bi/longest common subsequence between generated and reference text.
- BERT Score (Zhanget al. 2020): measures the similarity between the generated text and the reference text using contextualized word embeddings from a transformer-based model.

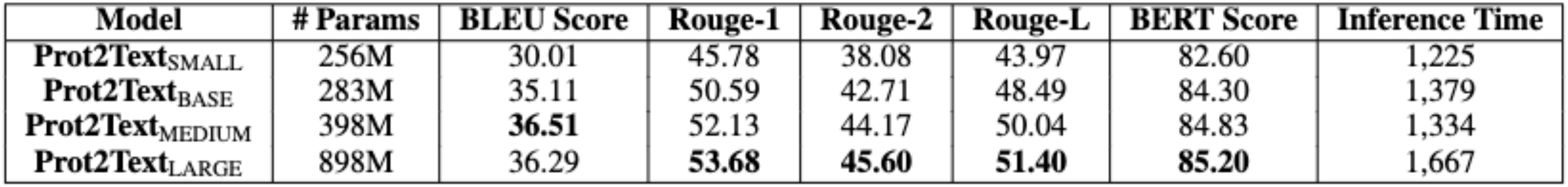
# Prot2Text: Experimental Results (ongoing)



| Model | # Params | BLEU Score | Rouge-1 | Rouge-2 | Rouge-L | BERT Score |
|---|---|---|---|---|---|---|
| vanilla-Transformer | 225M | 15.75 | 27.80 | 19.44 | 26.07 | 75.58 |
| ESM2-35M | 225M | 32.11 | 47.46 | 39.18 | 45.31 | 83.21 |
| RGCN | 220M | 21.63 | 36.20 | 28.01 | 34.40 | 78.91 |
| RGCN + ESM2-35M | 255M | 30.39 | 45.75 | 37.38 | 43.63 | 82.51 |
| RGCN × vanilla-Transformer | 283M | 27.97 | 42.43 | 34.91 | 40.72 | 81.12 |
| **Prot2Text$_{\text{BASE}}$** | 283M | **35.11** | **50.59** | **42.71** | **48.49** | **84.30** |

- Test set results encoder models,
- unimodal encoders: vanilla-Transformer, ESM2-35M,and RGCN,
- multimodal encoders: RGCN × vanilla-Transformer, RGCN + ESM2-35.
- All models share the same GPT-2 decoder.
- Structure increases performance (cross attention)
- Prot2TextBASE achieves the highest performance across all evaluation metrics

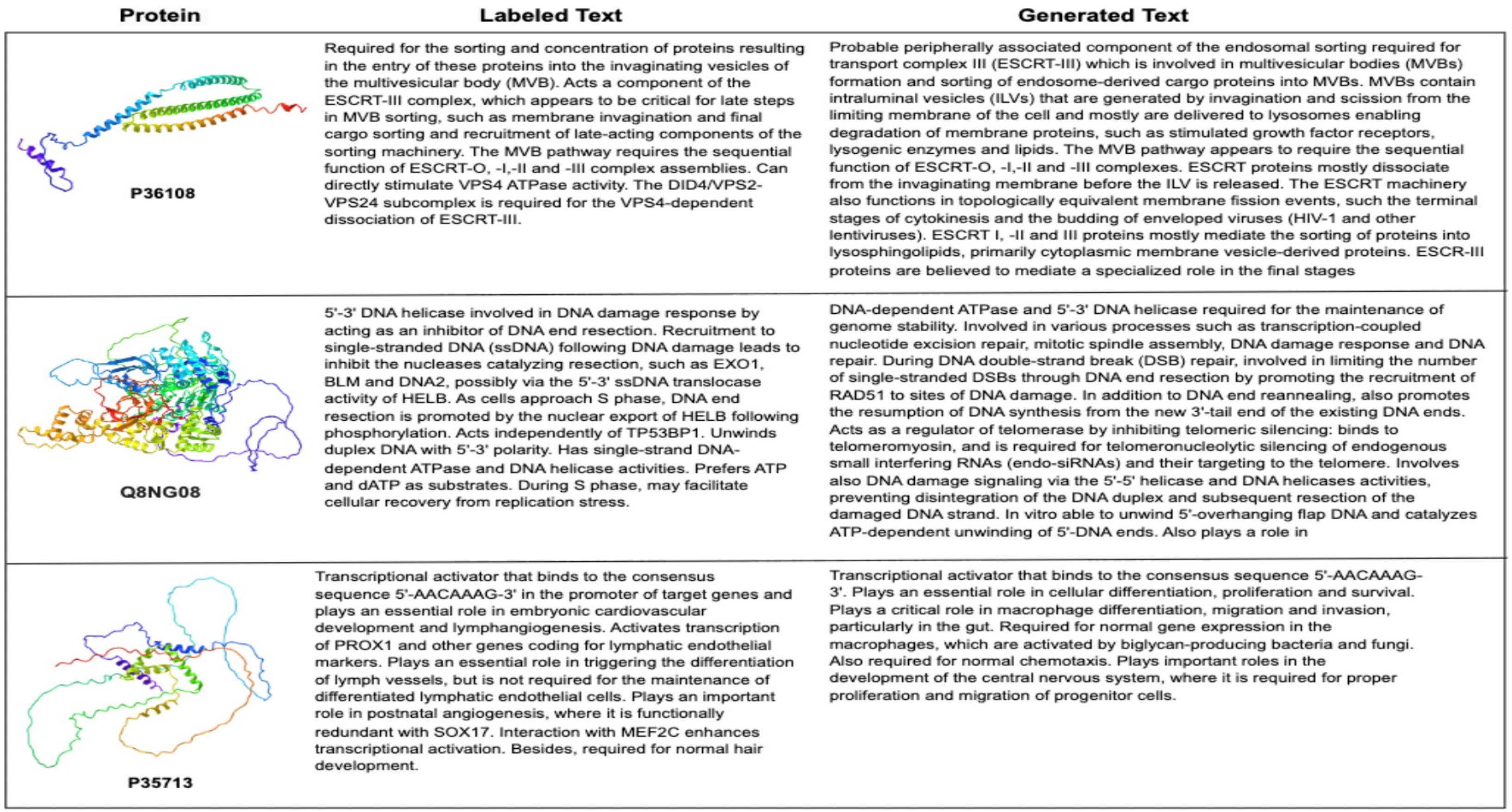
# Prot2Text: Experimental Resutls (ongoing)



| Model | # Params | BLEU Score | Rouge-1 | Rouge-2 | Rouge-L | BERT Score | Inference Time |
|---|---|---|---|---|---|---|---|
| **Prot2Text$_{\text{SMALL}}$** | 256M | 30.01 | 45.78 | 38.08 | 43.97 | 82.60 | 1,225 |
| **Prot2Text$_{\text{BASE}}$** | 283M | 35.11 | 50.59 | 42.71 | 48.49 | 84.30 | 1,379 |
| **Prot2Text$_{\text{MEDIUM}}$** | 398M | **36.51** | 52.13 | 44.17 | 50.04 | 84.83 | 1,334 |
| **Prot2Text$_{\text{LARGE}}$** | 898M | 36.29 | **53.68** | **45.60** | **51.40** | **85.20** | 1,667 |

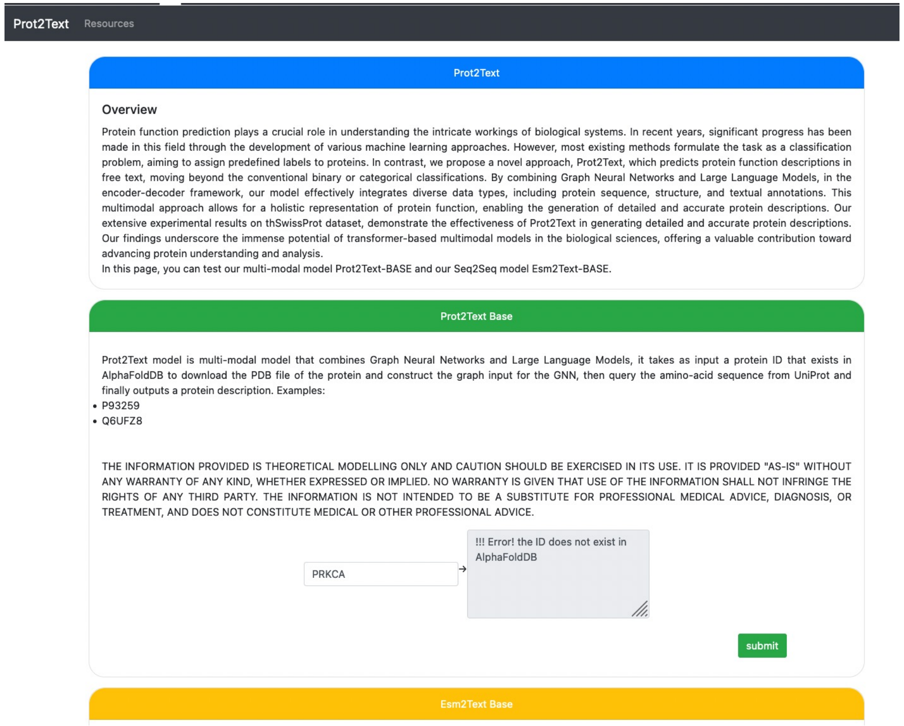
- Test set results for different size variations of Prot2Text.
- Larger models outperform their smaller counterparts across most evaluation metrics, indicating the benefits of employing larger language models in the Prot2Text framework.
- The Prot2TextBASE model, strikes an optimal balance between performance and computational efficiency. This configuration demonstrates improved performance compared to the smaller model while still maintaining reasonable computational costs.

# Prot2Text: Experimental Resutls (ongoing)



| Protein | Labeled Text | Generated Text |
|---|---|---|
| P36108 | Required for the sorting and concentration of proteins resulting in the entry of these proteins into the invaginating vesicles of the multivesicular body (MVB). Acts a component of the ESCRT-III complex, which appears to be critical for late steps in MVB sorting, such as membrane invagination and final cargo sorting and recruitment of late-acting components of the sorting machinery. The MVB pathway requires the sequential function of ESCRT-O, -I,-II and -III complex assemblies. Can directly stimulate VPS4 ATPase activity. The DID4/VPS2-VPS24 subcomplex is required for the VPS4-dependent dissociation of ESCRT-III. | Probable peripherally associated component of the endosomal sorting required for transport complex III (ESCRT-III) which is involved in multivesicular bodies (MVBs) formation and sorting of endosome-derived cargo proteins into MVBs. MVBs contain intraluminal vesicles (ILVs) that are generated by invagination and scission from the limiting membrane of the cell and mostly are delivered to lysosomes enabling degradation of membrane proteins, such as stimulated growth factor receptors, lysogenic enzymes and lipids. The MVB pathway appears to require the sequential function of ESCRT-O, -I,-II and -III complexes. ESCRT proteins mostly dissociate from the invaginating membrane before the ILV is released. The ESCRT machinery also functions in topologically equivalent membrane fission events, such the terminal stages of cytokinesis and the budding of enveloped viruses (HIV-1 and other lentiviruses). ESCRT I, -II and III proteins mostly mediate the sorting of proteins into lysosphingolipids, primarily cytoplasmic membrane vesicle-derived proteins. ESCR-III proteins are believed to mediate a specialized role in the final stages |
| Q8NG08 | 5'-3' DNA helicase involved in DNA damage response by acting as an inhibitor of DNA end resection. Recruitment to single-stranded DNA (ssDNA) following DNA damage leads to inhibit the nucleases catalyzing resection, such as EXO1, BLM and DNA2, possibly via the 5'-3' ssDNA translocase activity of HELB. As cells approach S phase, DNA end resection is promoted by the nuclear export of HELB following phosphorylation. Acts independently of TP53BP1. Unwinds duplex DNA with 5'-3' polarity. Has single-strand DNA-dependent ATPase and DNA helicase activities. Prefers ATP and dATP as substrates. During S phase, may facilitate cellular recovery from replication stress. | DNA-dependent ATPase and 5'-3' DNA helicase required for the maintenance of genome stability. Involved in various processes such as transcription-coupled nucleotide excision repair, mitotic spindle assembly, DNA damage response and DNA repair. During DNA double-strand break (DSB) repair, involved in limiting the number of single-stranded DSBs through DNA end resection by promoting the recruitment of RAD51 to sites of DNA damage. In addition to DNA end reannealing, also promotes the resumption of DNA synthesis from the new 3'-tail end of the existing DNA ends. Acts as a regulator of telomerase by inhibiting telomeric silencing: binds to telomeromyosin, and is required for telomeronucleolytic silencing of endogenous small interfering RNAs (endo-siRNAs) and their targeting to the telomere. Involves also DNA damage signaling via the 5'-5' helicase and DNA helicases activities, preventing disintegration of the DNA duplex and subsequent resection of the damaged DNA strand. In vitro able to unwind 5'-overhanging flap DNA and catalyzes ATP-dependent unwinding of 5'-DNA ends. Also plays a role in |
| P35713 | Transcriptional activator that binds to the consensus sequence 5'-AACAAAG-3' in the promoter of target genes and plays an essential role in embryonic cardiovascular development and lymphangiogenesis. Activates transcription of PROX1 and other genes coding for lymphatic endothelial markers. Plays an essential role in triggering the differentiation of lymph vessels, but is not required for the maintenance of differentiated lymphatic endothelial cells. Plays an important role in postnatal angiogenesis, where it is functionally redundant with SOX17. Interaction with MEF2C enhances transcriptional activation. Besides, required for normal hair development. | Transcriptional activator that binds to the consensus sequence 5'-AACAAAG-3'. Plays an essential role in cellular differentiation, proliferation and survival. Plays a critical role in macrophage differentiation, migration and invasion, particularly in the gut. Required for normal gene expression in the macrophages, which are activated by biglycan-producing bacteria and fungi. Also required for normal chemotaxis. Plays important roles in the development of the central nervous system, where it is required for proper proliferation and migration of progenitor cells. |

# Demonstration



Overview

Protein function prediction plays a crucial role in understanding the intricate workings of biological systems. In recent years, significant progress has been made in this field through the development of various machine learning approaches. However, most existing methods formulate the task as a classification problem, aiming to assign predefined labels to proteins. In contrast, we propose a novel approach, Prot2Text, which predicts protein function descriptions in free text, moving beyond the conventional binary or categorical classifications. By combining Graph Neural Networks and Large Language Models, in the encoder-decoder framework, our model effectively integrates diverse data types, including protein sequence, structure, and textual annotations. This multimodal approach allows for a holistic representation of protein function, enabling the generation of detailed and accurate protein descriptions. Our extensive experimental results on thSwissProt dataset, demonstrate the effectiveness of Prot2Text in generating detailed and accurate protein descriptions. Our findings underscore the immense potential of transformer-based multimodal models in the biological sciences, offering a valuable contribution toward advancing protein understanding and analysis.
In this page, you can test our multi-modal model Prot2Text-BASE and our Seq2Seq model Esm2Text-BASE.

Prot2Text model is multi-modal model that combines Graph Neural Networks and Large Language Models, it takes as input a protein ID that exists in AlphaFoldDB to download the PDB file of the protein and construct the graph input for the GNN, then query the amino-acid sequence from UniProt and finally outputs a protein description. Examples:

- P93259
- Q6UFZ8

THE INFORMATION PROVIDED IS THEORETICAL MODELLING ONLY AND CAUTION SHOULD BE EXERCISED IN ITS USE. IT IS PROVIDED "AS-IS" WITHOUT ANY WARRANTY OF ANY KIND, WHETHER EXPRESSED OR IMPLIED. NO WARRANTY IS GIVEN THAT USE OF THE INFORMATION SHALL NOT INFRINGE THE RIGHTS OF ANY THIRD PARTY. THE INFORMATION IS NOT INTENDED TO BE A SUBSTITUTE FOR PROFESSIONAL MEDICAL ADVICE, DIAGNOSIS, OR TREATMENT, AND DOES NOT CONSTITUTE MEDICAL OR OTHER PROFESSIONAL ADVICE.

PRKCA →

!!! Error! the ID does not exist in AlphaFoldDB

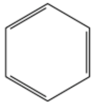
http://nlp.polytechnique.fr/

# Multimodal Gen AI for molecules

- Task: Generate or modify the structure of molecules based on text descriptions.
- Motivation:
  - Discovery in medicine and science is expensive.
  - huge amount of time and money for experts to design new molecules with the desired functionality every year.
  - deep learning tools are essential for facilitating molecule discovery.
- “Water is an oxygen hydride consisting of an oxygen atom that is covalently bonded to

  two hydrogen atoms” => H - O - H

- Molecule representations
  - SMILES(Simplified Molecular Input Line Entry System): string of compact encoding of molecular structures, making it easy to share and search.
  - Graph: each atom in the molecule corresponds to a node in the graph, bonds between atoms are represented as edges, where some attributes can be applied to nodes and edges.
  - else: SELFIES, 3-D structure, fingerprints..

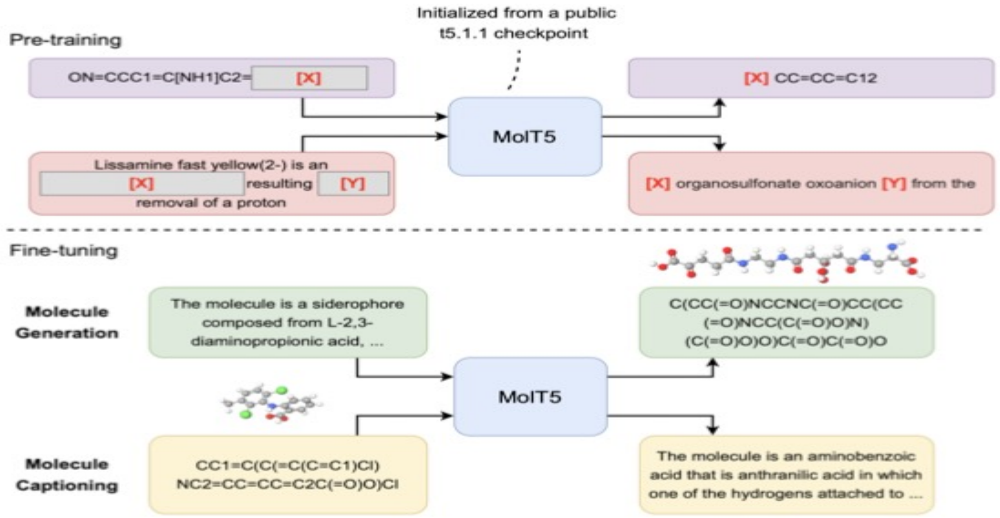
**Benzene:**  (C6H6)

- SMILES: c1ccccc1
- Graph: as adjacency matrix:

$$\begin{matrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{matrix}$$

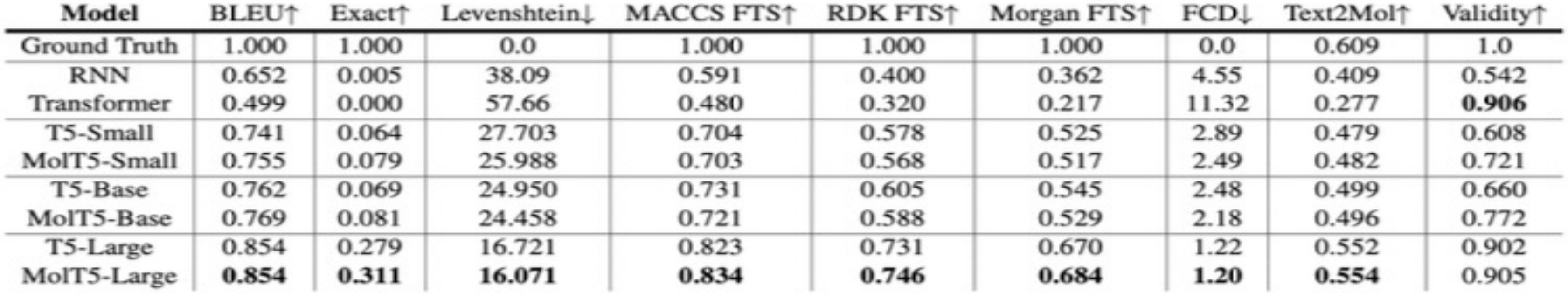
# MolT5[1]

- self-supervised learning framework for pre-training models on un-labeled natural language text and molecule strings.
- First pre-train MolT5 on both SMILES string and natural language using the “replace corrupted spans” objective. Fine-tuned for task of molecule captioning or generation.

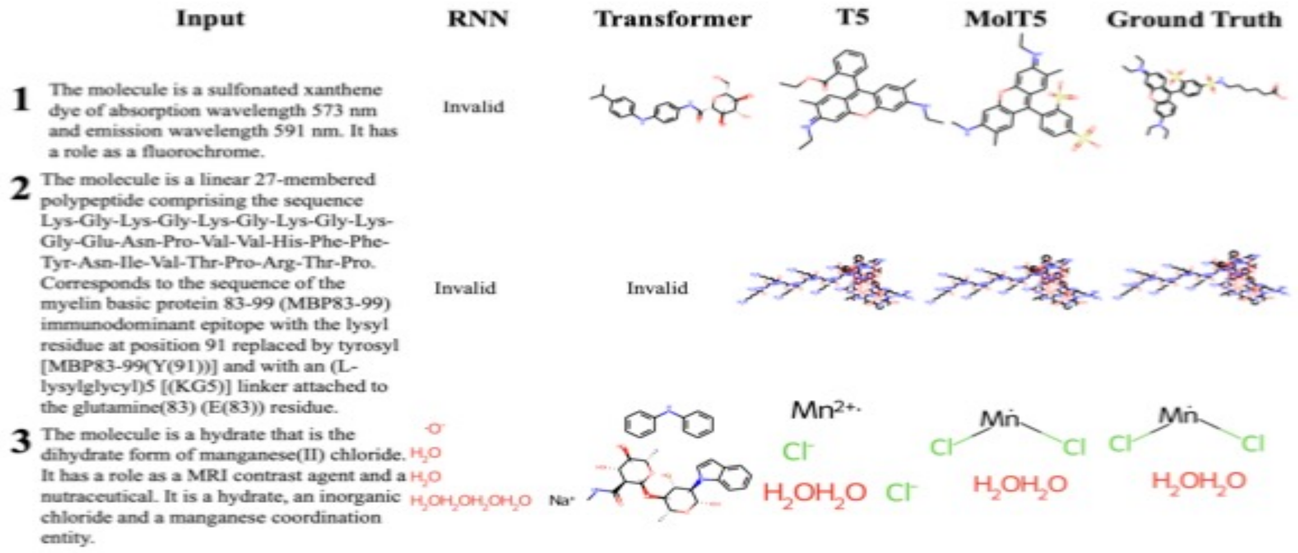

Initialized from a public t5.1.1 checkpoint
Pre-training
ON=CCC1=C[NH1]C2= [X]
Lissamine fast yellow(2-) is an [X] resulting [Y] removal of a proton
MolT5
[X] CC=CC=C12
[X] organosulfonate oxoanion [Y] from the
Fine-tuning
Molecule Generation
The molecule is a siderophore composed from L-2,3-diaminopropionic acid, ...
MolT5
C(CC(=O)NCCNC(=O)CC(CC(=O)NCC(C(=O)O)N)(C(=O)O)O)C(=O)C(=O)O
Molecule Captioning
CC1=C(C(=C(C=C1)Cl)NC2=CC=CC=C2C(=O)O)Cl
The molecule is an aminobenzoic acid that is anthranilic acid in which one of the hydrogens attached to ...

[1] Translation between Molecules and Natural Language, Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, Heng Ji, https://arxiv.org/abs/2204.11817

# MolT5[1]



| Model | BLEU↑ | Exact↑ | Levenshtein↓ | MACCS FTS↑ | RDK FTS↑ | Morgan FTS↑ | FCD↓ | Text2Mol↑ | Validity↑ |
|---|---|---|---|---|---|---|---|---|---|
| Ground Truth | 1.000 | 1.000 | 0.0 | 1.000 | 1.000 | 1.000 | 0.0 | 0.609 | 1.0 |
| RNN | 0.652 | 0.005 | 38.09 | 0.591 | 0.400 | 0.362 | 4.55 | 0.409 | 0.542 |
| Transformer | 0.499 | 0.000 | 57.66 | 0.480 | 0.320 | 0.217 | 11.32 | 0.277 | **0.906** |
| T5-Small | 0.741 | 0.064 | 27.703 | 0.704 | 0.578 | 0.525 | 2.89 | 0.479 | 0.608 |
| MolT5-Small | 0.755 | 0.079 | 25.988 | 0.703 | 0.568 | 0.517 | 2.49 | 0.482 | 0.721 |
| T5-Base | 0.762 | 0.069 | 24.950 | 0.731 | 0.605 | 0.545 | 2.48 | 0.499 | 0.660 |
| MolT5-Base | 0.769 | 0.081 | 24.458 | 0.721 | 0.588 | 0.529 | 2.18 | 0.496 | 0.772 |
| T5-Large | 0.854 | 0.279 | 16.721 | 0.823 | 0.731 | 0.670 | 1.22 | 0.552 | 0.902 |
| MolT5-Large | **0.854** | **0.311** | **16.071** | **0.834** | **0.746** | **0.684** | **1.20** | **0.554** | 0.905 |



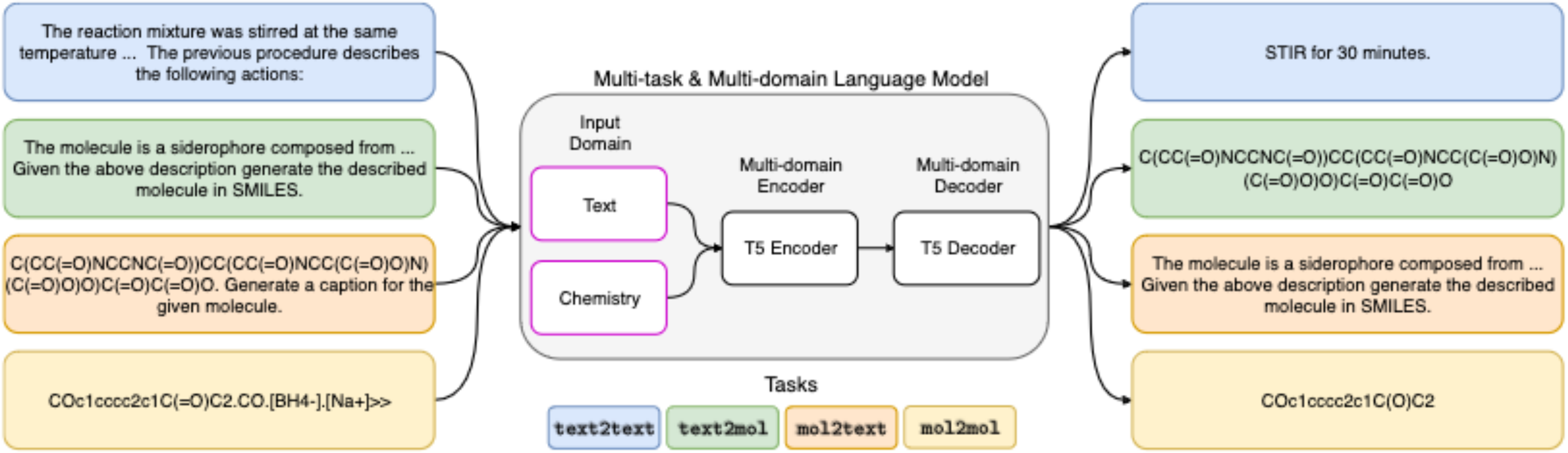
1 The molecule is a sulfonated xanthene dye of absorption wavelength 573 nm and emission wavelength 591 nm. It has a role as a fluorochrome.

2 The molecule is a linear 27-membered polypeptide comprising the sequence Lys-Gly-Lys-Gly-Lys-Gly-Lys-Gly-Lys-Gly-Glu-Asn-Pro-Val-Val-His-Phe-Phe-Tyr-Asn-Ile-Val-Thr-Pro-Arg-Thr-Pro. Corresponds to the sequence of the myelin basic protein 83-99 (MBP83-99) immunodominant epitope with the lysyl residue at position 91 replaced by tyrosyl [MBP83-99(Y(91))] and with an (L-lysylglycyl)5 [(KG5)] linker attached to the glutamine(83) (E(83)) residue.

3 The molecule is a hydrate that is the dihydrate form of manganese(II) chloride. It has a role as a MRI contrast agent and a nutraceutical. It is a hydrate, an inorganic chloride and a manganese coordination entity.

# Text+Chem T5[1]

- first multi-domain, multi-task LM for chemical and natural language domains
- Key Ideas:
  - Multi-tasking across multiple domains.
  - Weight sharing and information sharing across domains(encoders) .
  - efficient training strategy without the need for costly pre-training on large dataset and task-specific fine-tuning.



The reaction mixture was stirred at the same temperature ... The previous procedure describes the following actions:
The molecule is a siderophore composed from ... Given the above description generate the described molecule in SMILES.
C(CC(=O)NCCNC(=O))CC(CC(=O)NCC(C(=O)O)N)(C(=O)O)O)C(=O)C(=O)O. Generate a caption for the given molecule.
COc1cccc2c1C(=O)C2.CO.[BH4-].[Na+]>>
Multi-task & Multi-domain Language Model
Input Domain
Text
Chemistry
Multi-domain Encoder
T5 Encoder
Multi-domain Decoder
T5 Decoder
Tasks
text2text
text2mol
mol2text
mol2mol
STIR for 30 minutes.
C(CC(=O)NCCNC(=O))CC(CC(=O)NCC(C(=O)O)N)(C(=O)O)O)C(=O)C(=O)O
The molecule is a siderophore composed from ... Given the above description generate the described molecule in SMILES.
COc1cccc2c1C(O)C2

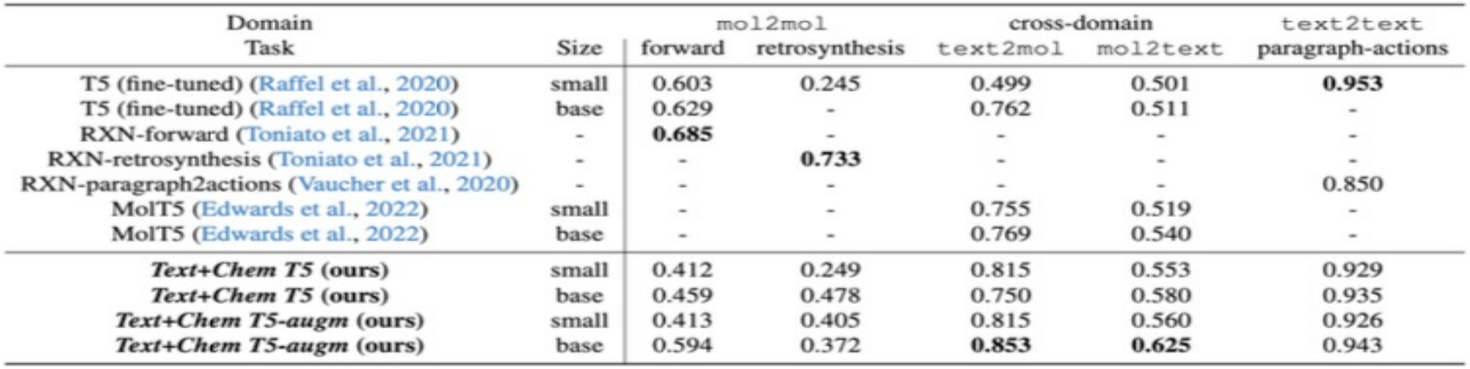
**Unifying Molecular and Textual Representations via Multi-task Language Modelling,** https://arxiv.org/abs/2301.12586

Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, Matteo Manica

Demo: https://huggingface.co/spaces/GT4SD/multitask-text-and-chemistry-t5

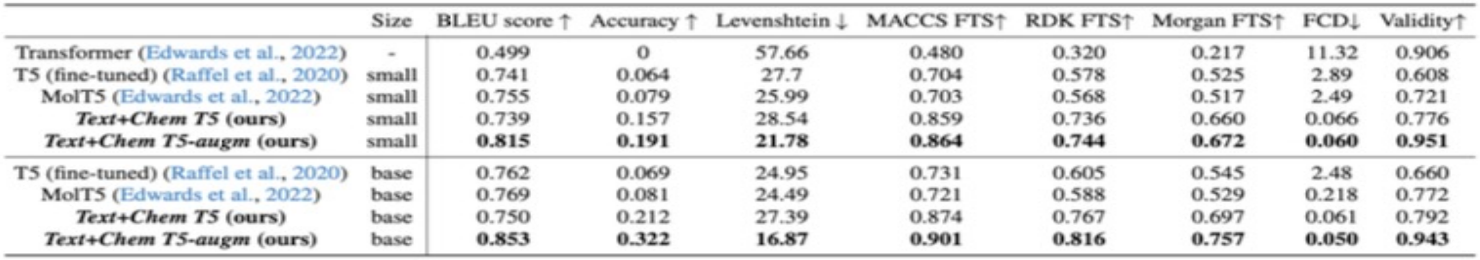
# Text+Chem T5

## Results(all):



| Domain<br>Task | Size | mol2mol<br>forward | <br>retrosynthesis | cross-domain<br>text2mol | <br>mol2text | text2text<br>paragraph-actions |
|---|---|---|---|---|---|---|
| T5 (fine-tuned) (Raffel et al., 2020) | small | 0.603 | 0.245 | 0.499 | 0.501 | **0.953** |
| T5 (fine-tuned) (Raffel et al., 2020) | base | 0.629 | - | 0.762 | 0.511 | - |
| RXN-forward (Toniato et al., 2021) | - | **0.685** | - | - | - | - |
| RXN-retrosynthesis (Toniato et al., 2021) | - | - | **0.733** | - | - | - |
| RXN-paragraph2actions (Vaucher et al., 2020) | - | - | - | - | - | 0.850 |
| MolT5 (Edwards et al., 2022) | small | - | - | 0.755 | 0.519 | - |
| MolT5 (Edwards et al., 2022) | base | - | - | 0.769 | 0.540 | - |
| ***Text+Chem T5* (ours)** | small | 0.412 | 0.249 | 0.815 | 0.553 | 0.929 |
| ***Text+Chem T5* (ours)** | base | 0.459 | 0.478 | 0.750 | 0.580 | 0.935 |
| ***Text+Chem T5-augm* (ours)** | small | 0.413 | 0.405 | 0.815 | 0.560 | 0.926 |
| ***Text+Chem T5-augm* (ours)** | base | 0.594 | 0.372 | **0.853** | **0.625** | 0.943 |

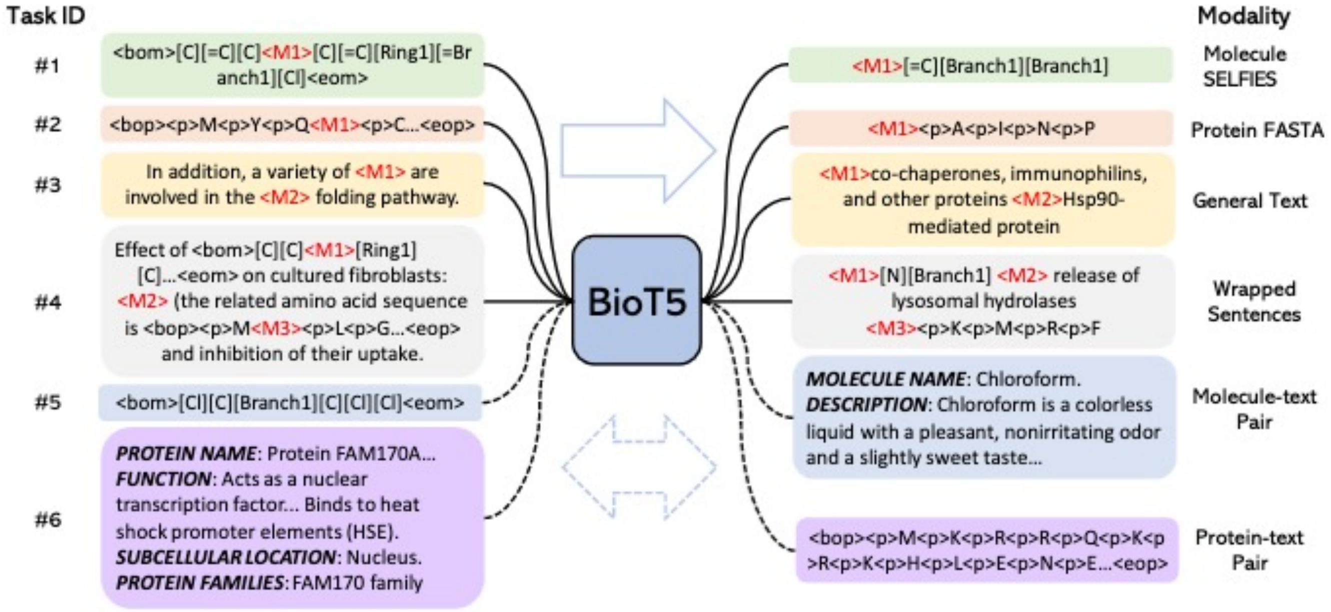
## Results(text2mol):



| | Size | BLEU score ↑ | Accuracy ↑ | Levenshtein ↓ | MACCS FTS↑ | RDK FTS↑ | Morgan FTS↑ | FCD↓ | Validity↑ |
|---|---|---|---|---|---|---|---|---|---|
| Transformer (Edwards et al., 2022) | - | 0.499 | 0 | 57.66 | 0.480 | 0.320 | 0.217 | 11.32 | 0.906 |
| T5 (fine-tuned) (Raffel et al., 2020) | small | 0.741 | 0.064 | 27.7 | 0.704 | 0.578 | 0.525 | 2.89 | 0.608 |
| MolT5 (Edwards et al., 2022) | small | 0.755 | 0.079 | 25.99 | 0.703 | 0.568 | 0.517 | 2.49 | 0.721 |
| ***Text+Chem T5* (ours)** | small | 0.739 | 0.157 | 28.54 | 0.859 | 0.736 | 0.660 | 0.066 | 0.776 |
| ***Text+Chem T5-augm* (ours)** | small | **0.815** | **0.191** | **21.78** | **0.864** | **0.744** | **0.672** | **0.060** | **0.951** |
| T5 (fine-tuned) (Raffel et al., 2020) | base | 0.762 | 0.069 | 24.95 | 0.731 | 0.605 | 0.545 | 2.48 | 0.660 |
| MolT5 (Edwards et al., 2022) | base | 0.769 | 0.081 | 24.49 | 0.721 | 0.588 | 0.529 | 0.218 | 0.772 |
| ***Text+Chem T5* (ours)** | base | 0.750 | 0.212 | 27.39 | 0.874 | 0.767 | 0.697 | 0.061 | 0.792 |
| ***Text+Chem T5-augm* (ours)** | base | **0.853** | **0.322** | **16.87** | **0.901** | **0.816** | **0.757** | **0.050** | **0.943** |

# Bio T5[1]

- BioT5 uses T5 model to incorporate modalities, in pre-training it uses:
- modality including molecule SELFIES, and general text independently.
- wrapped text from scientific corpus.
- Bidirectional translation for the molecule SELFIES-text pairs.



Task ID

#1 <bom>[C][=C][C]<M1>[C][=C][Ring1][=Branch1][Cl]<eom>

#2 <bop><p>M<p>Y<p>Q<M1><p>C...<eop>

#3 In addition, a variety of <M1> are involved in the <M2> folding pathway.

#4 Effect of <bom>[C][C]<M1>[Ring1][C]...<eom> on cultured fibroblasts: <M2> (the related amino acid sequence is <bop><p>M<M3><p>L<p>G...<eop> and inhibition of their uptake.

#5 <bom>[Cl][C][Branch1][C][Cl][Cl]<eom>

#6 ***PROTEIN NAME***: Protein FAM170A... ***FUNCTION***: Acts as a nuclear transcription factor... Binds to heat shock promoter elements (HSE). ***SUBCELLULAR LOCATION***: Nucleus. ***PROTEIN FAMILIES***: FAM170 family

BioT5

Modality

<M1>[=C][Branch1][Branch1] Molecule SELFIES

<M1><p>A<p>I<p>N<p>P Protein FASTA

<M1>co-chaperones, immunophilins, and other proteins <M2>Hsp90-mediated protein General Text

<M1>[N][Branch1] <M2> release of lysosomal hydrolases <M3><p>K<p>M<p>R<p>F Wrapped Sentences

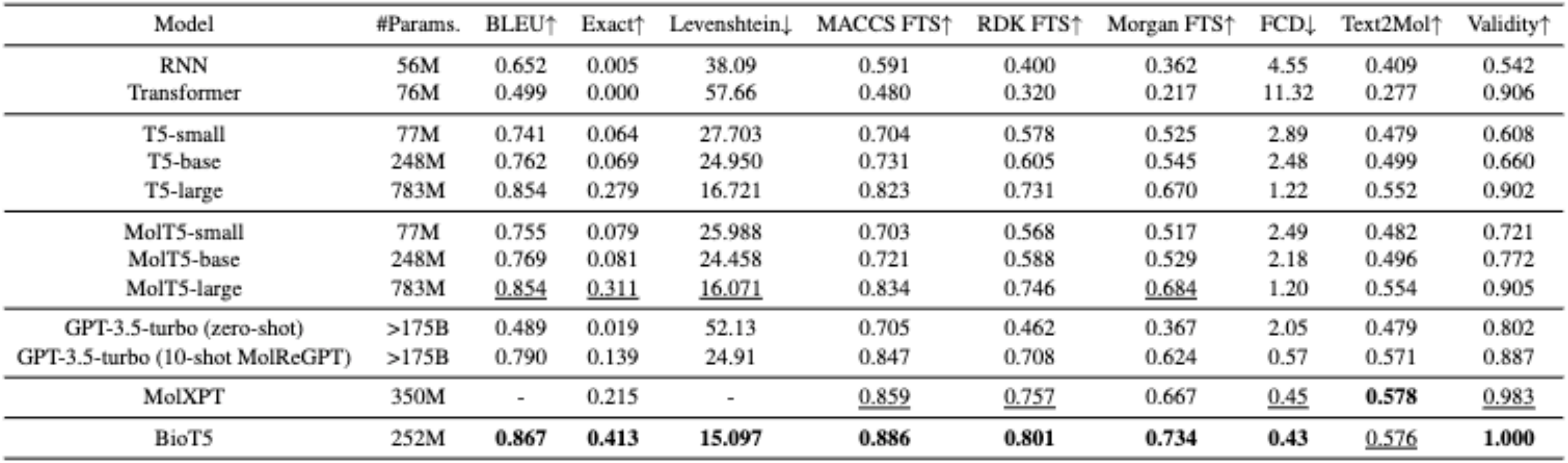
***MOLECULE NAME***: Chloroform. ***DESCRIPTION***: Chloroform is a colorless liquid with a pleasant, nonirritating odor and a slightly sweet taste... Molecule-text Pair

<bop><p>M<p>K<p>R<p>R<p>Q<p>K<p>R<p>K<p>H<p>L<p>E<p>N<p>E...<eop> Protein-text Pair

[1] BioT5: Enriching Cross-modal Integration in Biology with Chemical Knowledge and Natural Language Associations, https://arxiv.org/pdf/2310.07276.pdf

# Bio T5

- Text2mol task

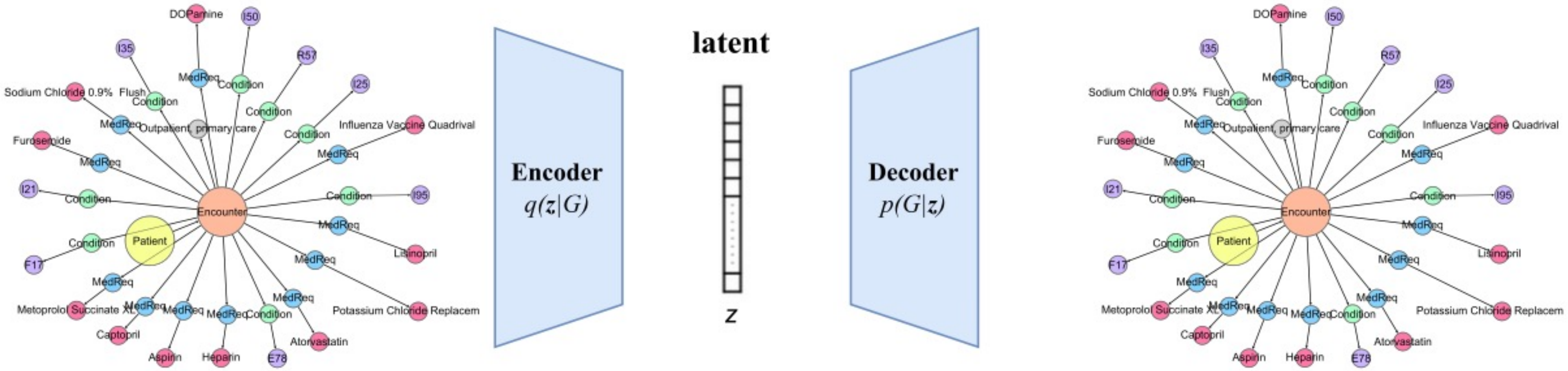


| Model | #Params. | BLEU↑ | Exact↑ | Levenshtein↓ | MACCS FTS↑ | RDK FTS↑ | Morgan FTS↑ | FCD↓ | Text2Mol↑ | Validity↑ |
|---|---|---|---|---|---|---|---|---|---|---|
| RNN | 56M | 0.652 | 0.005 | 38.09 | 0.591 | 0.400 | 0.362 | 4.55 | 0.409 | 0.542 |
| Transformer | 76M | 0.499 | 0.000 | 57.66 | 0.480 | 0.320 | 0.217 | 11.32 | 0.277 | 0.906 |
| T5-small | 77M | 0.741 | 0.064 | 27.703 | 0.704 | 0.578 | 0.525 | 2.89 | 0.479 | 0.608 |
| T5-base | 248M | 0.762 | 0.069 | 24.950 | 0.731 | 0.605 | 0.545 | 2.48 | 0.499 | 0.660 |
| T5-large | 783M | 0.854 | 0.279 | 16.721 | 0.823 | 0.731 | 0.670 | 1.22 | 0.552 | 0.902 |
| MolT5-small | 77M | 0.755 | 0.079 | 25.988 | 0.703 | 0.568 | 0.517 | 2.49 | 0.482 | 0.721 |
| MolT5-base | 248M | 0.769 | 0.081 | 24.458 | 0.721 | 0.588 | 0.529 | 2.18 | 0.496 | 0.772 |
| MolT5-large | 783M | 0.854 | 0.311 | 16.071 | 0.834 | 0.746 | 0.684 | 1.20 | 0.554 | 0.905 |
| GPT-3.5-turbo (zero-shot) | >175B | 0.489 | 0.019 | 52.13 | 0.705 | 0.462 | 0.367 | 2.05 | 0.479 | 0.802 |
| GPT-3.5-turbo (10-shot MolReGPT) | >175B | 0.790 | 0.139 | 24.91 | 0.847 | 0.708 | 0.624 | 0.57 | 0.571 | 0.887 |
| MolXPT | 350M | - | 0.215 | - | 0.859 | 0.757 | 0.667 | 0.45 | **0.578** | 0.983 |
| BioT5 | 252M | **0.867** | **0.413** | **15.097** | **0.886** | **0.801** | **0.734** | **0.43** | 0.576 | **1.000** |

# Conclusions

- Graph Generative AI
  - High potential topic with crucial applications: Power grid/telecoms, VLSI, social networks, chemistry/proteins/pharma
  - via LLMs – not promising currently – only for graph reasoning
  - Need for neural graph generators
- Graph Pretrained models
  - Different challenges
    - pretraining tasks are diverse, masking may not be enough
    - Decoder architecture – permutation invariance
    - Graph data loaders scaling
    - Prompting ? Domain depentent
    - Multimodality with graphs – architectural challenges
- Tasks for evaluation – potential
  - Variety of domain dependent metrics
  - genenerated graphs similarity is non trivial (graph kernels/embeddings)

# THANK YOU!



*Acknowledgements*

Dr. G. Nikolentzos, H.Abdine, M. Chatzianastassis, G. Shang, Y. Zhang