Molecule Design and generation with Graph neural networks

Daoud Brahim
Neit Mohamed Aymen
Djellab Ahmed Abdeljalil
Remili Khalil

1st January 2024

presentation content:

- Introduction To generative AI
- Introduction To Molecular Generation
- Discuss the chosen papers
- Code implementation and review
- Conclusion

What is generative AI?

Generative AI, or genAI, refers to systems that can generate new content, be it text, images, music, or even videos. Traditionally, AI/ML meant three things: supervised, unsupervised, and reinforcement learning. Each gives insights based on clustering output.

Non-generative AI models make calculations based on input (like classifying an image or translating a sentence). In contrast, generative models produce “new” outputs such as writing essays, composing music, designing graphics, and even creating realistic human faces that don’t exist in the real world.

how is generative AI done?

For generative AI, models are trained to recognize patterns in data and then use these patterns to generate new or similar data .

Applications of generative AI (and their controversies):

Art and design

Natural language processing (NLP)

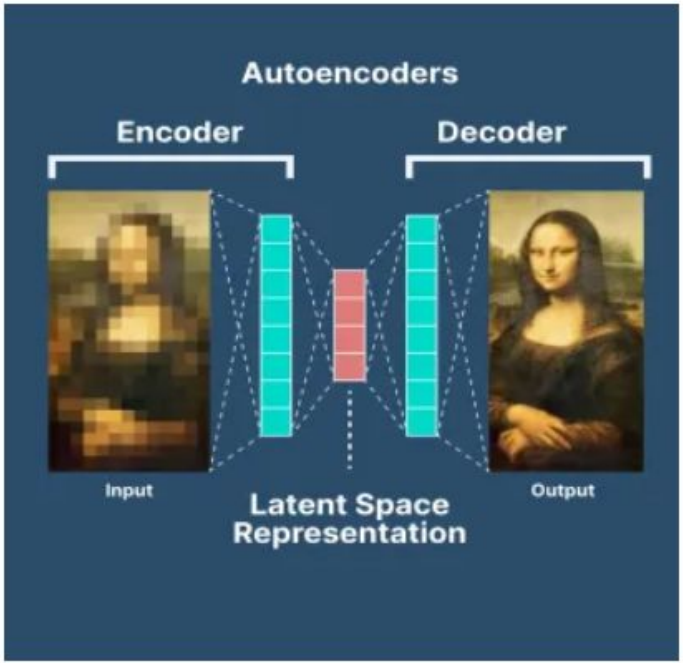
Medicine and drug discovery

Gaming

Marketing and advertising

VAE v/s GAN

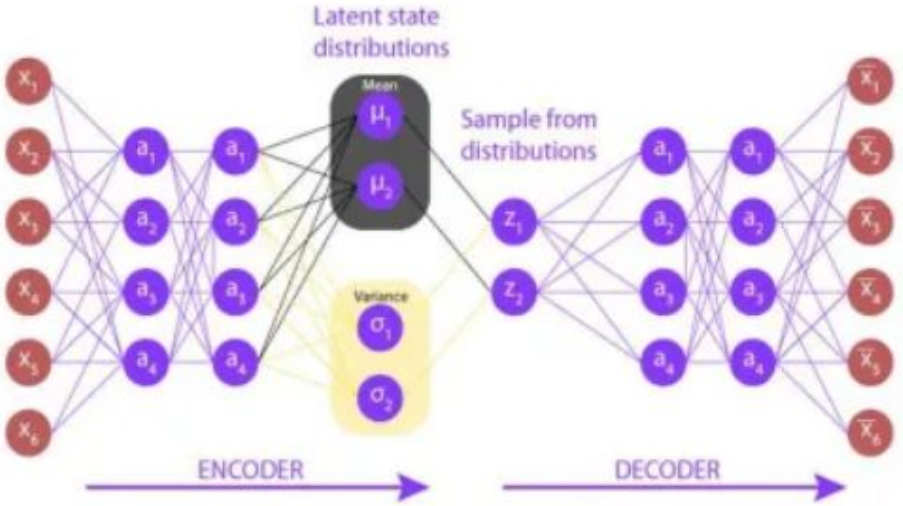
Autoencoders and Variational Autoencoders :



Autoencoders

VAE v/s GAN

Autoencoders and Variational Autoencoders :



Variational Autoencoders

VAE v/s GAN

Autoencoders and Variational Autoencoders :

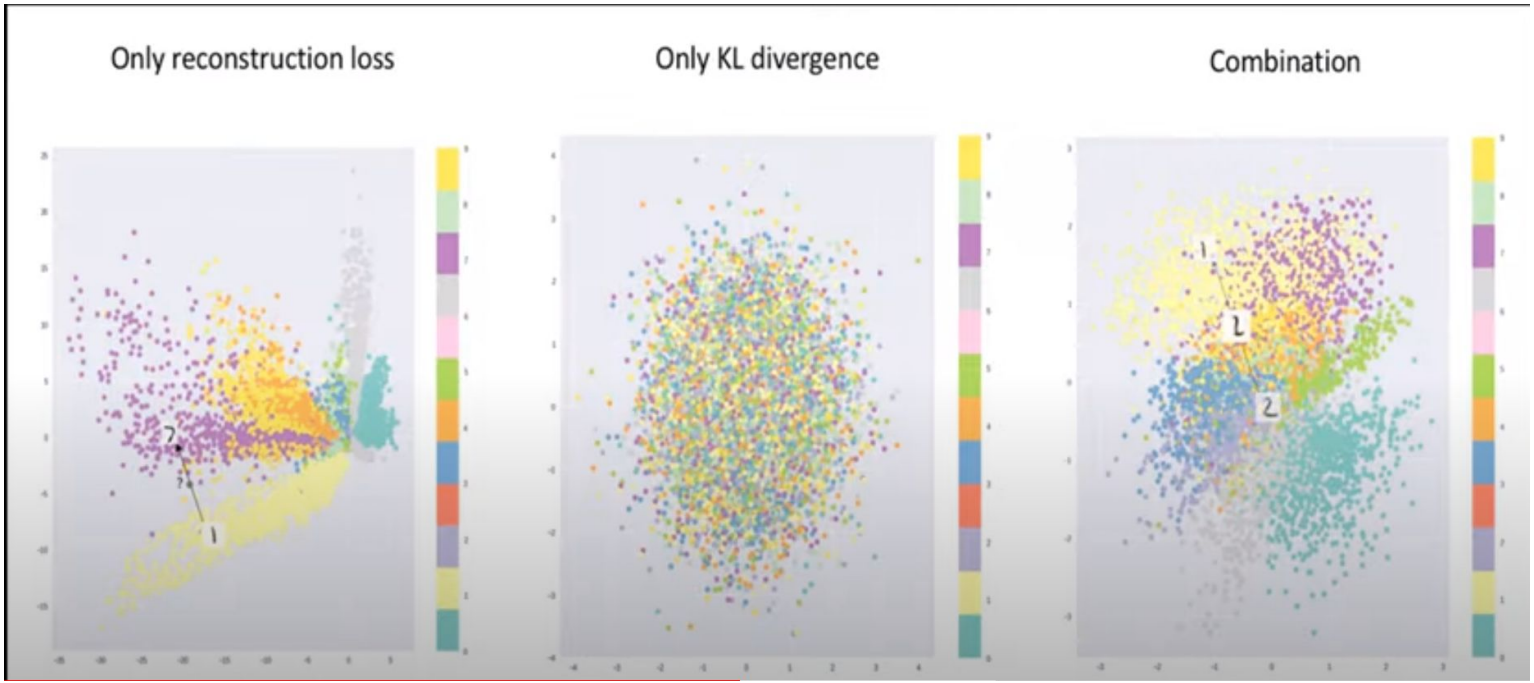
$$\text{VAE Loss} = \text{Reconstruction Loss} + \beta \times \text{KL Divergence}$$

$$ReconstructionLoss = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2$$

$$KLDivergence = -\frac{1}{2}\sum_{j=1}^{J}(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2)$$

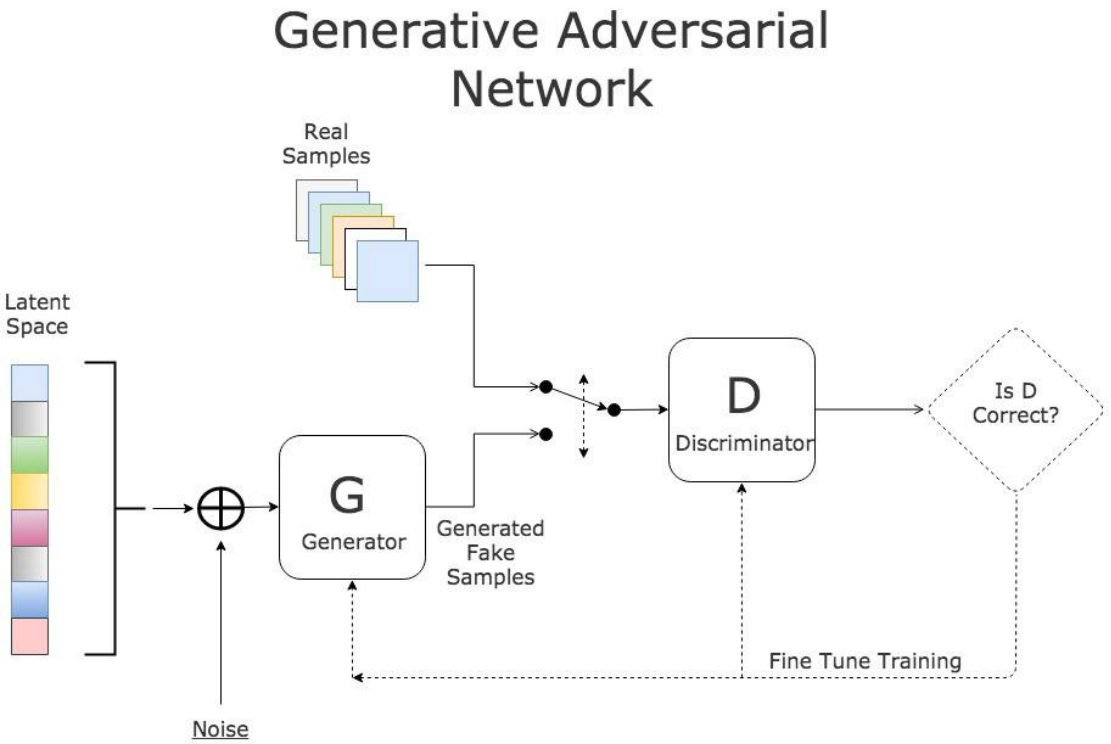
VAE v/s GAN

Autoencoders and Variational Autoencoders :



VAE v/s GAN

Generative Adversarial Networks



VAE v/s GAN

VAE	GAN
In terms of training, VAE has an enforced tradeoff between mixing and power of reconstruction generation. So it's simpler to train.	GAN wants to synchronize the discriminator with the generator during the training to achieve greater results. Therefore, it is more complex to train.
VAE maximizes the probability of generated output with respect to input to get an output by compressing the input to latent space.	GAN finds a point in the generator discriminator to help achieve fake data as one tries to deceive the other.
VAE generates a blurry image compared to GAN as latent vector is generated by encoder.	GAN accomplishes the task to generate non blurry images as latent vectors come from random noise.
Has error function to be minimized - KL divergence and reconstruction error	Has two loss functions - generator's loss and discriminator loss.

Molecular Generation

Motivation :

On average, it takes ten years and costs $2.6 billion dollars to take a drug from the point of understanding the root cause of a disease to its availability in the marketplace. A large portion of this time and effort/cost is because we are literally looking for a needle in a haystack. We are looking for the one molecule that can turn off a disease at the molecular level in a solution space of between 10^{30} to a google (yes, 10^{100}) synthetically feasible molecules. The chemical solution space is too vast to be efficiently screened for the particular molecule of interest. Pharmaceutical compound repositories contain only a fraction of the synthetically feasible molecules for research in a wet lab.

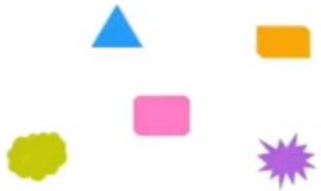
Drug discovery is challenging due to the large number of properties on needs to *simultaneously* optimize.

Molecular properties to optimize

- Binding to a known target
- Target selectivity
- Novelty
- Physico-chemical properties
  - Stability
  - Solubility
- ADMET properties
  - Adsorption
  - Desorption
  - Metabolism
  - Excretion
  - Toxicity
- Synthetic accessibility
- Production cost
- etc

Large number of interdependent properties

Is it the right mechanism of action?



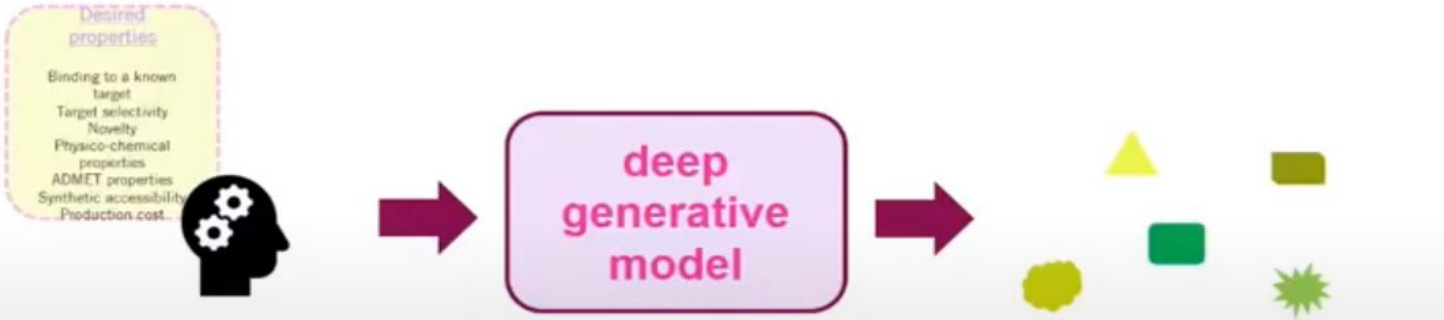
Space of drug-like molecules: $10^{23} - 10^{60}$

The traditional drug discovery process involves…

- Screening large libraries of compounds
- Humans proposing changes to promising molecules
- 3–5 years in initial drug discovery phase

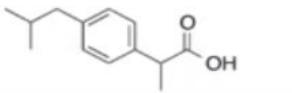
We can use deep generative models…

…to speed up drug discovery by generating molecules in promising areas of chemical space.



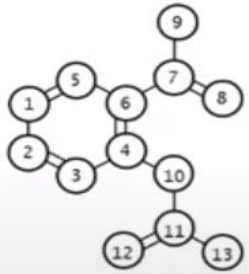
(Two) main classes of molecular generative models:

1. String-based approaches
2. Graph-based approaches
3. 3D approaches



Ibuprofen

string → CC(C)Cc1ccc(cc1)C(C)C(O)=O

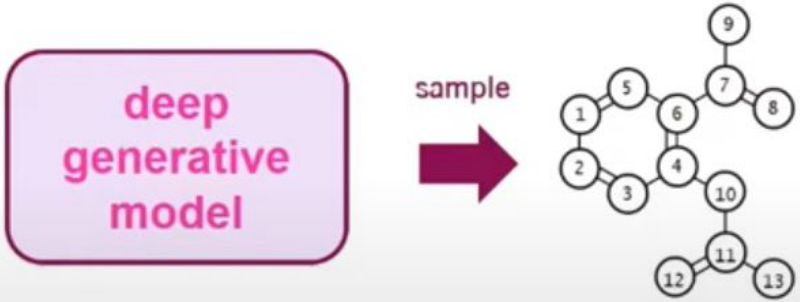


graph

Figure taken from
Segler et al. arXiv:1701.01329. **2017**.

Two main generation schemes:

1. **Single-shot**
2. Iterative



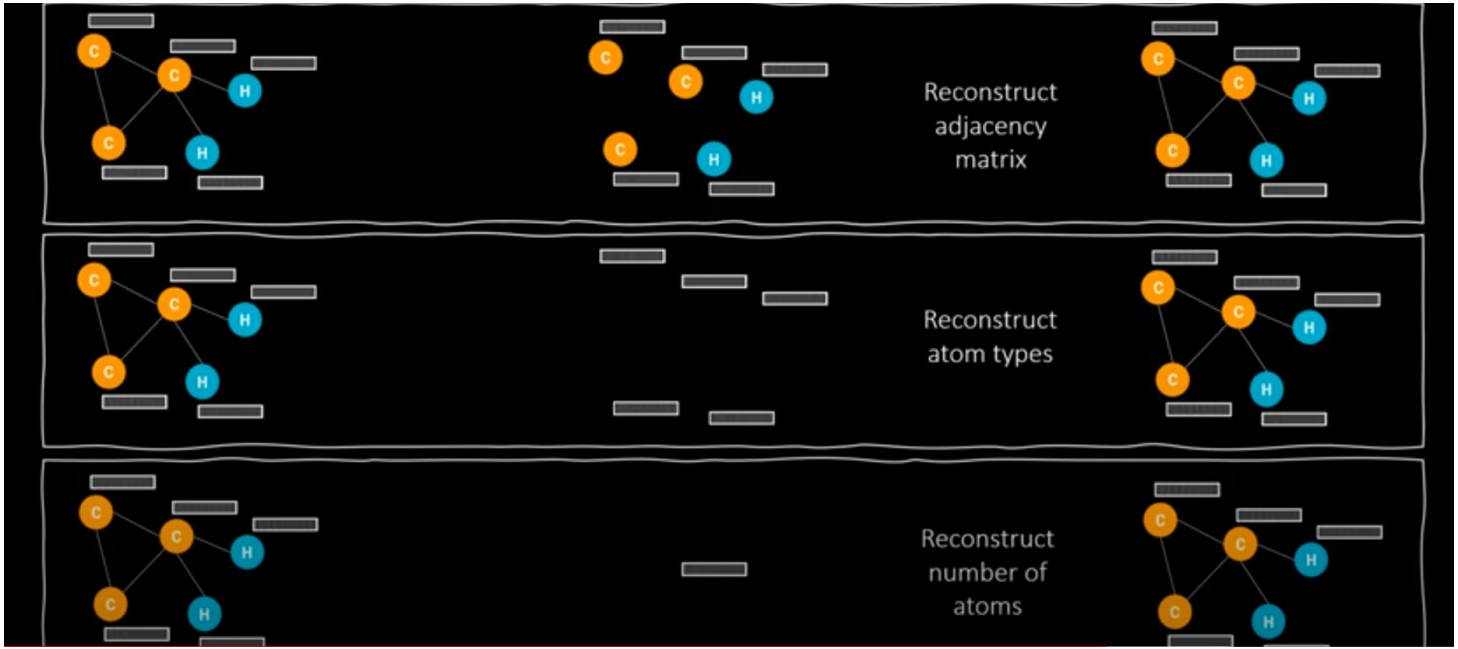
Two main generation schemes:

1. Single-shot
2. **Iterative**

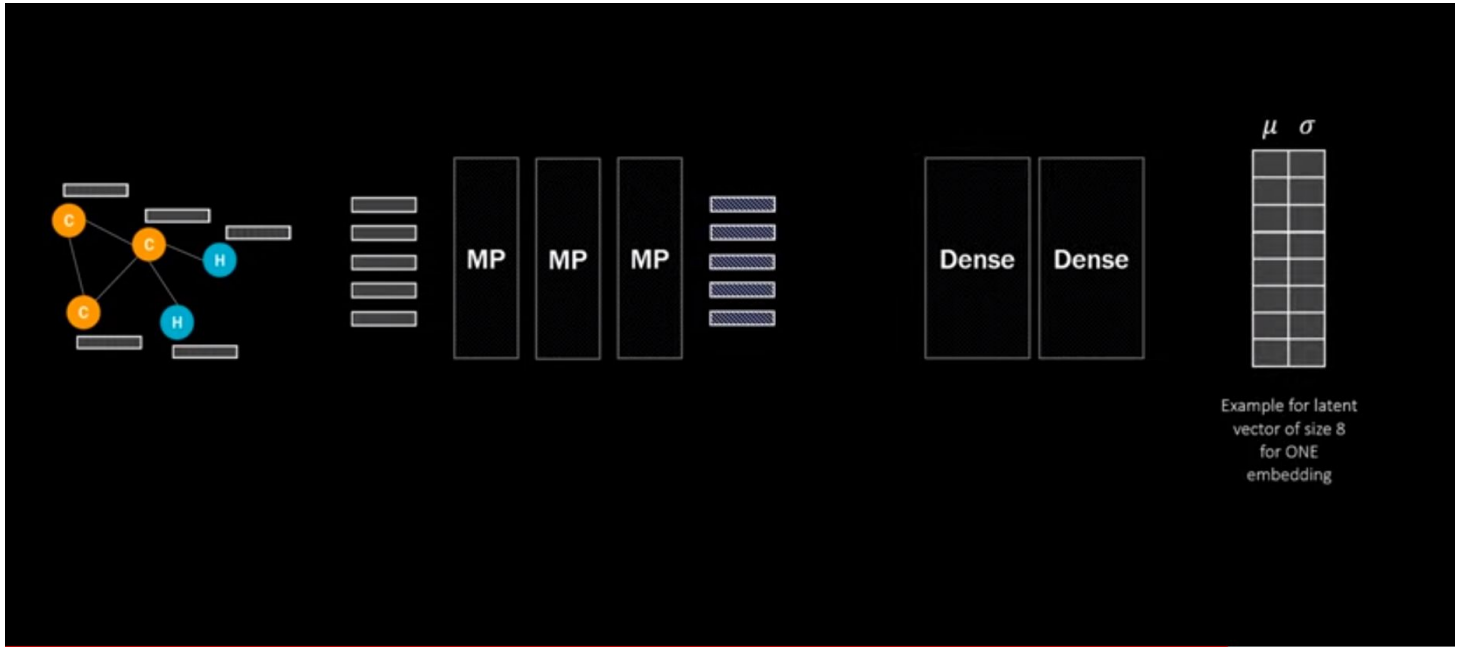
approches chosen by the study

- VEAs and Molecular generation and design approaches
- Graph representation
- one-shot generation scheme

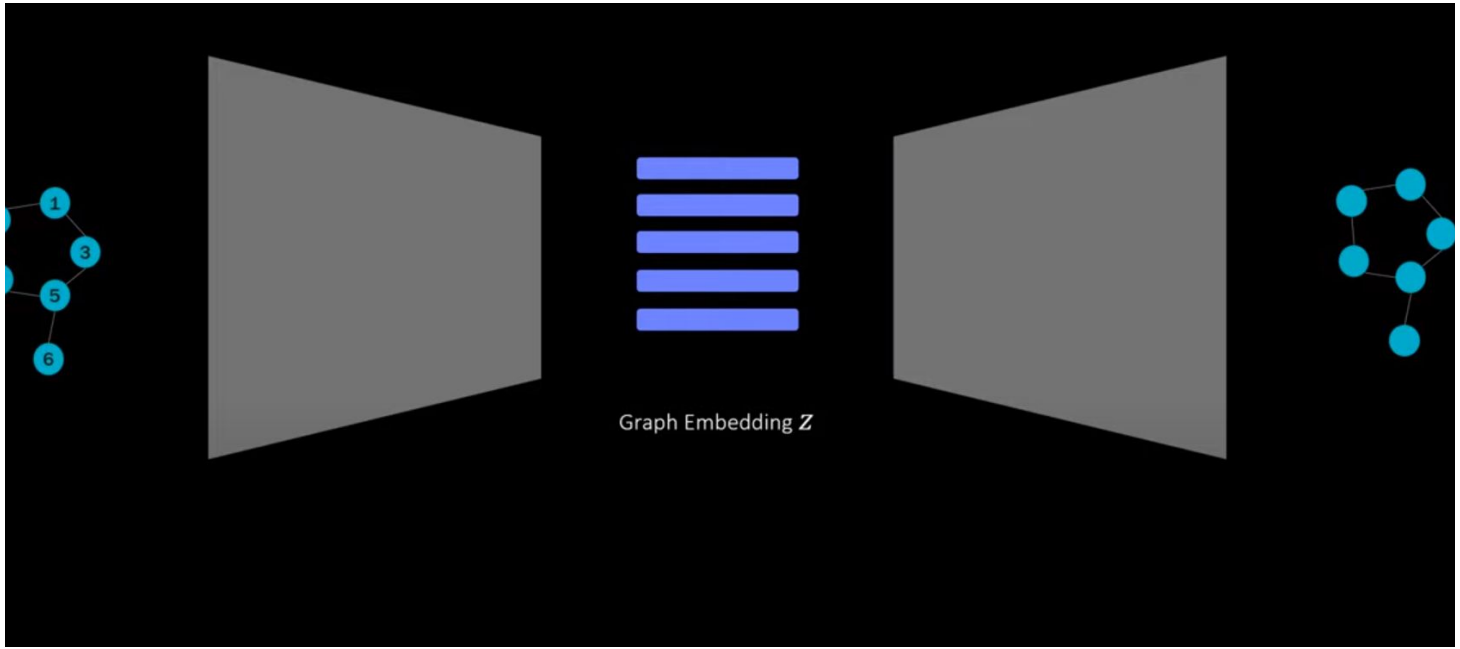
Reconstruction Variation



Node Level laten Vector



Node Level laten Vector



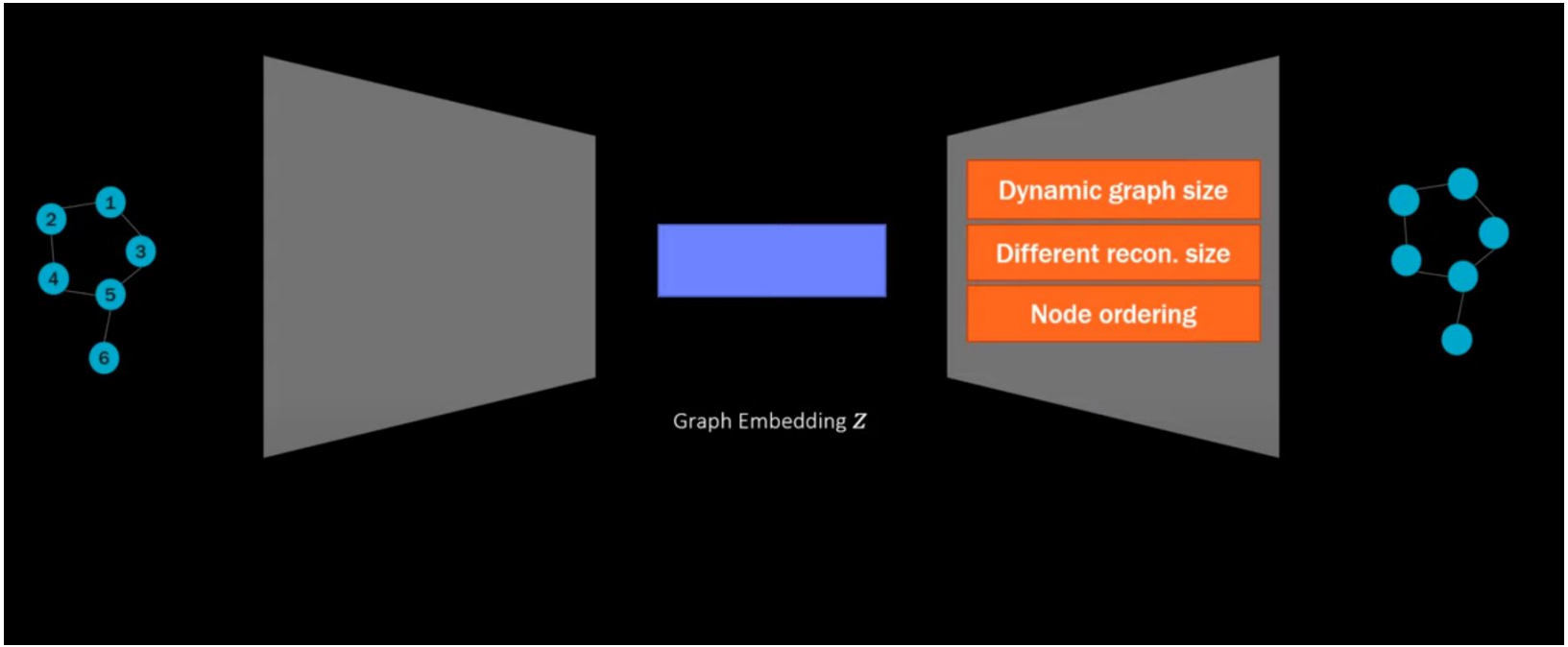
Limits of Node Level laten Vector :

- **Outputs several means and variances for every nodes**
- **Information of molecule distributed**
- **No global information about molecule**

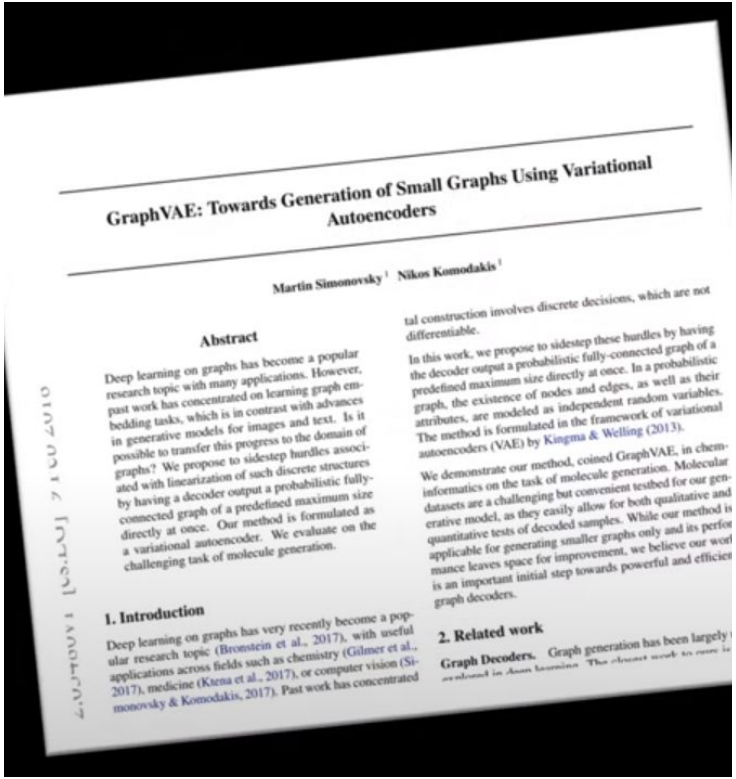
Suggested Solution : Use one laten vector by the encoder and use it for the sampling

challenge : How to map back in the Decoder

Faced Problems :



Paper Discussion



GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders

Martin Simonovsky [1] Nikos Komodakis [1]

Abstract

Deep learning on graphs has become a popular research topic with many applications. However, past work has concentrated on learning graph embedding tasks, which is in contrast with advances in generative models for images and text. Is it possible to transfer this progress to the domain of graphs? We propose to sidestep hurdles associated with linearization of such discrete structures by having a decoder output a probabilistic fully-connected graph of a predefined maximum size directly at once. Our method is formulated as a variational autoencoder. We evaluate on the challenging task of molecule generation.

1. Introduction

Deep learning on graphs has very recently become a popular research topic (Bronstein et al., 2017), with useful applications across fields such as chemistry (Gilmer et al., 2017), medicine (Ksena et al., 2017), or computer vision (Simonovsky & Komodakis, 2017). Past work has concentrated on learning graph embedding tasks, which is in contrast with advances in generative models for images and text. Is it possible to transfer this progress to the domain of graphs? We propose to sidestep hurdles associated with linearization of such discrete structures by having a decoder output a probabilistic fully-connected graph of a predefined maximum size directly at once. Our method is formulated as a variational autoencoder. We evaluate on the challenging task of molecule generation.

tal construction involves discrete decisions, which are not differentiable.

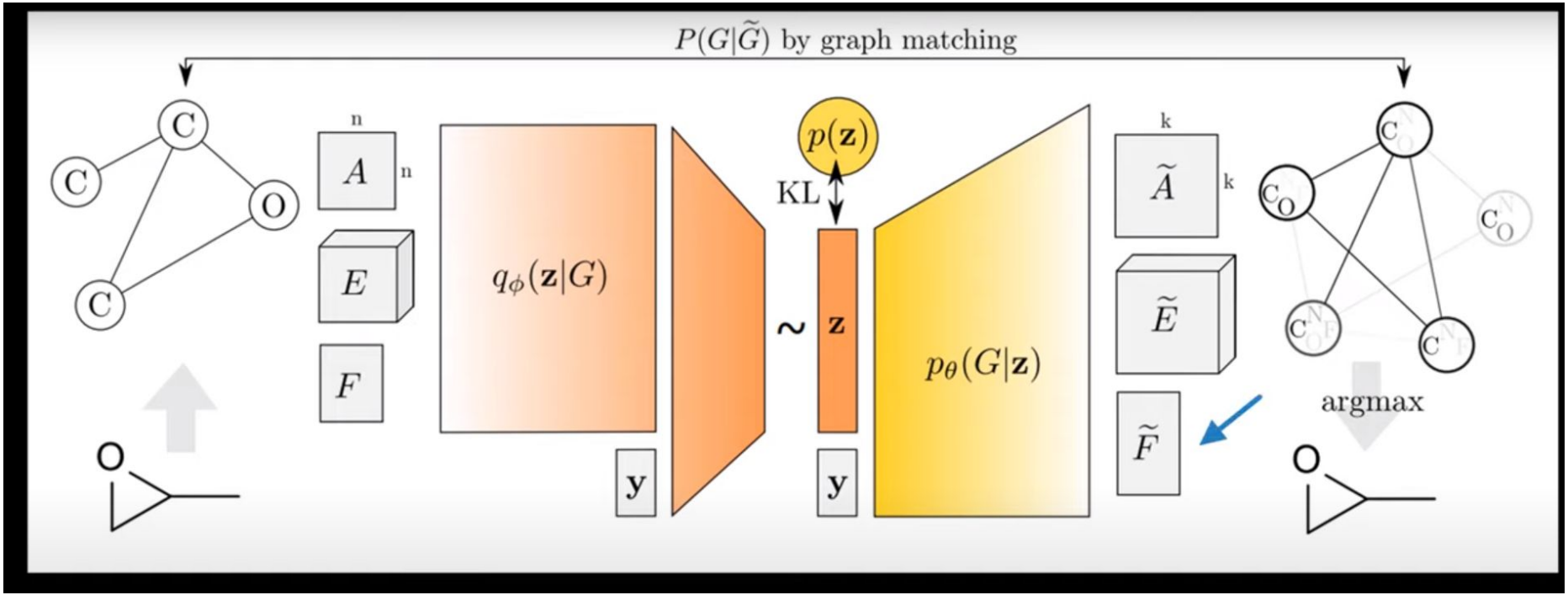
In this work, we propose to sidestep these hurdles by having the decoder output a probabilistic fully-connected graph of a predefined maximum size directly at once. In a probabilistic graph, the existence of nodes and edges, as well as their attributes, are modeled as independent random variables. The method is formulated in the framework of variational autoencoders (VAE) by Kingma & Welling (2013).

We demonstrate our method, coined GraphVAE, in cheminformatics on the task of molecule generation. Molecular datasets are a challenging but convenient testbed for our generative model, as they easily allow for both qualitative and quantitative tests of decoded samples. While our method is applicable for generating smaller graphs only and its performance leaves space for improvement, we believe our work is an important initial step towards powerful and efficient graph decoders.

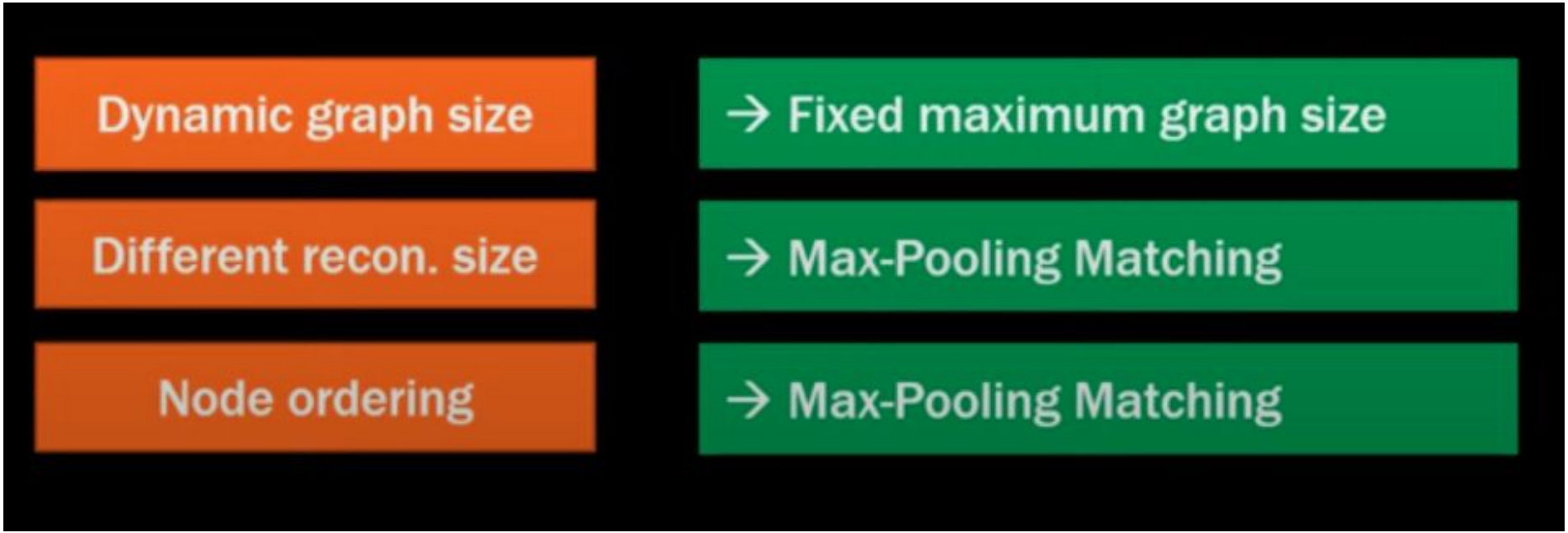
2. Related work

Graph Decoders. Graph generation has been largely

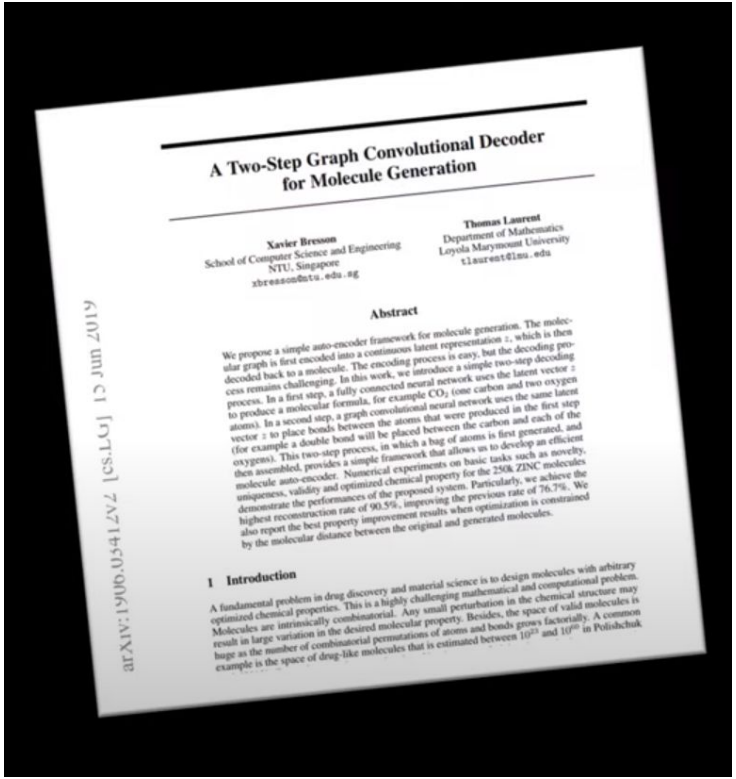
Paper Discussion



Paper Discussion



Paper Discussion



A Two-Step Graph Convolutional Decoder for Molecule Generation

Xavier Bresson
School of Computer Science and Engineering
NTU, Singapore
xbresson@ntu.edu.sg

Thomas Laurent
Department of Mathematics
Loyola Marymount University
tlaurent@lmu.edu

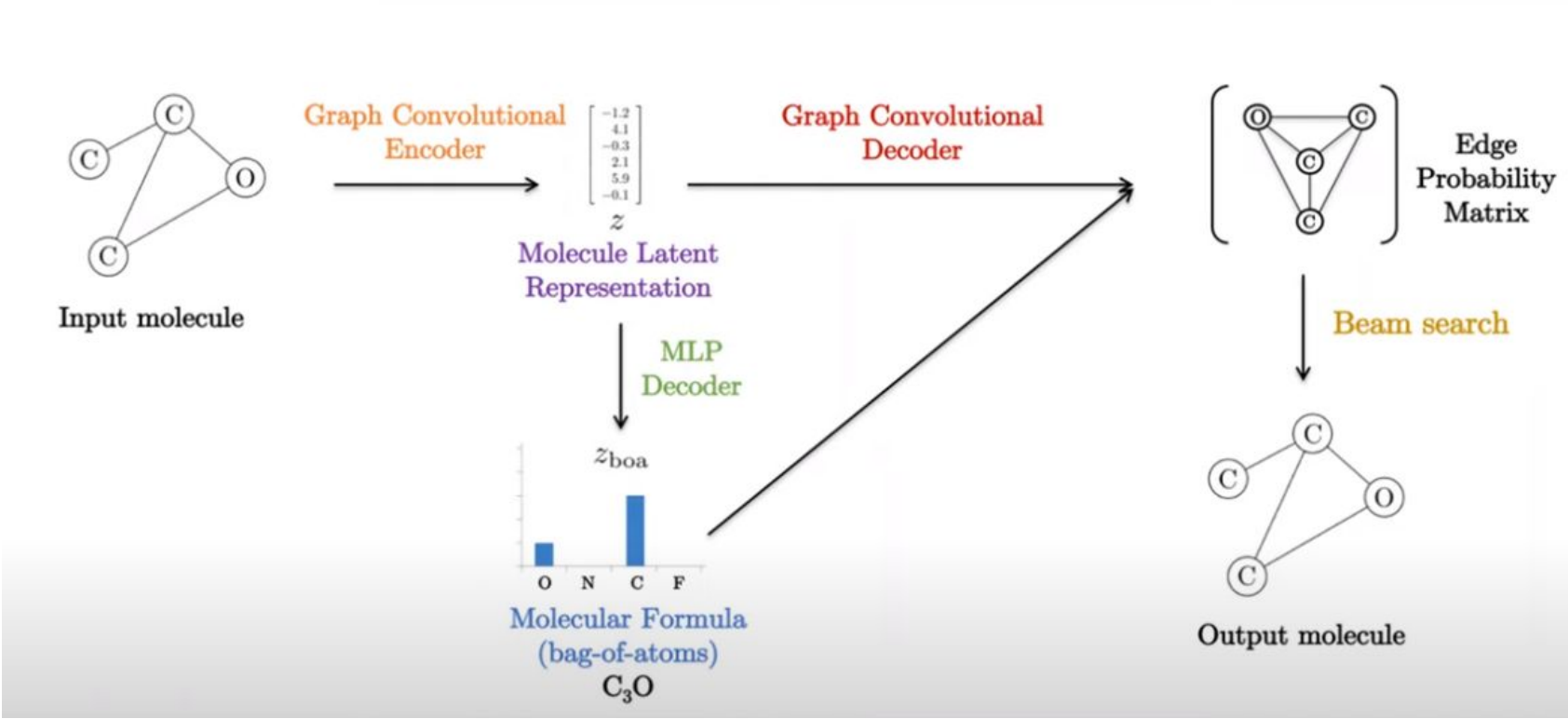
Abstract

We propose a simple auto-encoder framework for molecule generation. The molecular graph is first encoded into a continuous latent representation z, which is then decoded back to a molecule. The encoding process is easy, but the decoding process remains challenging. In this work, we introduce a simple two-step decoding process. In a first step, a fully connected neural network uses the latent vector z to produce a molecular formula, for example CO_2 (one carbon and two oxygen atoms). In a second step, a graph convolutional neural network uses the same latent vector z to place bonds between the atoms that were produced in the first step (for example a double bond will be placed between the carbon and each of the oxygens). This two-step process, in which a bag of atoms is first generated, and then assembled, provides a simple framework that allows us to develop an efficient molecule auto-encoder. Numerical experiments on basic tasks such as novelty, uniqueness, validity and optimized chemical property for the 250k ZINC molecules demonstrate the performances of the proposed system. Particularly, we achieve the highest reconstruction rate of 90.5%, improving the previous rate of 76.7%. We also report the best property improvement results when optimization is constrained by the molecular distance between the original and generated molecules.

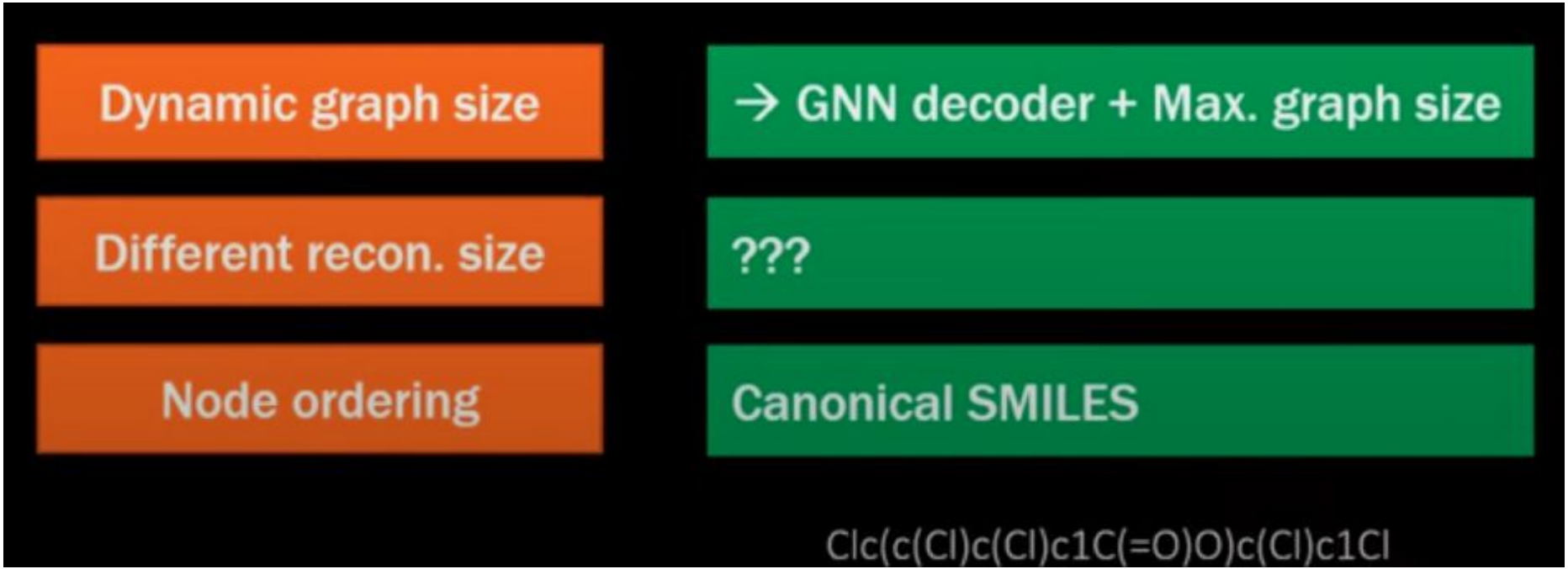
1 Introduction

A fundamental problem in drug discovery and material science is to design molecules with arbitrary optimized chemical properties. This is a highly challenging mathematical and computational problem. Molecules are intrinsically combinatorial. Any small perturbation in the chemical structure may result in large variation in the desired molecular property. Besides, the space of valid molecules is huge as the number of combinatorial permutations of atoms and bonds grows factorially. A common example is the space of drug-like molecules that is estimated between 10^{23} and 10^{60} in Polishchuk

Paper Discussion



Paper Discussion



Paper Discussion

Permutation-Invariant Variational Autoencoder for Graph-Level Representation Learning

Robin Winter [1 2] Frank Noé [2] Djork-Arné Clevert [1]

Abstract

Recently, there has been great success in applying deep neural networks on graph structured data. Most work, however, focuses on either node- or graph-level supervised learning, such as node, link or graph classification or node-level unsupervised learning (e.g. node clustering). Despite its wide range of possible applications, graph-level unsupervised learning has not received much attention yet. This might be mainly attributed to the high representation complexity of graphs, which can be represented by $n!$ equivalent adjacency matrices, where n is the number of nodes. In this work we address this issue by proposing a permutation-invariant variational autoencoder for graph structured data. Our proposed model indirectly learns to match the node ordering of input and output graph, without imposing a particular node ordering or performing expensive graph matching. We demonstrate the effectiveness of our proposed model on various graph reconstruction and generation tasks and evaluate the expressive power of extracted representations for downstream graph-level classification and regression.

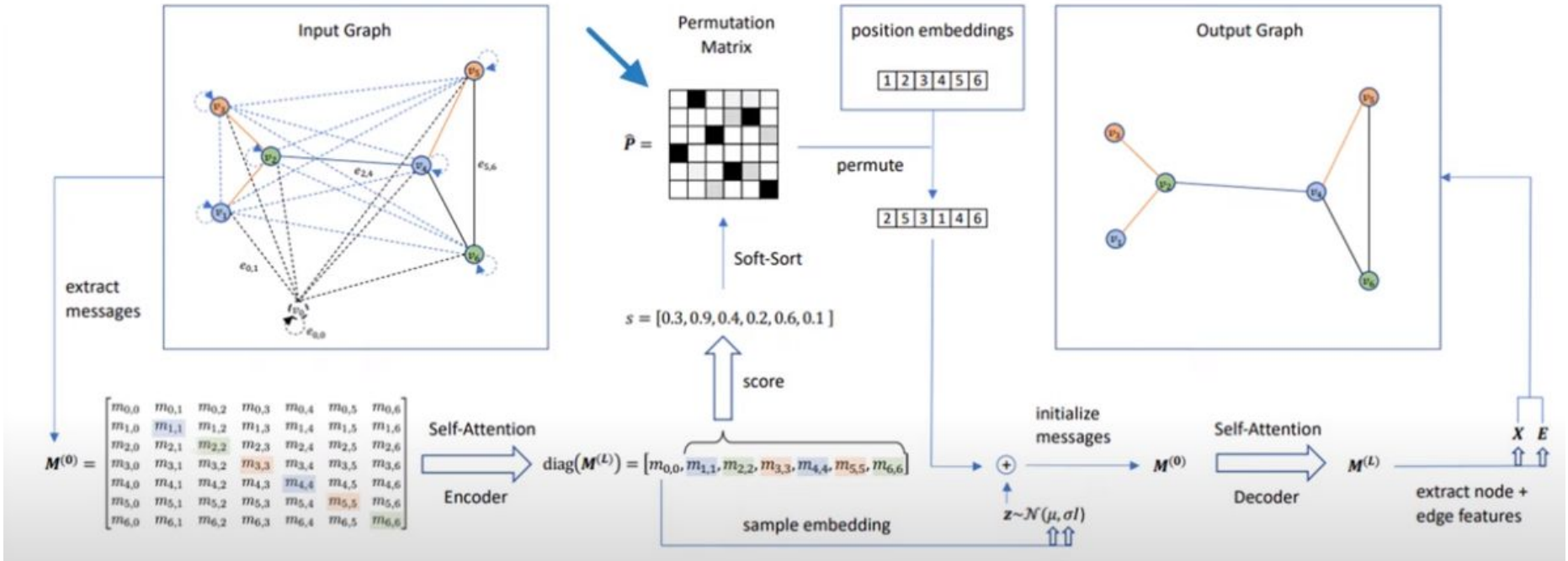
1. Introduction

Graphs are an universal data structure that can be used to describe a vast variety of systems from social networks to quantum mechanics (Hamilton et al., 2017). Driven by the success of Deep Learning in fields such as computer vision and natural language processing, there has been an increasing interest in applying deep neural networks on non-euclidean, graph structured data as well (Bronstein et al., 2017; Wu et al., 2020). Most notably, generalizing Convolutional Neural Networks and Recurrent Neural Networks to

to significant advances on task such as molecular property prediction (Duvenaud et al., 2015) or question-answering (Li et al., 2015). Research on unsupervised learning on graphs mainly focused on node-level representation learning, which aims at embedding the local graph structure into latent node representations (Cao et al., 2016; Wang et al., 2016; Kipf & Welling, 2016; Qiu et al., 2018; Pan et al., 2018). Usually, this is achieved by adopting an autoencoder framework where the encoder utilizes e.g. graph convolutional layers to aggregate local information at a node level and the decoder is used to reconstruct the graph structure from the node embeddings. Graph-level representations are usually extracted by aggregating node-level features into a single vector, which is common practice in supervised learning on graph-level labels (e.g. molecular property prediction) (Duvenaud et al., 2015). Unsupervised learning of graph-level representations, however, has not yet received much attention, despite its wide range of possible applications, such as feature extraction or pre-training for graph-level classification/regression tasks and graph matching or similarity ranking. A possible reason for this might be the inherent invariance of graphs with respect to the ordering of nodes within the graph. In general, a graph with n nodes, can be represented by $n!$ equivalent adjacency matrices, each corresponding to a different node ordering. Since the general structure of a graph is invariant to the order of their individual nodes, a graph-level representation should also not depend on the order of the nodes in the input representation of a graph. This poses a problem for most neural network architectures which are by design not invariant to the order of their inputs. Even if carefully designed in a permutation invariant way, it is not straight-forward to train e.g. an autoencoder, due to the ambiguous reconstruction objective, requiring the same discrete ordering of input and output graphs for comparison.

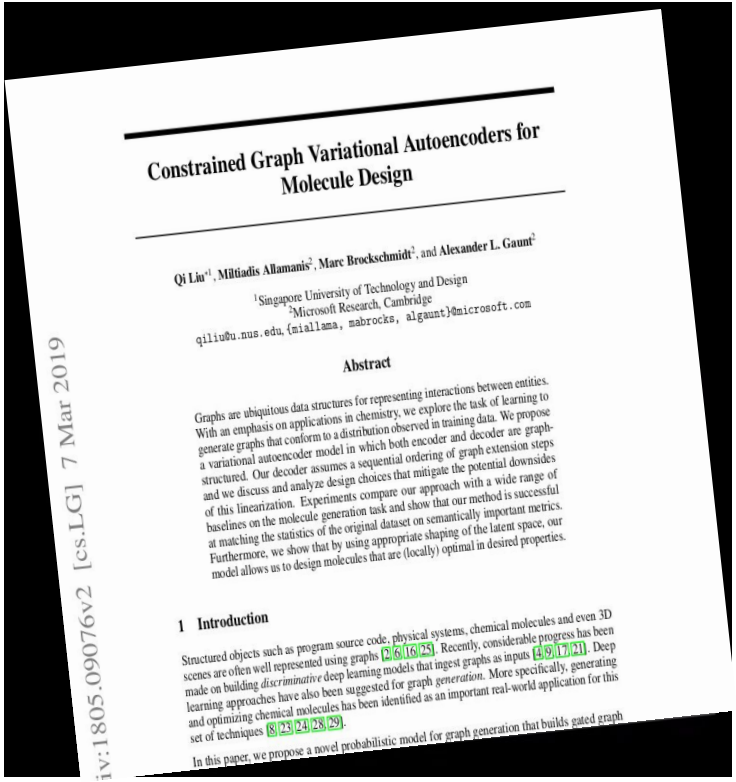
In this work we propose a graph autoencoder architecture that is by design invariant to the ordering of the nodes in

Paper Discussion



Paper Discussion

Paper Discussion



Constrained Graph Variational Autoencoders for Molecule Design

Qi Liu[*1], Miltiadis Allamanis[2], Marc Brockschmidt[2], and Alexander L. Gaunt[2]

[1]Singapore University of Technology and Design
[2]Microsoft Research, Cambridge
qiliu@u.nus.edu, {miallama, mabrocks, algaunt}@microsoft.com

Abstract

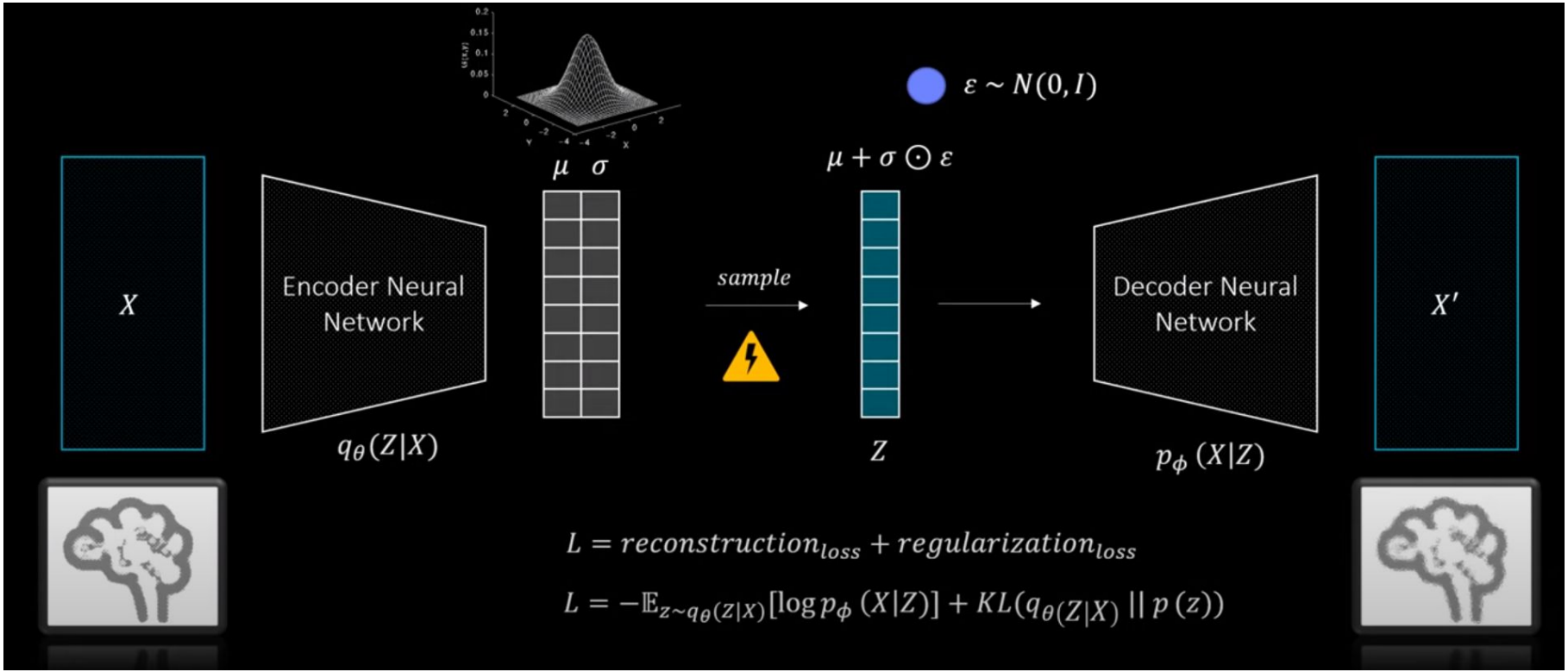
Graphs are ubiquitous data structures for representing interactions between entities. With an emphasis on applications in chemistry, we explore the task of learning to generate graphs that conform to a distribution observed in training data. We propose a variational autoencoder model in which both encoder and decoder are graph-structured. Our decoder assumes a sequential ordering of graph extension steps and we discuss and analyze design choices that mitigate the potential downsides of this linearization. Experiments compare our approach with a wide range of baselines on the molecule generation task and show that our method is successful at matching the statistics of the original dataset on semantically important metrics. Furthermore, we show that by using appropriate shaping of the latent space, our model allows us to design molecules that are (locally) optimal in desired properties.

1 Introduction

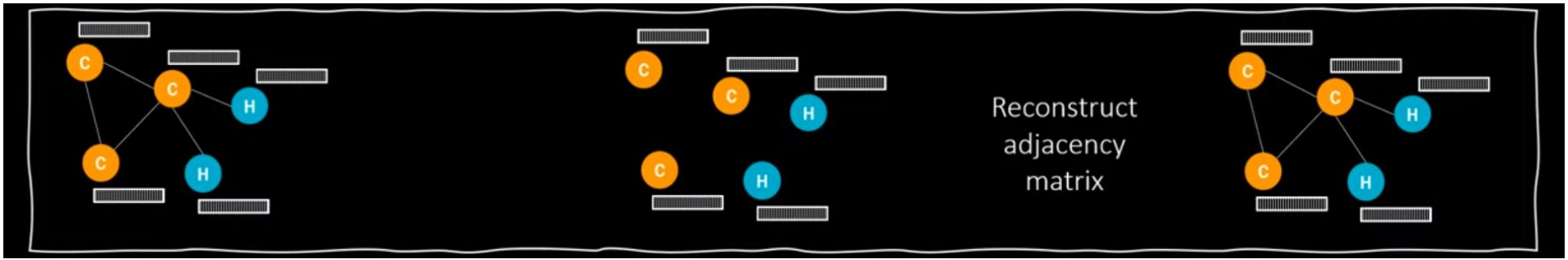
Structured objects such as program source code, physical systems, chemical molecules and even 3D scenes are often well represented using graphs [2, 6, 16, 25]. Recently, considerable progress has been made on building *discriminative* deep learning models that ingest graphs as inputs [4, 9, 17, 21]. Deep learning approaches have also been suggested for graph *generation*. More specifically, generating and optimizing chemical molecules has been identified as an important real-world application for this set of techniques [8, 23, 24, 28, 29].

In this paper, we propose a novel probabilistic model for graph generation that builds gated graph

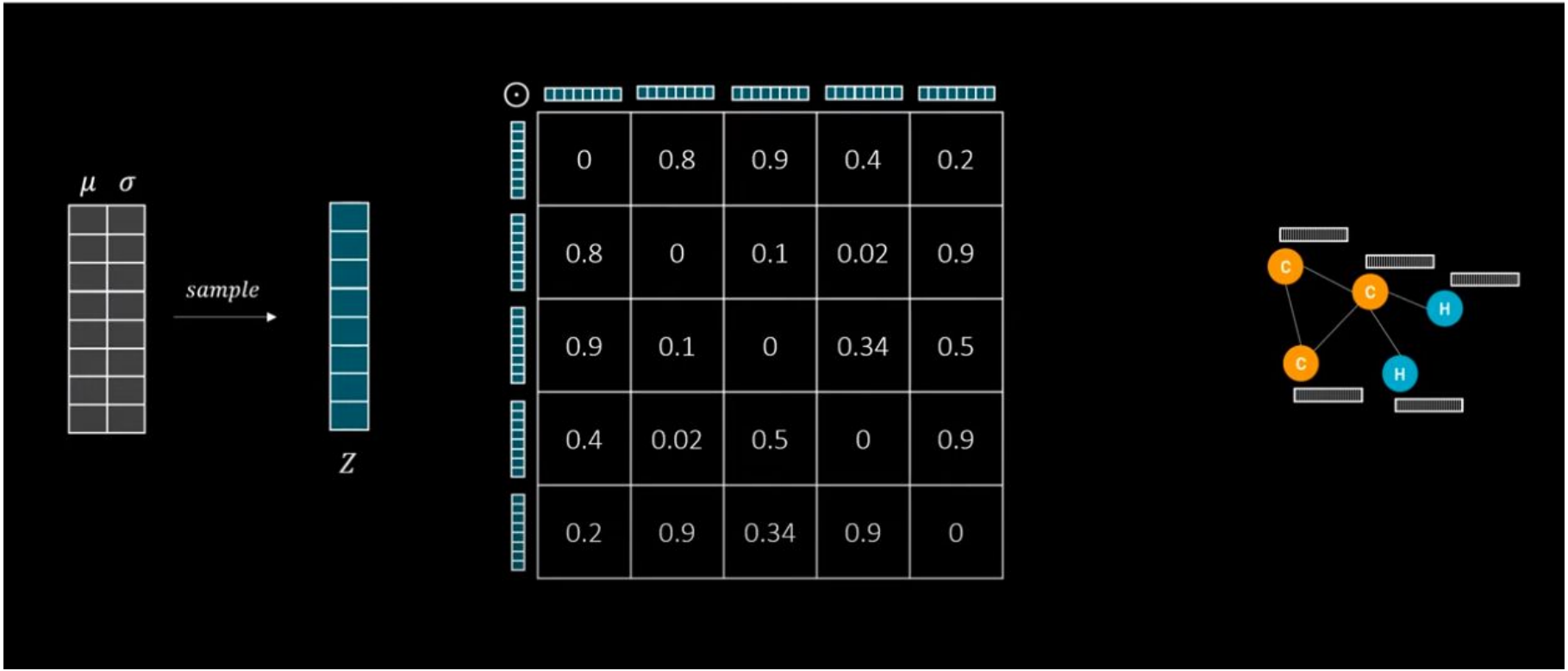
Paper Discussion



Paper Discussion



Paper Discussion



Implement the Graph VAE

Task to be Done :

- The reconstructed part is the Adjacency Matrix
- Use the reparametrization Tric
- Use GNN Instead of MLP in Decoder
- Use Different GNN (GCN and GAT) with VAE
- Compare and Plot results

Code Review

Conclusion

In conclusion, GraphVAEs mark a pivotal development in generative models for graph-structured data. They adeptly merge Variational Autoencoder principles with the complexities of graph data, enabling efficient learning and generation of diverse graph structures. This innovation holds significant promise for fields reliant on graph analysis, such as molecular design and social network analysis. As the technology progresses, GraphVAEs are poised to unlock new frontiers in graph-based machine learning and data science.