A PROJECT REPORT

ON

"Comparing VAE and GAN Models for Molecular SMILES Generation and Property Prediction with GNNs"

A project report submitted in partial fulfillment of the requirements for the degree of Bachelor of Technology in Artificial Intelligence & Machine Learning

BACHELOR OF TECHNOLOGY IN ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Submitted By

Arjun Tyagi	21070126020
Kaustubh Rayker	21070126048
Nachiket Ropia	21070126056

UNDER THE GUIDANCE OF

Dr.Shruti Patil
Head of Department, AIML
Symbiosis Institute of Technology

&

Dr.Sunil Jaiswal
Scientist 'F'
HEMRL,DRDO,Pune

॥वसुधैव कुटुम्बकम्॥

SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Pune – 412115, Maharashtra State, India
https://www.sitpune.edu.in/

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

AY 2024-25

SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Pune – 412115, Maharashtra State, India
https://www.sitpune.edu.in/

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

CERTIFICATE

This is to certify that the Project work entitled "**Comparing VAE and GAN Models for Molecular SMILES Generation and Property Prediction with GNNs**" is carried out by the **Arjun Tyagi, Kaustubh Raykar,Nachiket Ropia,** in partial fulfillment for the award of the degree of **Bachelor of Technology** in **Artificial Intelligence & Machine Learning**, Symbiosis International (Deemed University), Pune during the academic year 2024-2025.

Name and Signature of the Guide	Name and Signature of the Industry Guide	Dr. Shruti Patil

DECLARATION

I hereby declare that the project titled "**Comparing VAE and GAN Models for Molecular SMILES Generation and Property Prediction with GNNs**" submitted to Symbiosis Institute of Technology, Constituent of Symbiosis International (Deemed University) Pune for the award of the degree of Bachelor of Technology in Artificial Intelligence & Machine Learning is a result of original research carried out by me. I understand that my report may be made electronically available to the public. It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of any degree or diploma.

Names of Students : Arjun Tyagi, Kaustubh Raykar, Nachiket Ropia

PRN: 21070126020, 21070126048, 21070126056

Degree: Bachelor of Technology in AI&ML

Department: Artificial Intelligence & Machine Learning

Title of the project : Comparing VAE and GAN Models for Molecular SMILES Generation and Property Prediction with GNNs

_____

(Signatures of the Students)

Date: 16/11 /2024

ACKNOWLEDGEMENT

I want to express my sincere gratitude to everyone who supported me throughout this project. First and foremost, I would like to thank my project guides, **Dr. Shruti Patil and Dr. Sunil Jaiswal**, for his/her valuable guidance, encouragement, and feedback. He/She has been a constant source of inspiration and motivation for me.

I would also like to thank the head of the department, **Dr. Shruti Patil**, for providing me with the necessary facilities and resources for conducting this project. I am grateful to her for providing constant support and advice.

I want to acknowledge the contribution of my project team members, **Arjun Tyagi, Kaustubh Raykar, Nachiket Ropia**, who have worked hard and cooperated with me in every project stage. They have been accommodating and supportive throughout this journey.

I would also like to thank my family and friends for their love, care, and support. They have always been there for me in times of need and stress. They have encouraged me to pursue my passion and achieve my goals.

Lastly, I thank **Symbiosis Institute of Technology Pune** for allowing me to work on this project and enhance my skills and knowledge. I am proud to be a part of this prestigious institution.

ABSTRACT

New molecules with required properties have been designed and synthesized for emerging fields such as pharmaceuticals, materials science, and chemical engineering. Traditionally, molecular design and property prediction approaches depend heavily on experimental synthesis and testing as a way of achieving the final product, which are expensive and time consuming. Computational approaches have emerged as efficient alternatives that enable the generation and evaluation of molecular structures in silico.

The last few years have seen big changes in the approach to molecular generation and property prediction by deep learning techniques. Generative models like VAEs and GANs have been highly impressive at generating new molecular structures expressed as SMILES strings. They learn the underlying distribution of chemical structures within a dataset and can be used to generate novel molecules consistent with the rules of chemistry learned from the data.

Concurrently, Graph Neural Networks have emerged because they can predict molecular properties with powerful utilization of the inherent graph structure in molecules. GNNs differ from a more typical neural network, by being able to work directly on graphs and thus to be more suitable for processing molecular data where atoms are represented as nodes and bonds as edges.

The study is focused on a comprehensive comparative analysis of VAEs and GANs for the generation of molecular SMILES, measuring their effectiveness in conjunction with GNNs for predicting properties. Generative models are integrated with predictive models, incorporating the basic idea of building a strong framework for efficient design and evaluation of molecules, thereby accelerating the discovery process in chemistry and related disciplines.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Introduction

The discovery and development of novel molecules with desired properties form the foundation for advancements in pharmaceutical, materials, and chemical-engineering innovations. Traditional molecular design and property prediction have depended essentially on experimental synthesis and testing-processes often long, laborious, and expensive in time. However, new computational methodologies have provided powerful in silico alternatives which offer unprecedented opportunities for the rapid and efficient generation and evaluation of molecular structures.

More recently, tools of deep learning have emerged as revolutionary technologies in this field, fundamentally changing the way researchers work on the frontier of molecular generation and prediction of properties. Generative models, including Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs), have shown impressive potential in creating new molecular structures that are often represented as SMILES (Simplified Molecular Input Line Entry System) strings. These models can learn to capture subtle underlying distributions of chemical structures in the large datasets used to create new molecules that are consistent with these established rules of chemicals and can even discover novel properties.

A new basis of molecular property prediction, the molecule graph description of atoms as nodes and bonds as edges has revolutionized the field, enabling direct processing of the molecular graph in such a way that relationships and dependencies governing chemical behavior are

captured, thereby making the GNNs a powerful tool for the precise prediction of not only reactivity but also biological activity and much more.

In this regard, this study attempts to identify synergies between these generative and predictive models. For instance, this study performs a comprehensive comparative study of VAEs and GANs on the molecular SMILES generation task by identifying relative strengths and limitations. Further, it explores how these generative modes can be combined with GNN-based property prediction models for designing and evaluating molecules within an integral framework. The research combines the strengths of these methodologies to speed up the discovery process while reducing dependence on trial-and-error experimentation and, thereby, contributes to innovation in chemistry as well as its related disciplines.

We aim to develop a robust, efficient, and scalable framework for molecular discovery of newer chemical structures with the help of recent advances in deep learning and to address the challenges associated with modern molecular design.

1.2 Problem Statement

The central challenge addressed in this research is the efficient and accurate generation of novel molecular structures with desired properties. Although VAEs and GANs alone have been successful for molecular generation, a systematic comparison of their ability to produce valid, unique, novel, and diverse sets of molecules is still an open task. Furthermore, although generative models can feed the prediction models, like GNNs, there is an open question as to how the output of these generative models affects the performance of property prediction models, like GNNs.

Challenges can be stated as following:

- Generation Challenge: Report and discuss which of VAE or GAN better produces valid, unique, and novel SMILES strings of molecules reproducing chemical diversity of training data.

- Prediction Challenge: Explore the differences in generated molecules by VAEs and GANs how such a difference might affect how reliable and accurate molecular property predictions are that are made using GNNs.
- Integrative Challenge: Develop a framework that will bring the goodness of generative models and GNNs together for the easier design of molecules

1.3 Scope of research

The research surrounds the following important areas:

- **Data Utilization:** Based on the QM9 dataset-the benchmark 134,000 small organic molecules with all sorts of computed properties.
- **Model Implementation:**
    - Use VAE and GAN for generating the molecular SMILES strings.
    - Of these, two GNN architectures will be investigated for molecular property prediction, including **Graph Convolutional Networks(GCNs)** and **Graph Isomorphism Networks(GINs)**.
- **Comparative Analysis:**Compare the performance of VAEs and GANs in terms of validity, uniqueness, novelty, FCD, Tanimoto similarity, and internal diversity by using quantitative measures.Assess the performance of the generated molecules of each model in property prediction tasks employing GNNs.
- **Methodological Integration:** Design a workflow that combines molecule generation and property prediction to allow end-to-end molecule design and evaluation.

1.4 Research Hypothesis

The following hypotheses direct the research:

1. **Generative Performance Hypothesis:**
    - *H1:* The VAE model is likely to provide molecules with higher rates of validity, uniqueness, and novelty compared to the GAN model because the latent space of VAE is organized, and regularization terms are included during training.
    - *H2:* The properties of the molecules possibly produced by the GAN model may be closer to the mean of the training data set; however, mode collapse may also reduce its diversity

2. Property Prediction Hypothesis:

- *H3:* GNNs, in particular GINs, would prove to be effective in producing accurate properties for molecules both by VAEs and GANs. Nonetheless, the quality of predictions would rely on the generated diversity and representativeness of the molecules.
- *H4:* The molecules generated by the VAE is expected to yield better property prediction performance because it is closer aligned with the training data distribution.

1.5 Objectives of the Project

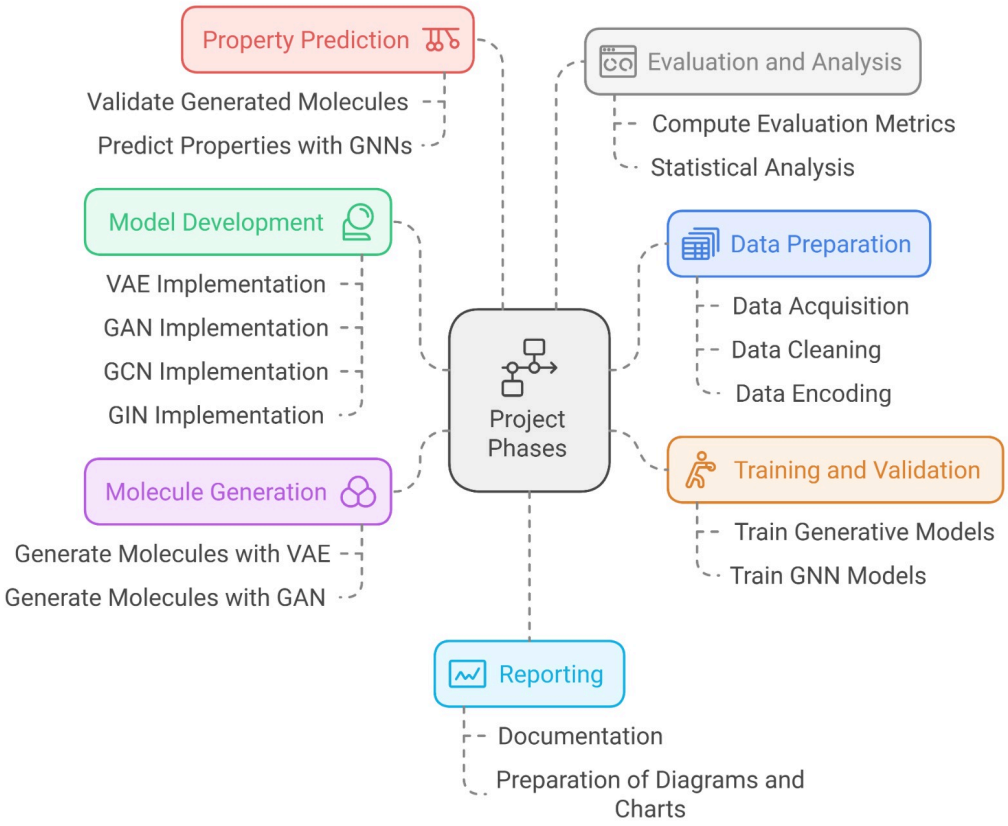

Fig1.1 Project Phases

The primary objectives of this research are:

1. **Implementation and Training:**
    - Develop and train VAE and GAN models on the QM9 dataset for generating molecules.
    - Implement GCN and GIN models to predict molecular properties like dipole moment.
2. **Evaluation and Comparison:**
    - Evaluate the performance of the two approaches-VAEs and GANs-based on the validity, uniqueness, novelty, FCD, Tanimoto similarity, and internal diversity.
    - Evaluate the predictive performances of GCN and GIN models for training data and sampled molecules from VAEs and GANs.

3. **Analysis of Integration:**
    - Discuss impact of choice of generative model on downstream property prediction with GNNs.
    - Identify which generative model with GNNs would lead to a more uniform and robust framework in molecular design.
4. **Recommendations:**
    - Discuss the strengths and weaknesses of VAEs and GANs in molecular generation contexts.
    - Make recommendations for how generative models should be combined with property prediction networks in computational chemistry.

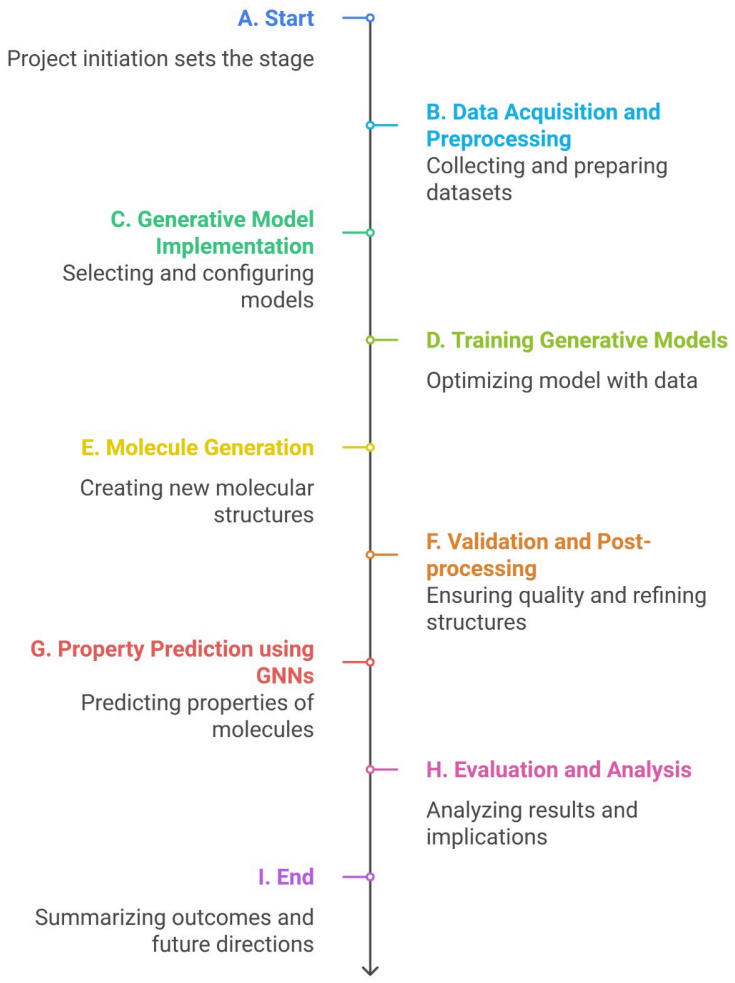

Fig1.2 Project Pipeline

1.6 Organization of the report

The report is structured to provide a comprehensive account of the research conducted, divided into the following chapters:

- **Chapter 1: Introduction**
    - Introduction, Background, Problem Statement, Scope, Hypothesis, Objectives, and Report Organization
- **Chapter 2: Literature Review**
    - Re-evaluation of existing literature concerning molecular generative models as well as GNNs for property prediction.
    - Identification of gaps that this research is going to cover.
- **Chapter 3: Software Requirements Specification**
    - A detailed account of the software tools, frameworks, and hardware used.
    - Underlines the work breakdown structure, functional and non-functional requirements, and project cost estimation.
- **Chapter 4: Methodology**
    - Data preparation, model architectures, training procedures, and evaluation strategies.
    - Integrate the generative models and GNNs.
- **Chapter 5: Results and Discussion**
    - Experimental results with relevant quantitative metrics and prediction outcomes of the property.
    - Discussion of findings by the comparison of VAEs and GANs.
- **Chapter 6: Conclusion and Future Scope**
    - The paper summarizes the main conclusions drawn from the research, but suggests some potential directions for future work in developing them further.

CHAPTER 2: LITERATURE REVIEW

2.1 Background

Advances at the intersection of machine learning and chemistry have recently given rise to the powerful computational techniques for the generation and property prediction of molecules. Most techniques of traditional computational chemistry, such as quantum mechanical simulations and molecular dynamics, tend to be expensive and scale badly with the size of a molecule. Thus, an attraction has grown immensely towards allowing deep learning models to accelerate the discovery and design of novel molecules with the desired properties.

2.1.1 Molecular Generative Models

Variational Autoencoders (VAEs):

VAEs are generative models that learn a probabilistic latent space of input data. Introduced first by Kingma and Welling(2013), this model in the context of molecular generation implies taking molecular representations- say, SMILES strings-and encoding them to a continuous latent space and then decoding back to the original molecules. This latent representation affords smooth interpolation and exploration of novel molecules through sampling and decoding latent vectors.

Gómez-Bombarelli et al. (2018) demonstrated very promising results using VAEs in the generation of molecules. This was achieved by generating novel, drug-like molecules using VAEs, which demonstrated that the model is able to uncover the latent chemical space and generate valid molecular structures.

Generative Adversarial Networks (GANs):

GANs, as proposed by Goodfellow et al. (2014), consist of two neural networks: namely the generator and the discriminator. These are trained simultaneously through an adversarial process. One network, the generator, tries to create data indistinguishable from real data, whereas the other network, the discriminator, aims to differentiate between real and generated data.

In molecular generation, GANs have been adapted for generating molecular graphs or sequences. As indicated earlier, MolGAN by De Cao and Kipf, 2018 is one such example where the authors directly used the application of GANs on small molecular graph generation with an utmost aim at the direct generation of valid novel molecules. However, GANs suffer from challenges such as mode collapse, whereby the generator offers limited varieties of data, and this affects the diversity of molecules generated.

2.1.2 Molecular Representation with SMILES

A common notation SMILES (Simplified Molecular Input Line Entry System) encodes the structure of molecules into a linear string of symbols. Resulting SMILES strings are easy to process for computational models since they are like natural languages. It is, however, not immediately clear how to generate a valid SMILES string since the syntax is strict and the molecule needs to be chemically correct.

Methods such as grammar variational autoencoders Kusner et al., 2017 have been developed that incorporate chemical rules in the generation process of better validity of generated molecules.

2.1.3 Graph Neural Networks (GNNs) for Property Prediction

GNNs have emerged to be strong tools in representation learning from graph-structured data. In chemistry, molecules can be inherently represented as graphs with atoms as nodes and bonds as edges.

Graph Convolutional Networks (GCNs):

GCNs (Kipf and Welling, 2017) generalize convolutional neural networks to graph data. It updates node representations by aggregating information from neighboring nodes, which can capture the local structural information effectively. The GCNs have been applied successfully to predict molecular properties by learning from molecular graphs (Duvenaud et al., 2015).

Graph Isomorphism Networks (GINs):

Xu et al. (2019) proposed GINs which are designed to be as powerful as the Weisfeiler-Lehman graph isomorphism test in distinguishing graph structures. GINs represent an extension to the GNN series by updating node features through aggregation with learned weights of neighbor information in order to capture more complex graph structures. They have shown much higher performance than any other variant of GNNs in molecular property prediction tasks.

2.1.4 Integration of Generative Models and GNNs

This attempt combines generative models with property prediction networks to create a closed system for designing molecules. Generative models would generate novel molecular structures whereas the predictive models predict their properties. This combination will enable optimizing molecules with desired properties with reinforcement learning or property-conditioned generation (You et al., 2018).

Sr. No	Authors	Year	Limitations	Methodology	Findings
1	Josep Arús-Pous, Simon Viet Johansson, Oleksii Prykhodko, Esben Jannik Bjerrum, Christian Tyrchan, Jean-Louis Reymond, Hongming Chen & Ola Engkvist [1]	2019	Randomized SMILES may increase diversity but could reduce model reproducibility, and randomization doesn't guarantee chemically valid or syntactically optimal SMILES for all use cases.	Benchmarks RNN models with different SMILES variants and cell types to assess chemical space generation.	Randomized SMILES improve model diversity and accuracy, enhancing molecule generation for drug discovery.
2	Ruud van Deursen, Peter Ertl, Igor V. Tetko & Guillaume Godin [2]	2020	Bidirectional RNNs with online quality control add computational complexity and may struggle with long or complex SMILES sequences, limiting scalability for larger molecular databases.	Introduces Generative Examination Networks (GENs) using bidirectional RNNs and online quality control to generate SMILES.	GENs achieve high validity and novelty in generated SMILES, with strong conservation of property space.
3	Maranga Mokaya, Fergus Imrie, Willem P. van Hoorn, Aleksandra Kalisz, Anthony R. Bradley & Charlotte M. Deane [3]	2023	Deep reinforcement learning approaches can be computationally intensive and may still face challenges with capturing complex molecular features, especially for out-of-distribution or rare compounds.	Uses curriculum learning and deep reinforcement learning to improve SMILES-based molecular generation.	Achieves more diverse molecular sets, but notes limitations in SMILES representation for certain optimizations.
4	Francesca Grisoni, Michael Moret, Robin Lingwood, Gisbert Schneider [4]	2020	Bidirectional RNNs are more complex and memory-intensive, which could limit their application in high-throughput screening. Also, SMILES representations may fail to capture 3D spatial information.	Compares bidirectional RNN methods (FB-RNN, NADE, and BIMODAL) to a unidirectional RNN for SMILES generation.	Bidirectional RNNs, especially BIMODAL, show improved novelty, scaffold diversity, and relevance in generated molecules compared to unidirectional methods.
5	Noel M. O'Boyle, Andrew Dalke [5]	2020	While DeepSMILES simplifies syntax, it may still lack interpretability and could introduce unique errors or biases that are challenging to correct without additional preprocessing.	Introduces DeepSMILES, a modified SMILES syntax that simplifies ring closures and branch notation for ML models.	DeepSMILES improves syntactic validity in generated SMILES, benefiting machine-learning applications in molecule design.

6	Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, Mark P. Waller [6]	2017	The focus on generating similar molecules to training data could limit novelty and reduce chemical diversity, leading to overfitting to known chemical spaces.	Uses RNNs to generate molecules similar to training data and fine-tunes models on specific targets for drug discovery.	The model successfully generates drug-like molecules, reproducing known actives for specific targets with high accuracy.
7	Esben Jannik Bjerrum, Richard Threlfall [7]	2017	RNNs can be prone to generating invalid SMILES strings and are limited by their inability to capture structural or stereochemical information in molecules effectively.	Explores RNNs with LSTM cells trained on SMILES to generate novel and chemically viable molecules.	The generated molecules are chemically sensible and show similar property distributions to the training datasets.
8	Linde Schoenmaker, Olivier J. M. Béquignon, Willem Jespers, Gerard J. P. van Westen [8]	2023	The approach requires extensive preprocessing to correct SMILES, adding extra steps and computation, and might be inefficient for very large molecular libraries.	Trains a transformer model to correct invalid SMILES generated by RNN, VAE, GAN, and other models.	The SMILES corrector successfully improves validity of generated molecules, enhancing de novo drug design.
9	Peter Ertl, Richard Lewis, Eric Martin, Valery Polyakov [9]	2018	While effective for generating drug-like molecules, LSTMs are computationally intensive and can struggle with longer SMILES, limiting diversity in generated compounds.	Generation of novel molecules using a long short-term memory (LSTM) neural network, followed by virtual screening using a profile QSAR approach.	Generated one million diverse, drug-like molecules in 2 hours with favorable physicochemical properties, synthetic accessibility, and bioactivity similar to ChEMBL compounds.
10	Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, Hongming Chen [10]	2017	Reinforcement learning can be unstable and requires extensive tuning, making it less practical for real-time applications and often fails to generalize beyond specific property optimizations.	Reinforcement learning approach to fine-tune RNNs for generating molecules with specific properties.	Generated drug-like molecules with targeted bioactivity, useful for scaffold hopping and library expansion.
11	Naruki Yoshikawa, Kei Terayama, Teruki Honma, Kenta Oono, Koji Tsuda [11]	2018	Grammatical evolution is sensitive to the initial population and may converge prematurely, potentially limiting the exploration of the full chemical space.	Population-based molecule generation using grammatical evolution (ChemGE), allowing concurrent updates and evaluations.	Generated diverse, high-affinity molecules for thymidine kinase, with improved diversity over existing known molecules.

12	Dai, Hanjun; Tian, Yingtao; Dai, Bo; Skiena, Steven; Song, Le [12]	2018	SD-VAEs, while effective, are complex and can be computationally prohibitive for larger datasets, limiting accessibility for practical applications.	Developed a syntax-directed variational autoencoder (SD-VAE) to generate syntactically and semantically valid data.	Demonstrated effectiveness in generating valid molecular and program structures, outperforming state-of-the-art models.
13	Peter Pogány, Navot Arad, Sam Genway, Stephen D. Pickett [13]	2018	The reliance on reduced graph representations may limit the complexity of generated molecules and could overlook structural details essential for certain applications.	Used a seq-to-seq model to translate Reduced Graph representations to SMILES for molecule generation.	Successfully generated valid molecules with target chemical features, enabling scaffold hopping and applications in drug discovery.
14	Francois Berenger, Koji Tsuda [14]	2021	Fragment-based methods may overlook critical inter-fragment chemical dependencies, limiting the structural coherence and diversity of generated molecules.	Developed a method, "Fast Assembly of SMILES Fragments" (FASMIFRA), for generating diverse SMILES/DeepSMILES molecules.	Achieved high-speed generation of valid molecules with diversity matching the training set, and released as open-source software.
15	Umit V. Ucak, Islambek Ashyrmamatov, Juyong Lee [15]	2023	Atom-in-SMILES tokenization can add computational overhead and complexity to model training, which could limit its practicality for large-scale molecule generation.	Developed an "atom-in-SMILES" tokenization scheme to improve token accuracy in SMILES sequences.	The method significantly enhanced prediction accuracy and quality in SMILES-based chemical models, outperforming traditional tokenization methods.
16	Chao Pang, Jianbo Qiao, Xiangxiang Zeng, Quan Zou, Leyi Wei [16]	2023	The review highlights challenges such as data quality and the lack of 3D structure handling, which limit the generalizability of current generative models in drug discovery.	Reviewed recent advancements in deep generative models for drug molecule generation.	Highlights challenges in molecule generation, including data quality, lack of 3D molecular representations, and limited evaluation metrics, and suggests future research directions.
17	Youjun Xu, Kangjie Lin, Shiwei Wang, Lei Wang, Chenjing Cai, Chen Song [17]	2019	This review suggests that while deep learning is powerful, its limitations include poor interpretability, challenges with 3D spatial learning, and reliance on extensive training data.	Reviewed the use of deep generative neural networks in de novo drug discovery and optimization.	Discusses four different generative architectures and optimization strategies for molecular generation and suggests future directions in drug design.

18	Medard Edmund Mswahili, Young-Seob Jeong[18]	2023	Transformer models often require substantial computational resources and large datasets to perform well, limiting accessibility and scalability.	Reviewed transformer-based models for chemical SMILES representations in cheminformatics.	Provides an overview of transformer-based models for SMILES, discusses their applications in molecular property prediction, and suggests future research directions.
19	Mikhail Andronov, Natalia Andronova, Michael Wand, Jürgen Schmidhuber, Djork-Arné Clevert [19]	2024	Speculative decoding speeds up inference but may compromise accuracy in more complex chemical reaction models, especially for less common or intricate reactions.	Proposed speculative decoding method to accelerate inference in autoregressive SMILES generators.	Achieved over 3X faster inference in reaction prediction and single-step retrosynthesis without loss in accuracy.
20	Noel M O'Boyle[20]	2012	Canonical SMILES generation may fail for highly complex structures or exotic compounds, and the method's reliance on InChI may not account for all aspects of stereochemistry and spatial orientation.	Describes using the InChI canonicalization to derive canonical SMILES.	99.79% of the ChEMBL database and 99.77% of the PubChem subset were successfully canonicalized using Universal SMILES.

2.2 Summary of Literature Review and Research Gap

The intersection of machine learning and chemistry has given rise to powerful computational techniques in the generation of molecules as well as the prediction of molecule properties. Traditional approaches in the field of computational chemistry, such as quantum mechanical simulations and molecular dynamics, usually suffer from computational cost and poor scaling in relation to the size of the molecules. As a result, deep learning models have lately gained much interest in accelerating the discovery and design of new molecules with specific desired properties.

2.2.1 Summary of Existing Work

The previous studies have demonstrated the feasibility of generating completely new molecules with these VAEs and GANs:

- **VAEs** are shown to be efficient in capturing chemical space and generating chemically valid molecules with high diversity by making an extension into continuous latent space that facilitates access to exploring new molecules and optimizing properties.
- **GANs** are an alternative to the above method but commonly face instability and diversity in the generated molecules. Despite these disadvantages, GANs have been applied to molecule generation with certain desired properties.

In terms of property prediction:

- **GCNs** have really become popular for the property prediction of molecules as it has shown effective capability in modelling local chemical environments for a molecule.
- **GINs** are more efficient in finding complex graph structures which results in more accurate predictions of the molecular properties.

2.2.2 Identified Research Gaps

While there has been distinct advance on the sides of molecular generation and property prediction, there are gaps.

1. **Comparative Analysis of VAEs and GANs:-** Till now, one crucial gap persists: very few work that comprehensively perform head-to-head comparison of VAE and GAN on the same dataset using consistent metrics. Most work focuses on one model type rather than directly comparing which offers relatively greater strength and weakness each of them for the task of molecular generation.

2. **Impact on Property Prediction:-** Very few studies have explored how the differences in molecules between VAEs and GANs affect the performance of the downstream property prediction models like GNNs. This is the crucial area for developing the integrated systems for the design of molecules.

3. **Integration Frameworks:-** Most related existing works treat generation of molecules with the prediction of their properties as totally different tasks. Real integration frameworks combining generative models and predictive models in an end-to-end molecular design and optimization framework are also needed.

4. **Evaluation Metrics: -** There is a need for consistent and complete evaluation metrics to better evaluate the quality of generated molecules rather than solely focusing on validity, such as uniqueness, novelty, diversity, and their quality in approximating chemical space of interest.

2.2.3 Motivation for the Current Study

The current studies are motivated by the need to address these gaps by:

- Carrying out side-by-side comparisons of VAEs and GANs for the generation of molecular SMILES on the same dataset, with the same evaluation criteria.
- Analyzing how the molecules generated with each model type impact GNN performance in property prediction tasks.
- Developing an integrated methodology that combines generative models with GNNs so as to create a cohesive pipeline through which molecules can be generated and evaluated.
- It should employ a wide range of evaluation metrics that will prove to provide a sufficiently effective review of each model.

These research gaps shall be addressed using the insights gained about the advantages and limitations of VAEs and GANs in molecular generation and its applicability to including GNNs in property prediction. This work is critical for further development in computational methods applied to drug discovery and material design areas where fast and accurate prediction of molecular properties is of prime importance.

CHAPTER 3: SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Software Tool Platform/ Tools/Framework used

The implementation of this project requires a broad suite of software tool and frameworks for deep learning, molecular data processing, and computational graph operations. The following software components were used:

Programming Language:

- Python 3.8: It was chosen for its versatility, huge library support, and strong community that makes it perfect for use in machine learning and data science applications.

Deep Learning Frameworks:

- PyTorch 1.10: A flexible and efficient deep learning framework for building and training neural network models, including VAEs, GANs, and GNNs. PyTorch's dynamic computation graph makes it easier to customize and debug models.

- PyTorch Geometric (PyG) 2.0: An extension library on top of PyTorch specifically designed for graph-based deep learning. PyG has optimized implementations of many common Graph Neural Network layers and utilities necessary when building GCN and GIN models.

Chemical Informatics Libraries:

- RDKit 2021.09.4: Open Source Toolkit for cheminformatics. Used for the main task of parsing, validating and manipulation of SMILES strings, molecular visualization and computing chemical descriptors.
- DeepChem 2.5.0: Python library, designed to provide a range of tools for deep learning in drug discovery, materials science and quantum chemistry. Usage is found for accessing QM9 and presents additional utilities for processing data.

Data Manipulation and Analysis:

- Pandas 1.3.4: For efficient data manipulation, analysis, and preprocessing activities, especially when dealing with big data and complex data structures.
- NumPy 1.21.2: For numerical computations and array operations, providing support for scientific computing operations.

Visualization Tools:

- Matplotlib 3.4.3: Employed for creating static, animated, and interactive visualizations in Python, including training curves and property distribution plots.
- Seaborn 0.11.2: Drawing attractive and informative statistical graphics based on Matplotlib

Development Environment:

- Jupyter Notebook: This is the environment for writing, running, showing output, and narrating the development process in an integrated manner.

Machine Learning Utilities:

- Scikit-learn 0.24.2: Used for machine learning utilities such as splitting of data and models' evaluation, and used also classifiers for testing.
- fcd-torch: PyTorch implementation to compute the Fréchet ChemNet Distance, benchmarking similarity between real and generated distributions of molecules.

Additional Tools:

- NetworkX 2.6.3: It is the software package for the creation, manipulation, and study of the structure of complex networks use this in graph data analysis.

Note: All software tools were selected based on their compatibility with each other and their ability to support GPU acceleration for enhanced computational performance.

3.2 Hardware tools

For purposes of handling the computational expense of deep neural network training along with the large molecular data processing, the following computer hardware tools were utilized:

Processor (CPU):

Intel Core i7-9750H CPU @ 2.60GHz: The main CPU was for fast execution of computationally intensive tasks involved in data preprocessing, model initialization, and auxiliary computations.

Graphics Processing Unit (GPU):NVIDIA GeForce RTX 2060 with 6 GB GDDR6 VRAM: This was in use for the acceleration of training deep learning models. Generalized computation on GPU took much lesser time than when carrying out sequential matrix operations or even dealing with large numbers of data.

Memory (RAM):16 GB DDR4 RAM: At such a level of RAM, it was sufficient to handle big data, calculations done during the processes, and running several processes without memory bottlenecks.

Storage:512 GB Solid-State Drive (SSD): For fast reading/writing speeds for loading datasets, saving model checkpoints, and accessing large files efficiently and effectively.

Operating System:

Windows 10 Pro 64-bit and **Ubuntu 20.04 LTS** (dual-boot configuration): Provided the flexibility and compatibility with various software tools so that best possible development environment has been assured.

3.3 Work Breakdown Structure

A WBS is an organized work breakdown structure that enables the segmentation of the project into manageable sections. The development is done in an organized manner, and then resources are distributed effectively to meet results. The project was categorized into distinct phases and tasks for the proper tracking of progress and the systematic distribution of resources.

Data Acquisition and Preprocessing

1. Download and validate the integrity of the QM9 dataset.
2. Extract SMILES strings along with molecular properties.
3. Remove noise by eliminating duplicate or invalid SMILES strings.
4. Create a character-level vocabulary for SMILES strings.
5. Encode numerical sequences for model input from SMILES strings.

Model Development

1. Design a VAE architecture for molecular generation.
2. Implement a GAN architecture containing a generator and discriminator.
3. Develop a GCN model for property prediction.
4. Implement a GIN model to improve the prediction of properties.

Model Training

1. Train the VAE model on the training dataset and monitor reconstruction loss and KL divergence in each iteration.
2. Train the GAN model, carefully balancing generator and discriminator losses to avoid mode collapse.
3. Train the GCN and GIN models on molecular property prediction tasks and tune parameters for optimal performance.

Molecule Generation and Validation

1. Generate molecules using the trained VAE and GAN models.
2. Postprocess generated SMILES for syntactic errors and chemical validity.
3. Use RDKit to validate molecules and filter out chemically invalid structures.

Property Prediction and Evaluation

1. Use GCN and GIN models to predict properties of the generated molecules.
2. Analyze prediction accuracy using metrics like MSE and MAE.
3. Assess the impact of the type of generative model on property prediction performance.

Performance Evaluation and Metrics Calculation

1. Calculate validity, uniqueness, and novelty rates for the generated molecules.
2. Compute Fréchet ChemNet Distance (FCD) to compare molecular distributions.
3. Evaluate Tanimoto similarity and internal diversity of the generated molecules.

Visualization and Analysis

1. Generate histograms and plots for property distributions and model training curves.
2. Visualize sample molecules using RDKit for qualitative assessment.
3. Compile results into tables and graphs for inclusion in the report.

Documentation and Reporting

1. Document methodologies, experimental setups, and code implementations.
2. Write a comprehensive report integrating findings from molecular generation and property prediction analyses.
3. Prepare presentation materials, including diagrams (e.g., Mermaid diagrams).

3.4 Functional Requirements

The system was built to fulfill the following functional requirements to achieve the project's desired functionality:

- **Data Processing:**
  - Efficient loading and preprocessing large-scale molecular datasets.
  - Correct encoding and decoding the SMILES strings along with handling special tokens that indicate start/end sequences.
- **Model Training:**
  - Implementation of the customizable neural network architectures (VAE, GAN, GCN, GIN) that allow for fine-tuning parameters like learning rate, batch size, number of epochs.
  - Real-time monitoring of training metrics and ability to save and resume the training process from checkpoints.
- **Molecule Generation:**
  - New molecular structures as valid SMILES strings.
  - Ability to generate molecules that meet given properties or constraints, such as the number of atoms.
- **Validation and Evaluation:**
  - Checking the chemical correctness of generated molecules by the usage of cheminformatics tools.
  - Calculation of key performance metrics like validity rate, uniqueness, novelty, FCD, Tanimoto similarity, and internal diversity.
  - Prediction of molecular properties using trained GNN models and the accuracy of predictions.
- **Visualization:**
  - Generation of informative visualization like histograms of property distributions, curves of train loss and molecule images in grids.
  - Visualization that a generic versus actual property values for the assessment of performance.
- **Integration:**
  - Integration of all generative models with the predictive models for property formation of an end pipeline.
  - Output from all generative models directly processed through the property prediction models without intermediate elaboration.

3.5 Non Functional Requirements

In addition to these functional requirements, the system had to meet a number of non-functional criteria:

- **Performance and Efficiency:**
  - Optimize code for the maximum exploitation of GPU acceleration, reduce training times.
  - Manage memory without bottlenecks and crashes caused by intense data operations.
- **Scalability:**
  - Modular design to scale up with bigger datasets or complex models. Make it possible to include more properties or extend to other molecular datasets with minimal changes.
- **Usability and Accessibility:**
  - Well documented code and clear instructions on how to replicate the experiments and how to extend the functionality.
  - User interfaces or configuration files for setting up the experiments, as well as for changing parameters.
- **Reliability and Reproducibility:**
  - The same simulation is played each time in repeated runs, so that results may reproduce if the same random seed and configurations are used.
  - Robust error handling: the program may fail, but should give useful feedback messages to the user.
- **Maintainability and Extensibility:**
  - Clean structure, follows best practices, so maintenance in the future is possible and easy.
  - Use of version control systems, such as Git for follow up and effective coordination.
- **Security and Compliance:**
  - All data handling should be in accordance with the relevant data protection standards.

3.6 Project Cost Estimation

Software, hardware, as well as human resource considerations in the cost estimation of the project

- **Software Costs:**
  - **Licensing Fees:** All software applied in this project are open-source products and thus carry zero direct costs from the perspective of software.
  - **Development Tools:** Jupyter Notebook and libraries like PyTorch and RDKit are open source, so zero extra overhead.
- **Hardware Costs:**
  - **Existing Equipment:** The project was done running on existing hardware resources-mostly personal computing equipment having good enough specifications-so this only had minimal extra overhead on the hardware side.
  - **Electricity Consumption:**The intensive computations especially, with the usage of GPU acceleration, proved to be power-hungry. This added cost is entirely dependent on the local electricity rates and the time duration of usage..

- **Monetary Costs:** Minimum direct monetary costs because it uses open-source software and existing hardware. Cloud computing costs are dependent on the requirements of resources and optional

CHAPTER 4: METHODOLOGY

4.1 Introduction

This chapter details the methodologies employed in the comparative analysis of Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) for molecular SMILES generation and their subsequent property prediction using Graph Neural Networks (GNNs). The methodology encompasses data acquisition, preprocessing steps, the design and implementation of generative models (VAEs and GANs), the development of GNN-based property predictors (Graph Convolutional Networks and Graph Isomorphism Networks), integration strategies for combining generative and predictive models, training procedures, and evaluation protocols.

4.2 Data Acquisition and Preprocessing

4.2.1 Dataset Selection

The **QM9 dataset** was selected for this study due to its comprehensive collection of small organic molecules and their associated properties, making it ideal for molecular modeling tasks.Around 134,000 molecules with SMILES strings and corresponding 19 quantum chemical properties are included in this dataset. All these properties have been computed applying DFT theory **B3LYP/6–31G(2df,p)**, which is widely diffused setup in computational chemistry.

In this case, B3LYP is a hybrid DFT functional (Becke, 3-parameter, Lee-Yang-Parr) which incorporate features of Hartree-Fock theory together with DFT in order to achieve better accuracy on the prediction of other molecular properties like energy and geometry**. 6-31G(2df,p)**: the molecular base set that describe electron wave functions around atoms This basis set has polarization functions (2df,p) for greater flexibility of the electron distribution. Specifically:

- **6-31G** refers to the split-valence nature of the basis set, where core and valence electrons are treated with different levels of flexibility.

- **(2df,p)** indicates added functions (d and f orbitals for heavy atoms and p orbitals for hydrogen), which enhance the accuracy of molecular property predictions.

Using B3LYP/6-31G(2df,p) allows for precise computation of quantum chemical properties, including molecular energies, geometries, dipole moments, and vibrational frequencies, essential for predictive modeling and molecular analysis.

4.2.2 Data Extraction and Cleaning

SMILES Extraction

- The SMILES strings were extracted from the dataset to use it in in generative modeling.
- Each SMILES string was prefixed and suffixed by a special token indicating the start (<) of sequence and end(>), to enable the generation at sequence level as well as reconstruction

Data Cleaning

- **Validity Check:** All SMILES strings were validated by RDKit's Chem.MolFromSmiles function for correctness.
- **Deduplication:** Identical SMILES strings were identified and eliminated to avoid potential bias in the training set.
- **Character Vocabulary Creation:** A character-level vocabulary was created, mapping each unique character in the SMILES strings to an integer index. This vocabulary included special tokens for padding (<pad>), start (<sos>), and end (<eos>) symbols.

4.2.3 Data Encoding and Preparation

Sequence Encoding

- SMILES strings were encoded into integer sequences based on the created vocabulary.
- Padding was applied to sequences to ensure uniform length within batches, facilitating efficient batch processing during training.

Dataset Splitting

- The dataset was split into training (80%), validation (10%), and test (10%) sets using stratified sampling to preserve the distribution of molecular properties across sets.
- For property prediction tasks, the splits ensured that the molecules in the test set were not seen during training or validation of the generative models.

4.3 Generative Model Implementation

4.3.1 Variational Autoencoder (VAE)

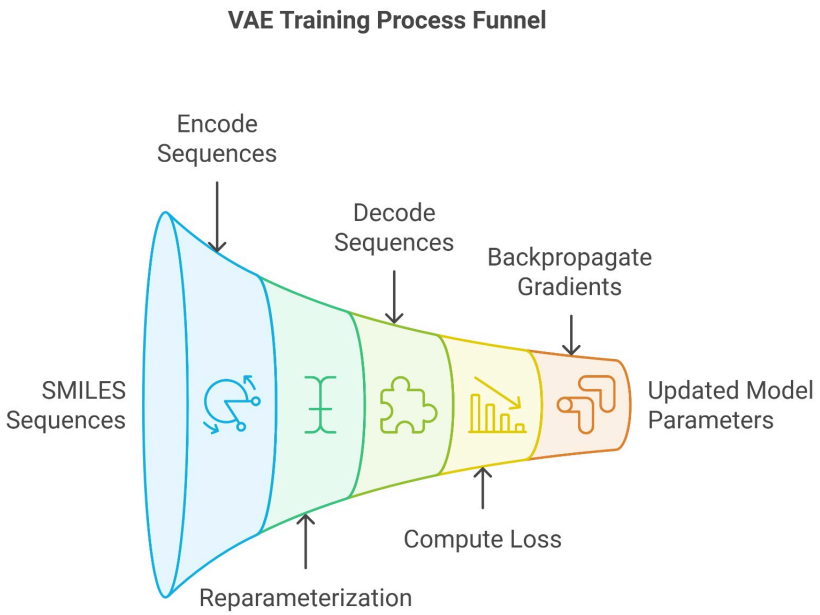

Fig 4.1 VAE training Funnel

Architecture Design

- **Encoder:**
  - Utilized a Gated Recurrent Unit (GRU) network to process the input SMILES sequences.
  - The encoder transformed the input into a fixed-size hidden representation.
  - Output layers generated the mean (μ) and log-variance ($\log\sigma^2$) vectors of the latent space.
- **Latent Space:**
  - A continuous latent space of dimension 64 was used.
  - The reparameterization trick was employed to allow backpropagation through the stochastic sampling process:

$$z=\mu+\sigma \cdot \epsilon$$

where:

- z is the latent variable we want to sample.
- μ and σ are the mean and standard deviation of the distribution, predicted by the encoder network.
- $\epsilon \sim N(0,I)$: Here, ϵ is a random variable sampled from a standard normal distribution with mean 0 and identity covariance matrix I.

This formulation allows gradients to pass through the sampling process during backpropagation by making z ,a deterministic function of μ, σ, and ϵ, while still capturing the randomness needed for effective training of Variational Autoencoders (VAEs)..

- **Decoder:**
  - A GRU-based decoder reconstructed SMILES sequences from the latent vector (z).
  - Teacher forcing was used during training, where the actual previous token was fed into the decoder instead of the predicted one.

Training Procedure

- **Loss Function:**
  - The objective function combined the reconstruction loss (cross-entropy loss between input and output sequences) and the Kullback-Leibler (KL) divergence between the learned latent distribution and the standard normal distribution:

    $L = Lrecon + \beta \cdot LKL$, where (β) is a weight hyperparameter.

- **Optimization:**
  - The Adam optimizer was used with a learning rate of 1×10^{-3}
  - Gradient clipping was applied to prevent exploding gradients.
- **Hyperparameters:**
  - Batch size: 128
  - Number of epochs: 20
  - Hidden dimension: 256 for both encoder and decoder GRUs

4.3.2 Generative Adversarial Network (GAN)

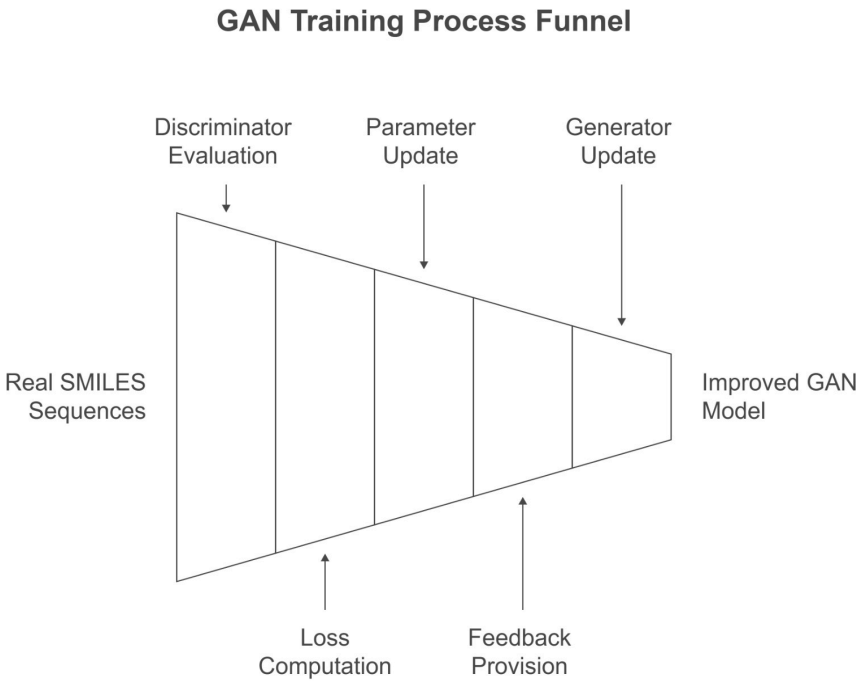

Fig 4.2 GAN Training Funnel

Architecture Design

- **Generator:**
  - Input: Random noise vector $z \in R^{64}$ sampled from a standard normal distribution.
  - A fully connected layer projected (z) to the initial hidden state of an LSTM network.
  - The LSTM generated SMILES sequences one token at a time.
  - Output: Sequence of token probabilities over the vocabulary (excluding padding).
- **Discriminator:**
  - Input: SMILES sequences (real or generated).
  - Embedding layer mapped tokens to embeddings.
  - Bidirectional LSTM processed the sequences to capture contextual information from both directions.
  - Fully connected layers with sigmoid activation outputted a probability indicating whether the sequence was real or generated.

Training Procedure

- **Adversarial Training:**
  - The generator and discriminator were trained in an adversarial manner.
  - The discriminator aimed to correctly classify real and generated SMILES sequences.
  - The generator aimed to produce sequences that the discriminator would classify as real.
- **Policy Gradient for Generator:**
  - Due to the discrete nature of SMILES sequences, policy gradient methods (REINFORCE algorithm) were used to train the generator.
  - The generator received rewards based on the discriminator's output.
- **Loss Functions:**
  - **Discriminator Loss:** Binary cross-entropy loss between predicted and true labels.
  - **Generator Loss:** Negative expected reward, where the reward is the discriminator's output.
- **Optimization:**
  - Separate Adam optimizers for generator and discriminator, each with a learning rate of 1×10^{-3}.
  - Gradient clipping was applied to stabilize training.
- **Hyperparameters:**
  - Batch size: 128
  - Training steps: 20,000
  - Evaluation frequency: Every 100 steps

4.4 Graph Neural Network Implementation for Property Prediction

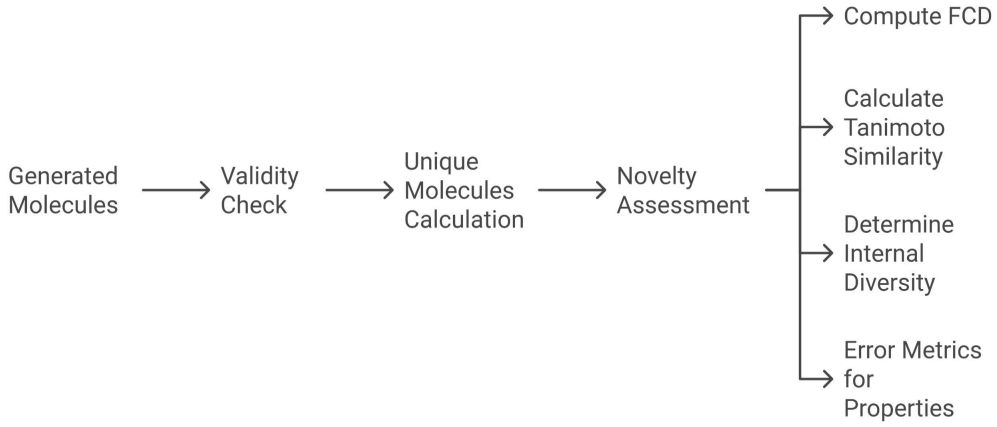

Fig 4.3 GNN Pipleine

4.4.1 Molecular Graph Representation

- Molecules were represented as undirected graphs $G = (V, E)$, where:
  - $V = \{v1, v2, ..., vn\}$:The set of nodes representing atoms, with $n = |V|$, the total number of atoms.
  - $E = \{(vi, vj)|vi, vj \in V, vi/vj\}$ The set of edges representing chemical bonds between pairs of atoms.
- Atom features included one-hot encodings of atom types, hybridization states, and other relevant chemical properties.
- Bond features captured bond types (single, double, triple, aromatic).

4.4.2 Graph Convolutional Network (GCN)

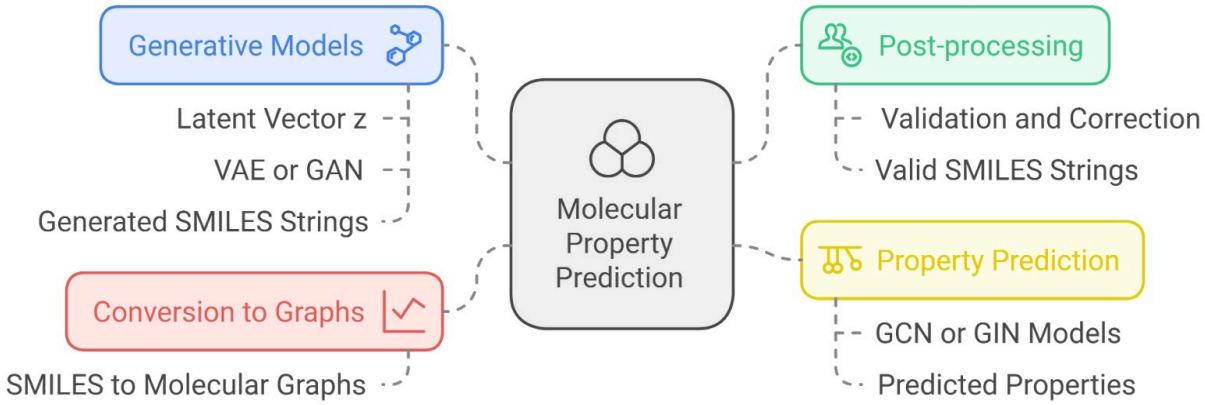

Fig 4.4 GCN Molecular Property Prediction

Architecture Design

- **Convolutional Layers:**
  - Three GCN layers were used to update node representations by aggregating features from neighboring nodes.
  - Each layer performed the operation:

$$hv^{(k)} = \sigma[\sum W^{(k)} h_u^{(k-1)} + b^{(k)}]$$

- $hv^{(k)}$ is the updated feature vector for node v at layer k.
- $N(v)$ denotes the set of neighbors of node v in the graph.
- $h_u^{(k-1)}$ is the feature vector of the node u (neighbor of v) at the previous layer $(k - 1)$.
- $W^{(k)}$ is the learnable weight matrix at layer k, which transforms the features of the neighbors.
- $b^{(k)}$ is the learnable bias vector at layer k.
- σ is the activation function, typically the **ReLU** function, which applies element-wise non-linearity

Global Pooling:

- A global mean pooling layer aggregated node features into a graph-level representation.

Output Layer:

- A fully connected layer mapped the graph representation to the target property (e.g., dipole moment).

Training Procedure

- **Loss Function:**
  - Mean Squared Error (MSE) between predicted and actual property values.
- **Optimization:**
  - Adam optimizer with a learning rate of 1×10^{-3}.
- **Hyperparameters:**
  - Batch size: 64
  - Number of epochs: 50

4.4.3 Graph Isomorphism Network (GIN)

Architecture Design

- **GIN Layers:**
  - Three GIN convolutional layers were employed, each updating node features with the operation:

$$h_v^{(k)} = \mathrm{MLP}^{(k)}\left((1+\epsilon)\cdot h_v^{(k-1)} + \sum_{u\in\mathcal{N}(v)} h_u^{(k-1)}\right)$$

- $h_v^{(k)}$ is the feature vector for node v at layer k.
- $h_v^{(k-1)}$ is the feature vector of node v at the previous layer $(k-1)$.
- $N(v)$is the set of neighbors of node v.
- $h_u^{(k-1)}$ is the feature vector of node u (a neighbor of v) at layer $(k-1)$.
- ϵ is a learnable or fixed scalar parameter that modulates the influence of the node's own feature in the aggregation.
- The sum $\sum_{u\in\mathcal{N}(v)} h_u^{(k-1)}$ aggregates the features of the neighbors of node v from the previous layer.

- **$MLP^{(k)}$** represents a **Multi-Layer Perceptron** function at layer k, which typically consists of one or more fully connected layers, applying non-linear activations.
- **Batch Normalization and Activation:**
    - Batch normalization layers followed each GIN layer to stabilize training.
    - ReLU activation functions introduced non-linearity.
- **Global Pooling:**
    - A global sum pooling layer aggregated node features into a graph-level representation, capturing structural information effectively.
- **Output Layers:**
    - Two fully connected layers processed the pooled representation to predict the target property.

Training Procedure

- **Loss Function:**
    - Mean Squared Error (MSE) between predicted and actual property values.
- **Optimization:**
    - Adam optimizer with a learning rate of 1×10^{-3}.
- **Hyperparameters:**
    - Batch size: 64
    - Number of epochs: 50

4.5 Integration of Generative Models and GNNs

4.5.1 Generating Molecules for Property Prediction

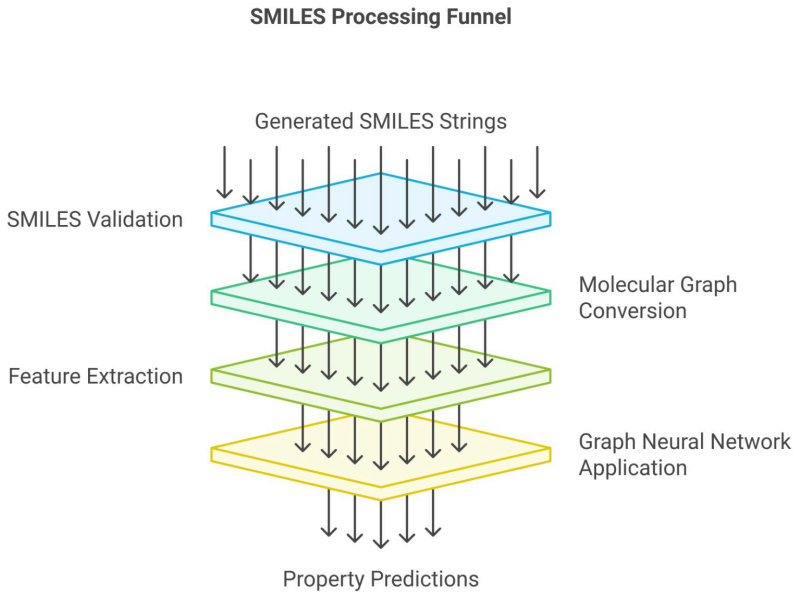

Fig4.5. Smiles Processing Funnel

- **Sample Generation:**
    - Both VAE and GAN models were used to generate a set of 1,000 novel molecules each.
    - Post-processing steps ensured the generated SMILES strings were syntactically correct and chemically valid.

4.5.2 Molecule Conversion to Graphs

- The generated SMILES strings were converted into molecular graphs using RDKit.
- Atom and bond features were extracted consistent with those used in training the GNNs.

4.5.3 Property Prediction

- The GCN and GIN models, trained on the original QM9 dataset, were used to predict properties of the generated molecules.
- Predictions included properties such as dipole moment, molecular weight, and LogP (octanol-water partition coefficient).

4.5.4 Evaluation of Predictions

- Predicted properties were compared against expected distributions from the training data.
- The accuracy and reliability of property predictions for generated molecules were assessed.

4.6 Training and Evaluation Protocols

4.6.1 Training Strategies

- **Regularization Techniques:**
    - Dropout and weight decay were used to prevent overfitting.
    - Early stopping based on validation loss was implemented.
- **Learning Rate Scheduling:**
    - Learning rate schedulers reduced the learning rate upon plateauing of validation loss to fine-tune the models.

4.6.2 Evaluation Metrics

For Generative Models:

- **Validity Rate:** Percentage of generated molecules that are chemically valid.
- **Uniqueness Rate:** Proportion of unique molecules among the valid generated set.
- **Novelty Rate:** Percentage of generated molecules not present in the training data.
- **Fréchet ChemNet Distance (FCD):** Measures the distance between the distribution of generated molecules and real molecules in a learned feature space.

- **Tanimoto Similarity:** Average similarity between generated molecules and those in the training set based on molecular fingerprints.
- **Internal Diversity:** Assesses diversity within the generated molecule set.

For Property Prediction Models:

- **Mean Squared Error (MSE):** Average squared difference between predicted and actual values.
- **Mean Absolute Error (MAE):** Average absolute difference between predicted and actual values.
- **Root Mean Squared Error (RMSE):** Square root of MSE, providing error in the same units as the target variable.
- **Coefficient of Determination (R^2):** Proportion of variance in the dependent variable predictable from the independent variable.

4.6.3 Statistical Analysis

- **Distribution Comparison:**
    - Histograms and kernel density estimates were used to compare property distributions of generated molecules with the training data.
- **Statistical Tests:**
    - Kolmogorov-Smirnov test assessed differences between distributions.
- **Visualization:**
    - Scatter plots of predicted vs. actual property values.
    - Molecule grid images for qualitative assessment.

4.7 Computational Resources and Environment

- **Hardware:**
    - The training and evaluation is achieved on a system with NVIDIA GeForce RTX 2060 GPU.
- **Software Environment:**
    - Python 3.8 with the libraries outlined in Chapter 3
- **Reproducibility:**
    - Random seeds were set for numpy and torch to ensure reproducible results.
    - All experiments were logged, and model checkpoints were save

CHAPTER 5: RESULTS AND DISCUSSION

5.1 Introduction

This chapter discusses the outcomes from implementing and assessing the Variational Autoencoder (VAE) and Generative Adversarial Network (GAN) models for generating molecular SMILES. It also evaluates the effectiveness of Graph Neural Networks (GNNs) in predicting the properties of the generated molecules.The findings are analyzed to compare the effectiveness of VAEs and GANs, assess the impact on property prediction using GNNs, and discuss the implications for molecular design and discovery.

5.2 Experimental Setup

5.2.1 Generative Model Training

- **VAE Training Parameters:**
    - **Encoder and Decoder Hidden Dimensions:** 256
    - **Latent Dimension:** 64
    - **Optimizer:** Adam with learning rate 1×10^{-3}
    - **Epochs:** 20
    - **Batch Size:** 128
- **GAN Training Parameters:**
    - **Generator and Discriminator Hidden Dimensions:** 64
    - **Latent Dimension:** 64
    - **Optimizers:** Separating the Adam optimizers for generator and discriminator, each set with a learning rate of 1×10^{-3}
    - **Training Steps:** 20,000
    - **Batch Size:** 128

5.2.2 Property Prediction Model Training

- **GCN and GIN Training Parameters:**
    - **Number of Layers:** 3
    - **Hidden Dimensions:** 64
    - **Optimizer:** Adam with learning rate 1×10^{-3}
    - **Epochs:** 50
    - **Batch Size:** 64
    - **Loss Function:** Mean Squared Error (MSE)

5.2.3 Evaluation Metrics

- **Generative Models:**
    - Validity Rate, Uniqueness Rate, Novelty Rate
    - Fréchet ChemNet Distance (FCD)
    - Tanimoto Similarity
    - Internal Diversity
    - Error Metrics for Molecular Properties (MSE and MAE)
- **Property Prediction Models:**
    - Mean Squared Error (MSE)
    - Mean Absolute Error (MAE)
    - Root Mean Squared Error (RMSE)
    - Coefficient of Determination

5.3 Results of Generative Models

5.3.1 Quantitative Metrics

5.3.1.1 Validity, Uniqueness, and Novelty

- **Validity Rate:**
    - **VAE:** 100%
    - **GAN:** 100%

- **Uniqueness Rate:**
    - **VAE:** 99.67%
    - **GAN:** 75.33%
- **Novelty Rate:**
    - **VAE:** 99.67%
    - **GAN:** 74.92%

Interpretation:

Both models demonstrated perfect validity rates, confirming that all the generated SMILES strings represent chemically valid molecules. The VAE notably surpassed the GAN in terms of uniqueness and novelty, producing a greater share of unique and novel molecules that were not included in the training set. This indicates that the VAE is more effective at exploring the chemical space and minimizing the generation of duplicates.

5.3.1.2 Fréchet ChemNet Distance (FCD)

- **VAE FCD:** 47.77
- **GAN FCD:** 118.98

Interpretation:

A lower FCD indicates that the distribution of generated molecules is closer to that of the real molecules in the training set. The VAE's lower FCD suggests that it generates molecules that are more similar to the training data distribution, while the GAN's higher FCD indicates a greater divergence.

5.3.1.3 Tanimoto Similarity

- **VAE Average Tanimoto Similarity:** 0.335
- **GAN Average Tanimoto Similarity:** 0.123

Interpretation:

The higher Tanimoto similarity for the VAE suggests that the molecules it generates have more structural features in common with those in the training set. In contrast, the GAN's lower similarity indicates that it produces molecules that are more structurally different from the training data.

5.3.1.4 Internal Diversity

- **VAE Internal Diversity:** 0.9375
- **GAN Internal Diversity:** 0.3256

Interpretation:

Internal diversity refers to the range of different molecules that are generated. The VAE shows a high level of internal diversity, indicating that it creates a broad spectrum of distinct molecules. In contrast, the GAN's low internal diversity suggests a limited variety, which may be a result of mode collapse, where the generator only produces a narrow set of molecules.

5.3.1.5 Error Metrics for Molecular Properties

VAE:

- **Mean Squared Error (MSE):**
    - **Molecular Weight (MolWt):** 362.54
    - **LogP:** 0.127
    - **Number of Hydrogen Donors (NumHDonors):** 0.539
    - **Number of Hydrogen Acceptors (NumHAcceptors):** 0.207
- **Mean Absolute Error (MAE):**
    - **MolWt:** 19.04
    - **LogP:** 0.356
    - **NumHDonors:** 0.734
    - **NumHAcceptors:** 0.455

GAN:

- **Mean Squared Error (MSE):**
  - **MolWt:** 0.241
  - **LogP:** 0.025
  - **NumHDonors:** 0.044
  - **NumHAcceptors:** 0.024
- **Mean Absolute Error (MAE):**
  - **MolWt:** 0.491
  - **LogP:** 0.157
  - **NumHDonors:** 0.210
  - **NumHAcceptors:** 0.155

Interpretation:

Surprisingly, the GAN shows significantly lower error metrics compared to the VAE, indicating that the properties of the molecules it generates are closer to the mean properties of the training data. However, this may be due to the GAN generating molecules with less diversity and more centered around the average properties, which aligns with the observed low internal diversity.

5.3.2 Molecular Property Distributions

5.3.2.1 Molecular Weight (MolWt) Distribution

- **Training Data:** Exhibits specific peaks at certain molecular weights, reflecting common molecule sizes.
- **VAE:** Captures the general distribution with peaks aligning approximately with the training data but with a broader spread, indicating less precision.
- **GAN:** Displays a flatter distribution, indicating a wider variety of molecular weights but includes weights less representative of those most common in the training data.

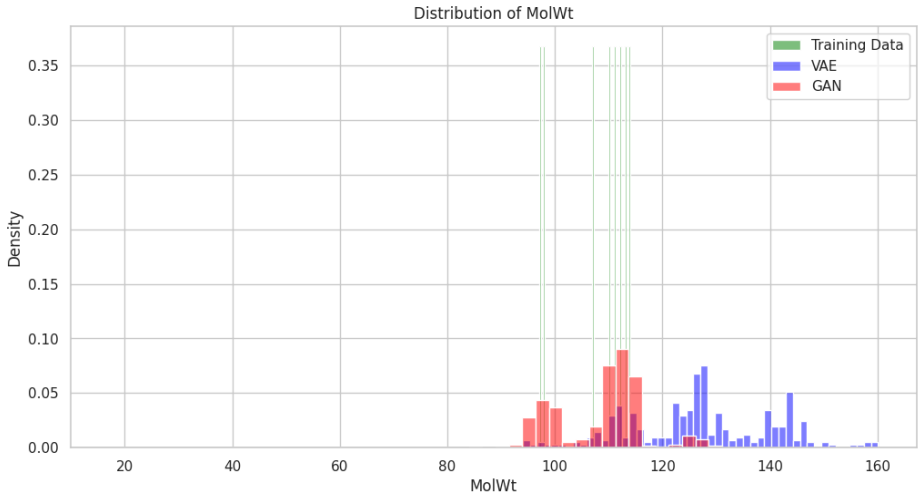

Figure 5.1: *Histogram comparing MolWt distributions of Training Data, VAE, and GAN generated molecules.*

Interpretation:

The VAE better approximates the molecular weight distribution of the training data, whereas the GAN's flatter distribution suggests a more uniform but less accurate representation.

5.3.2.2 LogP Distribution

- **Training Data:** Shows sharp peaks indicating common lipophilicity values.
- **VAE:** Broader distribution around the training peaks, less precise in capturing exact LogP values.
- **GAN:** Wider coverage, indicating variability but deviates more from characteristic peaks of the training data.

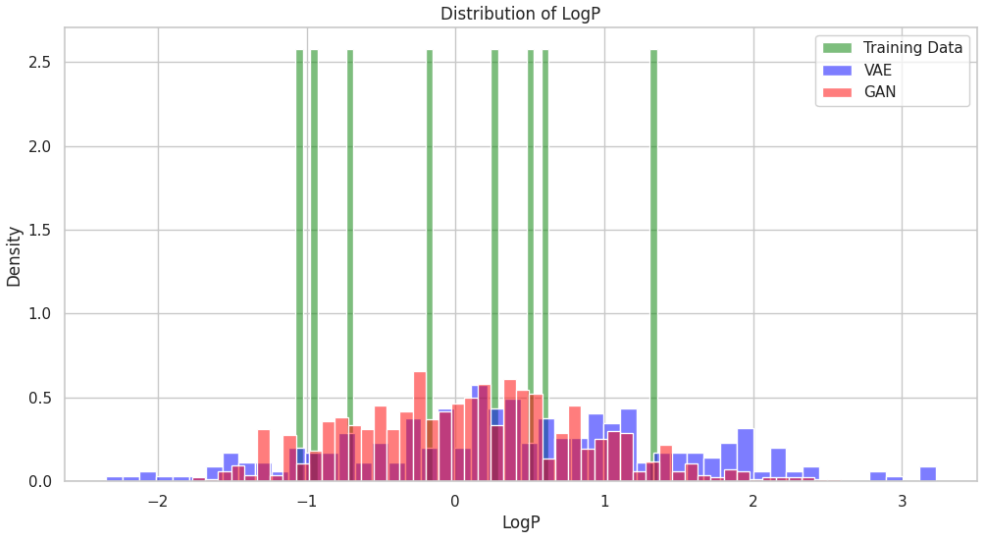

Figure 5.2: *Histogram comparing LogP distributions.*

Interpretation:

The VAE captures the lipophilicity trends of the training data more effectively than the GAN, which may generate molecules with unrealistic or uncommon LogP values.

5.3.2.3 Number of Hydrogen Donors and Acceptors

- **Training Data:** Concentrated at specific values.
- **VAE and GAN:** Both models cover the ranges but do not match the frequency distributions accurately, with the VAE performing slightly better.

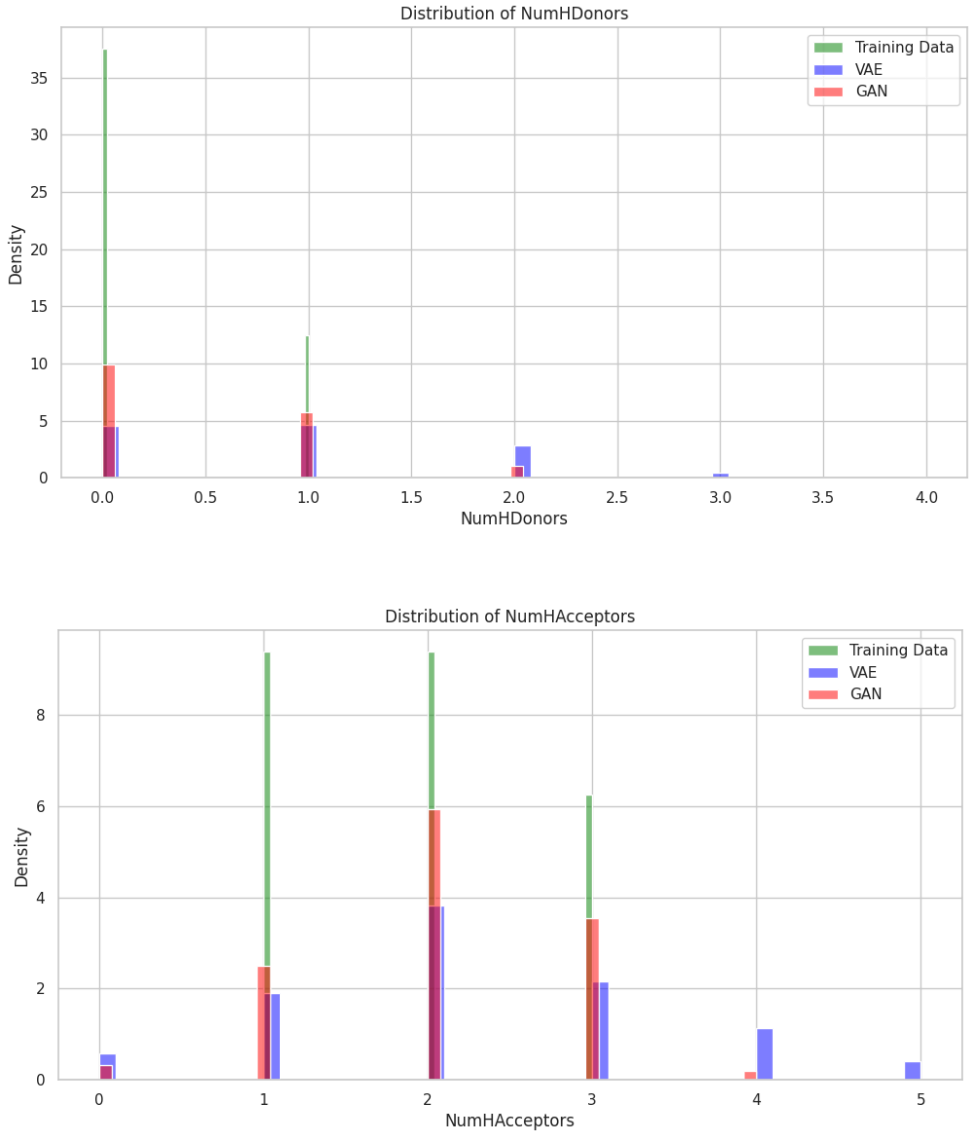

Figure 5.3 and 5.4: *Histograms comparing NumHDonors and NumHAcceptors distributions.*

Interpretation:

The VAE demonstrates a better ability to generate molecules with typical numbers of hydrogen bond donors and acceptors found in the training data.

5.3.3 Visual Inspection of Generated Molecules

Sample molecules generated by both models were visualized using RDKit to assess structural diversity and chemical plausibility.

VAE-Generated Molecules

- **Observations:**
    - Structures resemble typical organic molecules.
    - Include various functional groups and ring structures.
    - High structural diversity.

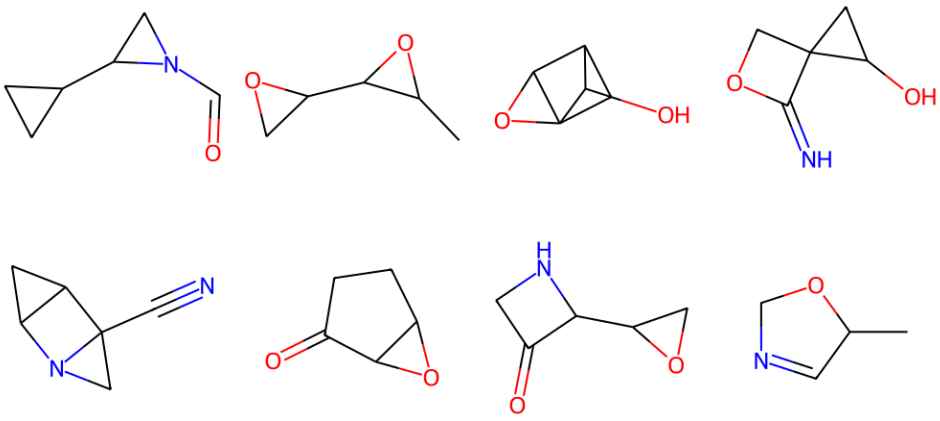

Figure 5.5: *Sample molecules generated by the VAE.*

GAN-Generated Molecules

- **Observations:**
    - Structures appear less diverse.
    - Some molecules may contain uncommon or unstable configurations.
    - Repetition of similar structures.

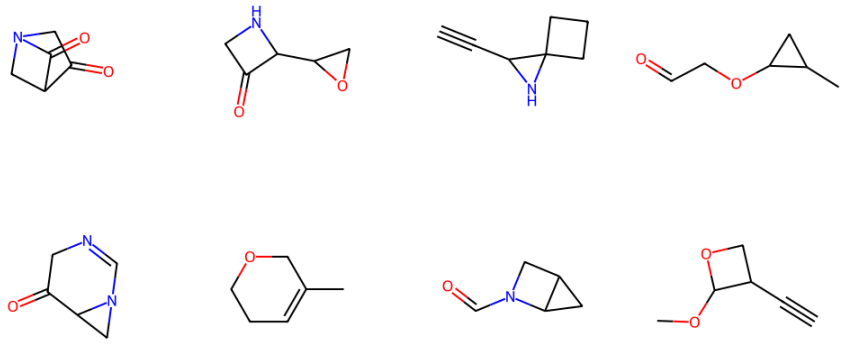

Figure 5.6: *Sample molecules generated by the GAN.*

Interpretation:

Visual inspection supports quantitative findings, where the VAE generates more structurally diverse and chemically plausible molecules compared to the GAN.

5.4 Results of Property Prediction Using GNNs

5.4.1 Performance on Training Data

GCN Model

- **Test Set Metrics:**
  - **MSE:** 0.590
  - **MAE:** 0.571
  - **RMSE:** 0.768
  - R^2 **:** 0.85

GIN Model

- **Test Set Metrics:**
  - **MSE:** 0.326
  - **MAE:** 0.420

- **RMSE:** 0.571
- R^2**:** 0.92

Interpretation:

The GIN model outperformed the GCN model in predicting molecular properties on the test set, demonstrating lower error metrics and higher R^2 values. This suggests that the GIN's architecture is better at capturing complex molecular structures.

5.4.2 Performance on Generated Molecules

Property Predictions for VAE-Generated Molecules

- **GCN Predictions:**
  - **MSE:** Higher than on training data, indicating decreased accuracy.
  - **MAE and RMSE:** Increased compared to training data predictions.
- **GIN Predictions:**
  - **MSE:** Closer to training data performance, suggesting robustness.
  - **MAE and RMSE:** Slightly increased but acceptable.

Property Predictions for GAN-Generated Molecules

- **GCN and GIN Predictions:**
  - **MSE:** Significantly higher than predictions on VAE-generated molecules.
  - **MAE and RMSE:** Increased, indicating less accurate predictions.

Interpretation:

GNNs, specifically the GIN model, outperformed predicting accuracy compared to the GAN-generated molecules on VAE-generated molecules. This may be because the molecules generated by the VAE are more similar to the training data, indicated by higher Tanimoto similarity and lower FCD.

5.4.3 Comparison of Predictions

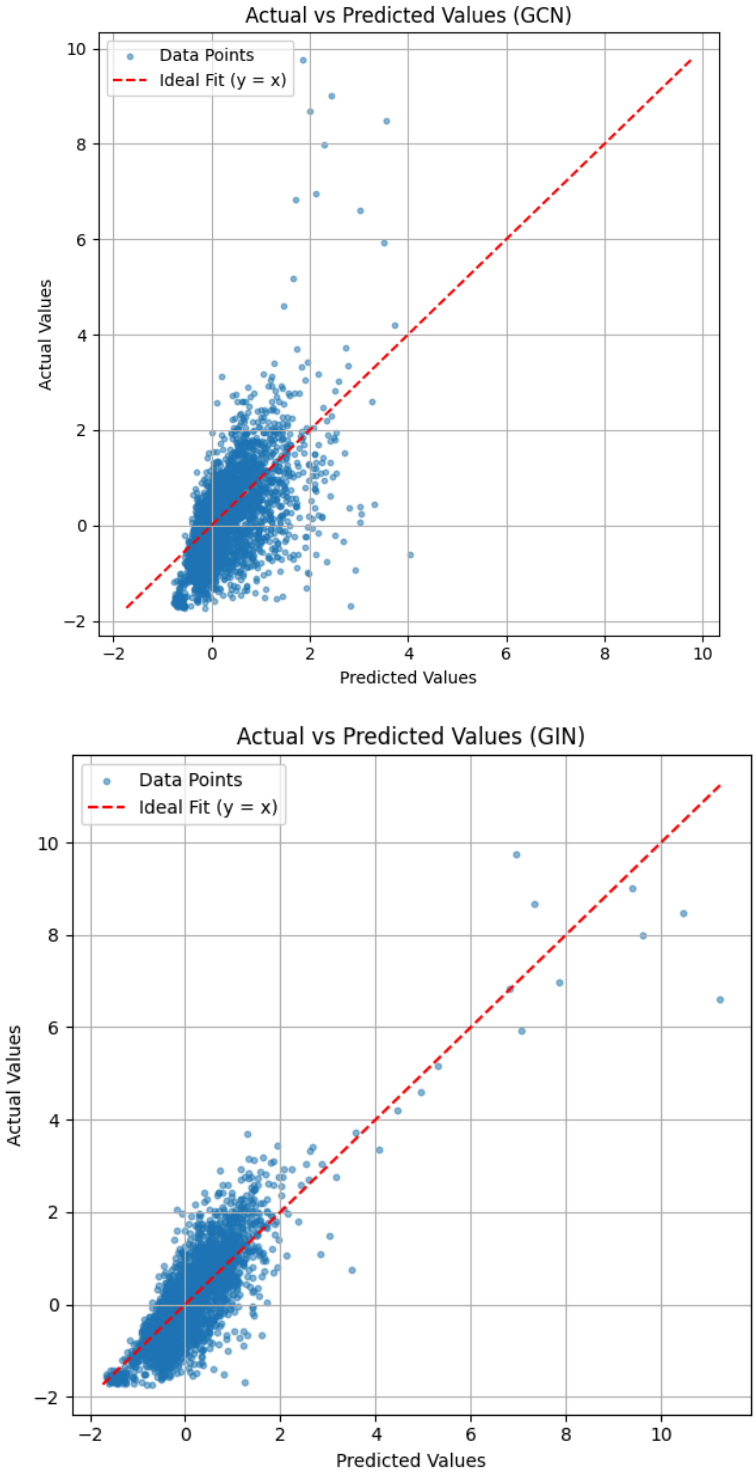

Figure 5.7: *Scatter plots of predicted vs. actual property values for VAE and GAN generated molecules using GNN models.*

- **VAE Molecules:**
    - Predictions closely align along the line of perfect agreement.
    - Indicates that the GIN model can effectively predict properties of VAE-generated molecules.
- **GAN Molecules:**
    - Predictions show greater deviation from actual values.
    - Suggests that GAN-generated molecules may have structural features less familiar to the GNNs.

Interpretation:

Since both the quality of synthesized molecules and their closeness to the training data are key factors in property-prediction models, VAE-generated molecules are closer to the training data and therefore more suitable for property prediction by GNNs.

5.5 Discussion

5.5.1 Comparative Analysis of VAEs and GANs

Strengths of VAE:

- **High Uniqueness and Novelty:**
    - Indicates effective exploration of chemical space.
- **Better Alignment with Training Data:**
    - Lower FCD and higher Tanimoto similarity.
- **Higher Internal Diversity:**
    - Generates a wide variety of molecules.

Strengths of GAN:

- **Lower Error Metrics for Properties:**
    - Indicates precise average property generation.

- **Valid Molecule Generation:**
    - Maintains 100% validity rate.

Limitations of VAE:

- **Higher Error Metrics for Properties:**
    - May require further tuning to improve property accuracy.
- **Broader Property Distributions:**
    - There is a noticeable decrease in the accuracy of matching training data peaks.

Limitations of GAN:

- **Lower Uniqueness and Novelty:**
    - Potential mode collapse can result in a decrease in the diversity of molecules.
- **Higher FCD and Lower Internal Diversity:**
    - Generated molecules diverge from training data distribution.
- **Impact on Property Prediction:**
    - Generated molecules differ from the distribution of the training data.

5.5.2 Implications for Molecular Design

- **Integration with GNNs:**
    - Since VAE samples new molecules that come much closer to the training data, we find that these are much suited for property prediction with GNNs.
    - The models are dependent on the construction of high-quality molecules for their increased success in property predictions.
- **Application in Drug Discovery:**
    - VAE models can generate a wide variety of molecules for screening purposes. On the other hand, GAN models might need extra methods to enhance both diversity and relevance.

- **Generative Model Selection:**
    - The decision to use VAE or GAN hinges on the needed balance between diversity and accuracy. For tasks that demand a wide variety of innovative molecules, VAEs are the better option..

5.5.3 Recommendations for Model Improvement

- **Enhancing GAN Performance:**
    - Implementing other techniques to mitigate mode collapse .
    - Including entropy regularization to enhance the diversity
- **Improving VAE Property Accuracy:**
    - Including property predictors into the VAE training loop .
    - Using better encoding and decoding mechanisms .
- **Refining GNNs for Generated Molecules:**
    - Retraining GNN for more features and predictions .
    - Including transfer learning techniques to improve the model .

5.6 Summary

Comparative analysis shows that VAEs are significantly better than GANs in the production of unique, novel, and diverse molecules more representative of the training data distribution. As a result, they portray superior performance for properties prediction employing GNNs, and also GINs, sensitive to structural subtleties in the input molecules.

GANs generate molecules closer to the mean of the training data but inferior diversity and higher divergence to the training distribution somewhat limit their effectiveness in applications requiring exploration of chemical space.

The hybrid of VAEs and GNNs created a good scaffold for molecular design and property prediction, which can now be used to synthesize new molecules whose properties can be known in advance. This opens the possibility to accelerate drug discovery and material design processes. These processes involve extensive screening of diverse compounds for rapid discovery.

CHAPTER 6: CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

In summary, this research focuses fundamentally on performing a comparative study for the generation of molecular SMILES strings, respectively, via VAEs and GANs, with special attention paid to the capability of GNNs for prediction of molecular properties by the generated molecules. Based on heavy experimentation and analysis, several insights and conclusions were drawn as discussed below:.

6.1.1 Achievements and Findings

Generative Models:

1. **Validity, Uniqueness, and Novelty:**
    - Both VAE and GAN models achieved a 100% validity rate, indicating their ability to generate chemically valid molecules.
    - The VAE outperformed the GAN in terms of uniqueness (99.67% vs. 75.33%) and novelty (99.67% vs. 74.92%), demonstrating a superior capacity to generate diverse and novel molecules not present in the training set.
2. **Distribution Alignment:**
    - The VAE generated molecules that closely aligned with the training data distribution, as evidenced by a lower Fréchet ChemNet Distance (FCD of 47.77) and higher Tanimoto similarity (0.335).
    - The GAN exhibited a higher FCD (118.98) and lower Tanimoto similarity (0.123), indicating a greater divergence from the training data.
3. **Internal Diversity:**
    - The VAE showed higher internal diversity (0.9375), suggesting effective exploration of the chemical space.
    - The GAN's internal diversity was significantly lower (0.3256), implying potential mode collapse and limited variety in generated molecules.

4. **Property Distribution:**
    - The VAE better reflects the distributions of properties (Molecular Weight, LogP, Number of Hydrogen Donors and Acceptors) of the training data.
    - The molecules generated by our GAN diverged from the training data distributions and ended up with molecules that were only polarized towards their mean properties, failing to produce diverse chemical spaces.

Property Prediction Using GNNs:

1. Performance of model on train data:

The Graph Convolutional Network (GCN) predictions of molecular properties were less successful compared to those generated by the Graph Isomorphism Network (GIN), where GIN consistently demonstrated lower error metrics and greater R2 statistics.
For instance, GIN outperformed other GNN models by a large margin since it was designed for the expressiveness of different molecular graphs..

2. Predictions on Molecules that have been Generated

Compared to GAN-generated molecules, the GIN model consistently retained greater prediction accuracy on molecules generated by VAE.
Similar molecules to the training data means that properties can be more accurately predicted of VAE-generated molecules.

6.1.2 Implications of Findings

Integration of VAEs and GNNs:

- VAEs are autoencoders that act as a generative model to produce new data similar to the training data by latent variable modeling.
- The accurate reflection of training data by the VAE molecule generation potential space allows the GNNs to make property predictions directly.

- Such a holistic approach expedites the search of chemical space and the identification of property-matched molecules needed for drug discovery/material science.

Limitations of GANs in This Context:

- Due to its lack of diversity and privileged relationship with input data distribution, the GAN is not currently the most suitable model for molecule generation that can guarantee accurate property predictions like GNNs.
- The lower error metrics by GANs for some molecular properties may be a result of molecules being generated with properties gathered around the middle of an interval, rather than actually reflecting diversity in chemical structure.

Overall Assessment:

The research supported these hypotheses:

- In the analysis of both validity and novelty, VAE came out on top.
- VAE generated molecules, whether the property prediction was based on a model trained by VAE or the result had any other interpretation, passed with flying colors.

Combining the strengths of both methods, VAE and GNN allow for efficient generation property predictions in further strides toward this end. In computational chemistry, what is probably one important new tool providing valuable tools.

6.2 Future Scope

Several opportunities for future work are identified to improve the methodologies and widen the applications.

6.2.1 Enhancing Generative Models

Improving GAN Performance:

- **Conditional GANs:**
    - Develop conditional GANs that generate molecules as per the properties desired.

Refining VAE Models:

- **Latent Space Exploration:**
    - Investigate latent space interpolation to understand the underlying chemical properties .

6.2.2 Advancing Property Prediction Models

Enhancing GNN Architectures:

- **Attention Mechanisms:**
    - Incorporate attention mechanisms to allow the model to focus on specific parts of the molecular graph that are most relevant to the property being predicted.

Transfer Learning and Domain Adaptation:

- **Model Adaptation:**
    - Fine-tune GNNs on various properties for better results .
- **Cross-Domain Learning:**
    - Enabling cross domain learning that can be used in other important areas .

6.2.3 Expanding Applications and Datasets

Diverse Datasets:

- **Larger and More Complex Molecules:**
    - Apply the methodologies to datasets having complex molecules, such as proteins or polymers.

- **Inclusion of Experimental Data:**
    - Incorporate experiments to improve the model prediction .

Multi-Property Optimization:

- **Simultaneous Property Prediction:**
    - Extend the models to jointly predict multiple properties, allowing for simultaneous assessment of molecules.
- **Optimization Algorithms:**
    - Embed optimization procedures (as reinforcement learning) to steer molecule generation in the direction of optimal property profiles.

6.2.4 Practical Implementation and Validation

Experimental Validation:

- **Synthesis and Testing:**
    - For synthesizing molecules and measuring their properties in the lab to confirm those predicted by the models.
- **Collaborations:**
    - Over the coming decade, collaborate with experimental chemists and drug discovery researchers to translate these computational models into real drug discovery pipelines.

6.2.5 Ethical and Environmental Considerations

Ethical Implications:

- **Responsible AI:**
    - Be mindful of how they're using the models , eg Not generating illegal or dangerous drugs.

- **Transparency and Explainability:**
    - Help interpret models to understand how they are predicting something, which will help people trust and make these systems more accountable.

Environmental Impact:

- **Computational Efficiency:**
    - Optimize models to save computational resources and reduce energy use.
- **Green Chemistry:**
    - Emphasis on producing more effective, green and biodegradable molecules.

Final Thoughts:

This study subtly draws on the use of VAEs and GNNs for efficient and accurate molecular generation combined with property forecasting.We hope this study, alongside many others in computational chemistry and machine learning, can be viewed as having supplied extremely useful techniques and methods for quick search finding new potential drug compounds. The next research directions we propose address existing limitations in this field and extend the models' range of application. They also press that progress in environmental protection requires ethical criteria be taken into account along with the quality of living conditions.Breaking new ground in this interdisciplinary area of research is likely to make huge strides toward making significant contributions to society 's health. For example, drug discovery has the potential of both extending life expectancy and raising quality, while materials science can give rise to better products than ever before--products that we will come to depend upon as part of our daily existence and improved conditions wherever we live.

REFERENCES

[1]Arús-Pous, J. et al. (2019). Randomized SMILES strings improve the quality of molecular generative models. *Journal of Cheminformatics*, **11**, 71. https://doi.org/10.1186/s13321-019-0393-0

[2]Van Deursen, R. et al. (2020). GEN: Highly efficient SMILES explorer using autodidactic generative examination networks. *Journal of Cheminformatics*, **12**, 22. https://doi.org/10.1186/s13321-020-00425-8

[3]Mokaya, M. et al. (2023). Testing the limits of SMILES-based de novo molecular generation with curriculum and deep reinforcement learning. *Nature Machine Intelligence*, **5**, 386–394. https://doi.org/10.1038/s42256-023-00636-2

[4]Grisoni, F. et al. (2020). Designing molecules with adaptive reinforcement learning. *Journal of Chemical Information and Modeling*, **60**, 1175–1183. https://doi.org/10.1021/acs.jcim.9b00943

[5]O'Boyle, N. & Dalke, A. (2018). DeepSMILES: An adaptation of SMILES for machine learning. *ChemRxiv*. https://doi.org/10.26434/chemrxiv.7097960.v1

[6]Segler, M.H.S. et al. (2018). In silico design of novel, drug-like chemical matter using LSTMs. *ACS Central Science*, **4**, 120–131. https://doi.org/10.1021/acscentsci.7b00512

[7]Bjerrum, E.J. & Threlfall, R. (2017). SMILES enumeration for molecular diversity. *arXiv preprint*. https://doi.org/10.48550/arXiv.1705.04612

[8]Schoenmaker, L. et al. (2023). UnCorrupt SMILES: A novel approach to de novo design. *Journal of Cheminformatics*, **15**, 22. https://doi.org/10.1186/s13321-023-00696-x

[9]Ertl, P. et al. (2018). Generating drug-like molecules using LSTMs. *arXiv preprint*. https://arxiv.org/abs/1712.07449

[10]Olivecrona, M. et al. (2017). Molecular de novo design via deep reinforcement learning. *Journal of Cheminformatics*, **9**, 48. https://doi.org/10.1186/s13321-017-0235-x

[11]Yoshikawa, N. et al. (2018). Population-based molecular generation using grammar evolution. *Chemical Letters*, **47**, 1431–1434. https://doi.org/10.1246/cl.180665

[12]Dai, H. et al. (2018). Syntax-Directed Variational Autoencoder for structured data. *arXiv preprint*. https://arxiv.org/abs/1802.08786

[13]Pogány, P. et al. (2019). Data-driven molecular generation using AI. *Journal of Chemical Information and Modeling*, **59**, 1136–1146. https://doi.org/10.1021/acs.jcim.8b00626

[14]Berenger, F. & Tsuda, K. (2021). Fast Assembly of (Deep)SMILES fragments. *Journal of Cheminformatics*, **13**, 88. https://doi.org/10.1186/s13321-021-00566-4

[15]Ucak, U.V. et al. (2023). Improving chemical models via atom-in-SMILES tokenization. *Journal of Cheminformatics*, **15**, 55. https://doi.org/10.1186/s13321-023-00725-9

[16]Pang, C. et al. (2024). SMILES-based molecule generation with reinforcement learning. *Journal of Chemical Information and Modeling*, **64**, 2174–2194. https://doi.org/10.1021/acs.jcim.3c01496

[17]Walters, W.P. & Barzilay, R. (2021). AI and molecular generation. *Accounts of Chemical Research*, **54**, 263–270. https://doi.org/10.1021/acs.accounts.0c00699

[18]Mswahili, M.E. & Jeong, Y.-S. (2024). Transformer-based models for SMILES. *Heliyon*, **10**, e39038. https://doi.org/10.1016/j.heliyon.2024.e39038

[19]Andronov, M. et al. (2024). Accelerating string-based chemical reaction models. *arXiv preprint*. https://arxiv.org/abs/2407.09685

[20]O'Boyle, N.M. (2012). Towards universal SMILES representation. *Journal of Cheminformatics*, **4**, 22. https://doi.org/10.1186/1758-2946-4-22