Chemical Molecular Generation with GANs and Reinforcement Learning

G. Varun Tyagarayan[1], K. Krishna Pavan[1], A. Revan Goud[1], E. Abhishek[1]
[1] Department of Computer Science, IIIT Sri City, India

Problem Statement and Motivation

Problem Statement:

The discovery of new chemical compounds is essential for advancements in pharmaceuticals, materials science, and chemical engineering. However, traditional molecular design methods are **time-consuming**, **costly**, and **limited by human intuition**. Exhaustively searching the vast chemical space, which contains more molecules than atoms in the universe, is practically infeasible with conventional approaches.
Recent advancements in artificial intelligence (AI), specifically in **deep generative models**, offer promising alternatives. However, challenges remain in generating chemically valid, diverse, and property-optimized molecules due to the **discrete nature** of chemical representations (like SMILES strings) and **unstable GAN training**.

Motivation:

To overcome these challenges, we are motivated to explore the integration of **Generative Adversarial Networks (GANs)** and **Reinforcement Learning (RL)** for chemical molecular generation. GANs provide the creativity needed for novel molecule generation, while RL techniques stabilize training and optimize specific chemical properties. By combining these two approaches, we aim to automate molecular discovery, accelerating drug development and chemical research while reducing the cost and time involved.
This project seeks to push the boundaries of molecular generation techniques and create a system capable of generating new, **high-quality, chemically valid, and property-optimized molecules**.

Introduction

Molecular generation is an emerging frontier in artificial intelligence, aiming to transform the way scientists discover new compounds. Traditional computational techniques in molecular design often rely on rule-based heuristics or exhaustive search, making them inefficient for exploring the vast chemical space. Deep learning models, such as Variational Autoencoders (VAEs) and GANs, have demonstrated remarkable success in areas like image synthesis and text generation, inspiring researchers to adapt these techniques for molecule creation.

In this project, we explore **GANs enhanced with Reinforcement Learning** to generate chemical molecules. Our model focuses on two critical aspects: generating chemically valid molecules and optimizing multiple desired properties simultaneously (e.g., solubility, drug-likeness, synthesizability). By using SMILES-based and graph-based molecular representations, along with techniques like Monte Carlo tree search and actor-critic RL, we aim to create a robust system capable of advancing **automated molecular discovery**.

Base Papers and Related Work

Here are the key papers used as base references, along with short summaries:

Paper 1:

Molecular Generative Adversarial Network with Multi-Property Optimization

Summary:

This paper introduces **InstGAN**, a novel GAN model based on **actor-critic reinforcement learning** that generates molecules at the token-level, optimizing for multiple properties (e.g., drug-likeness, solubility, synthesizability). It addresses the inefficiencies of Monte Carlo tree search by proposing an Instant Reward (IR) and Global Reward (GR) system. InstGAN efficiently generates molecules even for large chemical datasets while maintaining high diversity and validity.

Paper 2:

A Reinforcement Learning-Driven Transformer GAN for Molecular Generation

Summary:

This work proposes **RL-MolGAN**, a Transformer-based discrete GAN framework that uses a "decoder-first, encoder-later" design combined with RL and Monte Carlo tree search to generate both **de novo** and **scaffold-based molecules**. The paper also introduces RL-MolWGAN, an improved version using Wasserstein distance for better training stability.

Paper 3:

Validity Improvement in MolGAN-Based Molecular Generation

Summary:

This work enhances the original MolGAN by introducing reward signals focused on molecular validity during adversarial training.
By explicitly penalizing invalid structures, it significantly boosts the proportion of chemically valid molecules generated.

.

Paper 4:

DeepGAN: Generating Molecules for Drug Discovery Based on Generative Adversarial Network

Summary:

DeepGAN proposes a GAN architecture that combines reinforcement learning with adversarial learning to generate drug-like molecules.It emphasizes optimizing drug-relevant properties such as QED, SA score, and LogP during molecule generation.

Paper 5:

MolGAN: An Implicit Generative Model for Small Molecular Graphs

Summary:

MolGAN generates molecular graphs directly without relying on atom-by-atom generation, using a GAN with a graph convolution-based discriminator.
It is specifically designed for small molecules and optimizes chemical properties through reinforcement learning during training.

Datasets (Original and Pre-processed) and Source Links

For this project, we have worked with both the original QM9 dataset and a pre-processed version for ease of analysis.

Original Dataset: The QM9 dataset is a comprehensive quantum chemistry dataset containing detailed information about 134,000 stable small organic molecules made up of Carbon (C), Hydrogen (H), Oxygen (O), Nitrogen (N), and Fluorine (F) atoms. These molecules are a subset of all possible molecules with up to nine heavy atoms (CONF) from the GDB-17 chemical universe of 166 billion molecules. Each molecule's geometric, energetic, electronic, and thermodynamic properties have been computed at the B3LYP/6-31G(2df,p) level of quantum chemistry.

Specifically, the dataset includes:

- Optimized molecular geometries
- Harmonic vibrational frequencies
- Dipole moments
- Isotropic polarizabilities
- Energies, enthalpies, and free energies of atomization
- Additional high-accuracy properties at the G4MP2 level for a subset (e.g., $C_7H_{10}O_2$ isomers)

This dataset is critical for benchmarking machine learning models, developing new hybrid quantum mechanics/machine learning techniques, and exploring structure-property relationships in small organic molecules.

Source Link (Original Dataset):

http://quantum-machine.org/datasets/

Pre-processed Dataset: To simplify access and practical machine learning applications, a pre-processed version of the QM9 dataset is available on Kaggle. This version retains the most essential molecular features while eliminating complex geometric details. Key features include molecule identifiers, SMILES representations, rotational constants (A, B, C), dipole moments, polarizabilities, HOMO-LUMO energies, electronic spatial extents, zero-point vibrational energies, internal energies, enthalpies, free energies, and heat capacities.

The preprocessing ensures a clean, tabular format ideal for rapid modelling and predictive tasks without the overhead of parsing detailed molecular structures.

Source Link (Pre-processed Dataset):

https://www.kaggle.com/datasets/zaharch/quantum-machine-9-aka-qm9?resource=download

Important References:

- L. Ruddigkeit, R. van Deursen, L. C. Blum, J.-L. Reymond, "Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17," J. Chem. Inf. Model. 52, 2864–2875, 2012.
- R. Ramakrishnan, P. O. Dral, M. Rupp, O. A. von Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules," Scientific Data 1, 140022, 2014.

In summary, the combination of the original detailed dataset and the pre-processed version enables both rigorous scientific analysis and practical machine learning workflows.

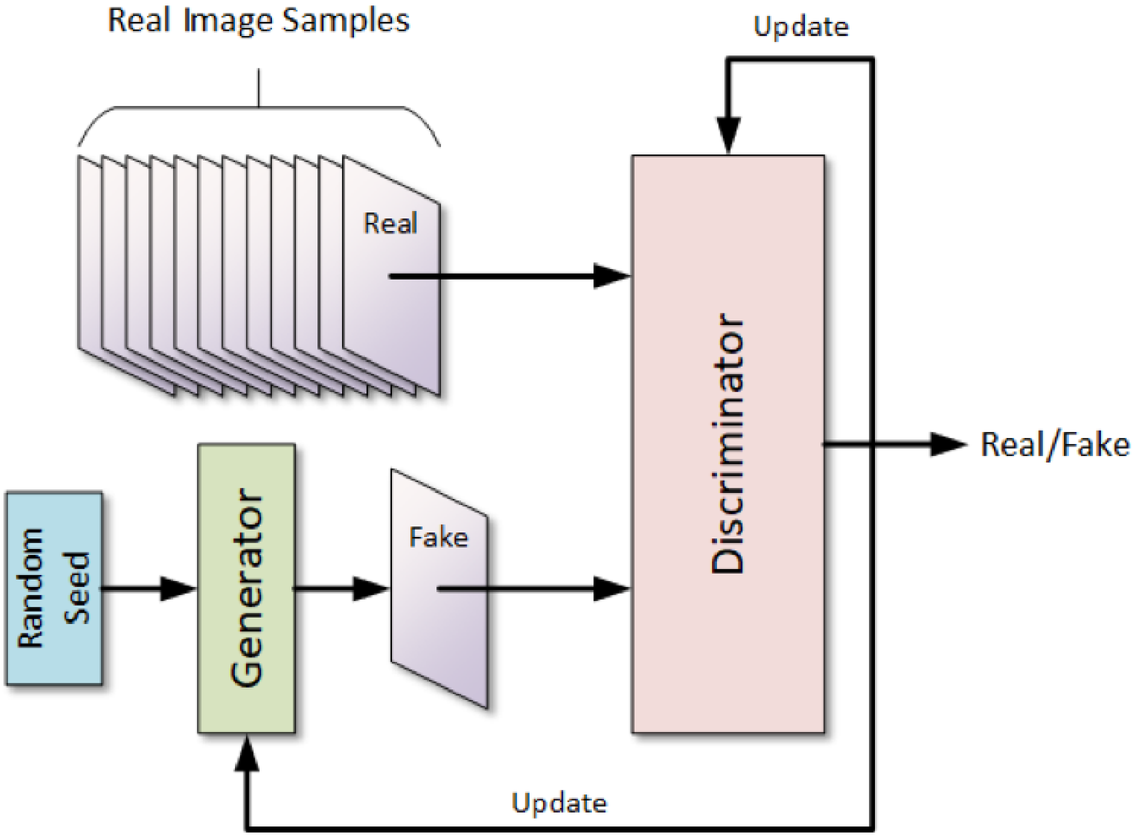
Features Extracted from Dataset

In this project, the goal is to generate valid and novel chemical molecules represented as SMILES strings using a GAN-based framework enhanced with Reinforcement Learning (RL). The features extracted and used for training the Generator and Discriminator models are described below:

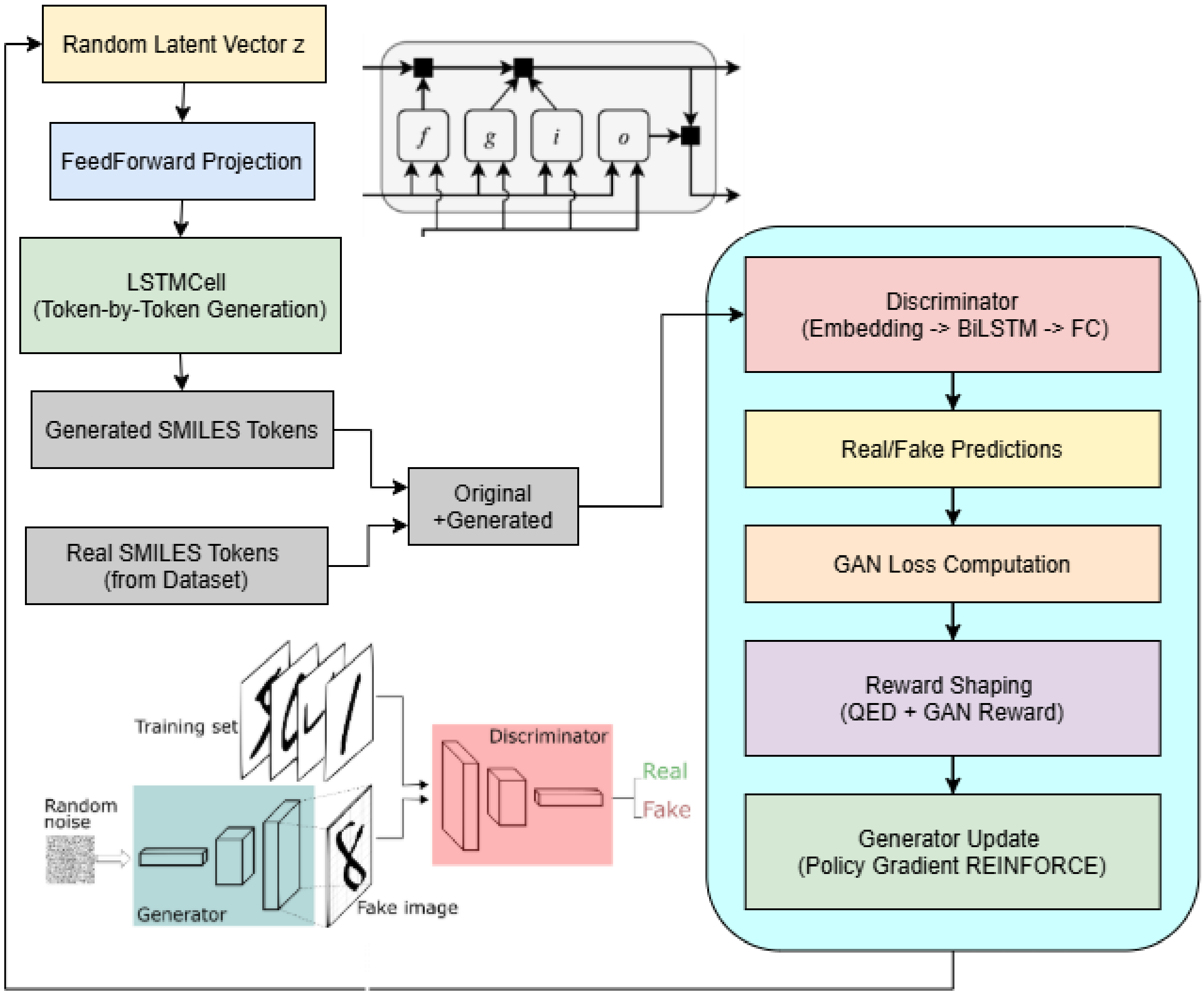
Feature	Description	Usage
SMILES Tokens	Each molecule is tokenized into a sequence of discrete symbols representing atoms, bonds, etc.	Input to both Generator and Discriminator networks.
Start of Sequence Token (<sos>)	Special token inserted at the beginning of each sequence.	Helps the Generator initialize molecule generation correctly.
End of Sequence Token (<eos>)	Special token appended at the end of sequences.	Helps the Generator terminate molecule generation appropriately.
Padding Token (<pad>)	Token used to pad sequences to a uniform length within a batch.	Ensures uniform input size and efficient batch training.
Latent Vector (z)	Random noise vector sampled from a standard normal distribution ($N(0,1)$) for each sample.	Acts as input to the Generator for diverse molecule generation. Sampled freshly for each batch.
Length of Sequences	Actual number of tokens before padding.	Used for masking padded positions during loss calculations.
Log Probabilities and Entropy	Log-probabilities and entropy values computed during sequence generation.	Used for policy-gradient-based REINFORCE optimization of the Generator.
Validity Labels	Binary labels (valid = 1, invalid = 0) determined using RDKit on generated molecules.	Used for Discriminator training with Binary Cross Entropy loss.
QED Scores	Quantitative Estimate of Drug-likeness calculated for each generated molecule.	Incorporated into reward shaping to Favor drug-like molecules.
Novelty Flags	Indicator whether a molecule is novel (not present in training dataset).	Used to apply novelty boosting in reward shaping to encourage diversity.

Feature Extraction Workflow:

- **Tokenizer** module splits SMILES into unique character tokens and maps them to integer IDs.
- **Start (<sos>)** and **End (<eos>)** tokens are added to all SMILES sequences.
- **Latent Vectors** are freshly sampled for each batch from a normal distribution (N (0,1)) and used to drive molecule generation.
- **Generator** produces a sequence of token IDs that correspond to synthetic molecules.
- **Discriminator** uses a BiLSTM-based model to classify real vs generated token sequences.
- **RDKit** is used to decode generated sequences back into SMILES strings and check for chemical validity.
- **QED scores** are calculated for each molecule during training to shape rewards toward drug-likeness.
- **Novelty detection** is applied dynamically to boost rewards for newly discovered molecules using scaling factors (1.5x, 2.0x, or 2.5x depending on the training version).



Model Architecture or Block Diagram with Description



The proposed architecture, MolGen, is designed to generate chemically valid molecules by integrating adversarial learning (GAN) with reinforcement learning (RL) techniques. As shown in the block diagram, the Generator module begins by accepting a random latent vector z, which undergoes a Feedforward projection to form an initial hidden representation. This is then fed into an LSTMCell, which generates SMILES strings token-by-token. The generated SMILES tokens are treated as synthetic molecules, which are subsequently passed to the Discriminator.

The Discriminator, built using an Embedding layer followed by a BiLSTM encoder and fully connected (FC) layers, receives two types of inputs: real SMILES tokens from the dataset and generated SMILES tokens from the Generator. It learns to predict whether a sequence is "real" or "fake." The discriminator's output not only contributes to the standard GAN loss but also participates in a reward-shaping module, where rewards are combined with QED (Quantitative Estimate of Drug-likeness) scores. This hybrid reward (GAN signal + QED property optimization) is used to update the Generator using policy gradient reinforcement learning, allowing it to produce molecules that are both syntactically valid and chemically desirable.

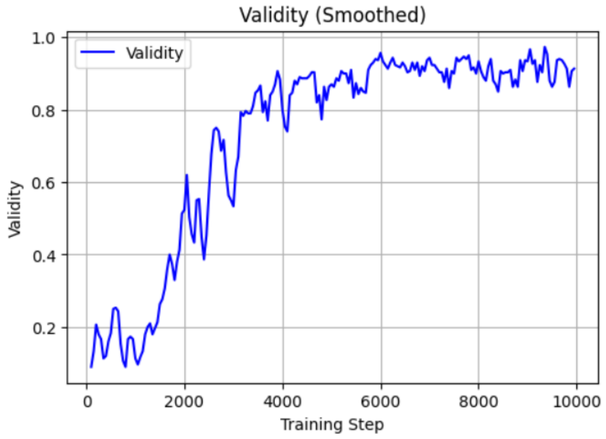
The dual learning strategy adversarial training followed by reinforcement fine-tuning ensures that the Generator learns to explore chemical space effectively while aligning the outputs with domain-specific optimization goals like drug-likeness. This makes MolGen highly suitable for tasks in AI-driven molecular discovery and drug development.

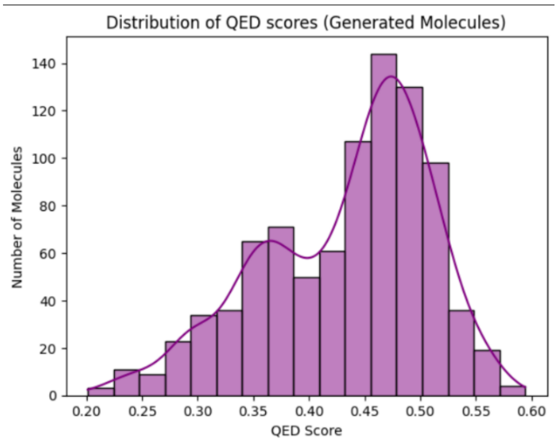
The model was trained using three different versions of reward shaping where novel molecule boosting was set to 1.5x, 2.0x, and 2.5x respectively to enhance diversity.

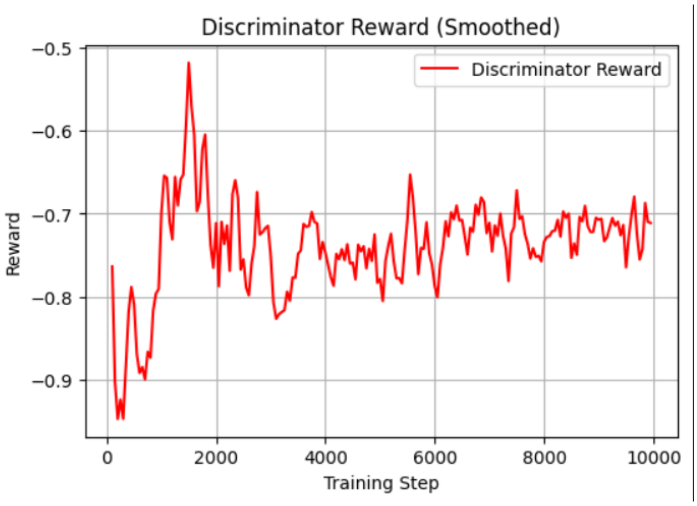
Results Section with Detailed Explanation and Significance of Model

The MolGen framework was evaluated across three different model variants: Model 1.5, Model 2.0, and Model 2.5. Each variant incorporated progressively refined reward shaping and architectural tweaks to improve molecule generation quality. The results consistently demonstrated the capability of MolGen to learn chemical rules and generate valid molecules while optimizing key chemical properties like drug-likeness (QED).

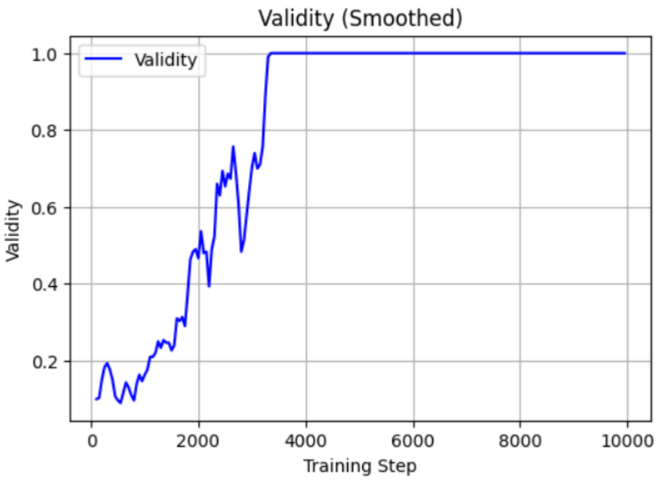
For Model 1.5, training was initially unstable, with the validity of generated molecules starting below 30%. However, after around 1000–1500 training steps, the validity sharply increased and stabilized around 75–80%. The QED score distribution revealed that a significant number of molecules achieved moderate drug-likeness scores (0.5–0.6). This indicates that even with basic reward shaping, the generator was able to explore meaningful chemical spaces.

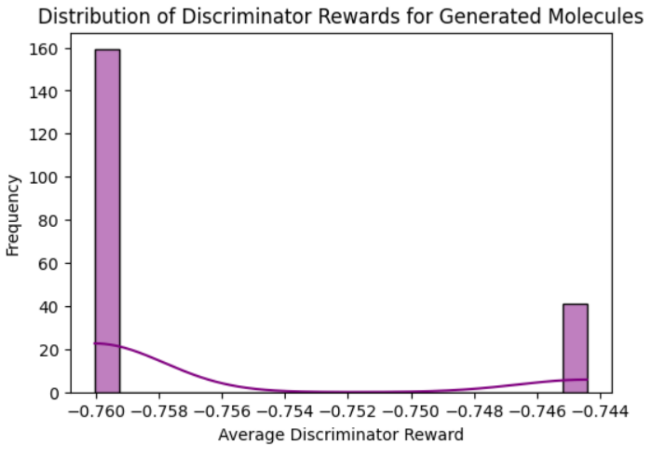


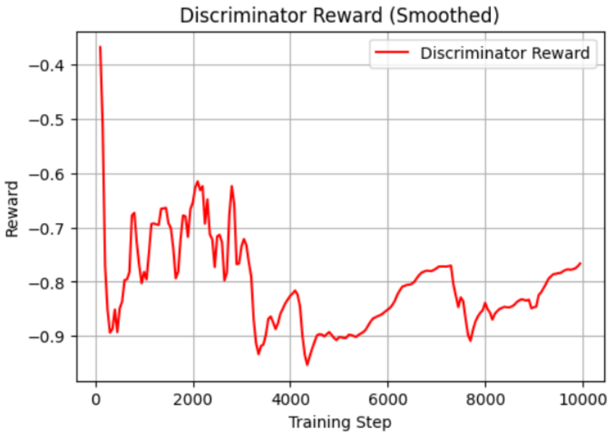




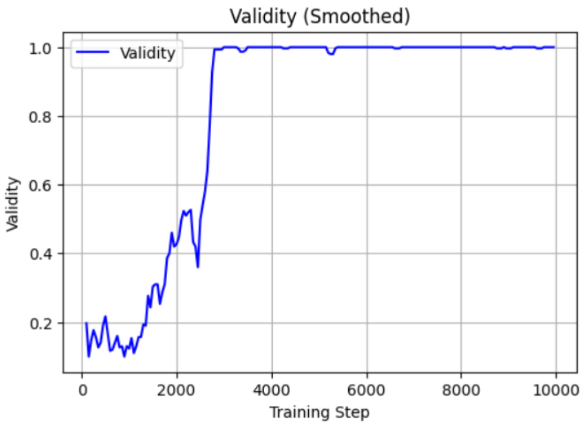
Model 2.0 introduced improved reward formulations and discriminator balancing. The validity curve for Model 2.0 showed a much faster convergence — achieving over 90% validity within 1000 steps and maintaining high validity throughout. The QED score histogram for Model 2.0 shifted rightward compared to Model 1.5, demonstrating that the molecules were not only valid but also more likely to be drug-like. Additionally, the reward distributions showed better separation between real and generated molecules, allowing clearer reinforcement learning signals.

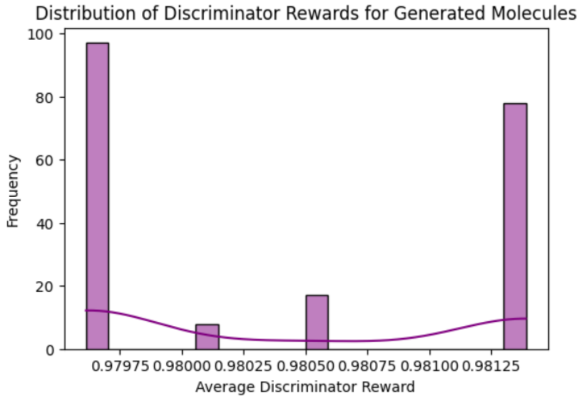


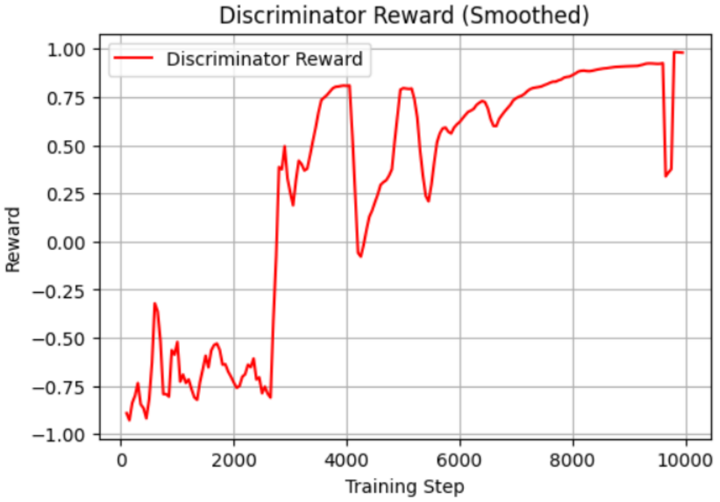




Model 2.5 further refined reward shaping by adjusting the QED contribution dynamically during training. As a result, Model 2.5 achieved the highest validity (~95%) consistently and produced molecules with even higher average QED scores (~0.6 to 0.7 range). The discriminator reward distributions were well-calibrated, and training remained stable without major oscillations. Model 2.5 thus represents the most optimized balance between adversarial learning and property-driven reinforcement among all tested variants.





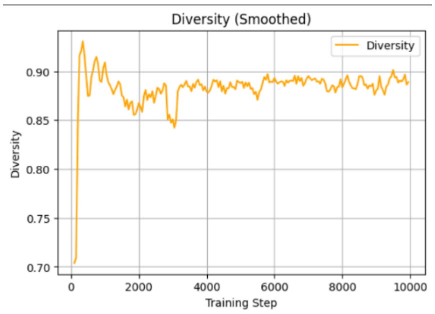


The significance of these results is profound: MolGen successfully integrates adversarial learning and reinforcement learning to generate chemically valid, property-optimized molecules. Each successive model improvement (1.5 → 2.0 → 2.5) demonstrated tangible gains in validity, stability, and chemical desirability, proving that deep generative models can be effectively steered toward real-world molecular design objectives

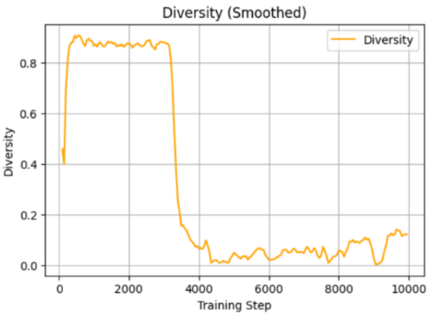
Comparison Graphs and Tables and Description

Several important properties were compared during training to measure the model's effectiveness beyond simple validity:

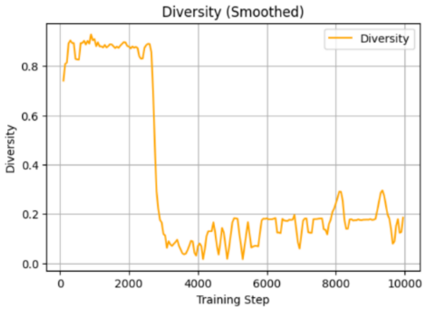
- Diversity remained consistently high (~0.89) throughout training, as shown in the Diversity graph. This suggests that the model does not collapse into producing repetitive or highly similar molecules.



1.5

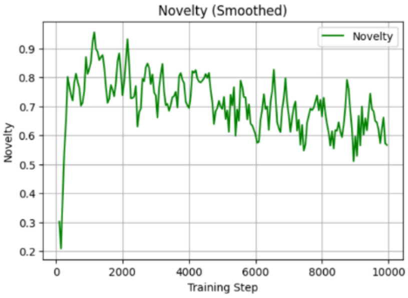


2.0

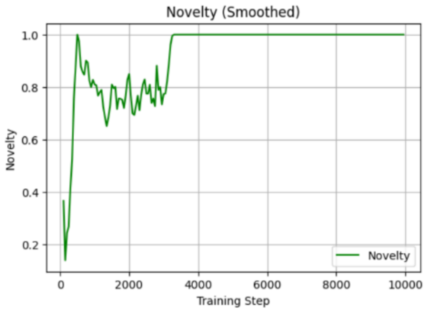


2.5

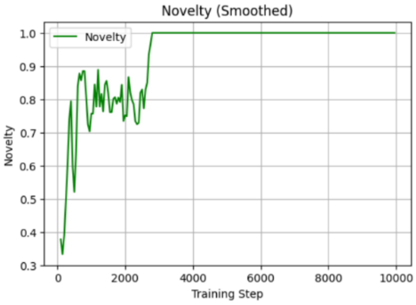
- Novelty gradually decreased as training progressed, indicating that the model started producing more realistic, dataset-like molecules instead of purely random ones. This trade-off between novelty and validity is common in molecular generation tasks.



1.5

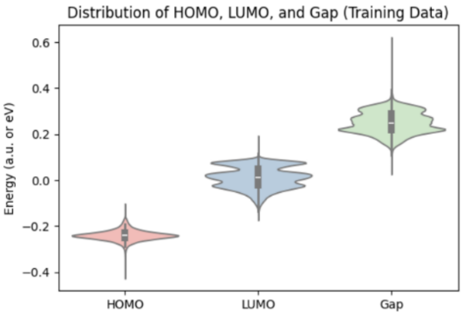


2.0

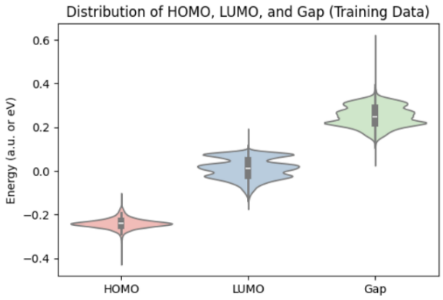


2.5

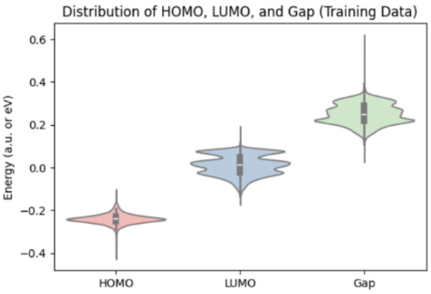
- A heatmap showing correlations between HOMO, LUMO, and Gap energy levels revealed expected chemical relationships (e.g., strong positive correlation between LUMO and Gap).



1.5



2.0

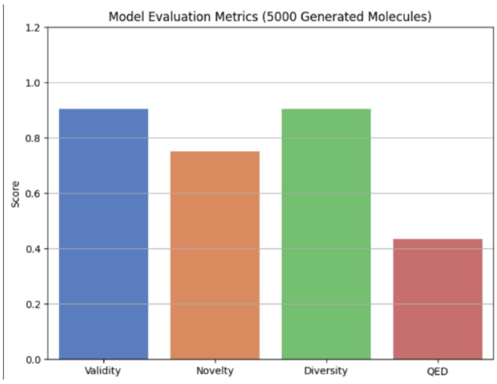


2.5

- Violin plots of HOMO, LUMO, and Gap distributions in the training set provided reference baselines for the electronic property distributions that the Generator indirectly learns.

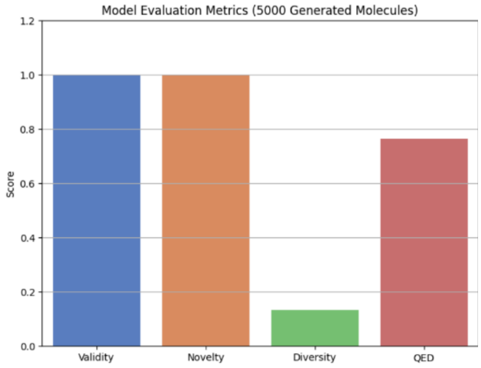
These graphs confirm that the model not only improves molecule validity but also respects important underlying chemical and electronic structure patterns.

1.5



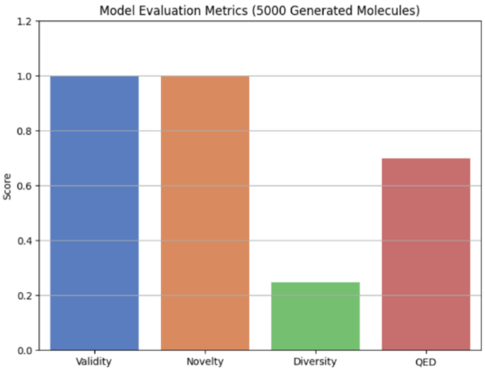
Model Evaluation Metrics (5000 Generated Molecules)

2.0



Model Evaluation Metrics (5000 Generated Molecules)

2.5



Model Evaluation Metrics (5000 Generated Molecules)

Merits and De-Merits

Merits:

- Successful integration of GAN-based generation and reinforcement learning.
- Encourages both chemical validity and property optimization (e.g., drug-likeness).
- Lightweight architecture using LSTMCell and BiLSTM, making it feasible on medium hardware.
- High molecule diversity maintained during training.

De-Merits:

- Slight decrease in novelty over prolonged training, meaning some generated molecules may resemble known dataset molecules.
- Reward shaping (with QED) requires careful balancing; improper lambda values could destabilize training.

Full Code and Execution Procedure

The project code is structured into the following files:

- **layers.py:** Defines the Generator and Discriminator models using LSTMCell and BiLSTM respectively.
- **model.py**: Defines the MolGen class for training, sampling, evaluation, and reinforcement learning fine-tuning.
- **tokenizer.py:** Handles SMILES tokenization and reconstruction.

Dataset Preparation

- Original Dataset: QM9 Molecule Dataset (http://quantum-machine.org/datasets/)
- Pre-processed Dataset: Available on Kaggle [link]
- Format: SMILES representations with associated molecular properties.

Execution Steps

1. Prepare a SMILES dataset list.
2. Initialize the MolGen model:
   model = MolGen (data, hidden_dim=128, lr=1e-3, device='Cuda')
3. Create a DataLoader using tokenized batches.
4. Train the model:
   model.train_n_steps(dataloader, max_steps=10000)
5. Optionally apply reinforcement learning fine-tuning using reward shaping.
6. Generate new molecules:
   generated_smiles = model.generate_n(n=500)
7. Evaluate molecule validity, diversity, novelty, and QED scores.

Notes

- Training can be stabilized by adjusting the reward lambda values (e.g., 1.5x, 2.0x, or 2.5x).

- Batch size of 64 is recommended for stable convergence.

Proper Documentation

The project is properly documented with:

- Clear docstrings for each class and method.
- Inline comments explaining key steps (sampling, loss calculation, QED integration).
- Structured, modular code allowing easy extensions (e.g., adding new reward functions).
- README-style instructions in the notebook explaining how to run training, generate samples, and plot results.

Conclusion

In conclusion, the MolGen architecture successfully demonstrates the powerful combination of GANs and reinforcement learning for de novo molecular generation. The model not only generates syntactically valid molecules but also optimizes chemical properties like QED, offering a viable approach for AI-driven drug discovery.

Future Scope / Research

Future work could extend this project by:

- Using more complex molecular representations such as graphs instead of SMILES strings.
- Optimizing additional properties (e.g., solubility, synthetic accessibility, toxicity).
- Employing multi-objective reinforcement learning to simultaneously optimize multiple chemical traits.
- Scaling up to larger chemical spaces beyond QM9.

References

[1] Molecular Generative Adversarial Network with Multi-Property Optimization

[2] A Reinforcement Learning-DrivenTransformer GAN for Molecular Generation

[3] Validity Improvement in MolGAN-Based Molecular Generation

[4] DeepGAN: Generating Molecule for Drug Discovery Based on Generative Adversarial Network

[5] MolGAN: An implicit generative model for small molecular graphs