MORLD - Chemical Molecule Generation Using LSTM and Reinforcement Learning

Indian Institute of Information Technology, Sri City

Abstract
This project explores generative modeling of chemical molecules through deep learning techniques. An LSTM network is trained to learn SMILES strings from a curated subset of the ZINC database. Reinforcement learning (RL) is subsequently employed to fine-tune the model toward generating molecules with optimized chemical properties. This approach facilitates the discovery of novel, valid, and property-optimized drug-like molecules.

1 Problem Statement and Motivation

The pharmaceutical industry faces persistent challenges in discovering novel chemical compounds with desirable biological activity. Traditional discovery methods are often time-consuming, resource-intensive, and costly.

Motivation: This project aims to leverage machine learning—specifically deep learning and reinforcement learning—to automate and enhance the molecular design process, providing innovative pathways to accelerate drug development.

Authors

- Edara Sri Sai Kaushal
- Perisetty Satya Amrutha Raja
- Are Ram Gopal Reddy
- Nilesh Ganja

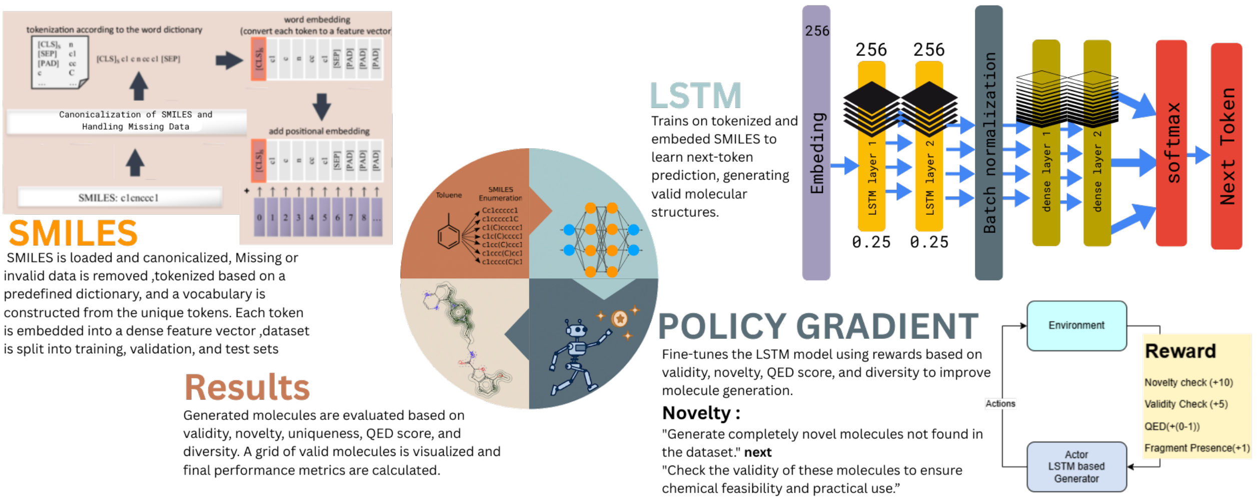
2 Introduction

SMILES (Simplified Molecular Input Line Entry System) is a text-based representation of chemical structures. Long Short-Term Memory (LSTM) networks can be trained to predict and generate valid SMILES strings. Further fine-tuning using reinforcement learning (RL) allows the model to generate molecules that optimize certain properties, such as drug-likeness or the presence of specific chemical fragments.

3 End-to-End Project Workflow

The following is the step-by-step architecture of the project pipeline, highlighting how data flows from raw SMILES strings to molecule generation and evaluation.

Workflow Diagram:



SMILES

SMILES is loaded and canonicalized, Missing or invalid data is removed ,tokenized based on a predefined dictionary, and a vocabulary is constructed from the unique tokens. Each token is embedded into a dense feature vector ,dataset is split into training, validation, and test sets

LSTM

Trains on tokenized and embeded SMILES to learn next-token prediction, generating valid molecular structures.

POLICY GRADIENT

Fine-tunes the LSTM model using rewards based on validity, novelty, QED score, and diversity to improve molecule generation.

Novelty :

"Generate completely novel molecules not found in the dataset." **next**
"Check the validity of these molecules to ensure chemical feasibility and practical use."

Results

Generated molecules are evaluated based on validity, novelty, uniqueness, QED score, and diversity. A grid of valid molecules is visualized and final performance metrics are calculated.

Figure 1: End-to-End Workflow for MORLD.

- **1. Raw Data Collection:** Downloaded the entire ZINC database containing 1 crore (10 million) SMILES strings.
- **2. Preprocessing & Canonicalization:** Canonicalized SMILES using RDKit and removed corrupted strings (`preprocess.py`).
- **3. Tokenization:** Tokenized SMILES strings to create a vocabulary (`tokenize_smiles.py`).
- **4. Dataset Creation:** Created 10 datasets of 10 lakh SMILES each.
- **5. Local Token Mapping:** Generated token-to-index mappings for each dataset.
- **6. Data Splitting:** Split datasets into 70% training and 30% testing (`split_data_tokens.py`).
- **7. Fragment Augmentation:** Replaced 30% of training molecules with molecules containing desired fragments during each epoch.
- **8. LSTM Model Training:** Trained LSTM on SMILES (`train_model.py`).
- **9. RL Fine-Tuning (REINFORCE):** Fine-tuned with a custom reward (`RLfinetune.py`).
- **10. Molecule Generation:** Generated SMILES using nucleus sampling (`testRLLSTM.py`).
- **11. Postprocessing & Evaluation:** Analyzed Validity, Novelty, Diversity, QED, and Uniqueness.
- **12. Visualization:** Created grid images of valid molecules.

Novelty Enforcement: During RL fine-tuning, a high reward was given to novel molecules, ensuring the generation of unique structures.

Reward Components:

- Novelty Bonus (+10)
- Validity (+5)
- Fragment Matching (+1)
- QED (continuous reward)

4 Datasets

Source: ZINC Database (https://zinc.docking.org/)

Original Data:

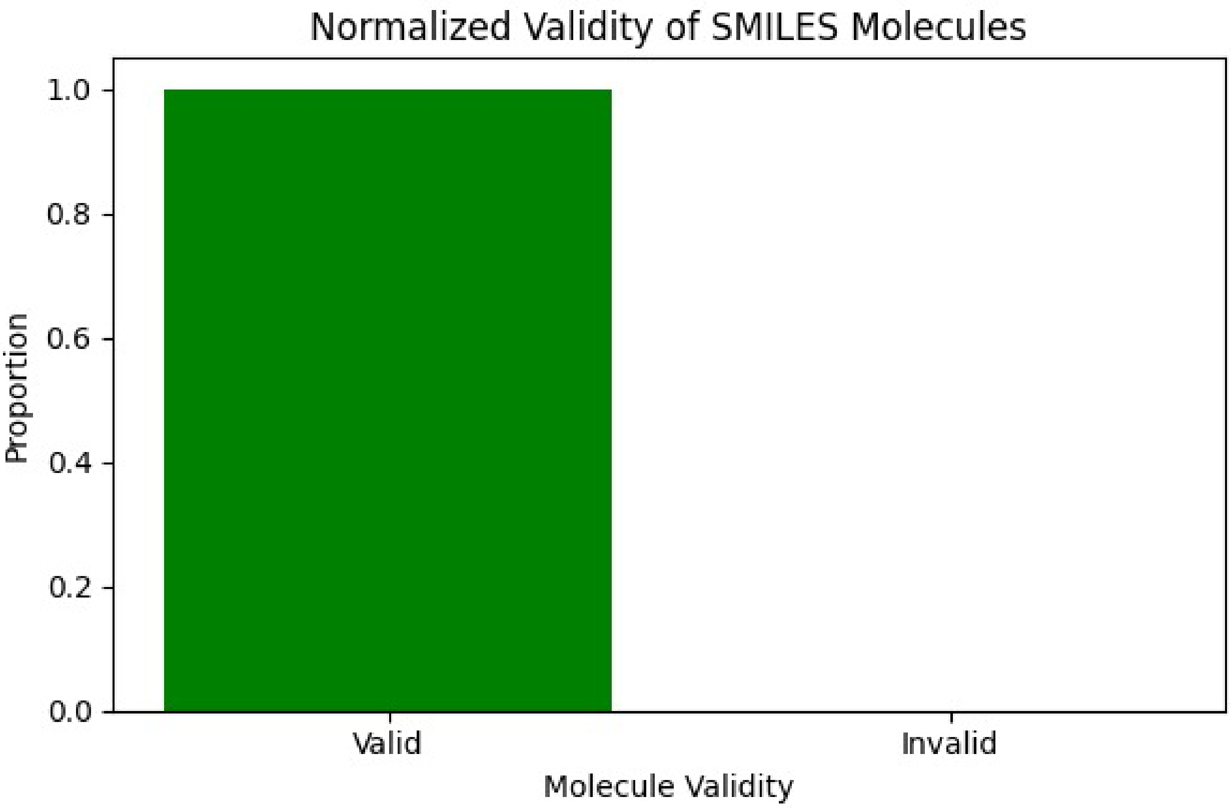
- 10 million SMILES strings.

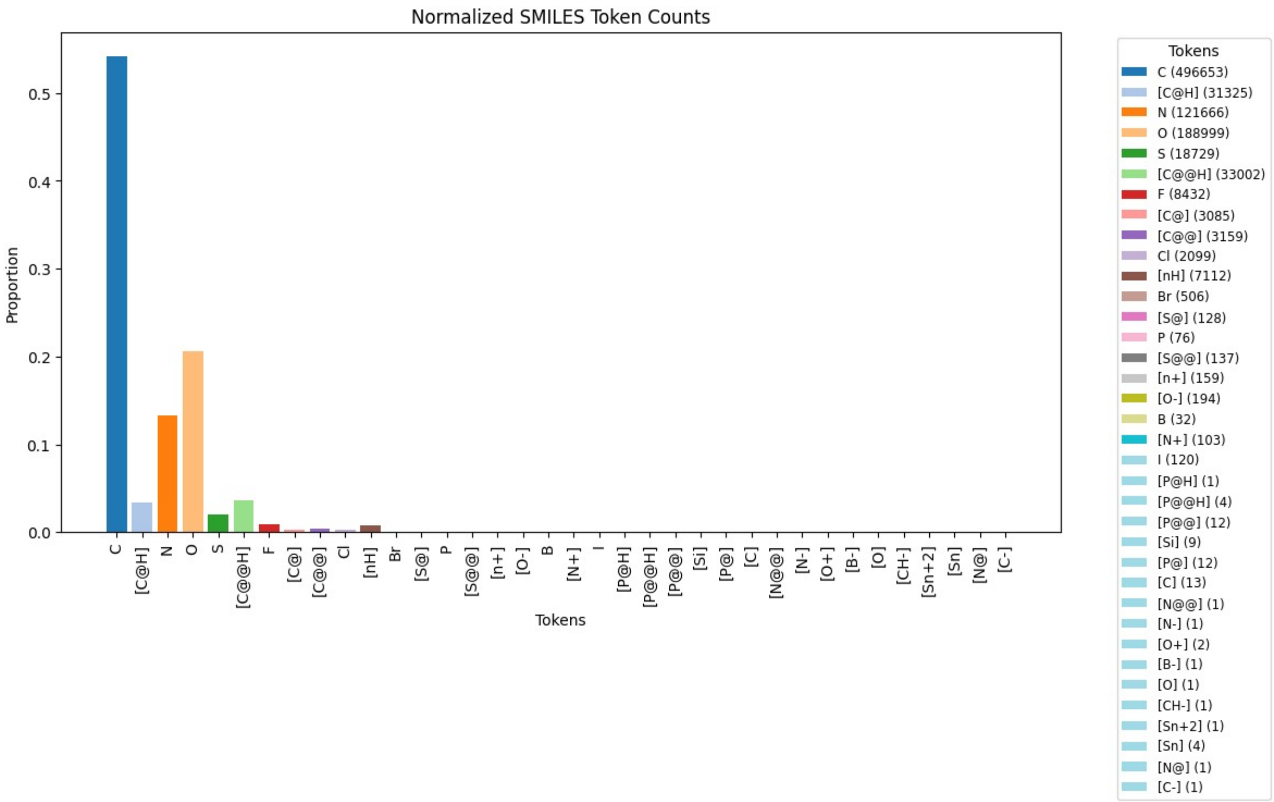
Preprocessed Data:

- Canonicalized SMILES strings.
- Global and local tokenization.
- 10 random datasets created.

5 Features Extracted from Dataset

- Character-level tokens.
- Functional group detection (C=O, N-H, O-H).
- QED score.
- Metrics: Validity, Uniqueness, Novelty.


Normalized Validity of SMILES Molecules
1.0
0.8
0.6
0.4
0.2
0.0
Proportion
Valid
Invalid
Molecule Validity


Normalized SMILES Token Counts
0.5
0.4
0.3
0.2
0.1
0.0
Proportion
C
[C@H]
N
O
S
[C@@H]
F
[C@]
[C@@]
Cl
[nH]
Br
[S@]
P
[S@@]
[n+]
[O-]
B
[N+]
I
[P@H]
[P@@H]
[P@@]
[Si]
[P@]
[C]
[N@@]
[N-]
[O+]
[B-]
[O]
[CH-]
[Sn+2]
[Sn]
[N@]
[C-]
Tokens
Tokens
C (496653)
[C@H] (31325)
N (121666)
O (188999)
S (18729)
[C@@H] (33002)
F (8432)
[C@] (3085)
[C@@] (3159)
Cl (2099)
[nH] (7112)
Br (506)
[S@] (128)
P (76)
[S@@] (137)
[n+] (159)
[O-] (194)
B (32)
[N+] (103)
I (120)
[P@H] (1)
[P@@H] (4)
[P@@] (12)
[Si] (9)
[P@] (12)
[C] (13)
[N@@] (1)
[N-] (1)
[O+] (2)
[B-] (1)
[O] (1)
[CH-] (1)
[Sn+2] (1)
[Sn] (4)
[N@] (1)
[C-] (1)

6 Model Architecture

LSTM Generator:

- Embedding Layer
- 2 LSTM Layers (hidden size 256)
- Dense Output Layer with Softmax

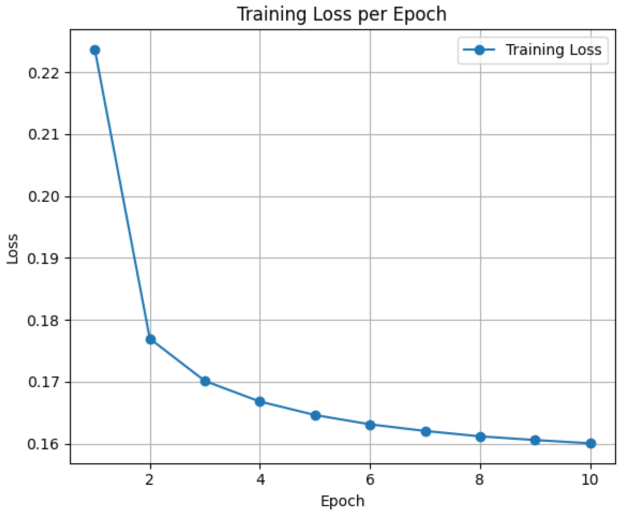
Reinforcement Learning Fine-Tuning:

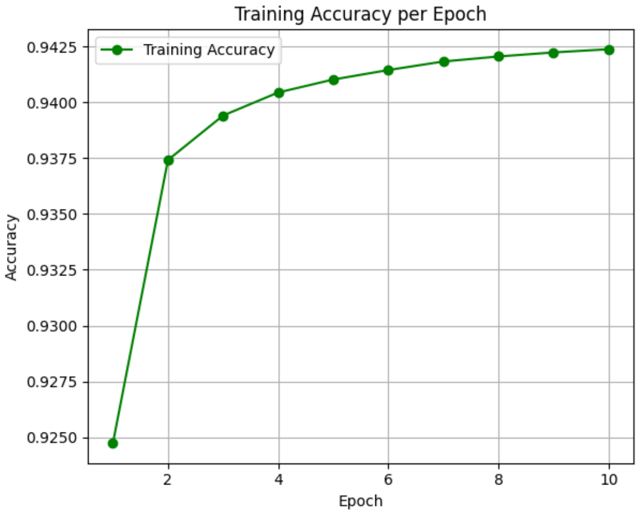
- Reward based on Validity, Novelty, Fragment Presence, and QED.
- Policy Gradient update (REINFORCE algorithm).

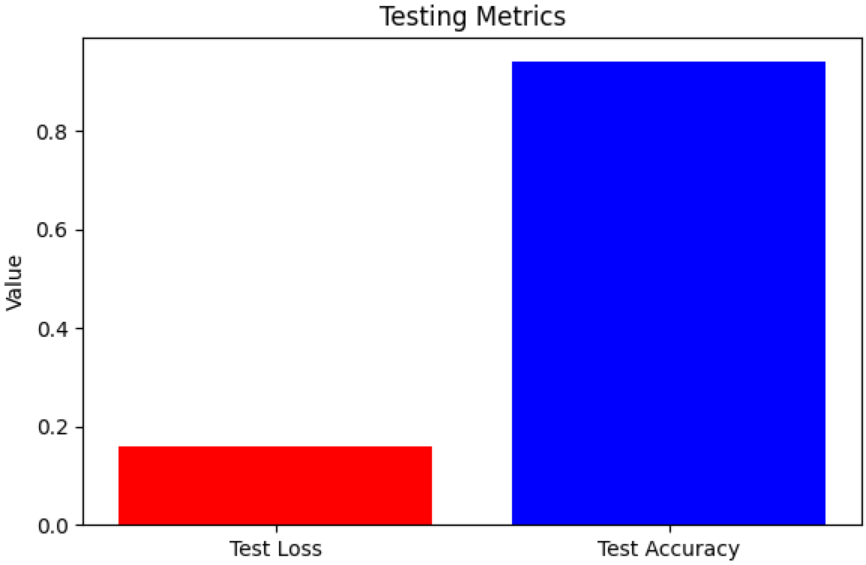
7 Results and Explanation

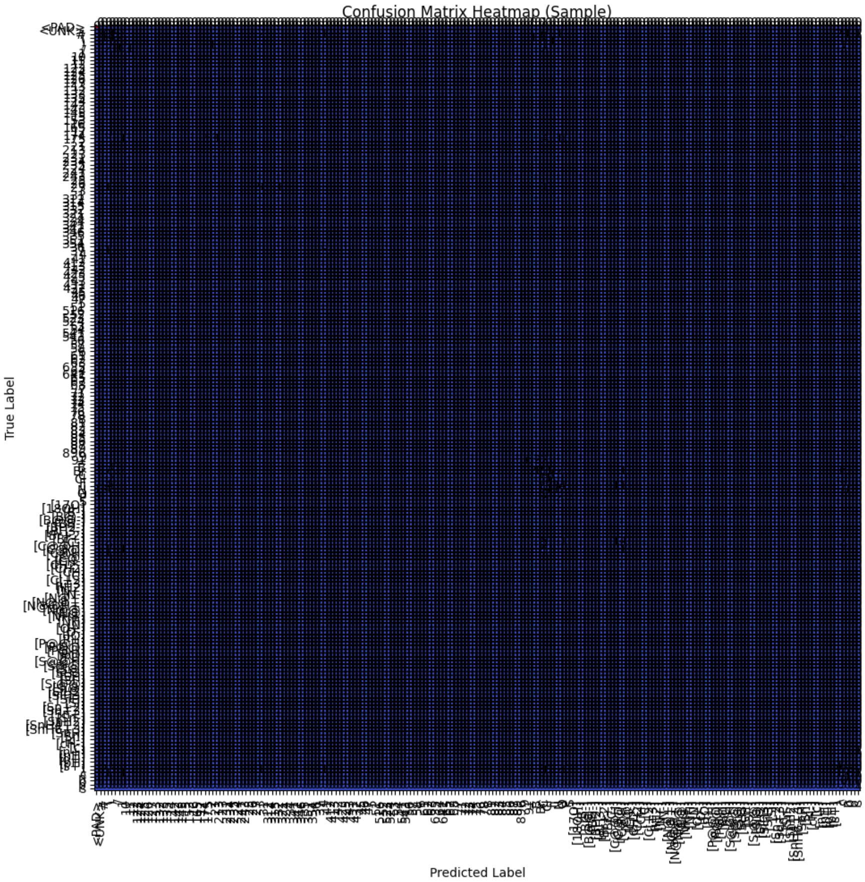
7.1 Post LSTM Results

- High syntactic validity in generated SMILES.



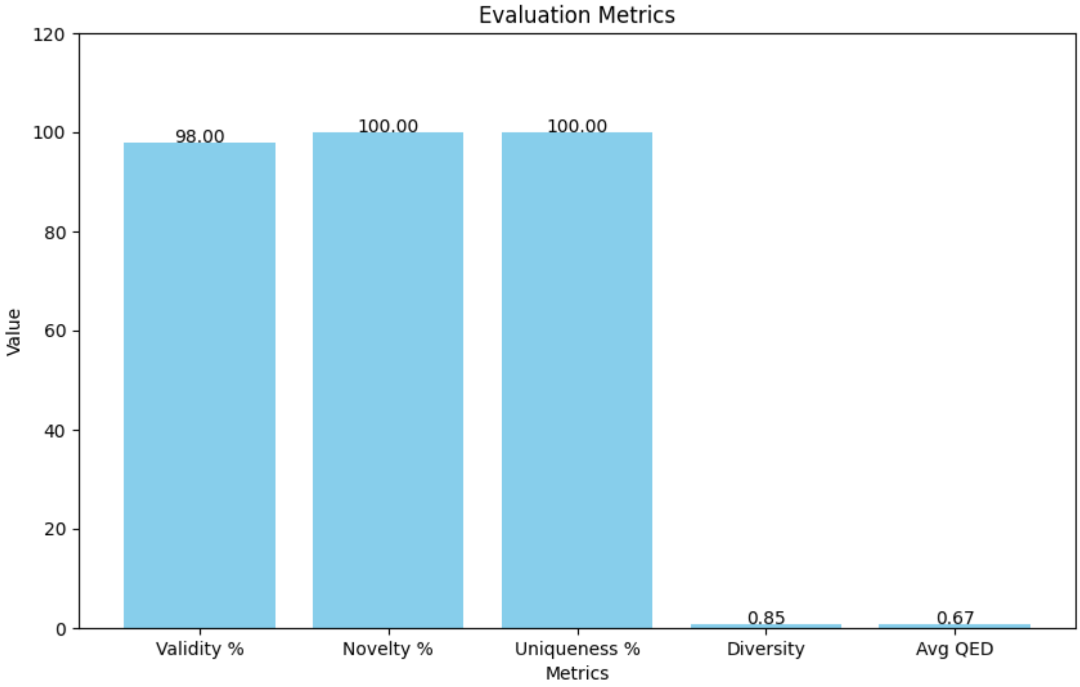


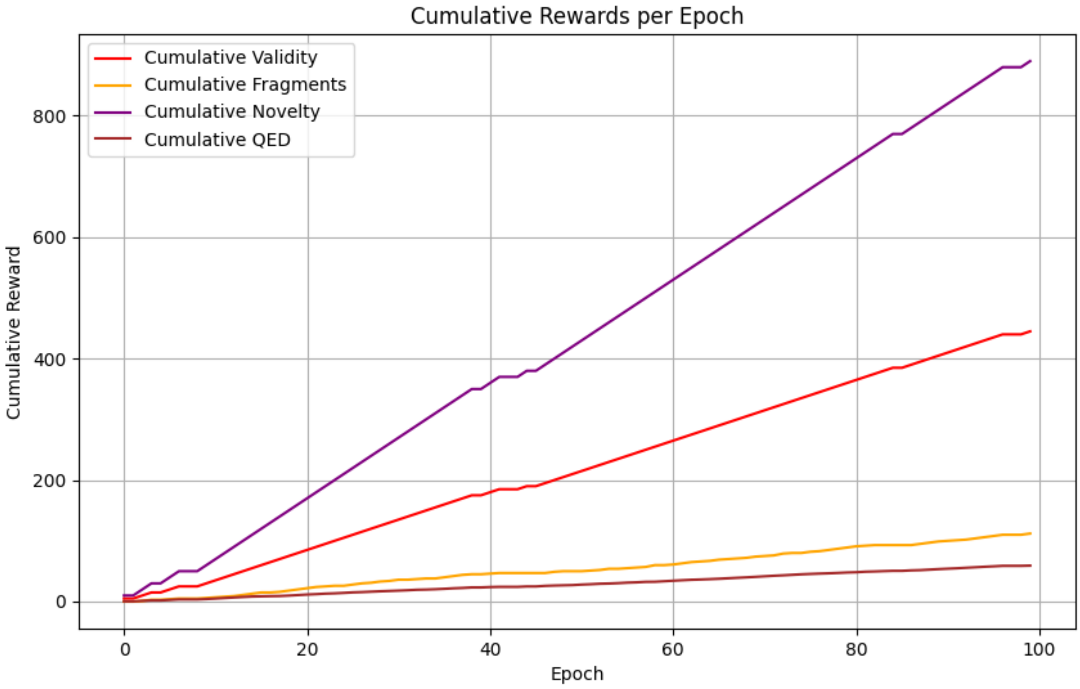


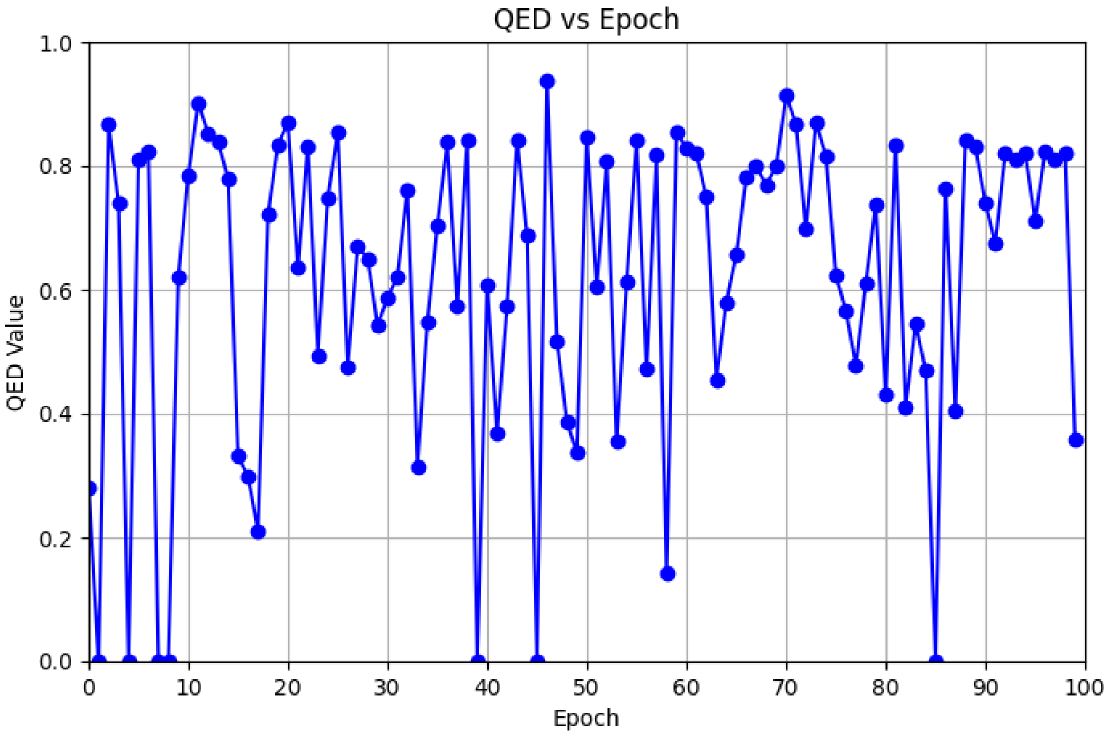


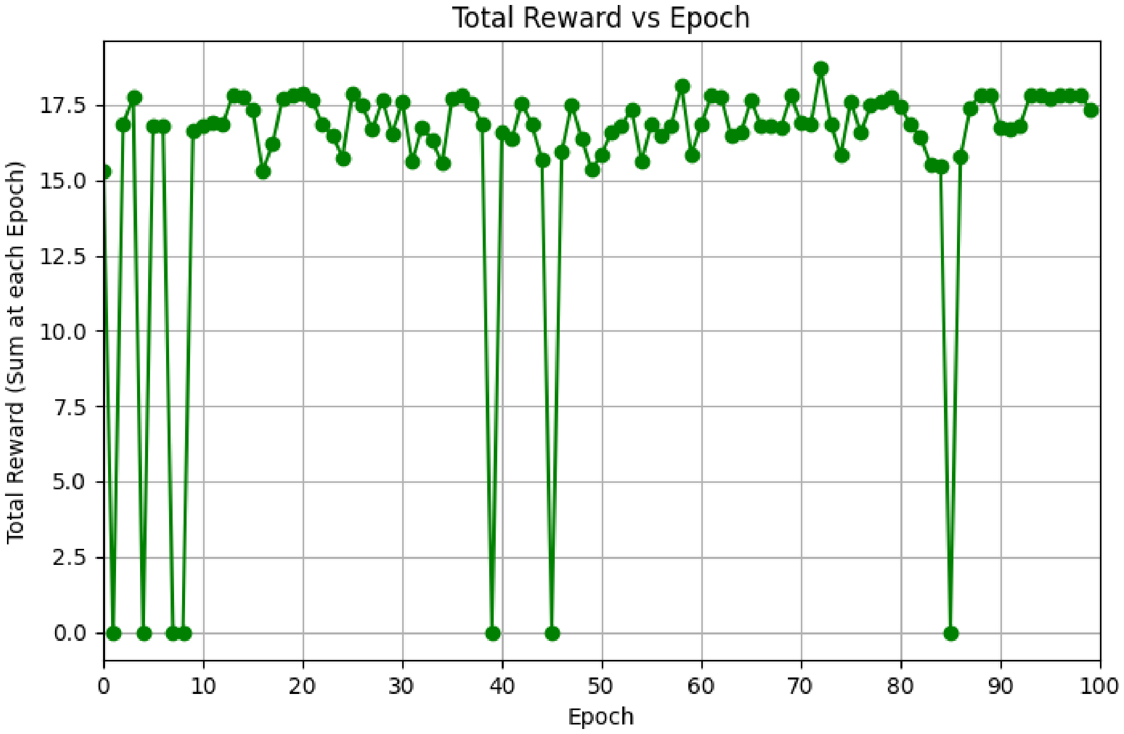
7.2 Results after RL Fine-Tuning

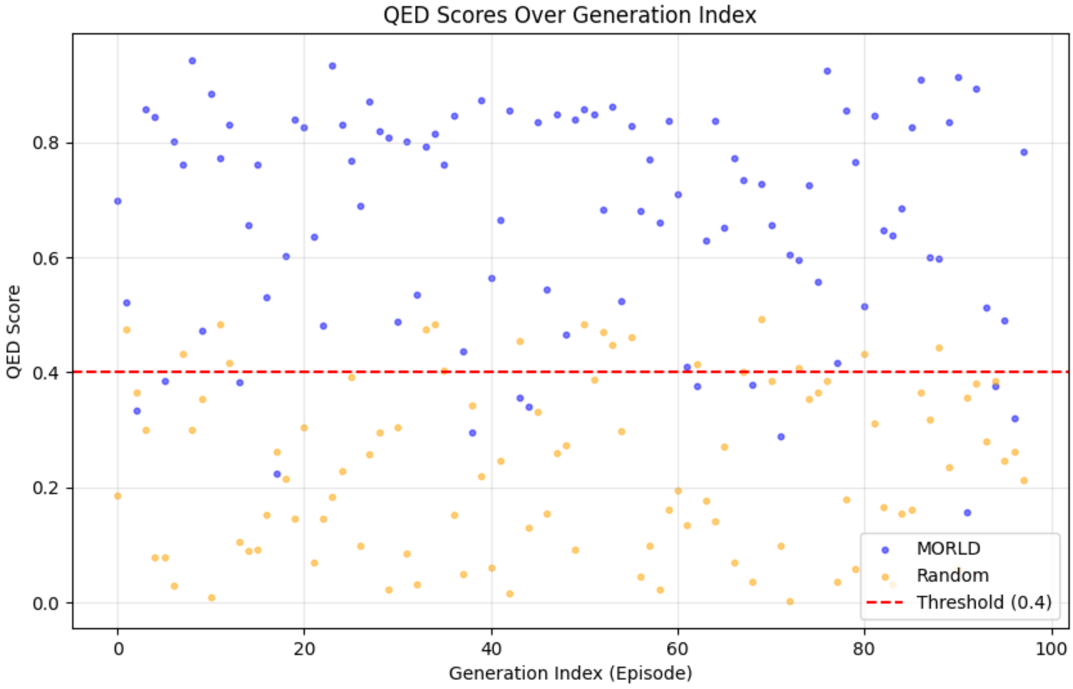
- Improved Novelty and QED scores.

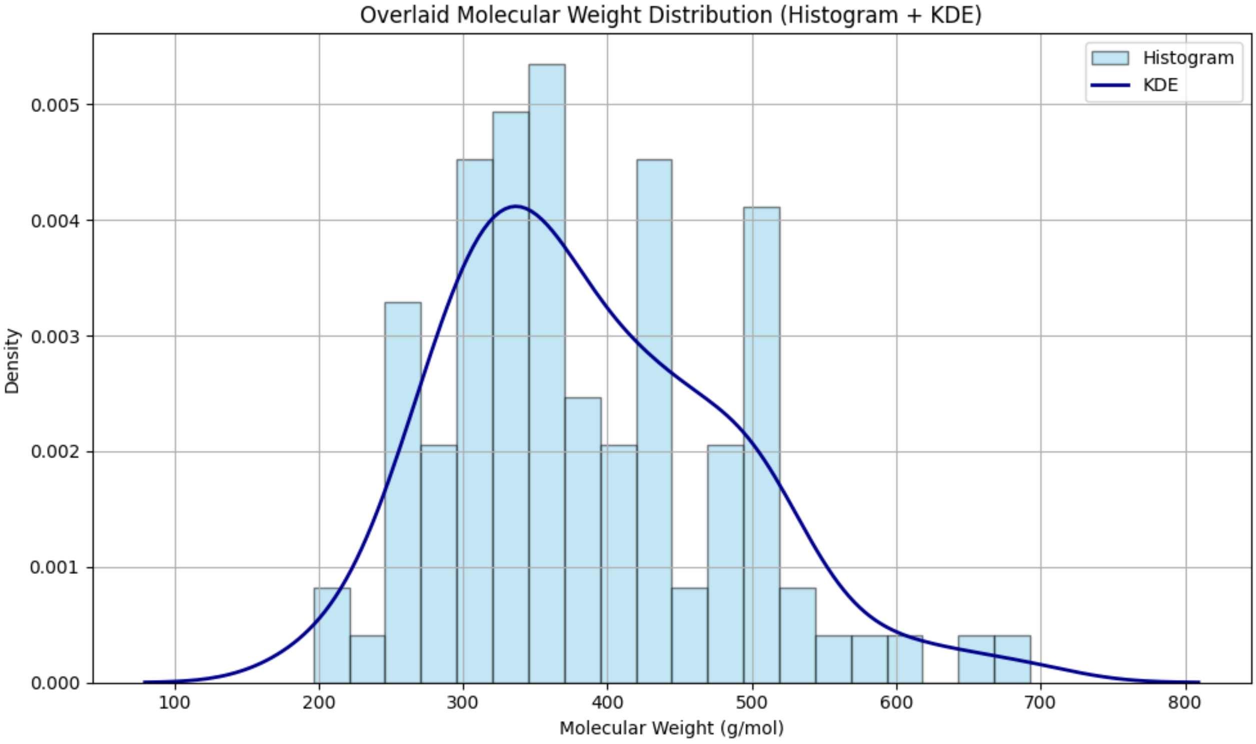

Evaluation Metrics
Value
120
100
80
60
40
20
0
98.00
100.00
100.00
0.85
0.67
Validity %
Novelty %
Uniqueness %
Diversity
Avg QED
Metrics

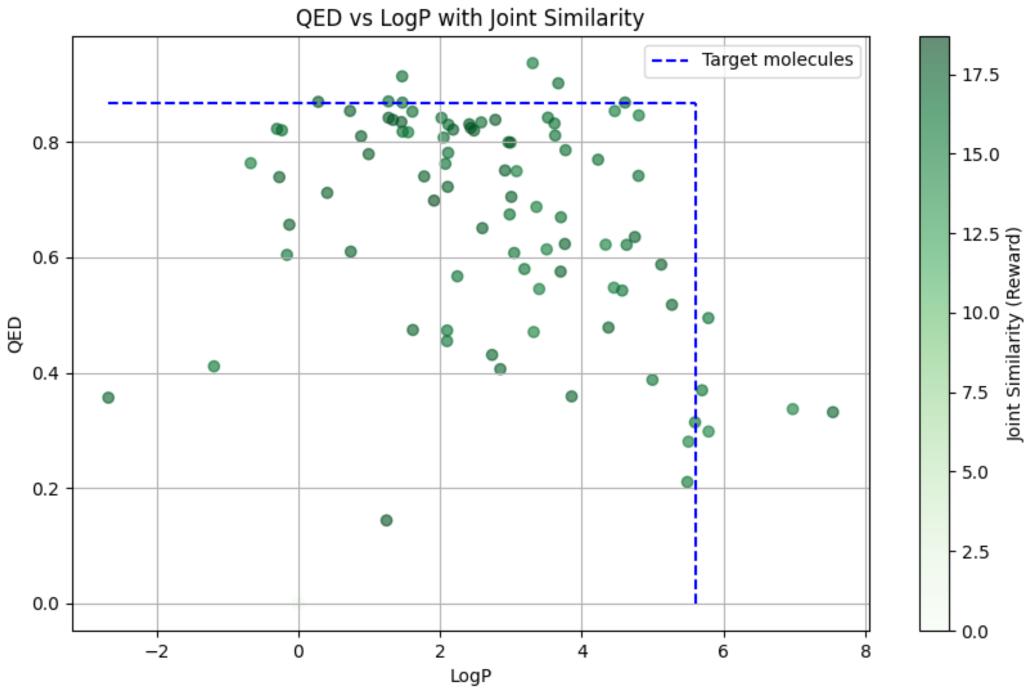

Cumulative Rewards per Epoch
Cumulative Validity
Cumulative Fragments
Cumulative Novelty
Cumulative QED
Cumulative Reward
800
600
400
200
0
0
20
40
60
80
100
Epoch


QED vs Epoch
QED Value
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
60
70
80
90
100
Epoch

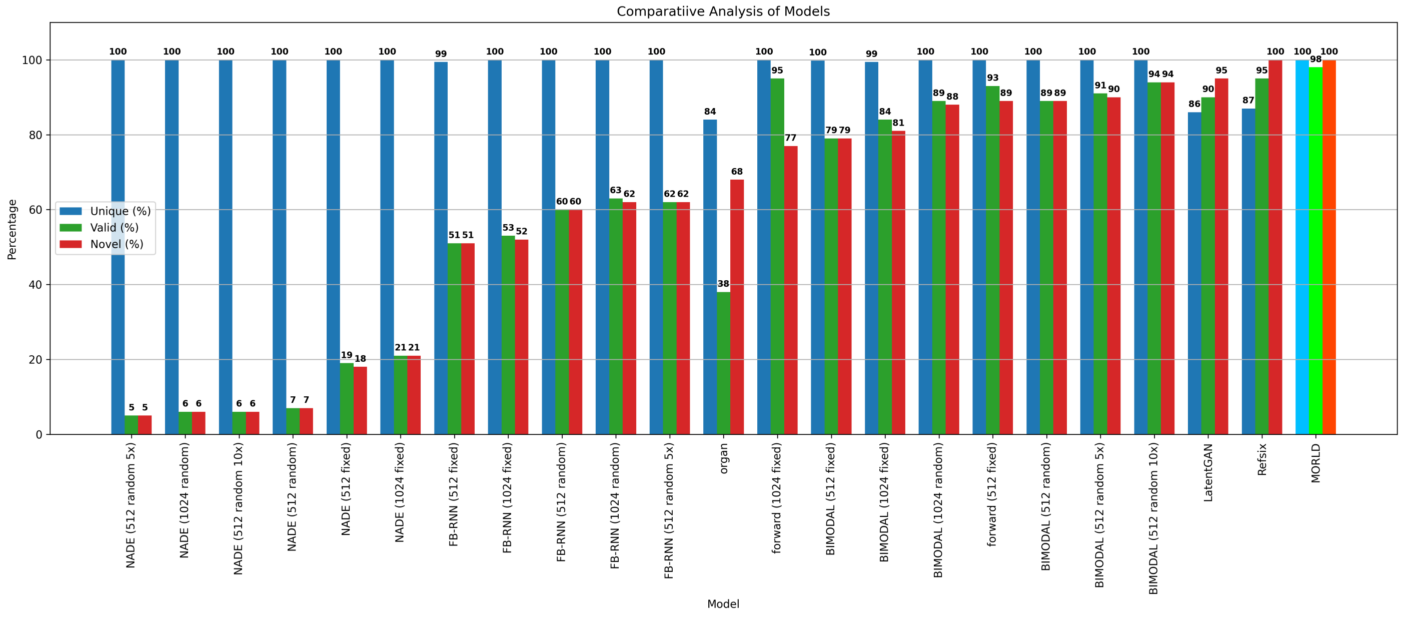

Total Reward vs Epoch
Total Reward (Sum at each Epoch)
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
10
20
30
40
50
60
70
80
90
100
Epoch


QED Scores Over Generation Index
QED Score
0.0
0.2
0.4
0.6
0.8
0
20
40
60
80
100
Generation Index (Episode)
MORLD
Random
Threshold (0.4)

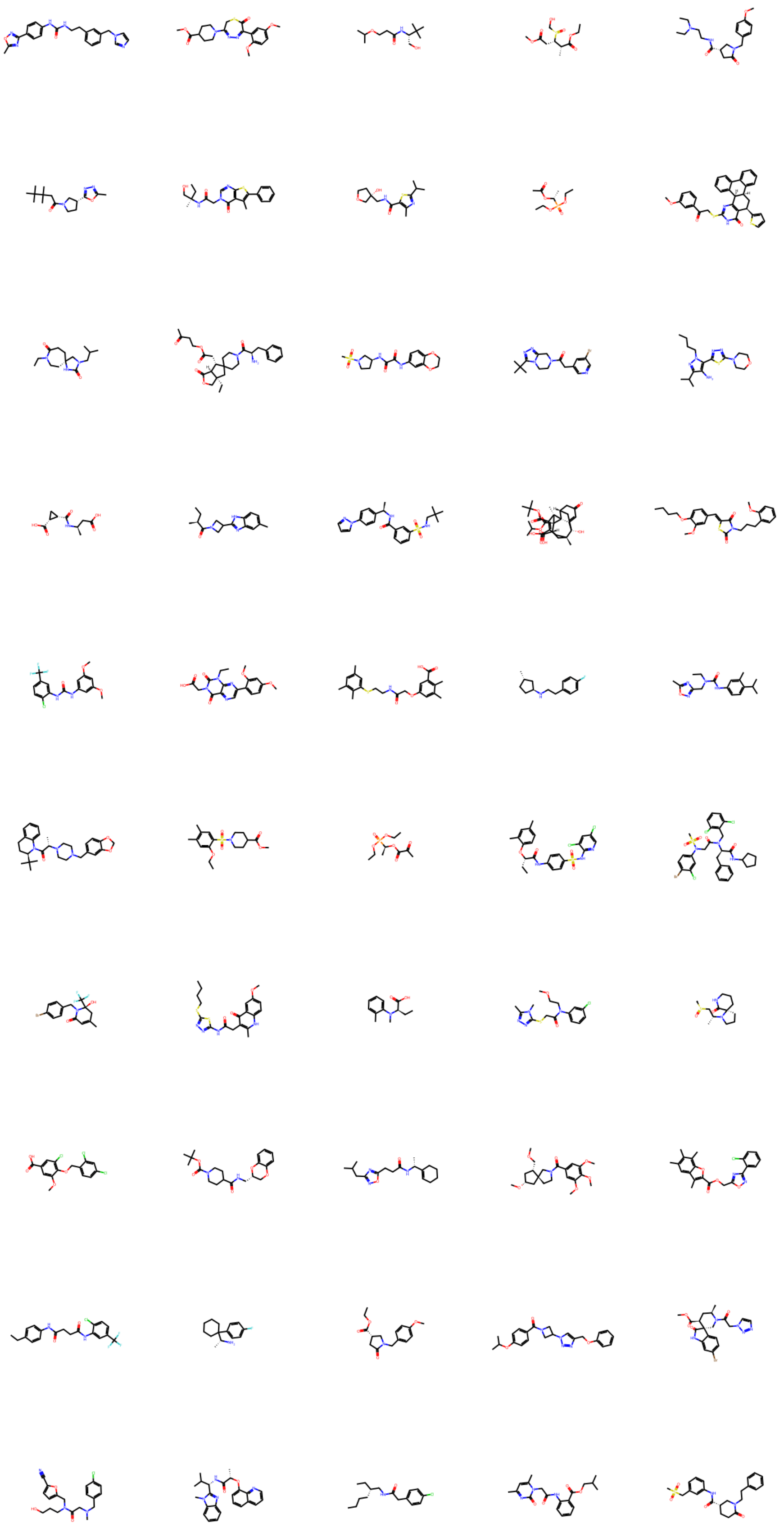

Overlaid Molecular Weight Distribution (Histogram + KDE)
Density
0.000
0.001
0.002
0.003
0.004
0.005
100
200
300
400
500
600
700
800
Molecular Weight (g/mol)
Histogram
KDE

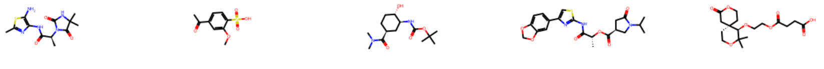


7.3 Comparative Analytics



8 Generated Molecules





9 Merits and De-merits

Merits:

- High validity rate.
- Reward customization possible.
- Accelerates drug discovery process.

De-merits:

- Computationally expensive.
- Requires domain-specific reward design.

10 Full Code and Execution Procedure

Code Structure:

- `codes/preprocess.py` - Preprocessing.
- `codes/tokenize_smiles.py` - Tokenization.
- `codes/split_data_tokens.py` - Splitting.
- `codes/train_model.py` - LSTM Training.
- `codes/RLfinetune.py` - Reinforcement Fine-Tuning.
- `codes/testRLLSTM.py` - Testing.

Execution Steps:

1. Install dependencies: `pip install rdkit-pypi torch numpy pandas`.
2. Preprocess: `python codes/preprocess.py`.
3. Tokenize: `python codes/tokenize_smiles.py`.
4. Split: `python codes/split_data_tokens.py`.
5. Train: `python codes/train_model.py`.
6. Fine-tune: `python codes/RLfinetune.py`.
7. Test: `python codes/testRLLSTM.py`.

11 Proper Documentation

All code is commented. Jupyter notebooks have markdowns explaining each stage. Outputs are saved.

12 Conclusion

Combining LSTM with reinforcement learning enables the generation of novel, valid molecules and accelerates drug discovery.

13 Future Scope

- Using Transformer architectures.
- Multi-objective optimization.
- Applying Mean Field Reinforcement Learning.

14 References

- Olivecrona, M., et al. (2017). Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*. https://jcheminf.biomedcentral.com/articles/10.1186/s13321-017-0235-x
- Arús-Pous, J., et al. (2020). SMILES-based deep generative scaffold decorator.
- ZINC Database (https://zinc.docking.org/)
- RDKit Documentation (https://www.rdkit.org)
- Blaschke, T., et al. (2020). Memory-assisted reinforcement learning for diverse molecular de novo design.
- Grisoni, F., et al. (2020). Bidirectional molecule generation.
- Wang, Q., et al. (2023). Molecular generation strategy using A2C RL.
- Jeon, W., & Kim, D. (2020). Autonomous molecule generation using RL and docking.