CSCI 447 MACHINE LEARNING

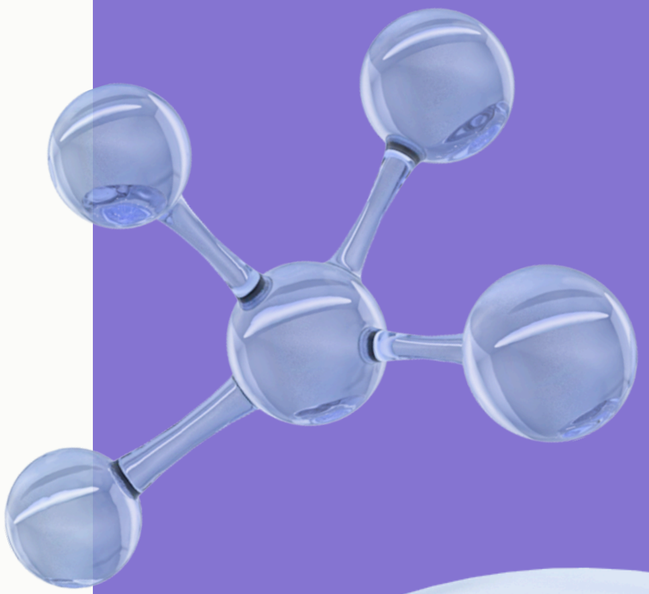
# GENERATIVE MODELS FOR LUMINESCENCE MOLECULES

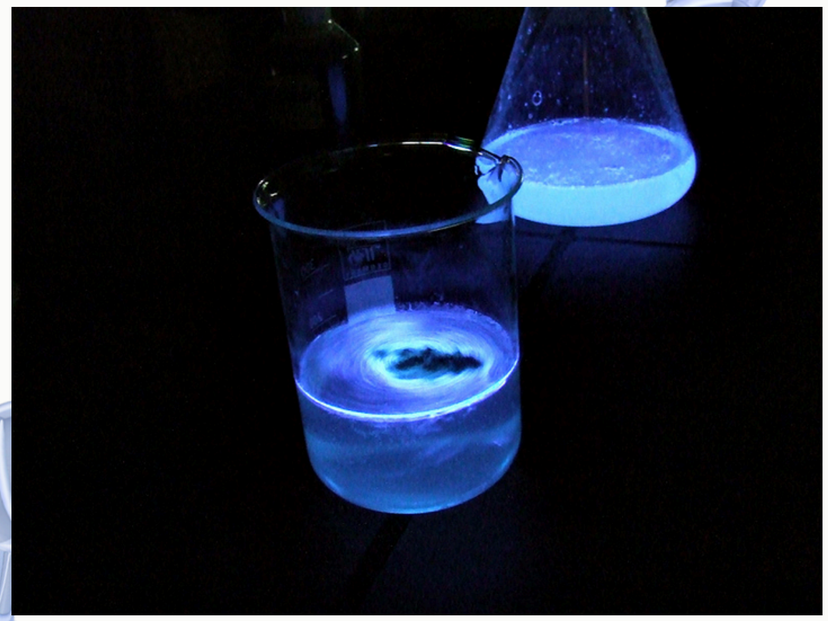
Dias Kuatbekov
Aruzhan Amangeldina
Alexandr Alpatov
Aiana Yergaliyeva
Nazerke Yeraliyeva

# Introduction

Luminescent molecules have applications ranging from organic light-emitting diodes (OLEDs) to bioimaging and fluorescent dyes.

- Organic Light Emitting Diodes are used in digital displays in TVs, smartphones and displays
- Bioimaging: non-invasive monitoring of biological processes





# What is luminescene?

- Luminescent materials emit photons upon absorption of energy in terms of electromagnetic waves.
- The extent to which it gives energy is described as quantum yield. Or, more technically, amount of photons absorbed/amount of photons emitted
- Such molecules come with their solvents
- A solvent can alter photophysical properties of a dissolved molecule

# Dataset

- "Experimental database of optical properties of organic compounds”
- First published in Nature Scientific Data in 2020.
- 20,236 chromophores under different solvents

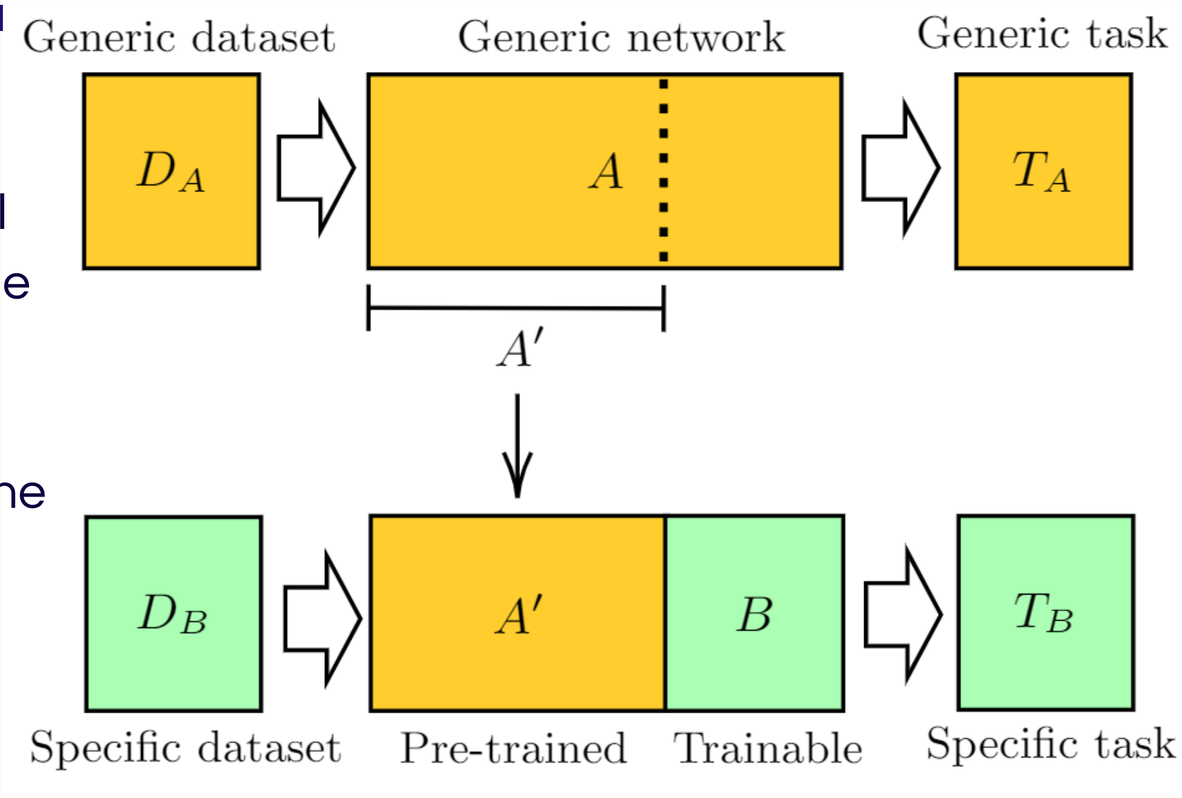
# Methodology

- **Model training**

Pre-train (if computing power allows it) a generative model on a big but general molecular dataset (Zinc, ChemBL). The goal is to let a model to learn a proper latent space where generic molecular structures live.
Use the pre-trained model to fine-tune it on our dataset. The goal is to reshape the latent space by letting the model obtain task-specific knowledge.

- **Model comparison**

Compare generative models based on established and custom metrics such as Uniqueness, Validity, Average Tanimoto Similarity to the training set, and Average Quantum Yield.

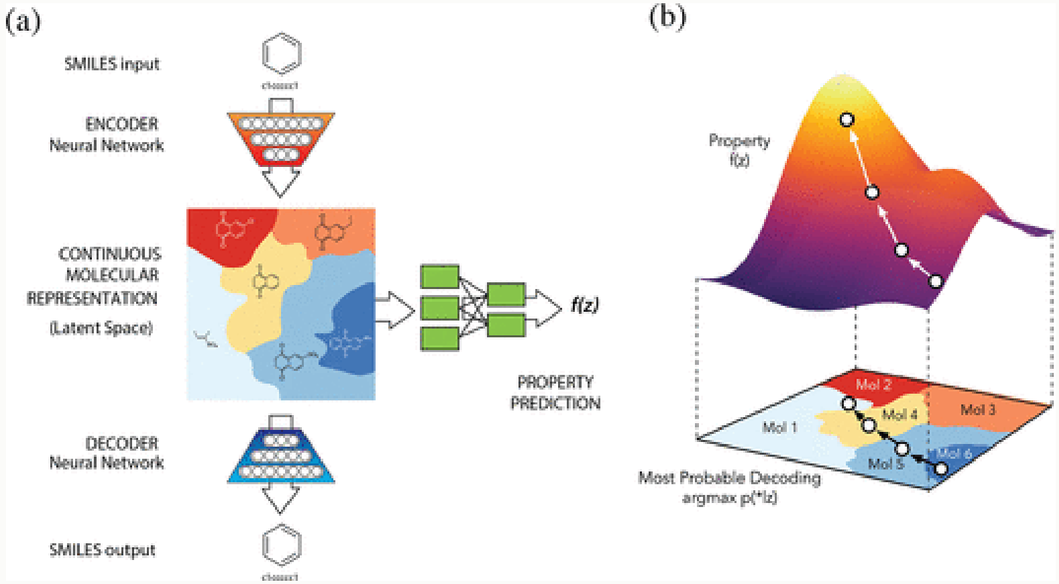


# Chemical VAE

chemVAE consistis of encoder and decoder.

- Encoder converts SMILES into a latent space vector
- Given a latent vector, decoder restores the SMILES string
- Encoding/Decoding done by GRU

Training Objective: minimization of Reconstruction Loss + KL divergence




https://pubs.acs.org/doi/full/10.1021/acscentsci.7b00572

# Leveraging VAE to molecule generation

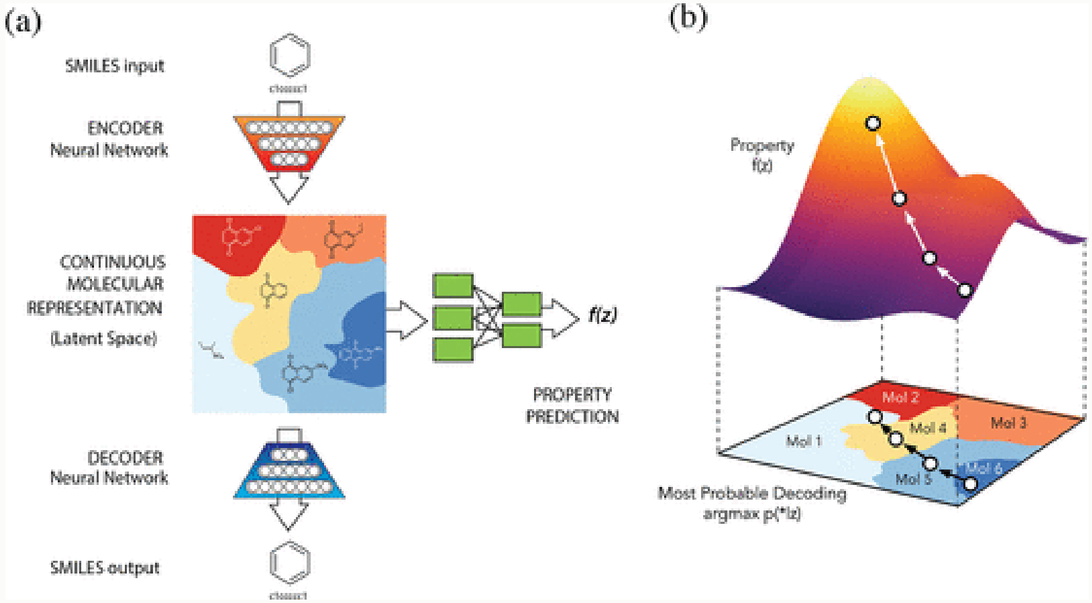
- encoder: CNN
- decoder: GRU (not traditional RNN)

initially trained on: ZINC dataset (250k)

data divided into tokens

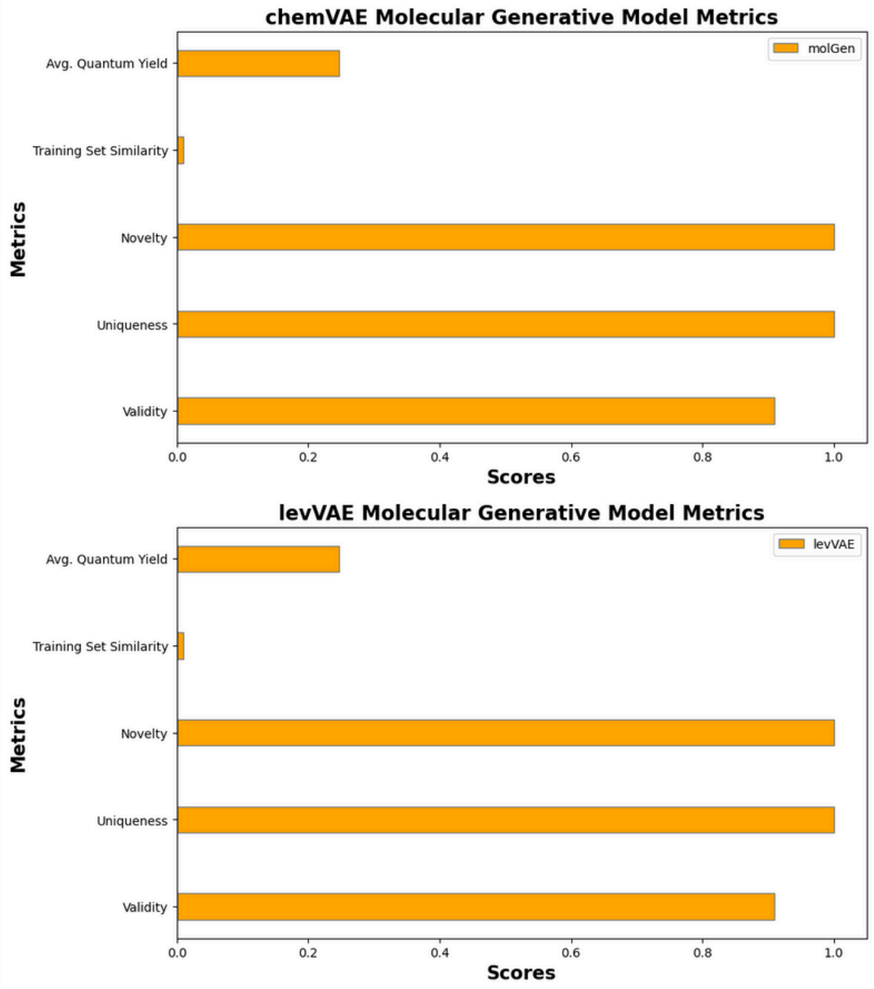
- GRU: remember things within each training iteration and make decision

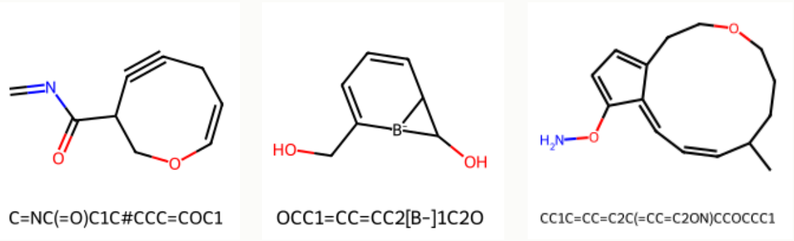
- Leverage - utilization of learned latent space representation to generate new data



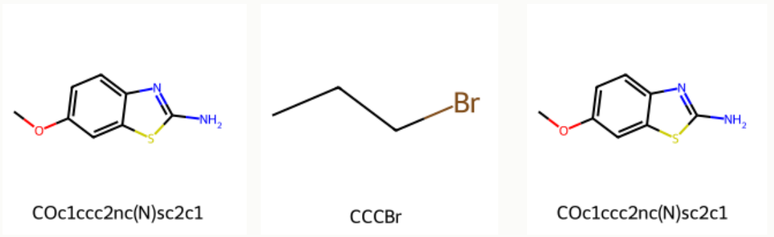

https://pubs.acs.org/doi/full/10.1021/acscentsci.7b00572

# VAE results




chemVAE molecules




levVAE molecules

# MolGen

Generator - generates SMILES strings:

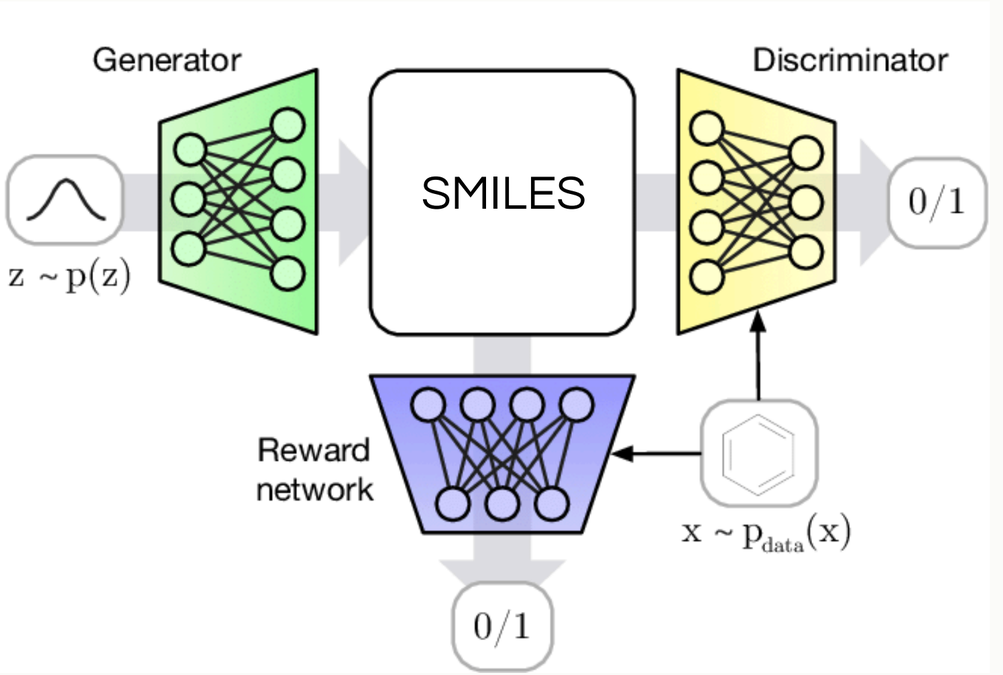
- takes  a  latent space vector z as input
- projects it to higher-dimensional space
- feeds it to LSTM cell to generate sequential output

Recurrent Discriminator - discriminates real and fake SMILES:

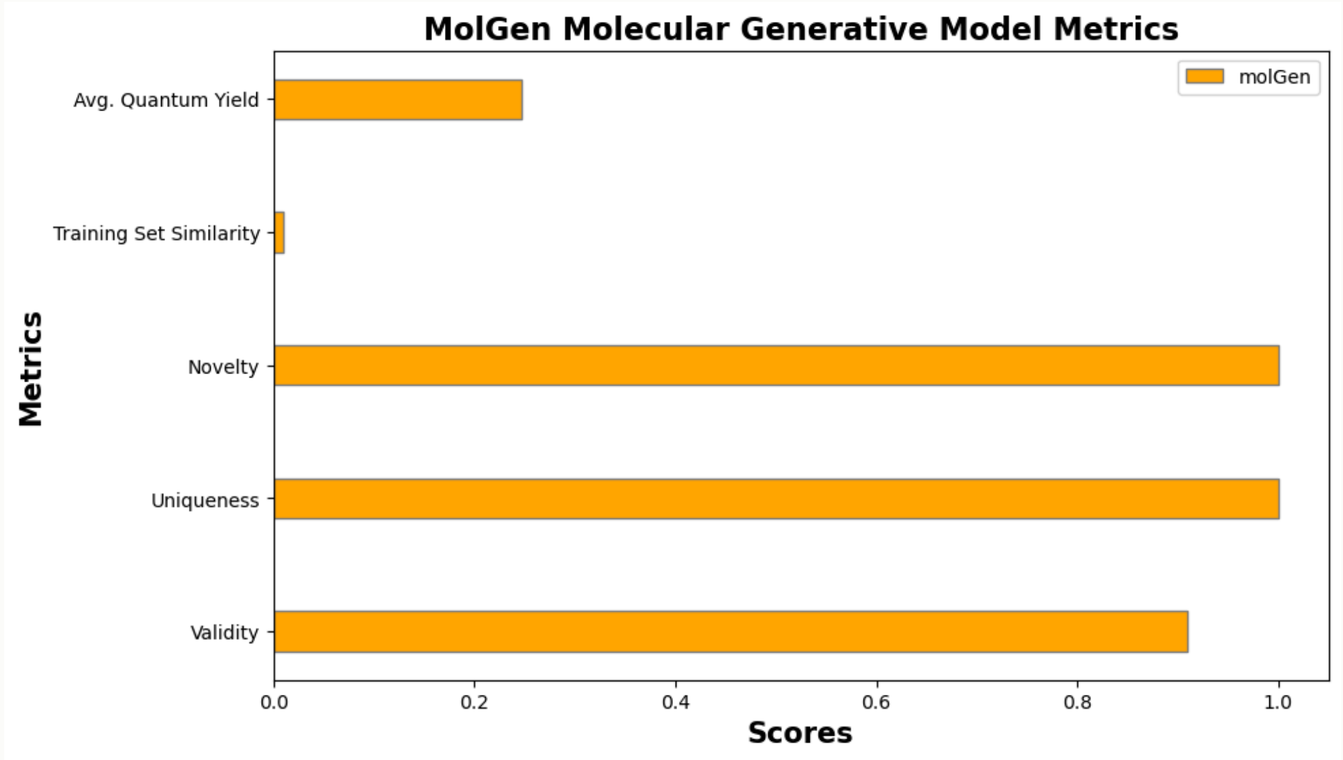
- calculates the probability that the sequence is real or not
- the probabilistic values are rewards for generator

Training:

- trained for 70000 steps
- evaluated every 2000 steps by checking the validity of generated SMILES using RDKit

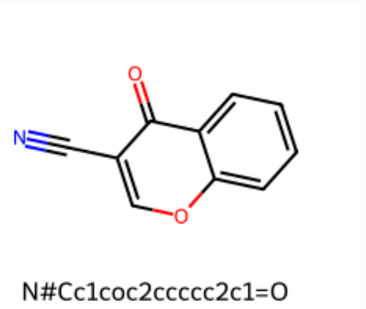


# MolGen

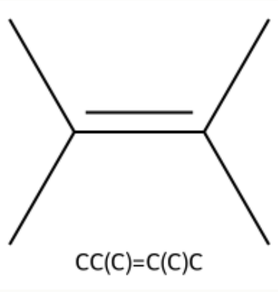




N#Cc1ccc(C#N)c(N)c1

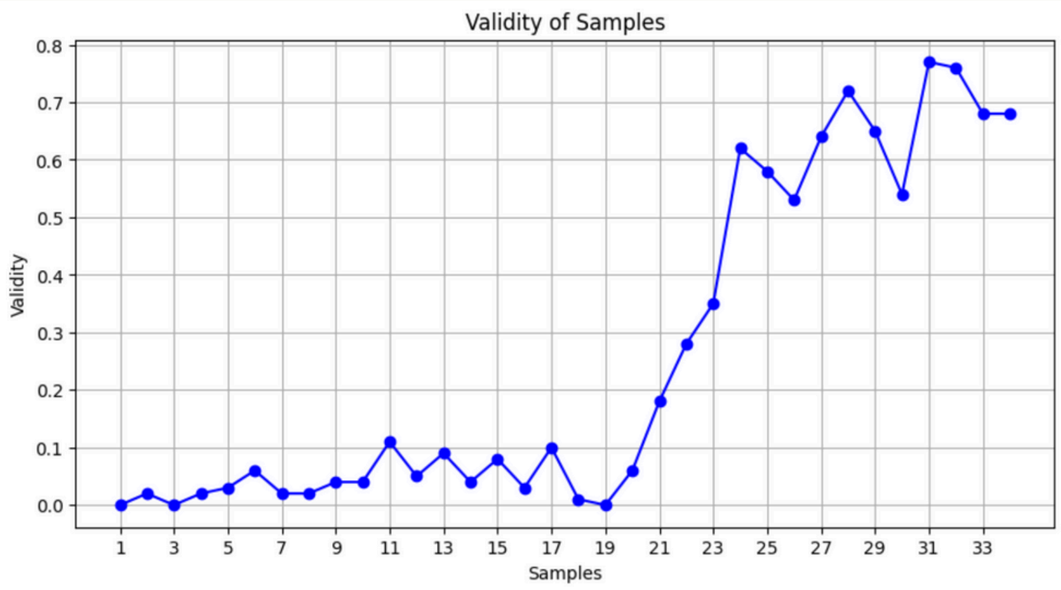



N#Cc1coc2ccccc2c1=O



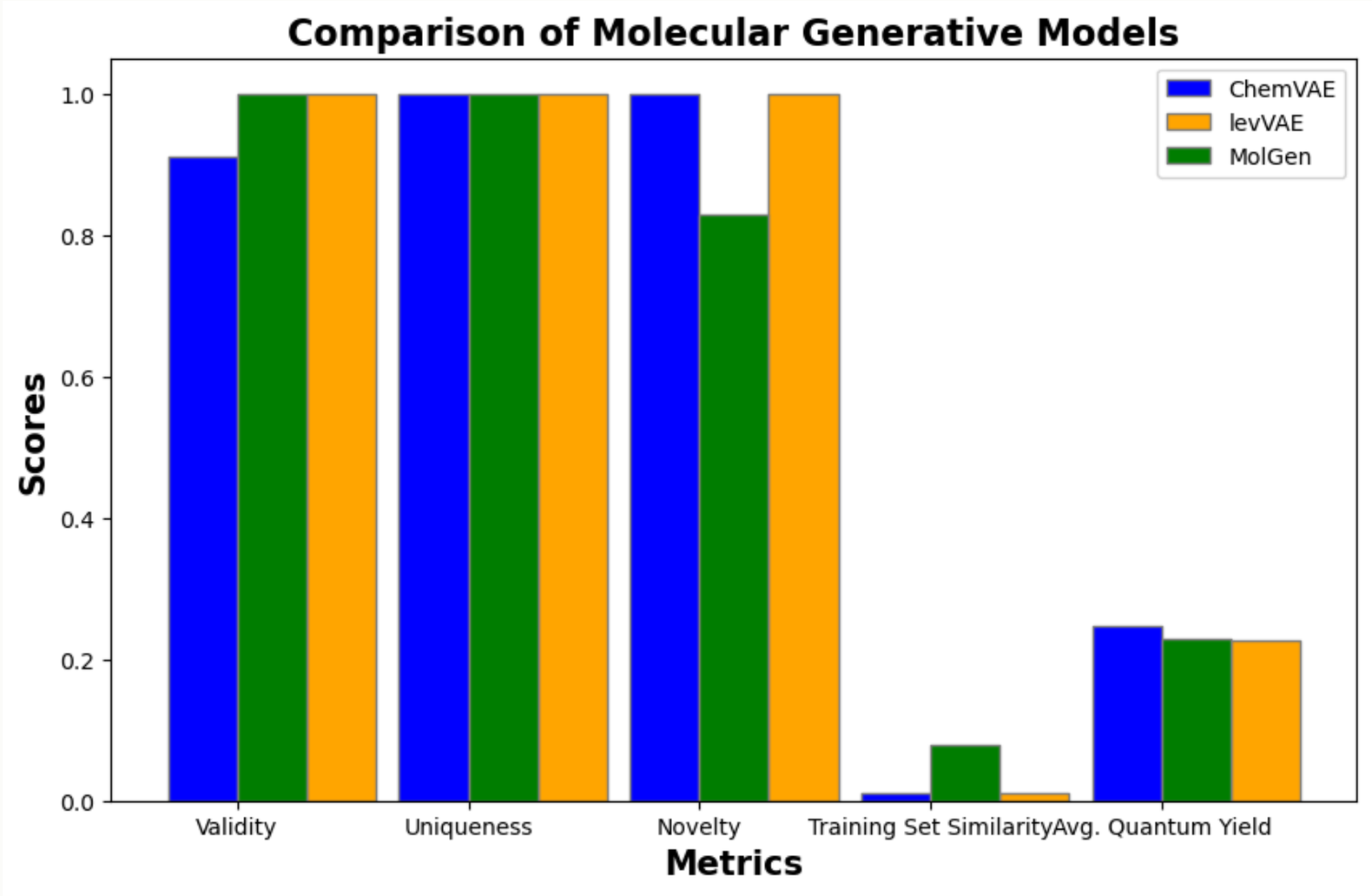
CC(C)=C(C)C

Examples of generated molecules



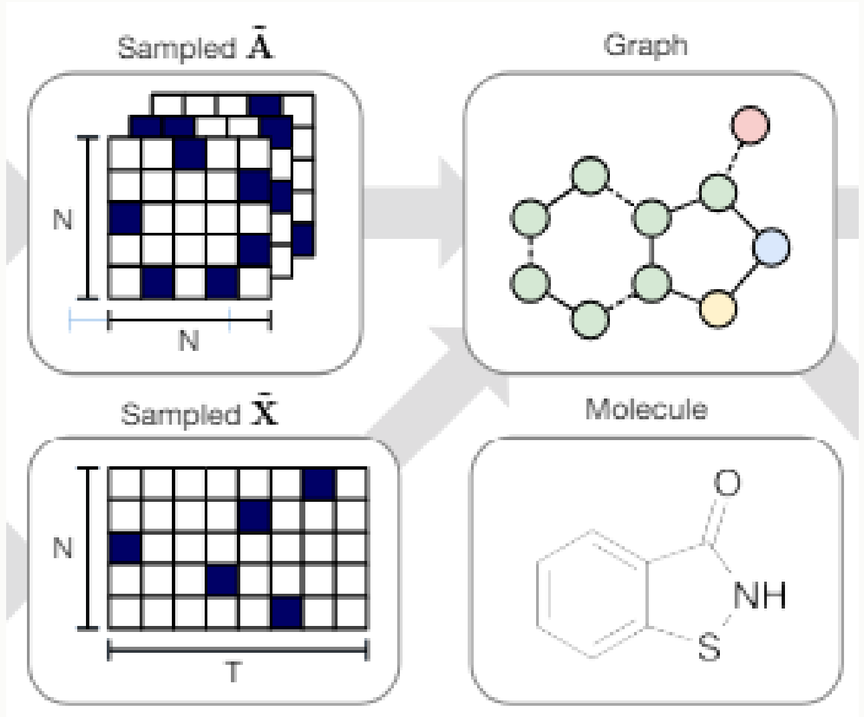

Training process

# Results


Comparison of Molecular Generative Models
ChemVAE
levVAE
MolGen
Scores
1.0
0.8
0.6
0.4
0.2
0.0
Validity
Uniqueness
Novelty
Training Set Similarity
Avg. Quantum Yield
Metrics

# Graph Representation

- Undirected graph with V nodes and E edges
- Each vertex in V - one-hot encoded vector Xi
- Each edge in E - one-hot encoded vector V in Eij
- All generated outputs are valid graphs
- No need for an expensive graph-matching procedure
- No order ambiguity

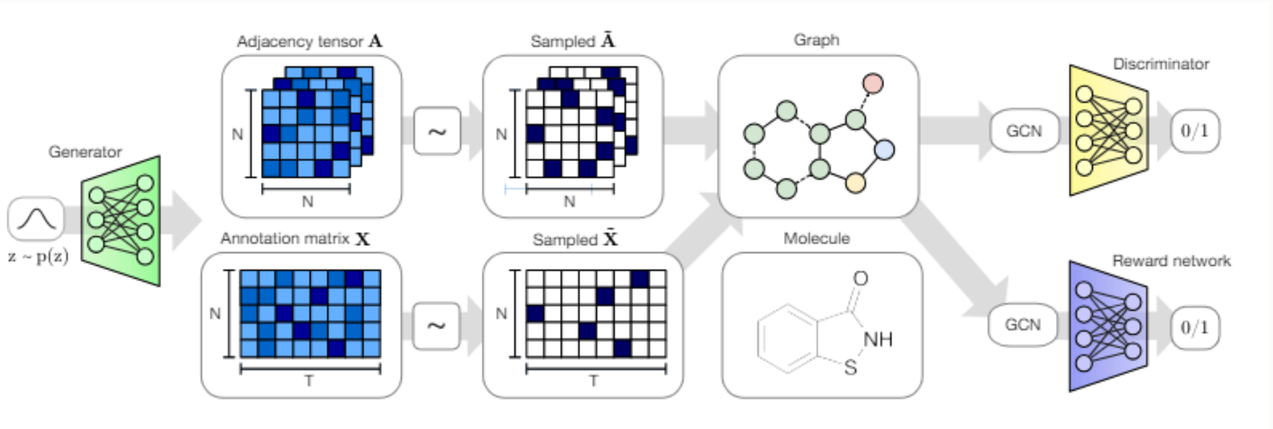


# MolGAN

An implicit, likelihoodfree generative model for small molecular graphs that circumvents the need for expensive graph matching procedures or node ordering heuristics of previous likelihood-based methods.

Advantages:

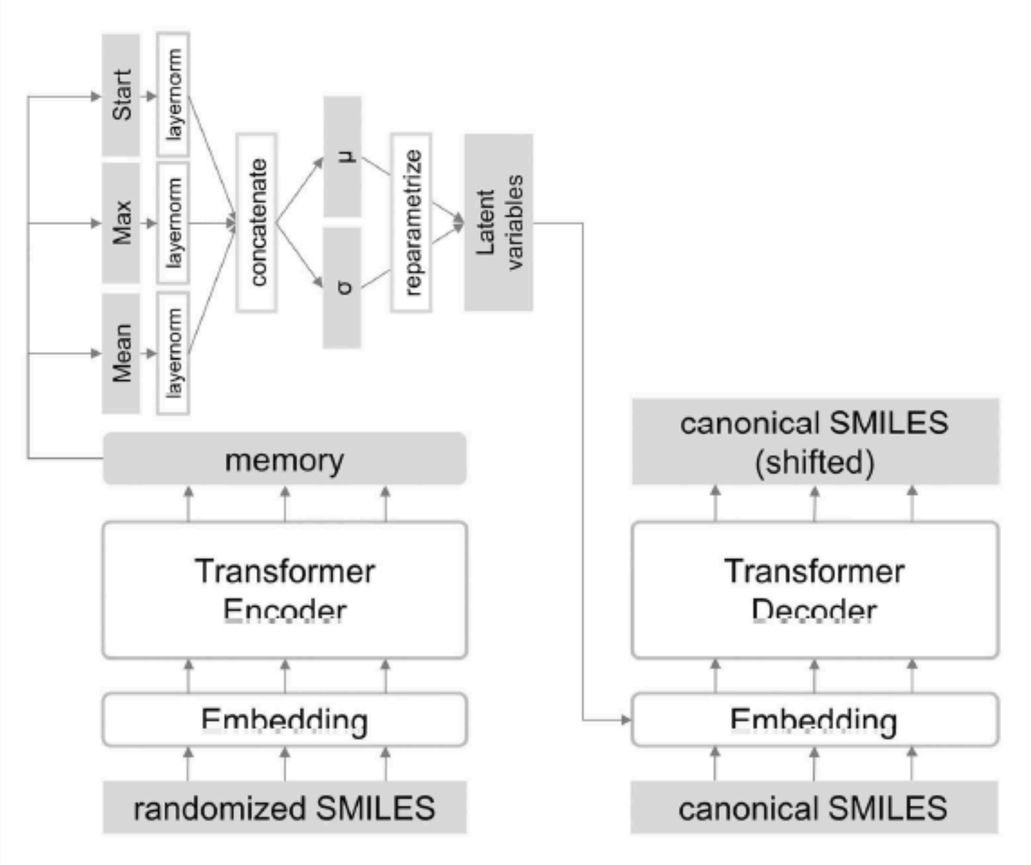
- Likelihood-free estimation
- Graph-based representation of the molecules
- Improved Loss function
- Deterministic Policy Gradient




| Algorithm | Valid | Unique | Novel |
|---|---|---|---|
| CharacterVAE | 10.3 | 67.5 | 90.0 |
| GrammarVAE | 60.2 | 9.3 | 80.9 |
| GraphVAE | 55.7 | **76.0** | 61.6 |
| GraphVAE/imp | 56.2 | 42.0 | 75.8 |
| GraphVAE NoGM | 81.0 | 24.1 | 61.0 |
| MolGAN | **98.1** | 10.4 | **94.2** |

https://arxiv.org/pdf/1805.11973

# TransformerVAE




- Randomized SMILES was inputted into the encoder, and distribution of latent variables were estimated from the pooled memory. Latent variables were added to the embedding of canonical SMILES to be decoded. Teacher forcing was used in the decoder during training, while beam search was used to generate new molecules.

  - trained on MOSES and ZINC-15
  - shows high validity, uniqueness, novelty

# TransformerVAE

- Valid - ratio of valid SMILES in all generated SMILES
- unique@1000 - ratio of SMILES which appear only once in 1000 valid generated SMILES
- FCD (Fréchet ChemNet Distance) - difference of generated molecules to test set in terms of distribution of molecules in the last layer activations of ChemNet
- SNN (nearest neighbor similarity) - average similarity of each generated molecule to its nearest molecule in test set
- Frag (fragment similarity) - cosine similarity of frequency of fragments between generated molecules and test set
- Scaf (scaffold similarity) - cosine similarity of frequency of scaffolds between generated molecules and test set
- Novelty - ratio of valid unique molecules which do not appear in training set

# TransformerVAE

| | | Our model | HMM | NGram | Combinatorial | CharRNN | AAE | VAE | JTN-VAE | LatentGAN |
|---|---|---|---|---|---|---|---|---|---|---|
| Valid(↑) | | 0.8761±0.0035 | 0.0760±0.0322 | 0.2376±0.0025 | **1.0000±0.0000** | 0.9748±0.0264 | 0.9368±0.0341 | 0.9767±0.0012 | **1.0000±0.0000** | **0.8966±0.0029** |
| unique@1000(↑) | | **1.0000±0.0000** | 0.6230±0.1224 | 0.9740±0.0108 | 0.9983±0.0015 | **1.0000±0.0000** | **1.0000±0.0000** | **1.0000±0.0000** | **1.0000±0.0000** | **1.0000±0.0000** |
| unique@10000(↑) | | **1.0000±0.0001** | 0.5671±0.1424 | 0.9217±0.0019 | 0.9909±0.0009 | 0.9994±0.0003 | 0.9973±0.0020 | 0.9984±0.0005 | 0.9996±0.0003 | |
| FCD(↓) | Test | 1.3380±0.0735 | 24.4661±2.5251 | 5.5069±0.1027 | 4.2375±0.0370 | **0.0732±0.0247** | 0.5555±0.2033 | 0.0990±0.0125 | | 0.2968±0.0087 |
| | TestSF | 1.7098±0.0740 | 25.4312±2.5599 | 6.2306±0.0966 | 4.5113±0.0274 | **0.5204±0.0379** | 1.0572±0.2375 | 0.5670±0.0338 | | 0.8281±0.0117 |
| SNN(↑) | Test | 0.4803±0.0025 | 0.3876±0.0107 | 0.5209±0.0010 | 0.4514±0.0003 | 0.6015±0.0206 | 0.6081±0.0043 | **0.6257±0.0005** | 0.5477±0.0076 | 0.5371±0.0004 |
| | TestSF | 0.4658±0.0026 | 0.3795±0.0107 | 0.4997±0.0005 | 0.4388±0.0002 | 0.5649±0.0142 | 0.5677±0.0045 | **0.5783±0.0008** | 0.5194±0.0070 | 0.5132±0.0002 |
| Frag(↑) | Test | 0.9927±0.0016 | 0.5754±0.1224 | 0.9846±0.0012 | 0.9912±0.0004 | **0.9998±0.0002** | 0.9910±0.0051 | 0.9994±0.0001 | | 0.9986±0.0004 |
| | TestSF | 0.9898±0.0018 | 0.5681±0.1218 | 0.9815±0.0012 | 0.9904±0.0003 | 0.9983±0.0003 | | **0.9985±0.0003** | 0.9947±0.0002 | 0.9972±0.0007 |
| Scaf(↑) | Test | 0.7159±0.0192 | 0.2065±0.0481 | 0.5302±0.0163 | 0.4445±0.0056 | 0.9242±0.0058 | | **0.9386±0.0021** | 0.8964±0.0039 | 0.8867±0.0009 |
| | TestSF | **0.1893±0.0081** | 0.0490±0.0180 | 0.0977±0.0142 | 0.0865±0.0027 | 0.1101±0.0081 | | 0.0588±0.0095 | 0.1009±0.0105 | 0.1072±0.0098 |
| IntDiv(↑) | | 0.8531±0.0013 | 0.8466±0.0403 | **0.8738±0.0002** | 0.8732±0.0002 | | 0.8557±0.0031 | 0.8558±0.0004 | 0.8551±0.0034 | 0.8565±0.0007 |
| Filters(↑) | | 0.9706±0.0005 | 0.9024±0.0489 | 0.9582±0.0010 | 0.9557±0.0018 | 0.9943±0.0034 | 0.9960±0.0006 | **0.9970±0.0002** | 0.9760±0.0016 | 0.9735±0.0006 |
| Novelty(↑) | | 0.9911±0.0005 | **0.9994±0.0010** | 0.9694±0.0010 | 0.9878±0.0008 | 0.8419±0.0509 | 0.7931±0.0285 | 0.6949±0.0069 | 0.9143±0.0058 | 0.9498±0.0006 |

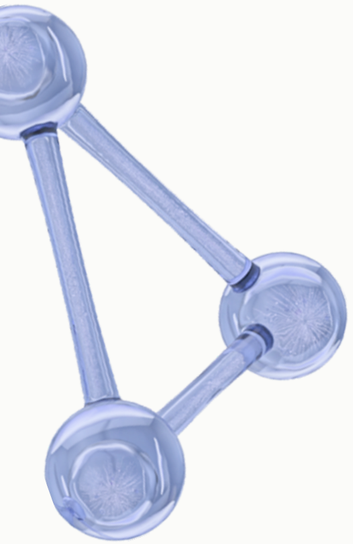
Generative performance of the Transformer VAE model, compared to baseline models in MolecularSets (↑)/ (↓) means higher/lower value is better.

# Limitations

- relatively small size of dataset, compared to benchmark datasets.
- not enough to train big models well
- models requires a lot of computational power to train
- the lack of consideration of synthesis difficulty and in vivo stability

# Conclusions

- Several models were trained to generate chromosomes
- Validity, uniqueness, novelty were predicted mostly well
- Models generated different from training set molecules
- Mediocre quantum yield properties were obtained
- Faced problems due to available computational power, small dataset size, outdated code



# Thank you for attention

# REFERENCES


- https://huggingface.co/jonghyunlee/ChemBERT_ChEMBL_pretrained
- https://github.com/urchade/molgen
- https://github.com/aspuru-guzik-group/chemical_vae
- https://arxiv.org/pdf/2402.11950
- https://github.com/mizuno-group/TransformerVAE/blob/main/usage.md
- https://arxiv.org/pdf/1805.11973