Group Meeting - February 18, 2021

Paper review & Research progress

Truong Son Hy [⋆]

[⋆]Department of Computer Science
The University of Chicago

Ryerson Physical Lab

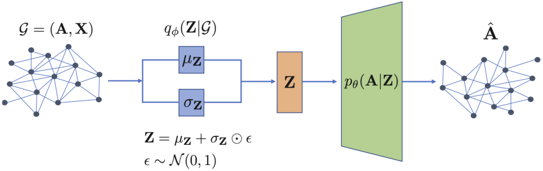
Summary

- **Molecular graph generation**
    - All-at-once method:
        - VAEs [1]
        - GANs [2]
        - (Graph) Normalizing Flows [3, 4]
    - Autoregressive method:
        - CGVAE [5]
        - Reinforcement Learning (RL) [6]
- **3D conformation generation**
    - Distance geometry (DG):
        - GraphDG [7]
        - Euclidean Distance Method (EDM) [8]
    - Generating points directly:
        - Boltzmann generator (BG) [9]
        - CVGAE [10]

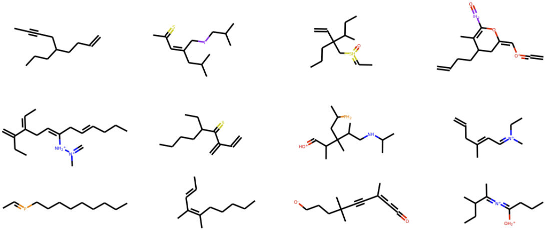
Slides can be found at (look for the date):

`http://people.cs.uchicago.edu/~hytruongson/Discussions-2021/`

VAEs (previous result)


$\mathcal{G} = (\mathbf{A}, \mathbf{X})$
$q_\phi(\mathbf{Z}|\mathcal{G})$
$\mu_\mathbf{Z}$
$\sigma_\mathbf{Z}$
$\mathbf{Z}$
$p_\theta(\mathbf{A}|\mathbf{Z})$
$\hat{\mathbf{A}}$
$\mathbf{Z} = \mu_\mathbf{Z} + \sigma_\mathbf{Z} \odot \epsilon$
$\epsilon \sim \mathcal{N}(0, 1)$

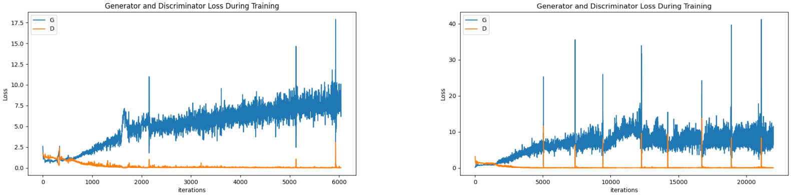
Examples generated by VAEs and higher order message passing on ZINC:



Problem: Unable to generate ring structures (e.g. Benzene).

Vanilla GANs (1)

Training for GAN (the mini-max game) is **tricky**.



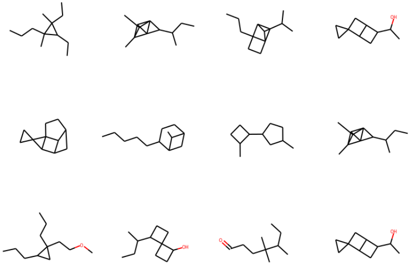
QM9 (left), ZINC (right).
Generator (G), Discriminator (D).

Problem: D does very well in distinguishing fake/non-fake examples, while G only generates crap molecules.
Solution: We need some way to initialize the Generator.

Vanilla GANs (2)

Thus, I apply the **pre-training** mixing VAE and GAN:

1. Start by training a VAE with an encoder and decoder for few epochs.
2. Then, take the decoder of VAE to be inside the generator of GAN. And continue training GAN.



Result: Validity 81.9%, Unique 14.64% and 23 long molecules among 5000 generated. Problem: Mode collapse.

MolGAN (1)

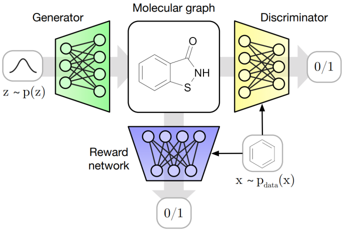


Figure 1. Schema of MolGAN. A vector *z* is sampled from a prior and passed to the generator which outputs the graph representation of a molecule. The discriminator classifies whether the molecular graph comes from the generator or the dataset. The reward network tries to estimate the reward for the chemical properties of a particular molecule provided by an external software.

Note:

- Improved WGAN with optimal transport: Wesserstein-1 distance (Kantorovich-Rubinstein duality).
- Gradient clipping.

MolGAN (2)

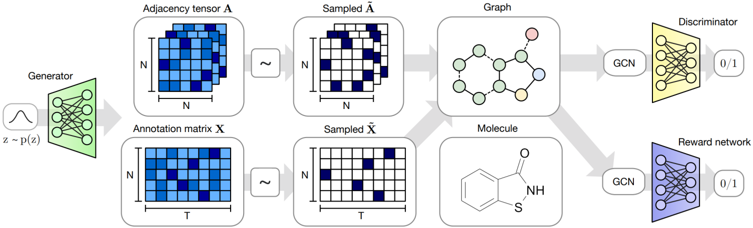
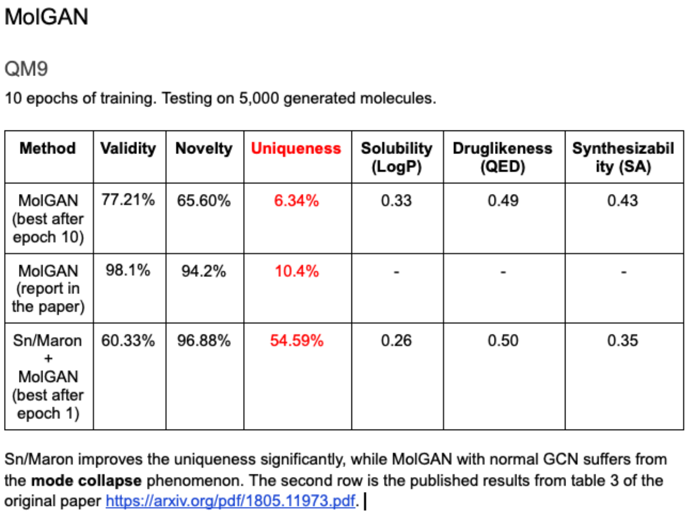

Adjacency tensor $\mathbf{A}$
Sampled $\tilde{\mathbf{A}}$
Graph
Discriminator
Generator
N
N
~
N
N
GCN
0/1
$z \sim p(z)$
Annotation matrix $\mathbf{X}$
Sampled $\tilde{\mathbf{X}}$
Molecule
Reward network
N
T
~
N
T
O
NH
S
GCN
0/1

Figure 2. Outline of MolGAN. From *left*: the generator takes a sample from a prior distribution and generates a dense adjacency tensor $\boldsymbol{A}$ and an annotation matrix $\boldsymbol{X}$. Subsequently, sparse and discrete $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{X}}$ are obtained from $\boldsymbol{A}$ and $\boldsymbol{X}$ respectively via categorical sampling. The combination of $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{X}}$ represents an annotated molecular graph which corresponds to a specific chemical compound. Finally, the graph is processed by both the discriminator and reward networks that are invariant to node order permutations and based on Relational-GCN (Schlichtkrull et al., 2017) layers.

Question

Can higher order message passing (e.g. global $\mathbb{S}_n$/Maron) improve the generation?

MolGAN (3)



MolGAN

QM9

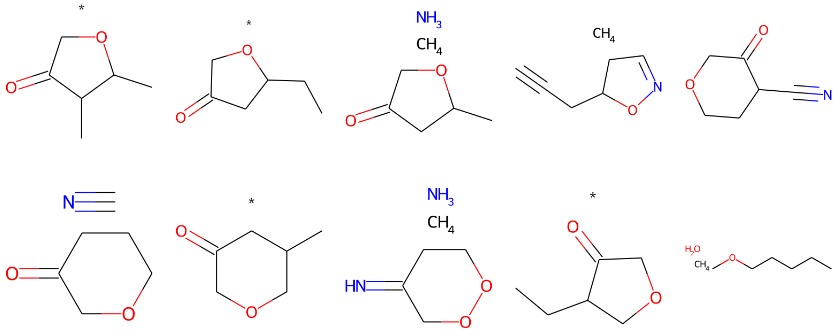
10 epochs of training. Testing on 5,000 generated molecules.

Method	Validity	Novelty	Uniqueness	Solubility (LogP)	Druglikeness (QED)	Synthesizability (SA)
MolGAN (best after epoch 10)	77.21%	65.60%	6.34%	0.33	0.49	0.43
MolGAN (report in the paper)	98.1%	94.2%	10.4%	-	-	-
Sn/Maron + MolGAN (best after epoch 1)	60.33%	96.88%	54.59%	0.26	0.50	0.35

Sn/Maron improves the uniqueness significantly, while MolGAN with normal GCN suffers from the **mode collapse** phenomenon. The second row is the published results from table 3 of the original paper https://arxiv.org/pdf/1805.11973.pdf.

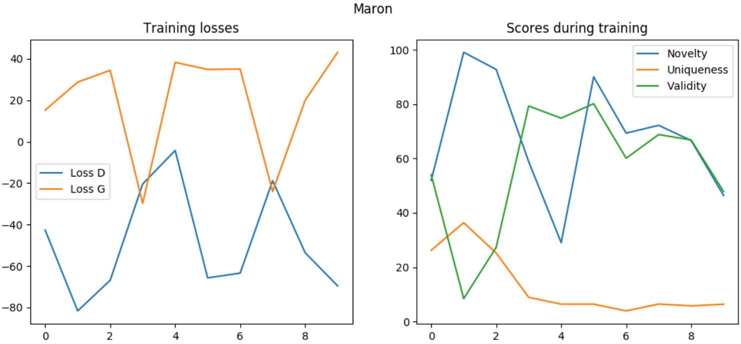
MolGAN (4)

$\mathbb{S}_n$/Maron + MolGAN generated molecules (this is all-at-once generation):



MolGAN (5)

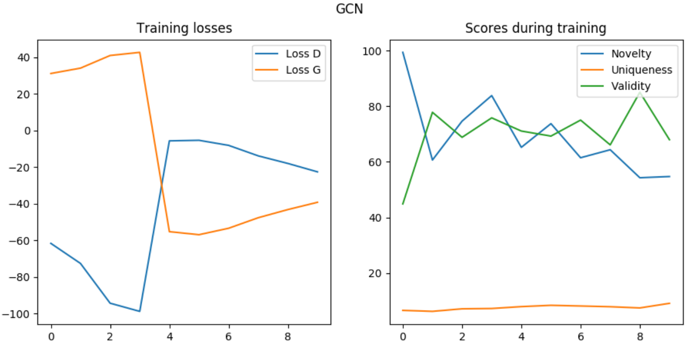
Training curves of $\mathbb{S}_n$/Maron + MolGAN:



The left figure is the training losses of the discriminator and the generator. On the right figure, the mode collapse happens given more epochs (training) – the uniqueness line.

MolGAN (6)

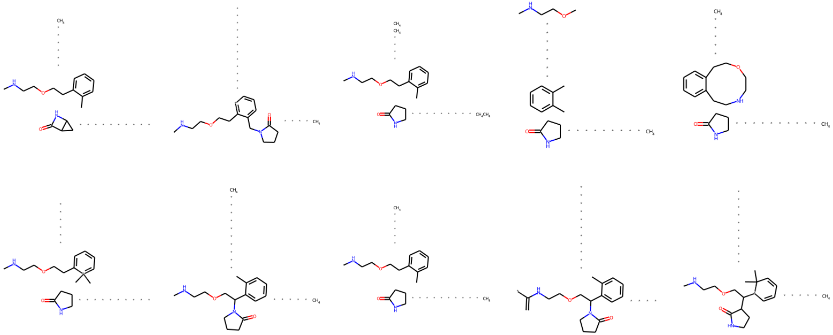
Training curves of MolGAN with the normal GCN:



The mini-max game training in this case seems to reach the **equilibrium** at the 10th epoch. The novelty and validity were always pretty good during the training. The uniqueness increases a bit, but it suffers heavily **mode collapse** since the begining. Next step: ZINC.

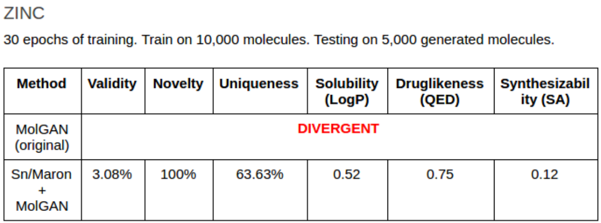
MolGAN (7)

Here are some generated molecules by Sn/Maron + MolGAN on ZINC dataset. The original MolGAN with normal GCN was **divergent**.



Sn/Maron + MolGAN could generate the Benzene ring and some long molecules. In some cases, it generates few valid molecules as disconnected components. It seems the model pickups a certain molecule and generate its nearest variants.

MolGAN (8)

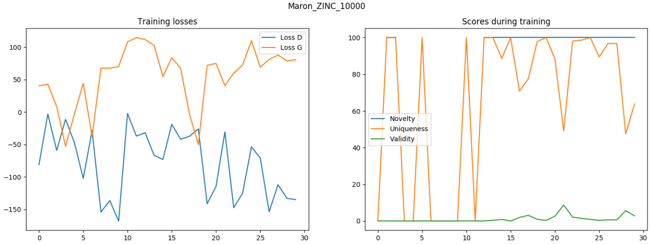


ZINC

30 epochs of training. Train on 10,000 molecules. Testing on 5,000 generated molecules.

Method	Validity	Novelty	Uniqueness	Solubility (LogP)	Druglikeness (QED)	Synthesizability (SA)
MolGAN (original)	DIVERGENT					
Sn/Maron + MolGAN	3.08%	100%	63.63%	0.52	0.75	0.12

The training in general is bad. It didn't reach any equilibrium, but it didn't become divergent (NaN) as the original one either.



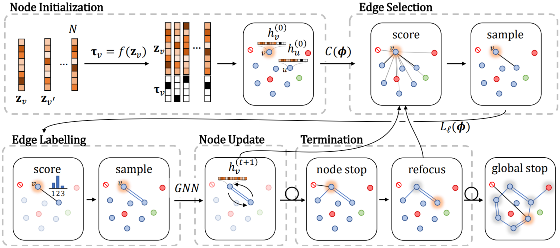
Maron_ZINC_10000

Training losses

Scores during training

Autoregressive (1)

Move to **autoregressive** approach (not RL) – Constrained Graph Variational Autoencoders for Molecule Design (NeurIPS 2018):



At first, we sample the each vertex latent z independently (so this is the first order), then iteratively we add new edge to the existing graph (given randomly selected node as the start). We apply second-order message passing (with gated recurrent architecture) to produce the probability of (u, v) where one vertex is in the existing graph and the another one is outside, and also the probability of its label.

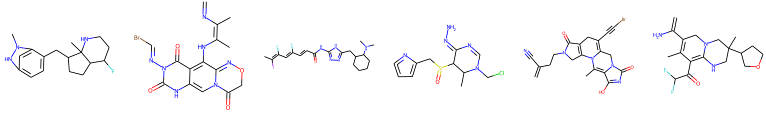
Autoregressive (2)

The difference between this architecture and RL approach is:

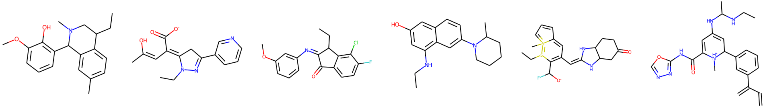
1. It uses VAE.
2. In the generation process, the RL selects a new atom type during the construction but this one basically has the atom types fixed at the beginning, we only select a new edge/bond.
3. RL is **unstable** and hard to train; only after we construct the whole molecular graph, we might get some rewards. And in RL, atom labels are not known, the model must decide to add an atom into the canvas or terminate.
4. It is more like **intimidation learning** in Eric Jonas's paper: actually we break down the generation process into multiple classifications, each classification is given an input as a partial graph and we have to predict the next edge (only the next edge, because the vertex/atom labels are known already).

Autoregressive (3)

I think it is easier for a graph nets in general to do the classification tasks rather than **all-at-once** generation. Much better generation!

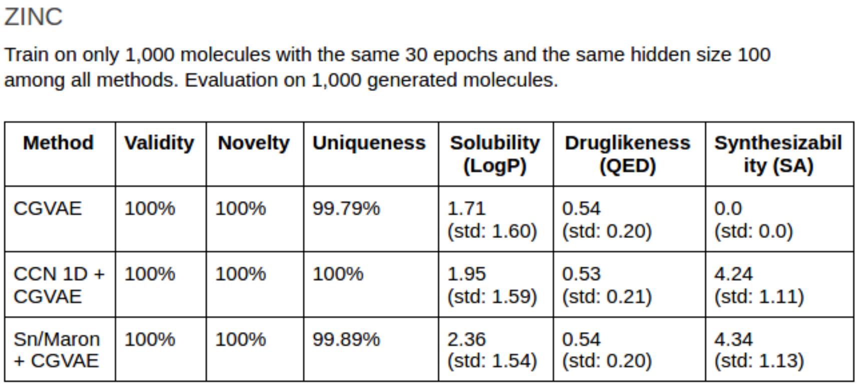


Examples generated with (global) Sn/Maron



Examples generated with (global) Sn/Maron + (local) CCN 1D
It seems to like Benzen rings (in almost every generated molecule)

Autoregressive (4)

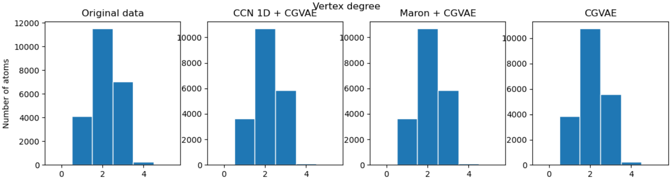


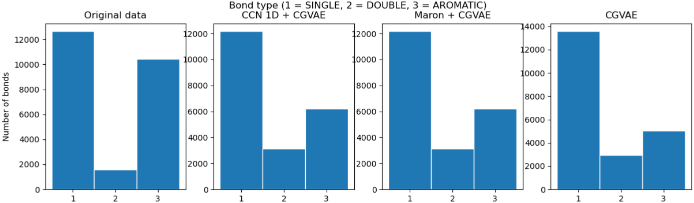
ZINC

Train on only 1,000 molecules with the same 30 epochs and the same hidden size 100 among all methods. Evaluation on 1,000 generated molecules.

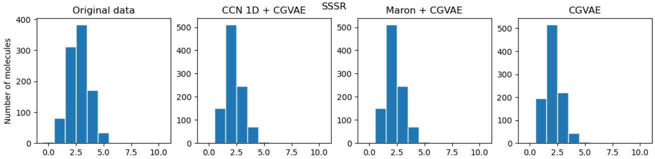
Method	Validity	Novelty	Uniqueness	Solubility (LogP)	Druglikeness (QED)	Synthesizability (SA)
CGVAE	100%	100%	99.79%	1.71 (std: 1.60)	0.54 (std: 0.20)	0.0 (std: 0.0)
CCN 1D + CGVAE	100%	100%	100%	1.95 (std: 1.59)	0.53 (std: 0.21)	4.24 (std: 1.11)
Sn/Maron + CGVAE	100%	100%	99.89%	2.36 (std: 1.54)	0.54 (std: 0.20)	4.34 (std: 1.13)

Autoregressive (5)

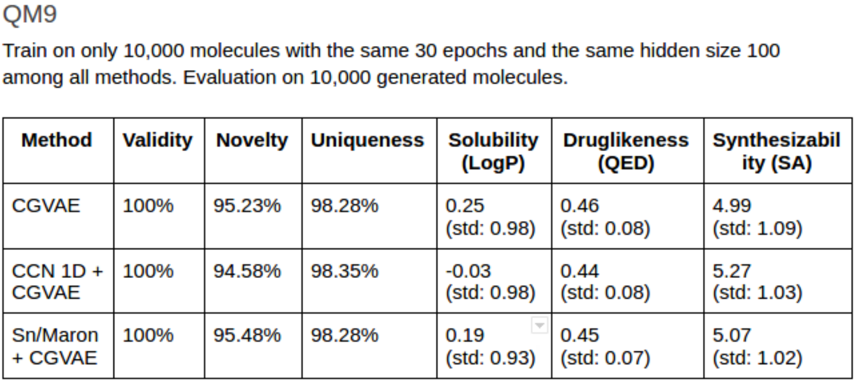




SSSR = smallest set of smallest rings



Autoregressive (6)

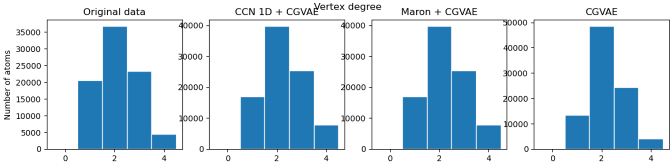


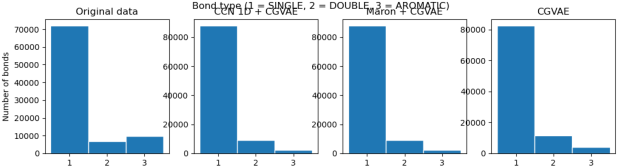
QM9

Train on only 10,000 molecules with the same 30 epochs and the same hidden size 100 among all methods. Evaluation on 10,000 generated molecules.

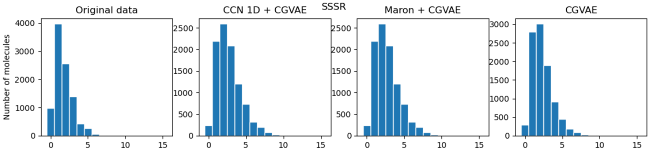
Method	Validity	Novelty	Uniqueness	Solubility (LogP)	Druglikeness (QED)	Synthesizability (SA)
CGVAE	100%	95.23%	98.28%	0.25 (std: 0.98)	0.46 (std: 0.08)	4.99 (std: 1.09)
CCN 1D + CGVAE	100%	94.58%	98.35%	-0.03 (std: 0.98)	0.44 (std: 0.08)	5.27 (std: 1.03)
Sn/Maron + CGVAE	100%	95.48%	98.28%	0.19 (std: 0.93)	0.45 (std: 0.07)	5.07 (std: 1.02)

Autoregressive (7)





SSSR = smallest set of smallest rings



Brainstorm

- **All-at-once method:**
  - Preserving permutation equivariance. Nice theory.
  - **But:** VAEs cannot generate ring structures. GANs and variants are hard to train and have the mode collapse phenomenon.
- **Autoregressive:**
  - Much better result!
  - **But:** Not preserving the equivariance.

Move forward

- Higher-order latent structure (exploiting the exchangability)?
- New datasets rather than QM9 and ZINC?
- We have encouraging incremental results, but we must target for a publication in the field.

Next time: **3D conformation generation**.

Reference – Molecular graph generation

[1] The general theory of permutation equivarant neural networks and higher order graph variational encoders
https://arxiv.org/abs/2004.03990

[2] MolGAN: An implicit generative model for small molecular graphs
https://arxiv.org/abs/1805.11973

[3] Variational Inference with Normalizing Flows
https://arxiv.org/abs/1505.05770

[4] Graph Normalizing Flows
https://arxiv.org/abs/1905.13177

[5] Constrained Graph Variational Autoencoders for Molecule Design
https://arxiv.org/abs/1805.09076

[6] Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation
https://arxiv.org/abs/1806.02473

Reference – 3D conformation generation

[7] A Generative Model for Molecular Distance Geometry
`https://arxiv.org/abs/1909.11459`
[8] Generating valid Euclidean distance matrices
`https://arxiv.org/abs/1910.03131`
[9] Boltzmann Generators – Sampling Equilibrium States of Many-Body Systems with Deep Learning
`https://arxiv.org/abs/1812.01729`
[10] Molecular Geometry Prediction using a Deep Generative Graph Neural Network
`https://www.nature.com/articles/s41598-019-56773-5`