# Group Meeting - January 29, 2021

## Paper review & Research progress

Truong Son Hy *

*Department of Computer Science
The University of Chicago

Ryerson Physical Lab

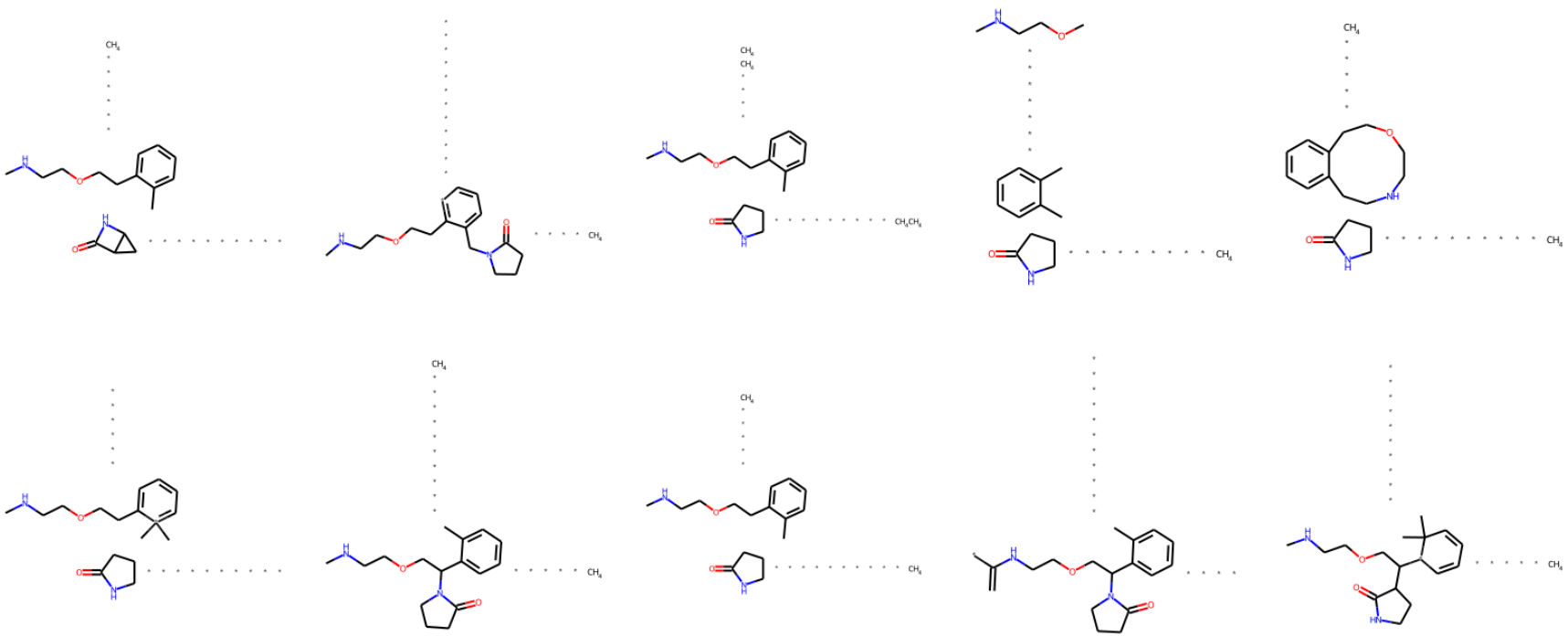
# Content

- Research update
- Literature review:
  1. **Molecular geometry prediction using a deep generative graph neural network** (Nature)
     `https://arxiv.org/abs/1904.00314`
     `https://www.nature.com/articles/s41598-019-56773-5`

  This is the work we can improve.

# Research update (1)

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.

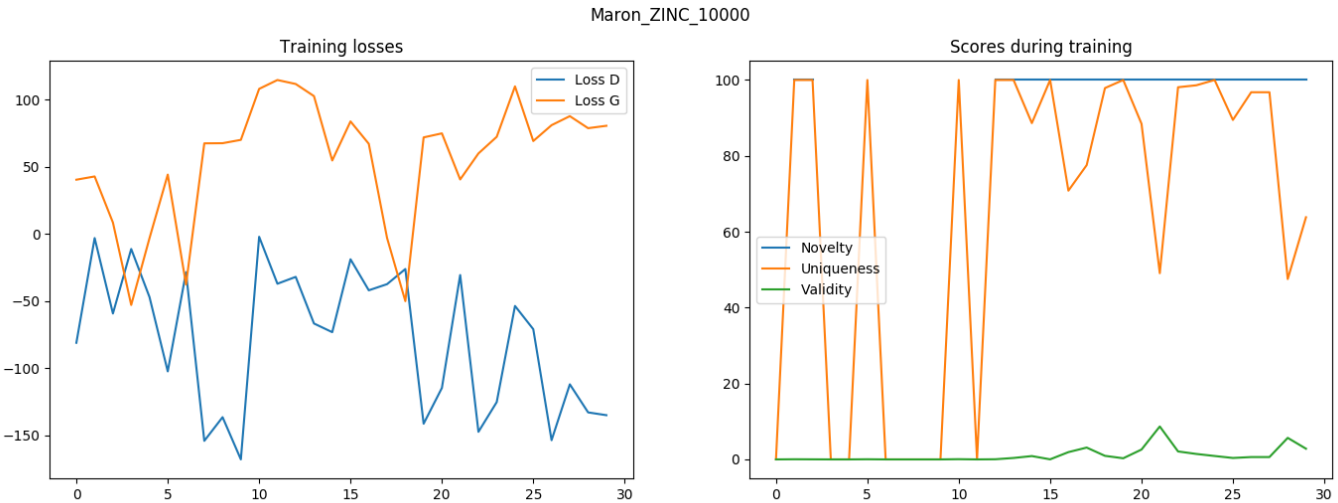
# Research update (2)

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.



# Research update (3)

**Next task:**

- Molecular geometry prediction/generation.
- I could reproduce the results from the released source code

  `https://github.com/nyu-dl/dl4chem-geometry`

  It is in TensorFlow 1.
- The Sn/Maron model is implemented. The graph in this case is **complete**, so second-order message passing makes sense here.
- We need SO(3)-equivariant model in TensorFlow 1. The sizes of QM9 and COD datasets are still small enough to be handled by FastCG (C++) library. I wrote the wrapper for it in TF 1 before.

# Paper 1

**Molecular geometry prediction using a deep generative graph neural network** (Nature)

Elman Mansimov, Omar Mahmood, Seokho Kang, Kyunghyun Cho

https://arxiv.org/abs/1904.00314

https://www.nature.com/articles/s41598-019-56773-5

# Introduction (1)

The 3D coordinates of atoms in a molecule are commonly referred to as the molecule's geometry or **conformation** that determines the reactions it participates in, the bonds it forms, and the interactions it has with other molecules.

## Applications of conformation generation

1. Generating 3D quantitative structure-activity relationships (QSAR).
2. Structure-based virtual screening.
3. Pharmacophore modeling.

## Computation issues

Conformations can be determined in a physical setting by:

- Instrumental techniques: X-ray crystallography.
- Experimental techniques.

But costly and time-consuming.

# Introduction (2)

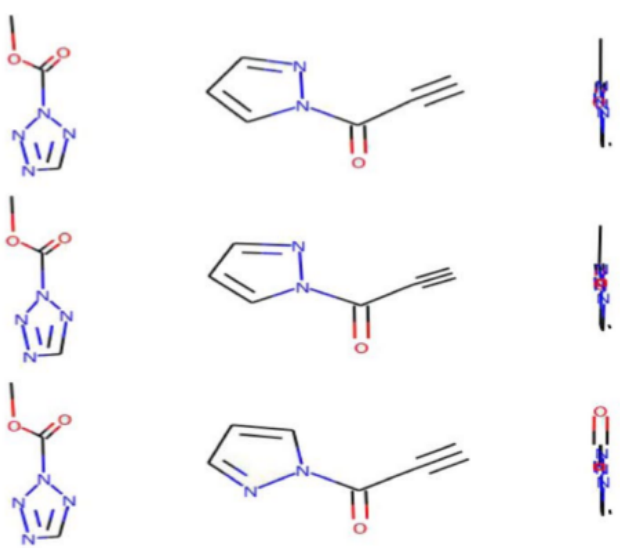
Computational method:

1. **Hand-design** a force field energy function to **approximate** the molecule's true potential energy, based on the molecule's atoms, bonds and coordinates.
2. The minimum of this energy function corresponds to the molecule's most stable configuration – **lowest-energy** conformations.
3. Hand-design functions might yield inaccurate approximations.

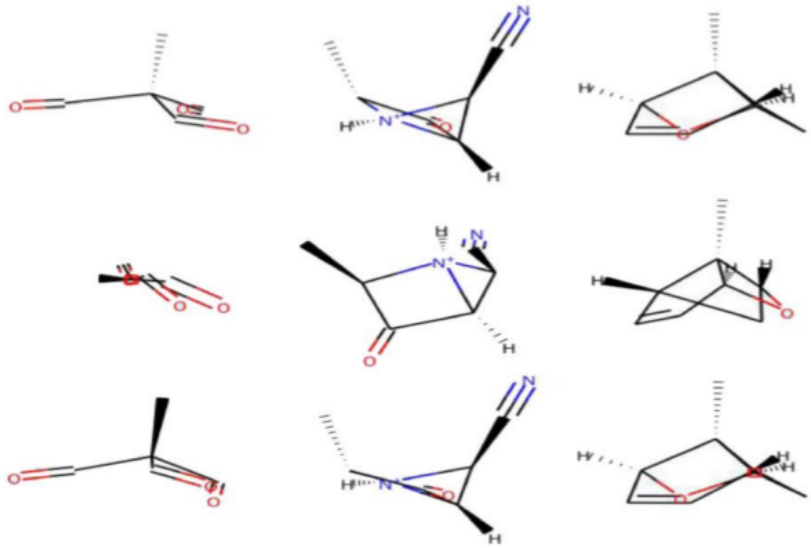
## Proposal

A deep generative graph neural network that **learns the energy function from data** in an end-to-end fashion **by generating molecular conformations** that are energetically favorable and more likely to be observed experimentally.

# Introduction (3)



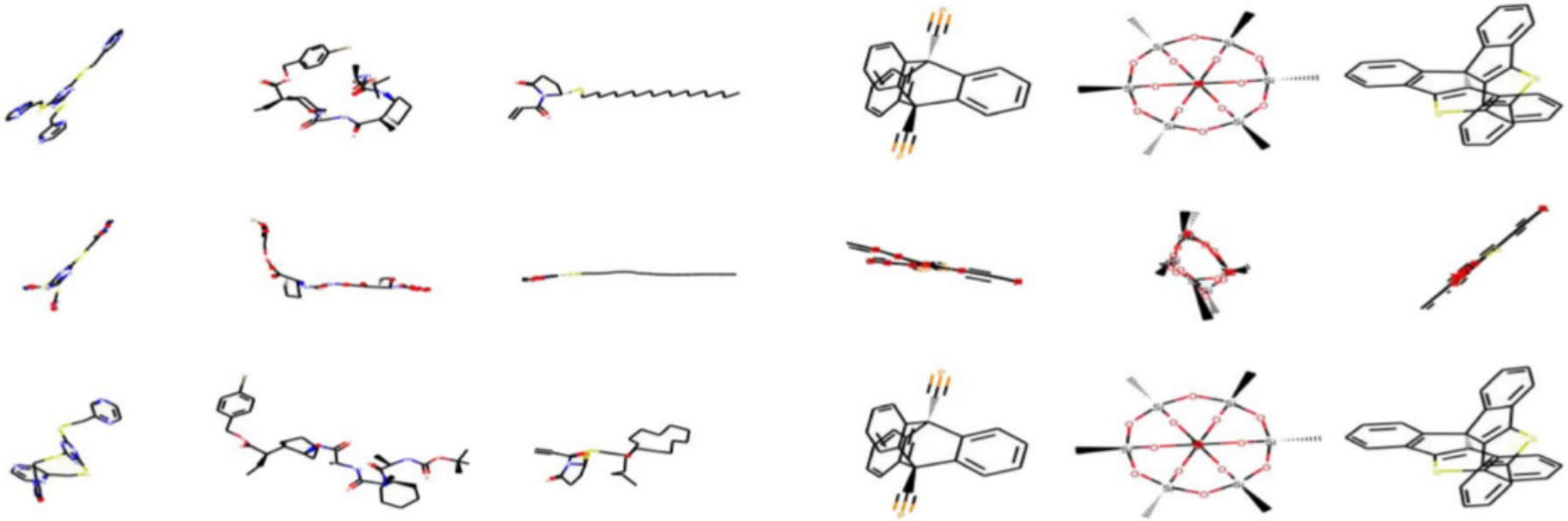
**(a)** QM9 greatest difference in favour of neural network predictions



**(b)** QM9 greatest difference in favour of ETKDG + MMFF predictions

**Figure 5.** This figure shows the three molecules in each dataset for which the differences between the RMSDs of the neural network predictions and the baseline ETKDG + MMFF predictions were greatest in favour of the neural network predictions ($max(RMSD_{CVGAE} - RMSD_{ETKDG+MMFF})$), and the three for which this difference was greatest in favour of the ETKDG + MMFF predictions ($max(RMSD_{ETKDG+MMFF} - RMSD_{CVGAE})$). The top row of each subfigure contains the reference molecules, the middle row contains the neural network predictions and the bottom row contains the conformations generated by applying MMFF to the reference conformations.

# Introduction (4)



**(c)** COD greatest difference in favour of neural network predictions

**(d)** COD greatest difference in favour of ETKDG + MMFF predictions

**Figure 5.** This figure shows the three molecules in each dataset for which the differences between the RMSDs of the neural network predictions and the baseline ETKDG + MMFF predictions were greatest in favour of the neural network predictions ($max(RMSD_{CVGAE} - RMSD_{ETKDG+MMFF})$), and the three for which this difference was greatest in favour of the ETKDG + MMFF predictions ($max(RMSD_{ETKDG+MMFF} - RMSD_{CVGAE})$). The top row of each subfigure contains the reference molecules, the middle row contains the neural network predictions and the bottom row contains the conformations generated by applying MMFF to the reference conformations.

# Conformation generation (1)

## Molecule

We consider a molecule as an undirected, **complete graph** $G = (V, E)$. There are $M$ vertices and $M(M-1)/2$ edges. Each atom and each edge are represented as a vector $v_i \in \mathbb{R}^{d_v}$ and $e_{ij} \in \mathbb{R}^{d_e}$, respectively.

## Plausible conformation

A plausible conformation corresponds to a **stable** configuration of a molecule. Molecular geometry prediction is the generation of a set of plausible conformations $X_a = (x_1^a, .., x_M^a)$, where $x_i^a \in \mathbb{R}^3$ is the 3D coordinates of the $i$-th atom in the $a$-th conformation.

**Note:**

$$\text{Stability} = \text{Local minima}$$

# Conformation generation (2)

We formulate conformation generation as finding (local) **minima** of an energy function $\mathcal{F}(X, G)$ defined on a pair of molecule graph and conformation:

$$\{X_1, .., X_S\} = \arg\min_X \mathcal{F}(X, G)$$

where $\mathcal{F}$ can be:

1. Universal Force Field (UFF)
2. Merck Molecular Force Field (MMFF)

Alternatively, we could do the **Gibbs sampling**:

$$\{X_1, .., X_S\} \sim p_{\mathcal{F}}(X|G)$$

where:

$$p_{\mathcal{F}}(X|G) = \frac{1}{\mathcal{C}(G)} \exp\{-\mathcal{F}(X, G)\}$$

where $\mathcal{C}$ is a normalizing constant, and $S$ is the number of conformations we generate for each molecule.

# Generative model approach (1)

## Conventional approach (e.g. graphdg, EDM, etc.)

1. Use **distance geometry** (DG) or its variants to randomly generate an initial conformation that satisfifes various geometric constraints.
2. Run the minimization many times. Because of the non-convexity of the energy function, each run is likely to end up in a unique local minimum. We collect them as a set of conformations.

## Data-driven generative approach

Dataset $\mathcal{D} = \{(G_1, X_1^*), .., (G_N, X_N^*)\}$ where $X_n^*$ is a **reference** conformation (energetically favorable) that might not correspond to the lowest energy. Optimization of learning an energy function:

$$\hat{F}(G, X) = \arg\max_{\mathcal{F}} \frac{1}{N} \sum_{n=1}^{N} \log p_{\mathcal{F}}(X_n^* | G_n)$$

# Generative model approach (2)

## Conditional Variational Graph Autoencoders

Introduce a set of latent variables $Z = \{z_1, .., z_M\}$ where $z_m \in \mathbb{R}^{d_z}$:

$$\log p(X|G) = \log \int p(X|Z, G)p(Z|G)dZ$$

We instead maximize the stochastic approximation of its lower bound:

$$\log p(X|G) \geq \mathbb{E}_{Z\sim Q(Z|G,X)}[\log p(X|Z, G)] - \mathcal{D}_{\mathrm{KL}}(Q(Z|G, X)||P(Z|G))$$

$$\approx \frac{1}{K}\sum_{k=1}^{K} \log p(X|Z^k, G) - \mathcal{D}_{\mathrm{KL}}(Q(Z|G, X)||P(Z|G))$$

where $Z^k$ is the $k$-th sample from the (approximate) posterior distribution $Q$. We assume the posterior $Q$ and prior $P$ to be normal.

# Generative model approach (3)

We are going to model the posterior $Q(Z|G, X)$ and prior $P(Z|G)$ by MPNN of $L$ layers:

$$h^{\ell}(v_i) = \mathbf{GRU}(h^{\ell-1}(v_i), J(h^{\ell-1}(v_i), h^{\ell-1}(v_{j\neq i}), h(e_{i,j\neq i}))$$

where:

- Node feature vector $h(v_i) \in \mathbb{R}^{d_h}$
- Edge feature matrix $h(e_{ij}) \in \mathbb{R}^{d_h \times d_h}$
- $J$ is a neural network aggregating information from the neighbors.

# Generative model approach (4)

## Prior parameterization

1. Initialize $h^0(v_i)$ and $h(e_{ij})$ as linear transformations from $v_i$ and $e_{ij}$:

$$h^0(v_i) = U_{\text{node}}^{\text{prior}} v_i, \qquad h(e_{ij}) = U_{\text{edge}}^{\text{prior}} e_{ij}$$

2. Diagonal covariance matrix:

$$\mu_i = W_\mu^{\text{prior}} h^L(v_i) + b_\mu^{\text{prior}}$$

$$\sigma_i^2 = \exp\{W_\sigma^{\text{prior}} h^L(v_i) + b_\sigma^{\text{prior}}\}$$

3. Prior distribution (in log):

$$\log p(Z|G) = \sum_{i=1}^{N} \sum_{j=1}^{3} -\frac{(\mu_{i,j} - z_{i,j})^2}{2\sigma_{i,j}^2} - \log \sqrt{2\pi\sigma_{i,j}^2}$$

# Generative model approach (5)

## Likelihood parameterization

We use a similar MPNN to model the likelihood distribution $P(X|Z,G)$ conditioned on both $G=(V,E)$ and the latent set $Z=\{z_1,..,z_M\}$.

1. Initialization:

$$h^0(v_i) = U_{\text{node}}^{\text{likelihood}} v_i, \qquad h(e_{ij}) = U_{\text{edge}}^{\text{likelihood}} e_{ij}$$

2. With a new set of parameters:

$$\{\theta_{\text{likelihood}}, W_\mu^{\text{likelihood}}, b_\mu^{\text{likelihood}}, W_\sigma^{\text{likelihood}}, b_\sigma^{\text{likelihood}}\}$$

The final mean and variance vectors are now in 3D.

# Generative model approach (6)

## Posterior parameterization

1. Computing the exact posterior $P(Z|G,X)$ is **intractable**, so we use a parameterized, approximate posterior $Q(Z|G,X)$ (amortized inference).
2. Parameterization by MPNN is similar, except the edge feature initialization:

$$h(e_{ij}) = U_{\text{edge}}^{\text{posterior}} \begin{bmatrix} e_{ij} \\ D(x_i^*) \end{bmatrix}$$

where $D(X^*)$ is the distance (proximity) matrix $D(X^*)$ of the reference 3D conformation $X^*$.

# Training & Inference (1)

The function aligns (by translation and rotation) the reference conformation to the predicted conformation and returns the aligned reference conformation $\hat{X} = R(X, X^*)$ such that it has the smallest distance RMSD:

$$\text{RMSD}(\hat{X}, X^*) = \sqrt{\frac{1}{M}\sum_{i=1}^{M} ||\hat{x}_i - x_i^*||^2}$$

**Note:** The authors didn't mention the detail of $R$. I think they used **Kabsh algorithm** based on the code.

## Likelihood

$$\log p(X|G, Z) = \sum_{i=1}^{M}\sum_{j=1}^{3} -\frac{(\mu_{i,j} - \hat{x}_{i,j}^*)^2}{2\sigma_{i,j}^2} - \log\sqrt{2\pi\sigma_{i,j}^2}$$

where:

$$\{\hat{x}_1^*, .., \hat{x}_M^*\} = R(\{\mu_1, .., \mu_M\}, X^*)$$

# Training & Inference (2)

## Kabsh algorithm

- Method for calculating the optimal rotation matrix that minimizes the RMSD between two paired sets of points.
- When both translation and rotation are performed, it is called the **orthogonal Procrustes algorithm**.

Given 2 sets of points $P$ and $Q$ in 3D as $N \times 3$ matrices.

1. Translation (to zero):

$$P \leftarrow P - \frac{1}{N}\sum_{p \in P} p, \qquad Q \leftarrow Q - \frac{1}{N}\sum_{q \in Q} q$$

2. Covariance matrix: $H = P^T Q$
3. Optimal translation: $R = (H^T H)^{1/2} H^{-1}$ (this can also be done by SVD)

# Training & Inference (3)

Original objective:

$$\mathcal{L} = \frac{1}{K} \sum_{k=1}^{K} \log p(X|Z^k, G) - \mathcal{D}_{\mathrm{KL}}(Q(Z|G, X)||P(Z|G))$$

Unconditional prior regularization:

$$\mathcal{L} = \log p(X|Z^1, G) - \mathcal{D}_{\mathrm{KL}}(Q(Z|G, X)||P(Z|G)) - \alpha \cdot \mathcal{D}_{\mathrm{KL}}(P(Z|G)||P(Z))$$

assuming $K = 1$ and $\alpha \geq 0$. The unconditional prior distribution $P(Z)$ is a factorized Normal distribution:

$$P(Z) = \prod_{i=1}^{M} \mathcal{N}(z_i|0, I)$$

# Training & Inference (4)

## Inference

Predicting molecular geometry:

1. Sample from the prior distribution $\tilde{Z} \sim P(Z|G)$.
2. Sample from the likelihood distribution $\tilde{X} \sim P(X|\tilde{Z}, G)$.

In practice, we fix the output variance $\sigma_{i,j}$ of the likelihood distribution to be 1 and take the mean set $\{\mu_1, .., \mu_M\}$ as a sample from the model.

# Experiments (1)

Two public datasets (the another one is private):

1. **QM9:**
   - 133,015 molecules containing of 9 heavy atoms of types C, N, O, F.
   - Each molecule is paired with a reference conformation from DFT.
   - Hold out 5,000 for each validation and testing.
2. **COD:**
   - Organic part of the COD dataset.
   - Containing no more than 50 heavy atoms of types B, C, N, O, F, Si, P, S, Cl, Ge, As, Se, Br, Te, and I.
   - 66,663 molecules in total with 3,000 hold out for each validation and testing.
   - Reference conformations are either from experiment or by DFT calculations.

# Experiments (2)

## The baseline for comparison:

- A conformation is created by **ETKDG** (using distance geometry).
- Then use either the energy function from **UFF** or **MMFF** to optimize the conformation.
- The baselines are referred to as **ETKDG** + **UFF** and **ETKDG** + **MMFF**.

## Two modes of Inference

1. Pure **CVGAE**: first sampling from the prior distribution and taking the mean vectors from the likelihood distribution.
2. **CVGAE** + **MMFF**: further optimize the generated samples by **MMFF**.

# Experiments (3)

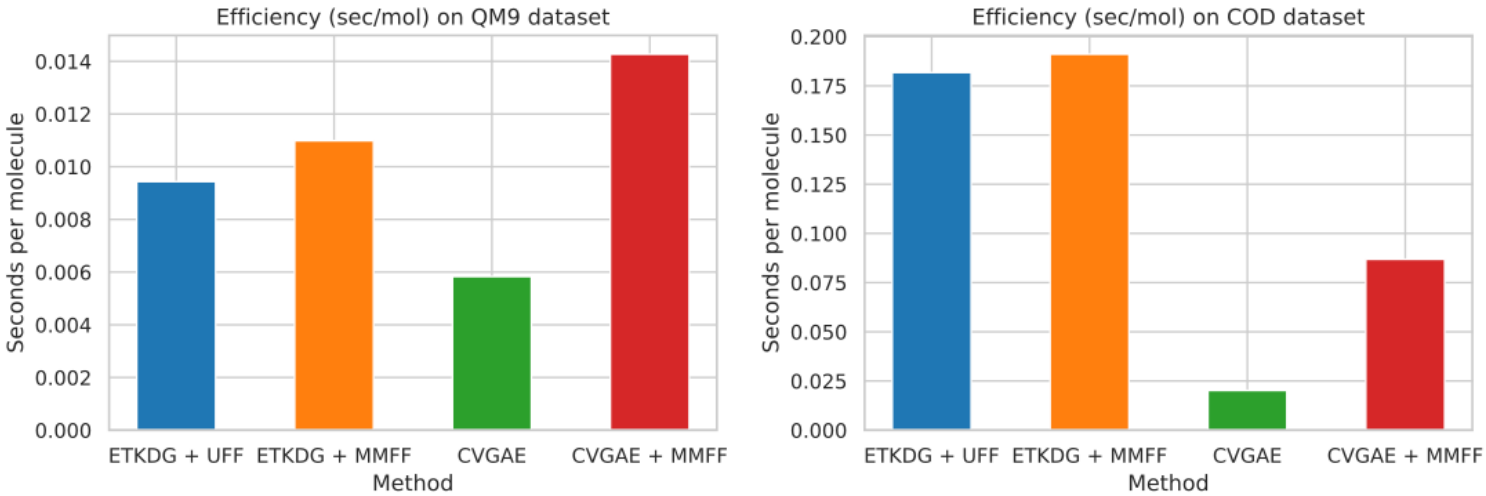
| Dataset | | ETKDG + Force Field | | CVGAE | CVGAE + Force Field |
|---|---|---|---|---|---|
| | | UFF | MMFF | | MMFF |
| QM9 | success per test set | 96.440% | 96.440% | 100% | 99.760% |
| | success per molecule | 98.725% | 98.725% | 100% | 98.684% |
| | mean | 0.425 | 0.415 | 0.390 | 0.367 |
| | std. dev. | 0.176 | 0.189 | 0.017 | 0.074 |
| | best | 0.126 | 0.092 | 0.325 | 0.115 |
| COD | success per test set | 99.133% | 99.133% | 100% | 95.367% |
| | success per molecule | 99.627% | 99.627% | 100% | 99.071% |
| | mean | 1.389 | 1.358 | 1.331 | 1.656 |
| | std. dev. | 0.407 | 0.415 | 0.099 | 0.425 |
| | best | 0.429 | 0.393 | 1.206 | 0.635 |
| CSD | success per test set | 97.400% | 97.400% | 100% | 99.467% |
| | success per molecule | 99.130% | 99.130% | 100% | 97.967% |
| | mean | 1.537 | 1.488 | 1.506 | 1.833 |
| | std. dev. | 0.421 | 0.418 | 0.115 | 0.434 |
| | best | 0.508 | 0.478 | 1.343 | 0.784 |

**Table 1.** Number of successfully processed molecules in the test set (success per test set 100), number of successfully generated conformations out of 100 (success per molecule ↑), median of mean RMSD (mean ↓), median of standard deviation of RMSD (std. dev. ↓) and median of best RMSD (best ↓) per molecule on QM9, COD and CSD datasets. ETKDG stands for Distance Geometry with experimental torsion-angle preferences. UFF and MMFF are force field methods and stand for Universal Force Field and Molecular Mechanics Force Field respectively. CVGAE stands for Conditional Variational Graph Autoencoder. CVGAE + Force Field represents running the MMFF force field optimization initialized by CVGAE predictions.

# Experiments (4)

**Table 2.** Conformation Diversity. Mean and std. dev. represents the corresponding mean and standard deviation of pairwise RMSD between at most 100 generated conformations per molecule.

| Dataset | | ETKDG + MMFF | CVGAE | CVGAE + MMFF |
|---|---|---|---|---|
| QM9 | mean | 0.400 | 0.106 | 0.238 |
| | std. dev. | 0.254 | 0.061 | 0.209 |
| COD | mean | 1.148 | 0.239 | 1.619 |
| | std. dev. | 0.699 | 0.181 | 0.537 |
| CSD | mean | 1.244 | 0.567 | 1.665 |
| | std. dev. | 0.733 | 0.339 | 0.177 |

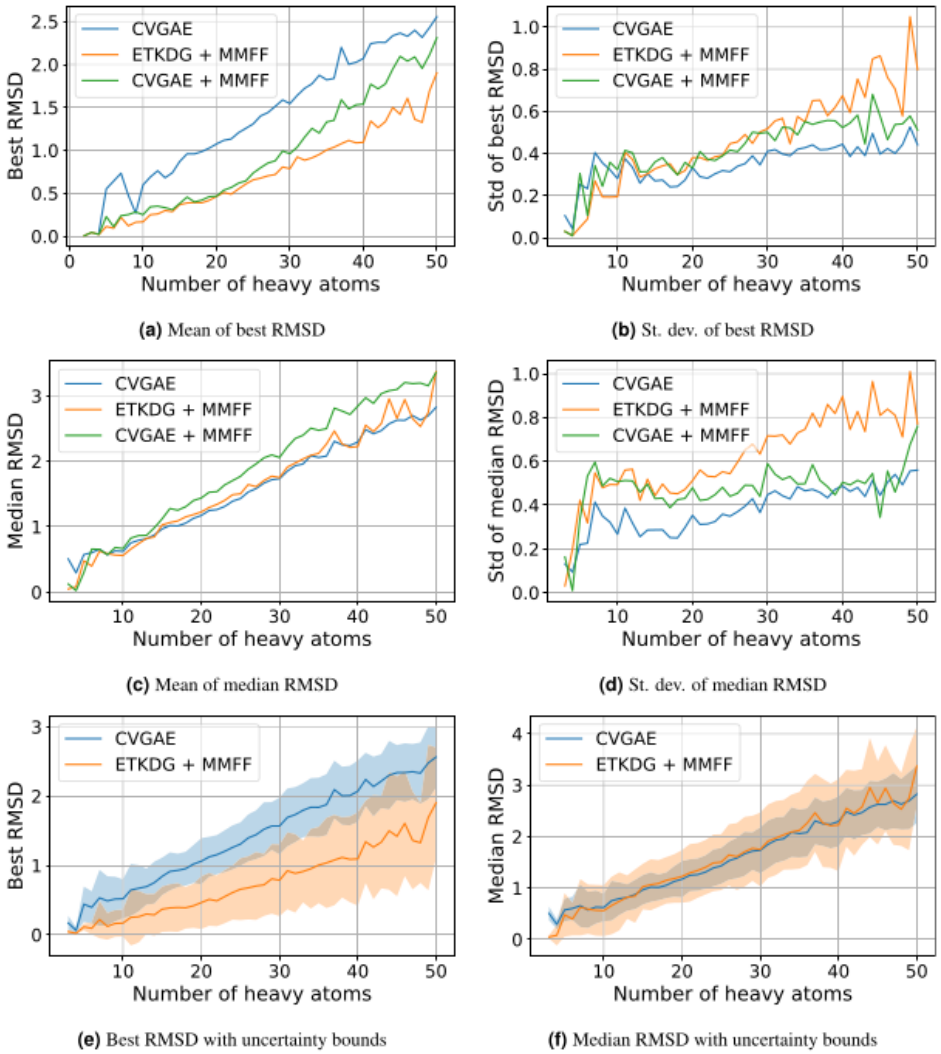



(a) QM9 dataset

(b) COD dataset

**Figure 2.** Computational efficiency of various approaches on QM9 and COD datasets.

# Experiments (5)




**(a)** Mean of best RMSD

**(b)** St. dev. of best RMSD

**(c)** Mean of median RMSD

**(d)** St. dev. of median RMSD

**(e)** Best RMSD with uncertainty bounds

**(f)** Median RMSD with uncertainty bounds

**Figure 3.** This figure shows the means and standard deviations of the best and median RMSDs on the union of COD and CSD datasets as a function of number of heavy atoms. The molecules were grouped by number of heavy atoms, and the mean and standard deviation of the median and best RMSDs were calculated for each group to obtain these plots. Groups at the left hand side of the graph with less than 1% of the mean number of molecules per group were omitted.

# Experiment with second-order message passing on QM9

| Method | Mean | STD |
|---|---|---|
| CVGAE (from paper) | 0.390 | 0.017 |
| CVGAE (rerun) | 0.606 | 0.057 |
| Sn/Maron + CVGAE | 0.659 | 0.060 |

**Note:**

1. The results of both the rerun and Sn/Maron are at 150th epoch. The result from the paper seems to be at 2,500th epoch (as seen in the code).
2. Due to time consuming, the network of Sn/Maron is scaled by half smaller in size.
3. I think there is an **algorithmic** bug in their MPNN implementation.
4. I think: Rotational-equivariant model will outperform these baselines given a smaller number of training examples.