Group Meeting - January 22, 2021

Paper review & Research progress

Truong Son Hy *

*Department of Computer Science
The University of Chicago

Ryerson Physical Lab

President Joe Biden

Weeping may endure for a night, but joy cometh in the morning.

Content

- Research update
- Literature review:
  1. **Learning Neural Generative Dynamics for Molecular Conformation Generation** (ICLR 2021)
     `https://openreview.net/pdf?id=pAbm1qfheGk`
  2. **Generating valid Euclidean distance matrices**,
     `https://arxiv.org/abs/1910.03131`

Research update (1)

MolGAN

QM9

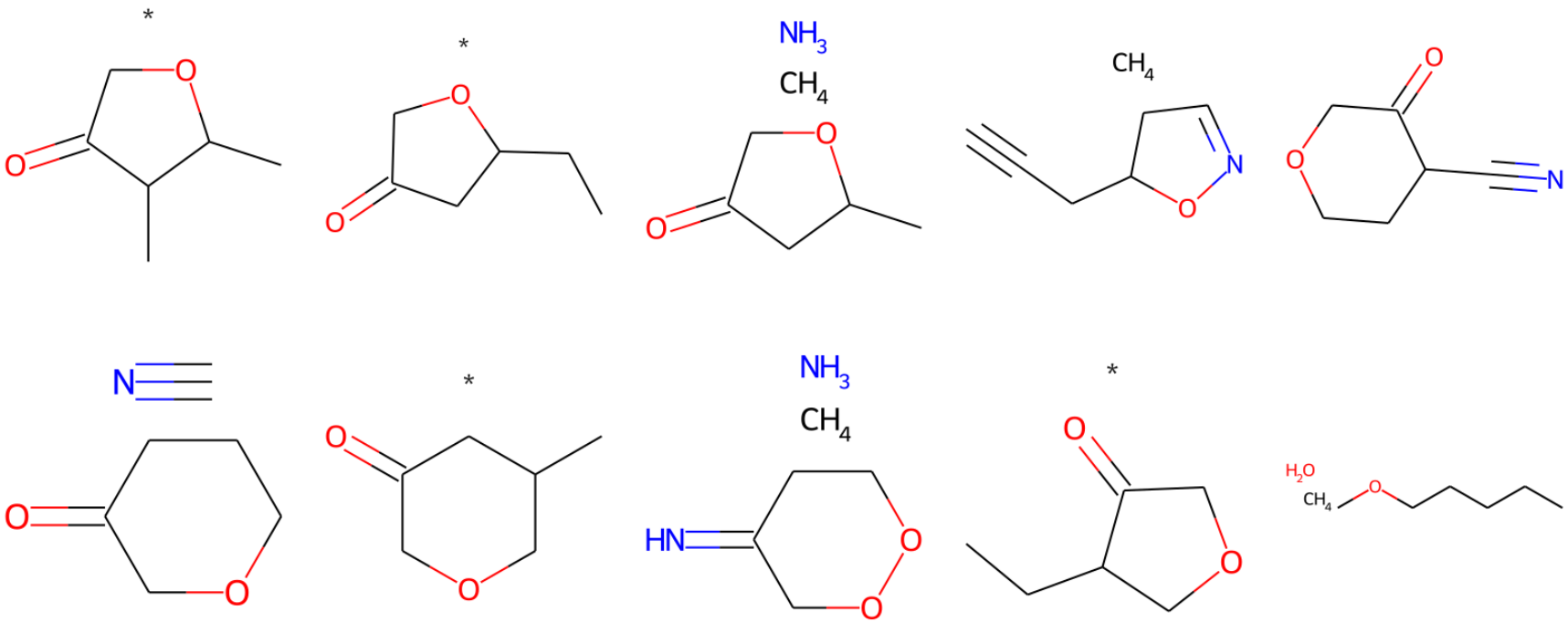
10 epochs of training. Testing on 5,000 generated molecules.

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.

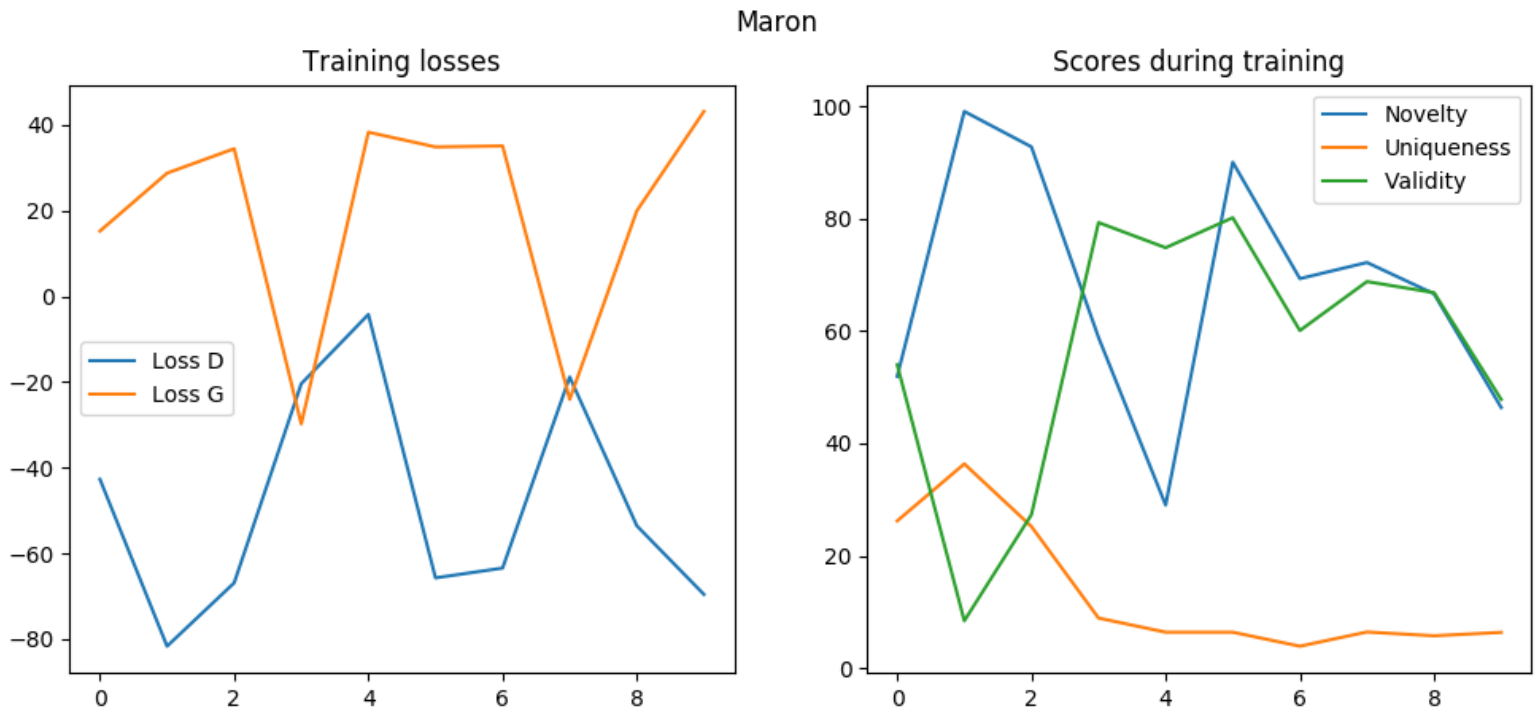
Research update (2)

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



Research update (3)

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.

Research update (4)

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.

Paper 1

Learning Neural Generative Dynamics for Molecular Conformation Generation (ICLR 2021)

Minkai Xu, Shitong Luo, Yoshua Bengio, Jian Peng, Jian Tang

`https://openreview.net/pdf?id=pAbm1qfheGk`

Note

The old version of this paper appeared in the NeurIPS 2020 workshop that I presented before. The authors then revived, added missing technical details and improved the evaluation.

Generative model (1)

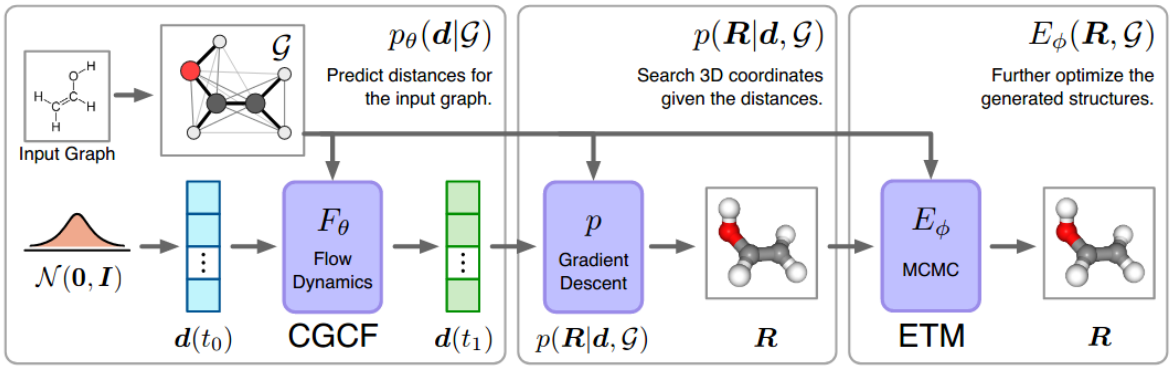


Figure 1: Illustration of the proposed framework. Given the molecular graph, we 1) first draw latent variables from a Gaussian prior, and transform them to the desired distance matrix through the Conditional Graph Continuous Flow (CGCF); 2) search the possible 3D coordinates according to the generated distances and 3) further optimize the generated conformation via a MCMC procedure with the Energy-based Tilting Model (ETM).

Ideas for the 1st component Conditional Graph Continuous Flow $p_\theta(\boldsymbol{d}|\mathcal{G})$:

- CGCF **breaks** permutation equivariance. The matrix $\boldsymbol{d}$ is flatten into a long vector. They treat $\boldsymbol{d}$ as a global graph representation.
- But we can address the permutation equivariance by:
  1. Graph normalizing flows.
  2. Second-order message passing $\boldsymbol{S}_n$/Maron, in which we treat $\boldsymbol{d}_{ij}$ as pair-wise features.
  3. The decoder can be EDM method of Frank Noe (that I will present right after).

Generative model (2)

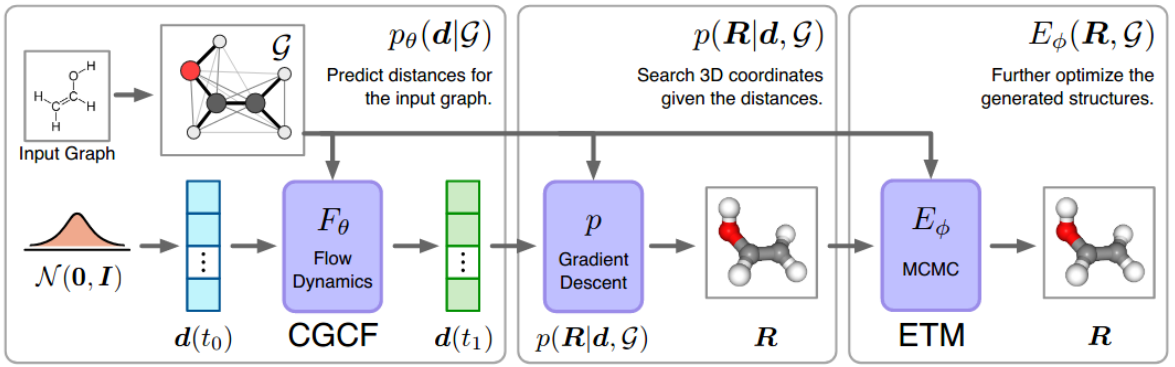


Figure 1: Illustration of the proposed framework. Given the molecular graph, we 1) first draw latent variables from a Gaussian prior, and transform them to the desired distance matrix through the Conditional Graph Continuous Flow (CGCF); 2) search the possible 3D coordinates according to the generated distances and 3) further optimize the generated conformation via a MCMC procedure with the Energy-based Tilting Model (ETM).

The 2nd component **closed-form** $p(\boldsymbol{R}|\boldsymbol{d},\mathcal{G})$:

- The genrated pair-wise distances can be converted into 3D structures through postprocessing methods such as **Euclidean Distance Geometry** (EDG) solved by MCMC or gradient descent:

$$p(\boldsymbol{R}|\boldsymbol{d},\mathcal{G}) = \frac{1}{Z}\exp\left\{-\sum_{e_{uv}\in\mathcal{E}}\alpha_{uv}(||\boldsymbol{r}_u-\boldsymbol{r}_v||_2-d_{uv})^2\right\}$$

- Before in the workshop paper, they used **Reinforcement Learning** (that can be unstable). In Frank Noe's EDM paper, they used the **Open Babel** tool.

Generative model (3)

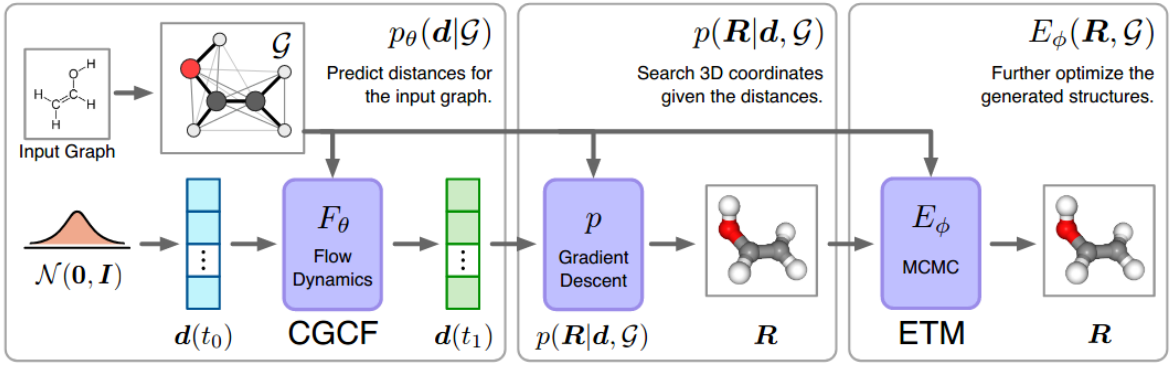


Figure 1: Illustration of the proposed framework. Given the molecular graph, we 1) first draw latent variables from a Gaussian prior, and transform them to the desired distance matrix through the Conditional Graph Continuous Flow (CGCF); 2) search the possible 3D coordinates according to the generated distances and 3) further optimize the generated conformation via a MCMC procedure with the Energy-based Tilting Model (ETM).

The 3rd component Energy-based Tiling Model (ETM) $E_\phi(\boldsymbol{R}, \mathcal{G})$:

- **SchNet** is used to model the long-range interactions between atoms explicitly.
- ETM is learned by **Noise Contrastive Estimation** (inspired by energy-based model):

$$\mathcal{L}_{\text{nce}}(\boldsymbol{R}, \mathcal{G}; \phi) = -\mathbb{E}_{p_{\text{data}}}\Big[\log \frac{1}{1+\exp(E_\phi(\boldsymbol{R}, \mathcal{G}))}\Big] - \mathbb{E}_{p_\theta}\Big[\log \frac{1}{1+\exp(-E_\phi(\boldsymbol{R}, \mathcal{G}))}\Big]. \quad (9)$$

Sampling

Employ a **two-stage** dynamic system to synthesize a possible conformation given the molecular graph representation $\mathcal{G}$:

1. **Stage 1:**
   - Draw a latent variable z from the Gaussian prior $\mathcal{N}(0, 1)$.
   - Pass z through the CNF to obtain a distance matrix $\boldsymbol{d}$.
   - Use gradient descent to maximize (find local maximum) probability of $p(\boldsymbol{R}|\boldsymbol{d}, \mathcal{G})$ with respect to $\boldsymbol{R}$.
   - An initial conformation $\boldsymbol{R}^{(0)}$ can be generated.
2. **Stage 2:** Further refine the initial conformation $\boldsymbol{R}^{(0)}$ with K steps of Langevin dynamics

$$\boldsymbol{R}_k = \boldsymbol{R}_{k-1} - \frac{\epsilon}{2}\nabla_{\boldsymbol{R}} E_{\theta,\phi}\left(\boldsymbol{R}|\mathcal{G}\right) + \sqrt{\epsilon}\omega, \omega \sim \mathcal{N}(0, \mathcal{I}),$$

$$\text{where } E_{\theta,\phi}(\boldsymbol{R}|\mathcal{G}) = -\log p_{\theta,\phi}(\boldsymbol{R}|\mathcal{G}) = E_{\phi}(\boldsymbol{R}, \mathcal{G}) - \log \int p(\boldsymbol{R}|\boldsymbol{d}, \mathcal{G}) p_{\theta}(\boldsymbol{d}|\mathcal{G}) \mathrm{d}\boldsymbol{d}. \tag{10}$$

Evaluation metric

Root-Mean-Square Deviation (RMSD):

$$\mathrm{RMSD}(\boldsymbol{R}, \hat{\boldsymbol{R}}) = \Big(\frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{R}_i - \hat{\boldsymbol{R}}_i\|^2\Big)^{\frac{1}{2}}, \tag{11}$$

Coverage (COV):

$$\mathrm{COV}(\mathbb{S}_g(\mathcal{G}), \mathbb{S}_r(\mathcal{G})) = \frac{1}{|\mathbb{S}_r|}\Big|\Big\{\boldsymbol{R} \in \mathbb{S}_r \Big| \mathrm{RMSD}(\boldsymbol{R}, \boldsymbol{R}') < \delta, \exists \boldsymbol{R}' \in \mathbb{S}_g\Big\}\Big|, \tag{12}$$

Matching (MAT):

$$\mathrm{MAT}(\mathbb{S}_g(\mathcal{G}), \mathbb{S}_r(\mathcal{G})) = \frac{1}{|\mathbb{S}_r|}\sum_{\boldsymbol{R}' \in \mathbb{S}_r} \min_{\boldsymbol{R} \in \mathbb{S}_g} \mathrm{RMSD}(\boldsymbol{R}, \boldsymbol{R}'). \tag{13}$$

where $\mathbb{S}_g(\mathcal{G})$ denotes the set of generated conformations for molecular graph $\mathcal{G}$, and $\mathbb{S}_r(\mathcal{G})$ denotes the reference set.

Experiments (1)

Table 1: Comparison of different methods on the COV and MAT scores. Top 4 rows: deep generative models for molecular conformation generation. Bottom 5 rows: different methods that involve an additional rule-based force field to further optimize the generated structures.

Dataset	GEOM-QM9				GEOM-Drugs			
Metric	COV* (%)		MAT (Å)		COV* (%)		MAT (Å)	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
CVGAE	8.52	5.62	0.7810	0.7811	0.00	0.00	2.5225	2.4680
GraphDG	55.09	56.47	0.4649	0.4298	7.76	0.00	1.9840	2.0108
CGCF	69.60	70.64	0.3915	0.3986	49.92	41.07	1.2698	1.3064
CGCF + ETM	**72.43**	**74.38**	**0.3807**	**0.3955**	**53.29**	**47.06**	**1.2392**	**1.2480**
RDKit	**79.94**	**87.20**	0.3238	**0.3195**	65.43	70.00	1.0962	1.0877
CVGAE + FF	63.10	60.95	0.3939	0.4297	83.08	95.21	0.9829	0.9177
GraphDG + FF	70.67	70.82	0.4168	0.3609	84.68	93.94	0.9129	0.9090
CGCF + FF	**73.52**	**72.75**	0.3131	0.3251	92.28	98.15	**0.7740**	**0.7338**
CGCF + ETM + FF	**73.54**	72.58	**0.3088**	**0.3210**	**92.41**	**98.57**	**0.7737**	0.7616

* For the reported COV score, the threshold δ is set as 0.5Å for QM9 and 1.25Å for Drugs. More results of COV scores with different threshold δ are given in Appendix H.

Experiments (2)

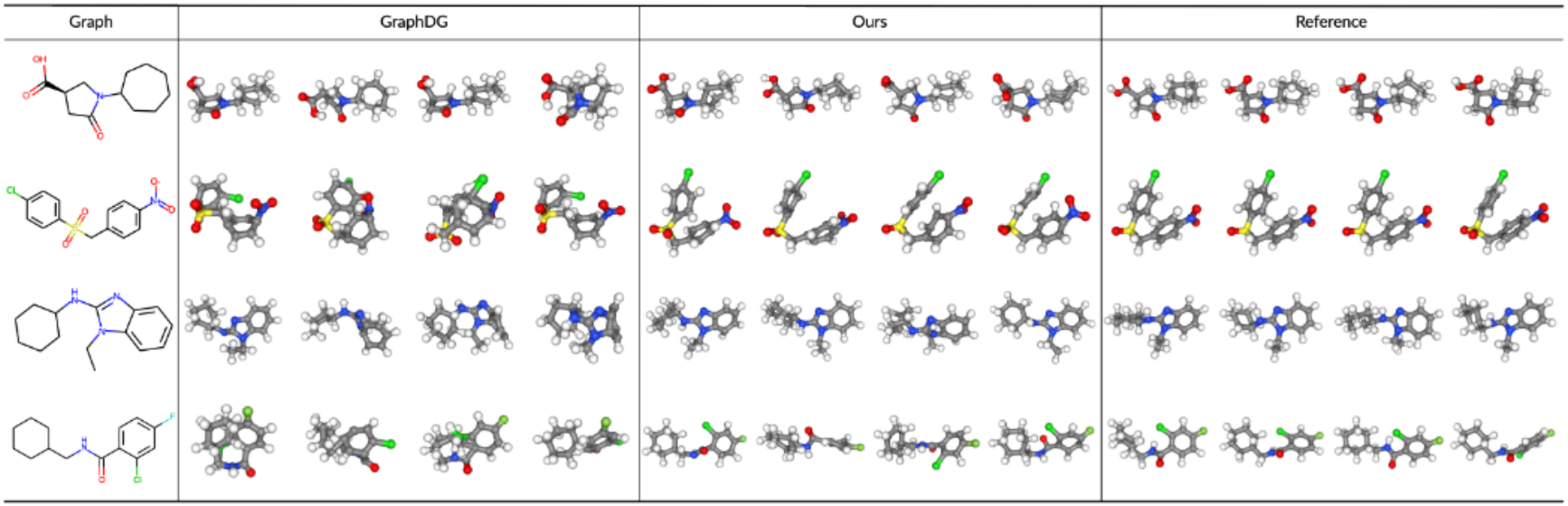


Figure 2: Visualization of generated conformations from the state-of-the-art baseline (GraphDG), our method and the ground-truth, based on four random molecular graphs from the test set of GEOM-Drugs. C, O, H, S and Cl are colored gray, red, white, yellow and green respectively.

Experiments (3)

Table 2: Comparison of distances density modeling with different methods. We compare the marginal distribution of single ($p(d_{uv}|\mathcal{G})$), pair ($p(d_{uv}, d_{ij}|\mathcal{G})$) and all ($p(\boldsymbol{d}|\mathcal{G})$) edges between C and O atoms. Molecular graphs $\mathcal{G}$ are taken from the test set of ISO17. We take two metrics into consideration: 1) **median** MMD between the ground truth and generated ones, and 2) **mean** ranking (1 to 3) based on the MMD metric.

	Single		Pair		All	
	Mean	Median	Mean	Median	Mean	Median
RDKit	3.4513	3.1602	3.8452	3.6287	4.0866	3.7519
CVGAE	4.1789	4.1762	4.9184	5.1856	5.9747	5.9928
GraphDG	0.7645	0.2346	0.8920	0.3287	1.1949	0.5485
CGCF	**0.4490**	**0.1786**	**0.5509**	**0.2734**	**0.8703**	**0.4447**
CGCF + ETM	0.5703	0.2411	0.6901	0.3482	1.0706	0.5411

Table 3: Conformation Diversity. Mean and Std represent the corresponding mean and standard deviation of pairwise RMSD between the generated conformations per molecule.

	RDKit	CVGAE	GraphDG	CGCF	CGCF +ETM
Mean	0.083	0.207	0.249	0.810	0.741
Std	0.054	0.187	0.104	0.223	0.206

Paper 2

Generating valid Euclidean distance matrices

Moritz Hoffmann, Frank Noé

`https://arxiv.org/abs/1910.03131`

Generating Euclidean distance matrices (1)

Goal

To generate Euclidean distance matrices $D \in \mathbb{EDM}^n \subset \mathbb{R}^{n \times n}$ without placing coordinates in Cartesian space. The output is invariant to translation and rotation.

Terminology:

- A matrix $D \in \mathbb{EDM}^n$ by definition if there exists the set of points $\boldsymbol{x}_1, .., \boldsymbol{x}_n \in \mathbb{R}^d$ such that $D_{ij} = ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2$ for all $i, j = 1, .., n$.
- The smallest integer $d > 0$ for which a set of n points in $\mathbb{R}^d$ exists that reproduces the matrix D is called the embedding dimension.

Generating Euclidean distance matrices (2)

Theorem

The connection between EDMs and positive semi-definite matrices:

$$D \in \mathbb{EDM}^n \Leftrightarrow -\frac{1}{2} JDJ \quad \text{positive semi-definite} \qquad (1)$$

where:

$$J = \mathbb{I} - \frac{1}{n} 1 1^T, \qquad 1 = (1, .., 1)^T \in \mathbb{R}^n$$

Generating Euclidean distance matrices (3)

The EDM D has a corresponding Gram matrix $M \in \mathbb{R}^{n \times n}$ by the relationship:

$$M_{ij} = \langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle_2 = \frac{1}{2}(D_{1j} + D_{i1} - D_{ij}) \qquad (2)$$

with $\boldsymbol{y}_k = \boldsymbol{x}_k - \boldsymbol{x}_1, (k = 1, .., n)$:

$$D_{ij} = M_{ii} + M_{jj} - 2M_{ij} \qquad (3)$$

Matrix M has a specfic structure:

$$M = \begin{bmatrix} 0 & 0^T \\ 0 & L \end{bmatrix} \qquad (4)$$

with $L \in \mathbb{R}^{(n-1)\times(n-1)}$ is symmetric and positive semi-definite.

Generating Euclidean distance matrices (4)

Eigenvalue decomposition of M:

$$M = USU^T = (U\sqrt{S})(U\sqrt{S})^T = YY^T$$

with:

$$S = \text{diag}(\lambda_1, .., \lambda_n), \qquad \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$$

Let d be the number of non-zero eigenvalues of M, then d is the embedding dimension of D, and the first d rows of Y reveals the coordinates $\{\boldsymbol{y}_k\}_{k=1}^n$.

Generating Euclidean distance matrices (5)

Algorithm:

- Suppose we have a **parameterized** arbitrary matrix $\tilde{L} \in \mathbb{R}^{(n-1)\times(n-1)}$.
- It can be transformed into a symmetric positive semi-definite matrix by any non-negative function $g(.)$:

$$L = g(\tilde{L}) = g\left(U \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{n-1} \end{pmatrix} U^\top\right) = U \begin{pmatrix} g(\lambda_1) & & \\ & \ddots & \\ & & g(\lambda_{n-1}) \end{pmatrix} U^\top \tag{5}$$

- Construct M as:

$$M = \begin{bmatrix} 0 & 0^T \\ 0 & L \end{bmatrix} \tag{4}$$

- Construct D as:

$$D_{ij} = M_{ii} + M_{jj} - 2M_{ij} \tag{3}$$

Generating Euclidean distance matrices (6)

Algorithm 1 Algorithm to train a generative neural network to (in general non-uniformly) sample Euclidean distance matrices based on the neural network G, where N_z is the dimension of the input vector, m the batch size, and n the number of points to place relative to one another.

Sample $\mathbf{z} \sim \mathcal{N}(0,1)^{m \times N_z}$, i.e., sample from a simple prior distribution,
Transform $X = G(\mathbf{z}) \in \mathbb{R}^{m \times (n-1) \times (n-1)}$ via a neural network G,
for $i = 1$ to m **do**
    Symmetrize $\tilde{L} \leftarrow \frac{1}{2}\left(X_i + X_i^\top\right)$
    Make positive semi-definite $L \leftarrow \mathrm{sp}(\tilde{L})$ with (5)
    Assemble $M = M(L)$ with (4)
    Assemble $D = D(M)$ with (3)
    Compute eigenvalues $\mu_1, \ldots, \mu_n$ of $-\frac{1}{2}JDJ$, see (1)
    $L_{\mathrm{edm}}^{(i)} \leftarrow \sum_{k=1}^{n} \mathrm{ReLU}(-\mu_k)^2$
    Compute eigenvalues $\lambda_1, \ldots, \lambda_n$ of M such that $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_n$
    $L_{\mathrm{rank}}^{(i)} \leftarrow \sum_{k=d+1}^{n} \lambda_k^2$
end for
$L \leftarrow \eta_1 \frac{1}{m} \sum_{i=1}^{m} L_{\mathrm{edm}}^{(i)} + \eta_2 \frac{1}{m} \sum_{i=1}^{m} L_{\mathrm{rank}}^{(i)}$
Optimize weights of G with respect to ∇L.

Wasserstein GAN:

$$\min_{G} \max_{C \in \mathcal{D}} \mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}_r}\left[C(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}_g}\left[C(\boldsymbol{x})\right], \tag{6}$$

where G is the generator as in Algorithm 1, and C is the critic (discriminator) network as SchNet.

Application (1)

Apply to a subset of QM9 dataset consisting of $6,095$ isomers with the chemical formula $C_7O_2H_{10}$:

1. Generate the Euclidean distance matrices along with the atom types.
2. From the EDM matrix, we infer bonds and bond order by lightly computational **Open Babel**.

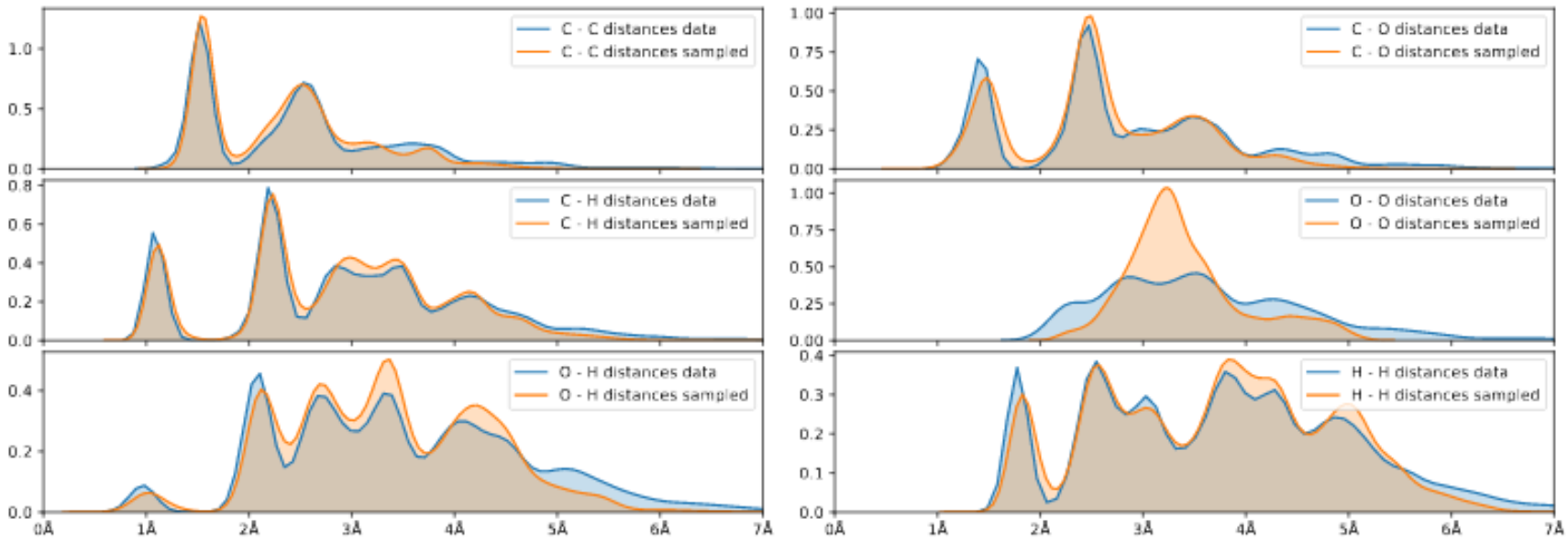


Figure 1: Distribution of pairwise distances between different kinds of atom type after training a Euclidean distance matrix WGAN-GP (Sec. 3) on the $C_7O_2H_{10}$ isomer subset of QM9.

Application (2)

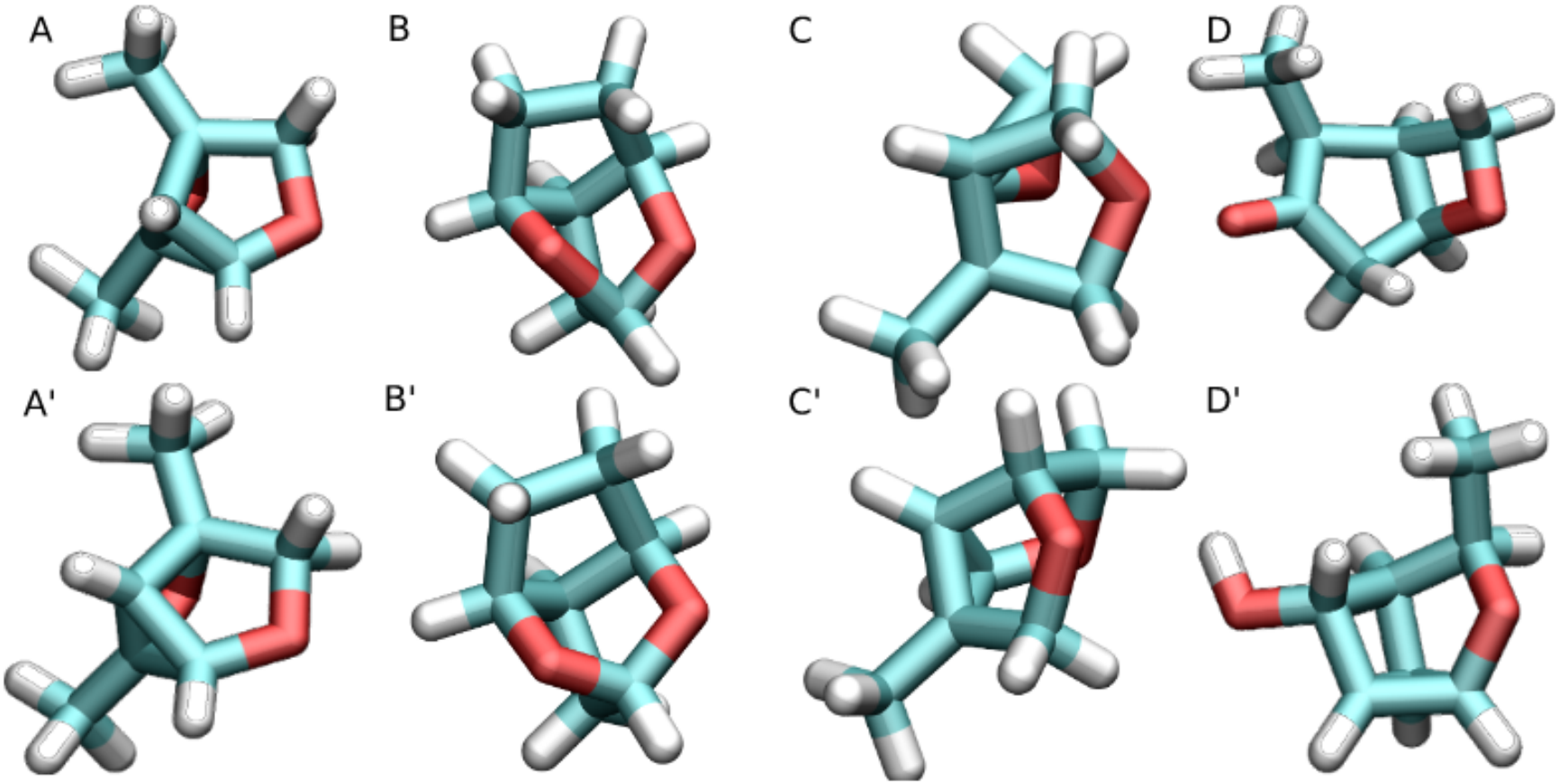


Figure 5: Sampled structures with the Euclidean distance matrix WGAN. Top row A to D are generated samples, bottom row A' to D' are closest matches from the QM9 database. Generated molecules A and B could be matched with A' and B' up to a maximum atom distance of 0.6 Å. Generated molecules C and D are new molecular structures with their closest matches C' and D', respectively.

Discussion questions

1. What are other tasks/datasets? In Chemistry (I don't know)? Point cloud generation?
2. Advantages/disadvantages against graph-based generation?
3. I think we can combine this with $\mathbb{S}_n$ and VAE:
   - The encoder would be graph-based message passing.
   - The decoder would be the Algorithm 1 (as the generator of GAN).
   - We can avoid the mode-collapse phenomenon.
4. Way to improve:

To this end, we apply the Hungarian algorithm [48] onto a cost matrix $C \in \mathbb{R}^{n\times n}$ for EDMs D_1, D_2 and type vectors $\mathbf{t}_1, \mathbf{t}_2 \in \mathbb{R}^n$ with

$$C_{i,j} = \begin{cases} \left|\frac{1}{n}\sum_{k=1}^{n}(D_1)_{i,k} - \frac{1}{n}\sum_{k=1}^{n}(D_2)_{j,k}\right| & \text{, if } (\mathbf{t}_1)_i = (\mathbf{t}_2)_j, \\ \infty & \text{, otherwise.} \end{cases} \tag{14}$$