Group Meeting - January 15, 2021

Paper review & Research progress

Truong Son Hy [*]

[*]Department of Computer Science
The University of Chicago

Ryerson Physical Lab

Content

1. **Generating valid Euclidean distance matrices**, `https://arxiv.org/abs/1910.03131`
2. Research update

Paper 1

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:

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

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:

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

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 $g(.)$:

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

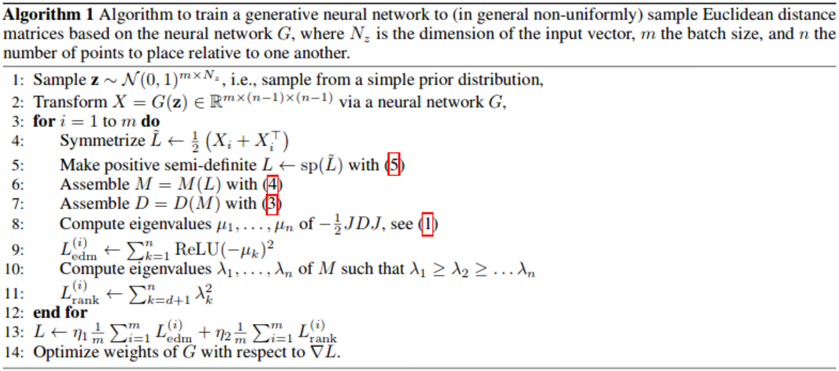
- Construct M as:

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

- Construct D as:

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

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.

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

Wasserstein GAN:

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

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**.

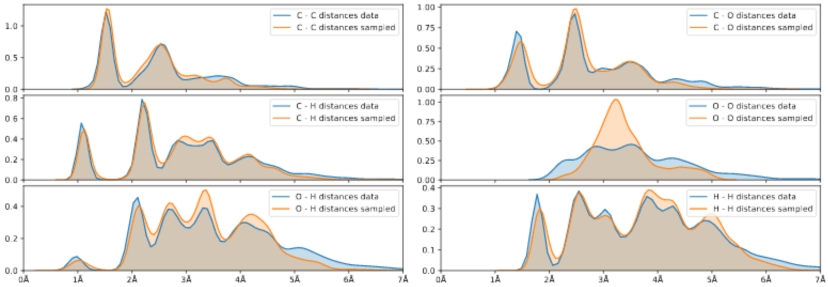


Figure 1: Distribution of pairwise distances between different kinds of atom type after training a Euclidean distance matrix WGAN-GP (Sec. 3) on the $C_7O_2H_{10}$ isomer subset of QM9.

Application (2)

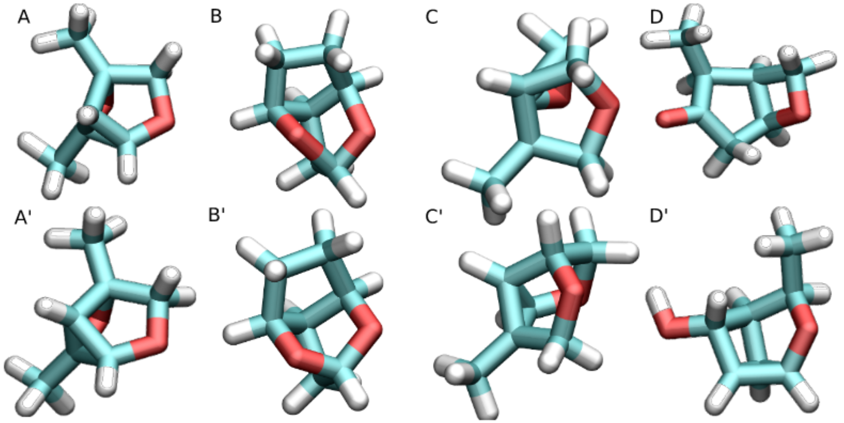
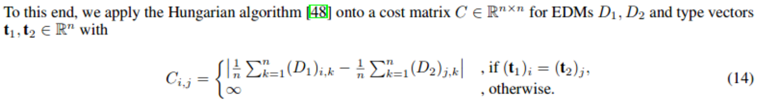


Figure 5: Sampled structures with the Euclidean distance matrix WGAN. Top row A to D are generated samples, bottom row A' to D' are closest matches from the QM9 database. Generated molecules A and B could be matched with A' and B' up to a maximum atom distance of 0.6 Å. Generated molecules C and D are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

Discussion questions

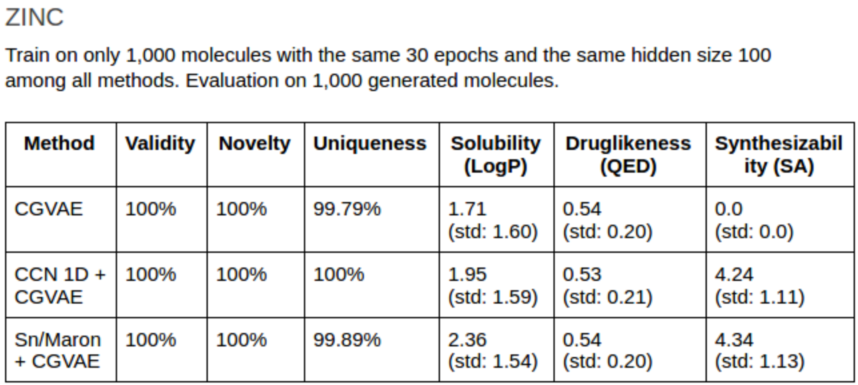
1. What are other tasks/datasets? In Chemistry (I don't know)? Point cloud generation?
2. Advantages/disadvantages against graph-based generation?
3. I think we can combine this with $\mathbb{S}_n$ and VAE:
    - The encoder would be graph-based message passing.
    - The decoder would be the Algorithm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

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

Research update – Graph-based molecular generation (1)

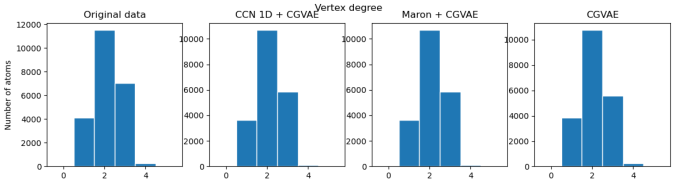


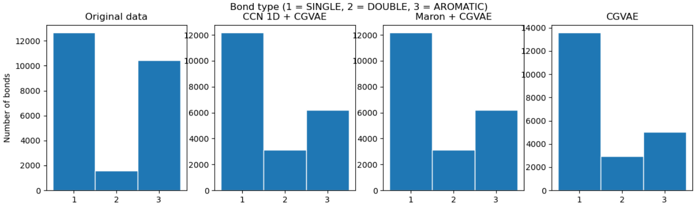
ZINC

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.

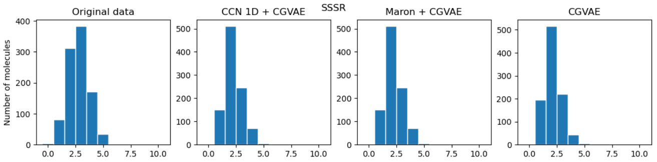
Method	Validity	Novelty	Uniqueness	Solubility (LogP)	Druglikeness (QED)	Synthesizability (SA)
CGVAE	100%	100%	99.79%	1.71 (std: 1.60)	0.54 (std: 0.20)	0.0 (std: 0.0)
CCN 1D + CGVAE	100%	100%	100%	1.95 (std: 1.59)	0.53 (std: 0.21)	4.24 (std: 1.11)
Sn/Maron + CGVAE	100%	100%	99.89%	2.36 (std: 1.54)	0.54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

Research update – Graph-based molecular generation (2)

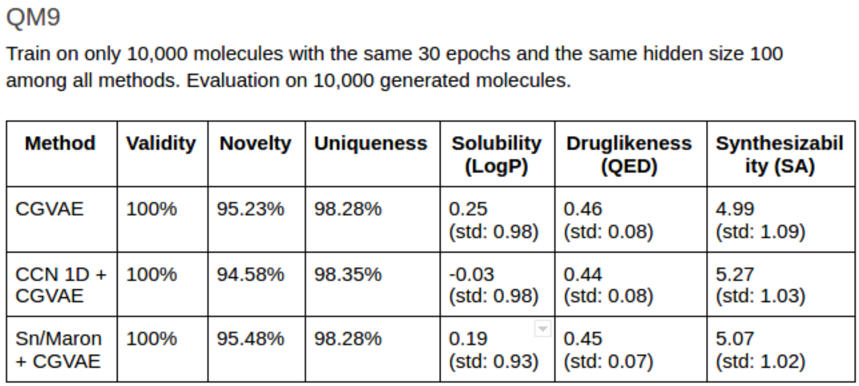




SSSR = smallest set of smallest rings



Research update – Graph-based molecular generation (3)

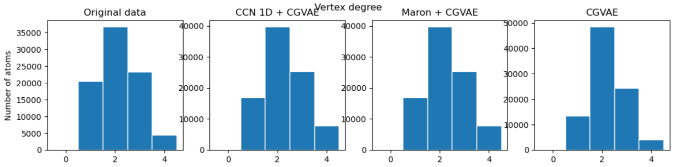


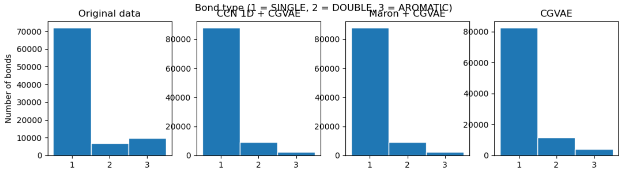
QM9

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.

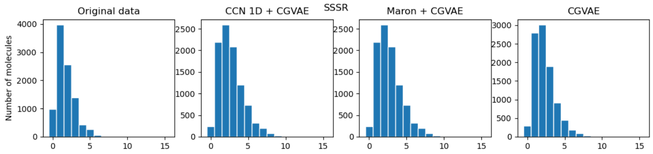
Method	Validity	Novelty	Uniqueness	Solubility (LogP)	Druglikeness (QED)	Synthesizability (SA)
CGVAE	100%	95.23%	98.28%	0.25 (std: 0.98)	0.46 (std: 0.08)	4.99 (std: 1.09)
CCN 1D + CGVAE	100%	94.58%	98.35%	-0.03 (std: 0.98)	0.44 (std: 0.08)	5.27 (std: 1.03)
Sn/Maron + CGVAE	100%	95.48%	98.28%	0.19 (std: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

Research update – Graph-based molecular generation (4)





SSSR = smallest set of smallest rings


SSSR
Original data
CCN 1D + CGVAE
Maron + CGVAE
CGVAE
Number of molecules