Group Meeting - December 11, 2020

Paper review & Research progress

Truong Son Hy *

*Department of Computer Science
The University of Chicago

Ryerson Physical Lab

Socrates

1. He is richest who is content with the least, for content is the wealth of nature.
2. Contentment is natural wealth, luxury is artificial poverty.
3. Prefer knowledge over wealth, for the one is transitory, the other perpetual.

Papers

Machine Learning for Molecules Workshop @ NeurIPS 2020
https://ml4molecules.github.io/papers2020/accepted.html

1. **Learning Latent Space Energy-Based Prior Model for Molecule Generation** (paper 41)
2. **Conditional generation of molecules from disentangled representations** (paper 52)
3. **A Probabilistic Model for Molecular Geometry Generation and Optimization** (paper 33)
4. **Flow-Based Models for Active Molecular Graph Generation** (paper 25)
5. **Masked Graph Modeling for Molecule Generation** (paper 13)

Paper 33

A Probabilistic Model for Molecular Geometry Generation and Optimization (paper 33)

https://ml4molecules.github.io/papers2020/ML4Molecules_2020_paper_33.pdf

Proposals (1)

Proposals

- Probabilistic model for molecular geometry generation and optimization.
- To address the issue of roto-translational invariance, instead of generating 3D coordinates:
  - employ VAEs to model the molecule's **inter-atomic distances**.
  - then generate the conformation using a distance geometry algorithm (heuristics).

Note

Instead of modeling the distribution wrt to 3D coordinates given a molecular graph $\mathcal{G}$, i.e. $p(\boldsymbol{R}|\mathcal{G})$ (that requires roto-translational invariance), they model $p(\boldsymbol{d}|\mathcal{G})$ and then define $p(\boldsymbol{R}|\mathcal{G})$ based on $p(\boldsymbol{d}|\mathcal{G})$.

Proposals (2)

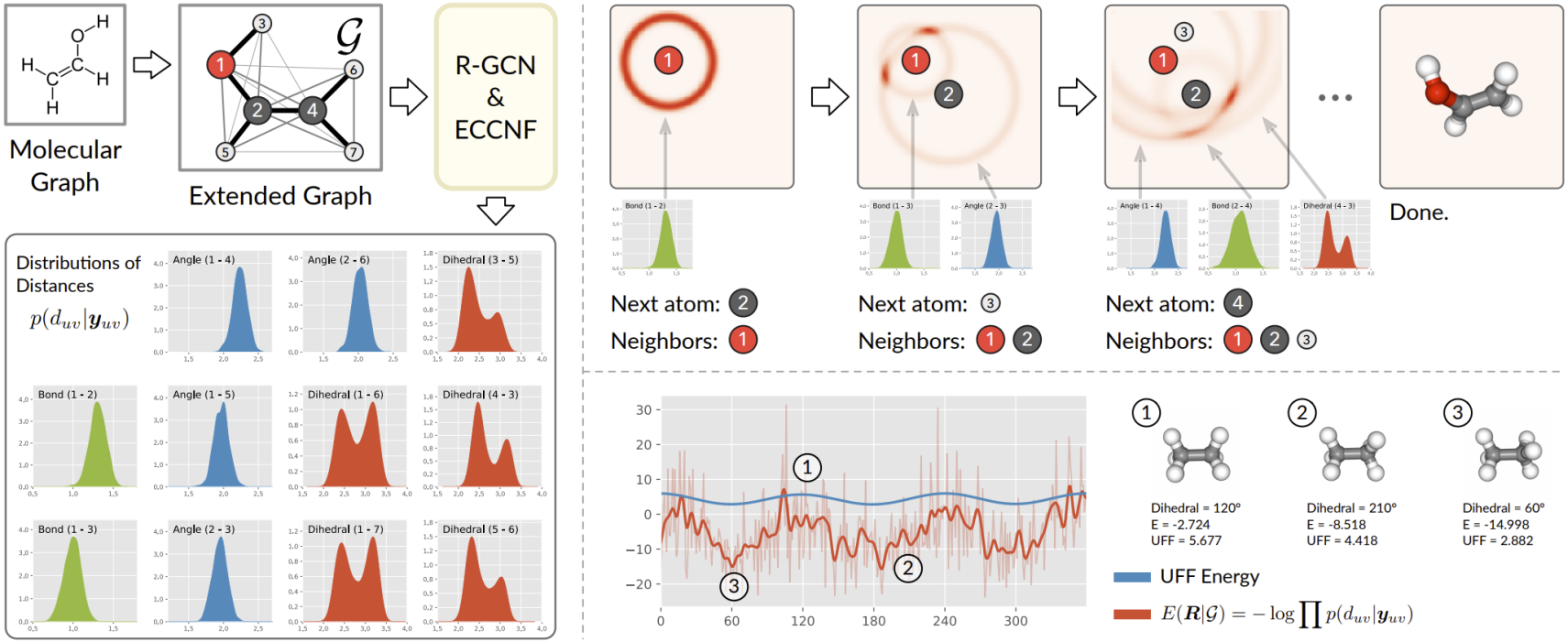


Figure 1: **(a) Left**: An illustration of our model. The model takes extended molecular graphs as input and leads to a set of distributions *w.r.t.* to distances. **(b) Top right**: An illustration of the auto-regressive conformation generation process. Atoms are placed one-by-one according to the probability distribution $p(\boldsymbol{R}_i|\boldsymbol{R}_{1:i-1}, \mathcal{G})$ which is based on $p(d_{uv}|\boldsymbol{y}_{uv})$. **(c) Bottom right**: Energy function defined in Eq. 7. Our energy function is positively correlated to the real molecular potential energy function, where equilibrium conformations are at lower energy levels.

Models (1)

Structural encoding with MPNNs

The distribution of distances $p(\boldsymbol{d}|\mathcal{G})$ is conditioned on the structural context of each edge in the graph $\mathcal{G}$:

- Nodewise features:

$$\boldsymbol{H} = \text{RGCN}(\mathcal{G}) \in \mathbb{R}^{|\mathcal{V}| \times F}$$

- Edge features:

$$\boldsymbol{y}_{uv} = \text{MLP}([\boldsymbol{H}_u \odot \boldsymbol{H}_v || \boldsymbol{\beta}_{uv}])$$

where $\boldsymbol{y}_{uv}$ is the feature of edge (u, v), $\boldsymbol{H}_u$ and $\boldsymbol{H}_v$ are node features of node u and v respectively, and $\boldsymbol{\beta}_{uv}$ is the embedding of edge (u, v) which contains information such as bond types.

Models (2)

Edge-conditioned continuous normalizing flow (EC-CNF)

$$d_{uv} = F(z(t_0), \boldsymbol{y}_{uv}) = z(t_0) + \int_{t_0}^{t_1} f(z(t), t, \boldsymbol{y}_{uv}) dt$$

$$\log p(d_{uv}|\boldsymbol{y}_{uv}) = \log p(F^{-1}(d_{uv}, \boldsymbol{y}_{uv})) - \int_{t_0}^{t_1} \text{trace}\left(\frac{\partial f}{\partial z(t)}\right) dt$$

where $z(t_0) \sim \mathcal{N}(0, 1)$ is the source distribution, f is the dynamic that transforms $z(t_0)$ to the target distribution.

Models (2)

Edge-conditioned continuous normalizing flow (EC-CNF)

$$d_{uv} = F(z(t_0), \boldsymbol{y}_{uv}) = z(t_0) + \int_{t_0}^{t_1} f(z(t), t, \boldsymbol{y}_{uv}) dt$$

$$\log p(d_{uv}|\boldsymbol{y}_{uv}) = \log p(F^{-1}(d_{uv}, \boldsymbol{y}_{uv})) - \int_{t_0}^{t_1} \text{trace}\left(\frac{\partial f}{\partial z(t)}\right) dt$$

where $z(t_0) \sim \mathcal{N}(0, 1)$ is the source distribution, f is the dynamic that transforms $z(t_0)$ to the target distribution.

Training Objective

MLE:

$$\mathcal{L} = \mathbb{E}_{(\mathcal{G}, \boldsymbol{R}) \sim p_{\text{data}}} \left[\sum_{(u,v) \in \mathcal{E}} \log p(||\boldsymbol{R}_u - \boldsymbol{R}_v||_2 | \boldsymbol{y}_{uv}(\mathcal{G})) \right]$$

Next step: geometry optimization

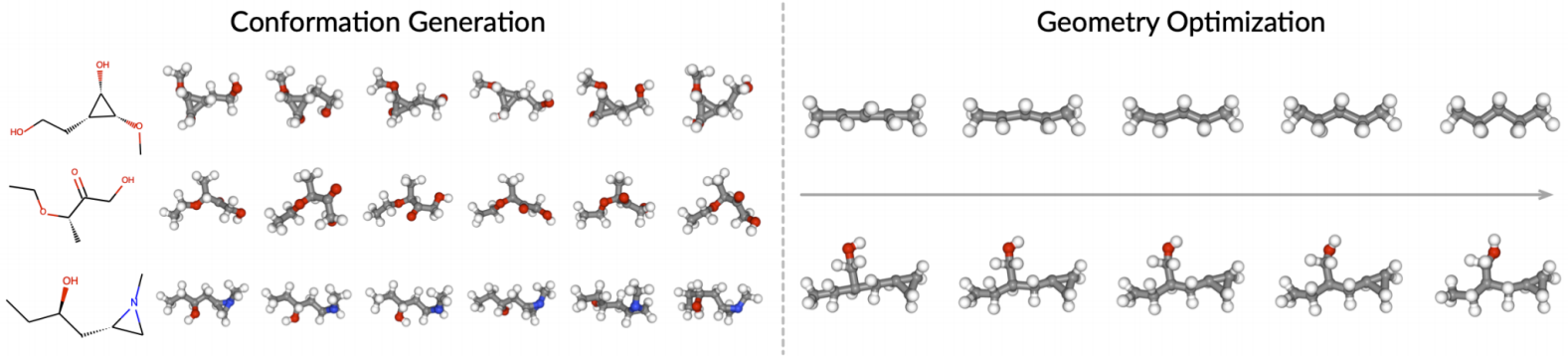


Figure 2: **(a) Left**: Generated conformations from our model. **(b) Right**: Visualization of the geometry optimization process.

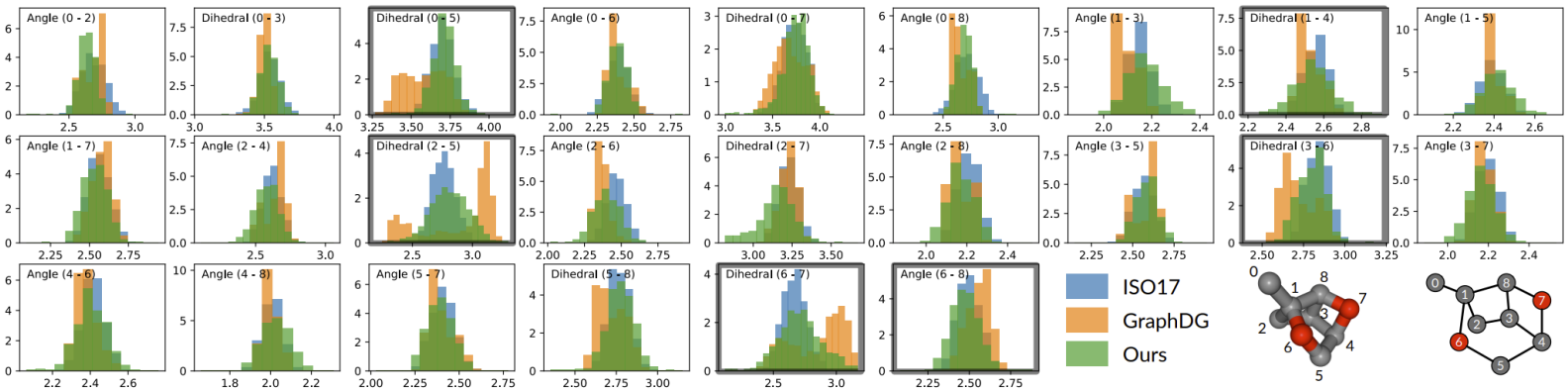


Figure 3: Marginal distributions $p(d_{uv}|\mathcal{G})$ of ground-truth (ISO17) and generated conformations from GraphDG and our model. We omit the distances of bonded atom-pairs which are trivial to model, and concentrate on angular and dihedral atom-pairs. Our model fits the target distribution better than GraphDG does, especially on the atom-pairs highlighted by black frames. (Note that following [4], the distances are computed from the generated 3D structures, not directly sampled from EC-CNF.)

Geometry optimization

Conformation generation

Generating 3D coordinates is thus done by sampling progressively from the following auto-regressive distribution:

$$p(\boldsymbol{R}_i|\boldsymbol{R}_{1:i-1},\mathcal{G}) = \prod_{j<i,(u_i,u_j)\in\mathcal{E}} p(||\boldsymbol{R}_i - \boldsymbol{R}_j||_2|\boldsymbol{y}_{u_i u_j}), \qquad \boldsymbol{R}_1 = 0$$

$$p(\boldsymbol{R}|\mathcal{G}) = \prod_{i>1} p(\boldsymbol{R}_i|\boldsymbol{R}_{1:i-1},\mathcal{G})$$

Geometry optimization

Conformation generation

Generating 3D coordinates is thus done by sampling progressively from the following auto-regressive distribution:

$$p(\boldsymbol{R}_i|\boldsymbol{R}_{1:i-1},\mathcal{G}) = \prod_{j<i,(u_i,u_j)\in\mathcal{E}} p(||\boldsymbol{R}_i - \boldsymbol{R}_j||_2|\boldsymbol{y}_{u_i u_j}), \qquad \boldsymbol{R}_1 = 0$$

$$p(\boldsymbol{R}|\mathcal{G}) = \prod_{i>1} p(\boldsymbol{R}_i|\boldsymbol{R}_{1:i-1},\mathcal{G})$$

Geometry optimization

Minimize the energy function defined based on the distributions wrt distances:

$$E(\boldsymbol{R}|\mathcal{G}) = -\log \prod_{(u,v)\in\mathcal{E}} p(||\boldsymbol{R}_u - \boldsymbol{R}_v||_2|\boldsymbol{y}_{uv})$$

Experiment

Table 1: Assessment of the accuracy of the distributions over distances generated compared to the ground-truth. We evaluate the distance 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 heavy atoms. Molecular graphs $\mathcal{G}$ are taken from the test set of ISO17. **Median** and **mean** MMDs between the ground truth and generated distributions are reported.

	Single		Pair		All	
	Mean	Median	Mean	Median	Mean	Median
RDKit [20]	3.4513	3.1602	3.8452	3.6287	4.0866	3.7519
CVGAE [3]	4.1789	4.1762	4.9184	5.1856	5.9747	5.9928
GraphDG [4]	0.7645	0.2346	0.8920	0.3287	1.1949	0.5485
Ours	**0.5042**	**0.1627**	**0.6086**	**0.2150**	**1.0239**	**0.5172**

Note

Nothing useful! Overfiting the distance distribution is meaningless. We care more about new molecules. The experiment must be:

1. Generate the Euclidean distance matrix (like Frank Noe).
2. Generate new 3D structure $\rightarrow$ molecular graph.

Paper 41

Learning Latent Space Energy-Based Prior Model for Molecule Generation (paper 41)

Bo Pang, Tian Han, Ying Nian Wu

https://ml4molecules.github.io/papers2020/ML4Molecules_2020_paper_41.pdf

Energy-based prior model (1)

Proposals

1. Language-model (LM) based, similar to the one in NLP. In constrast, our approach is considered as graph-based model.
2. Energy-based prior.

Let $x \in \mathbb{R}^D$ be the input SMILES string and $z \in \mathbb{R}^d$ be the latent variable. The generative model is described as:

$$z \sim p_\alpha(z), \qquad x \sim p_\beta(x|z)$$

- In normal VAE: $p_\alpha(z) = \mathcal{N}(0, 1)$ – isotropic Gaussian.
- In this work:

$$p_\alpha(z) = \frac{1}{Z(\alpha)} \exp(f_\alpha(z)) p_0(z)$$

where $p_0(z)$ is a reference distribution, assumed to be isotropic Gaussian; $f_\alpha(z)$ is the negative energy modeled by an MLP.

Energy-based prior model (2)

$$p_\alpha(z) = \frac{1}{Z(\alpha)} \exp(f_\alpha(z)) p_0(z)$$

Basic idea:

- The latent z is sampled from an isotropic Gaussian.
- And then transform further under an MLP, and normalize by:

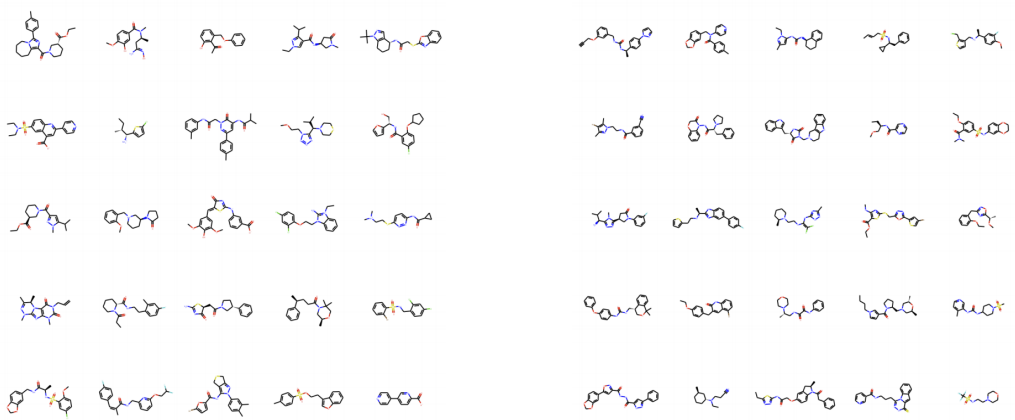
$$Z(\alpha) = \int \exp(f_\alpha(z)) p_0(z) dz = \mathbb{E}_{p_0}[\exp(f_\alpha(z))]$$

- Sampling done by MCMC.

The decoder is an autoregressive model:

$$p_\beta(x|z) = \prod_{t=1}^{T} p_\beta(x^{(t)}|x^{(1)}, .., x^{(t-1)}, z)$$

Energy-based prior model (3)



(a) ZINC (b) Generated

Figure 1: Sample molecules taken from the ZINC dataset (a) and generated by our model (b).

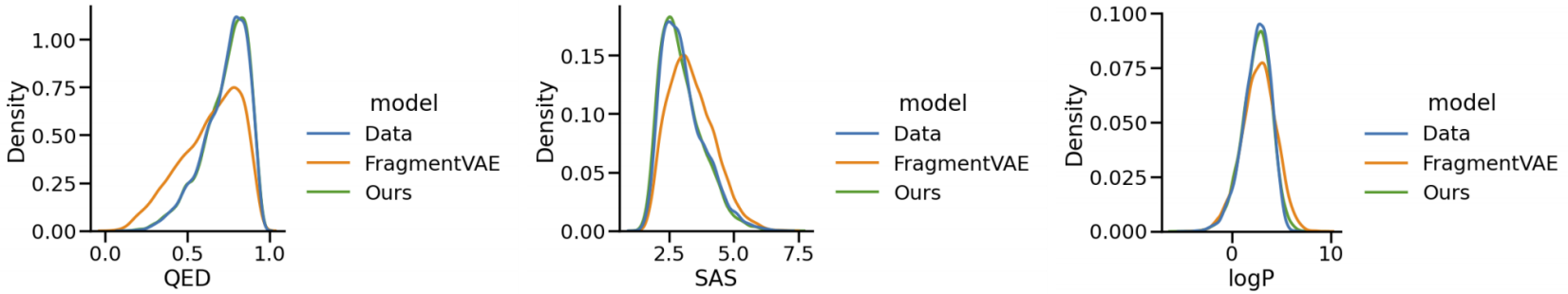


Figure 2: Distributions of molecular properties of data and 10,000 random samples from FragmentVAE and our model.

QED = drug likeness, SAS = synthetic accessiblity score, logP = octanol/water partition coefficient (solubility)

Energy-based prior model (4)

Model	Model Family	Validity w/ check	Validity w/o check	Novelty	Uniqueness
GraphVAE (Simonovsky et al., 2018)	Graph	0.140	-	1.000	0.316
CGVAE (Liu et al., 2018)	Graph	1.000	-	1.000	0.998
GCPN (You et al., 2018)	Graph	1.000	0.200	1.000	1.000
NeVAE (Samanta et al., 2019)	Graph	1.000	-	0.999	1.000
MRNN (Popova et al., 2019)	Graph	1.000	0.650	1.000	0.999
GraphNVP (Madhawa et al., 2019)	Graph	0.426	-	1.000	0.948
GraphAF (Shi et al., 2020)	Graph	1.000	0.680	1.000	0.991
ChemVAE (Gomez-Bombarelli et al., 2018)	LM	0.170	-	0.980	0.310
GrammarVAE (Kusner et al., 2017)	LM	0.310	-	1.000	0.108
SDVAE (Dai et al., 2018)	LM	0.435	-	-	-
FragmentVAE (Podda et al., 2020)	LM	**1.000**	-	0.995	0.998
Ours	LM	0.955	-	**1.000**	**1.000**

Table 1: Performance obtained by our model against LM-based and graph-based baselines.

Paper 25

Flow-Based Models for Active Molecular Graph Generation (paper 25)

Nathan C. Frey, Bharath Ramsundar

`https://ml4molecules.github.io/papers2020/ML4Molecules_2020_paper_25.pdf`

Flow-based model

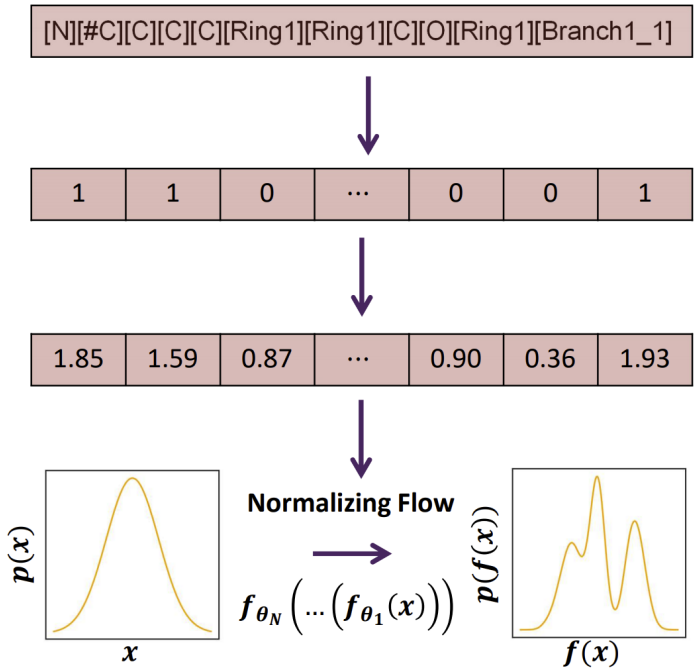
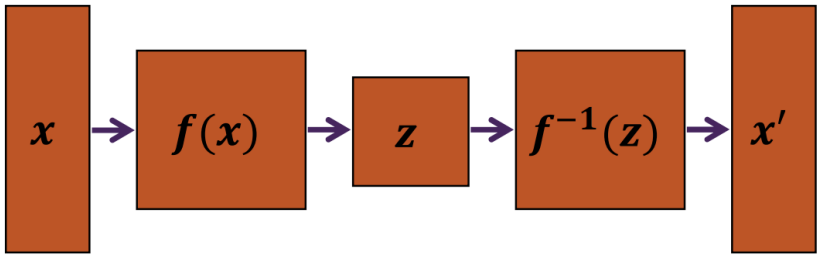


Figure 1: Dequantized one-hot encodings of SELFIES representations are inputs to the normalizing flow. The normalizing flow maps a simple base distribution to a complex target distribution.



Question: SELFIES instead of SMILES? SELFIES seems to encode more structural information, and similar to Morgan fingerprint.

Metric

Drug likeness – QED

$$\text{QED} = \exp\left(\frac{1}{n}\sum_{i=1}^{n} \ln d_i\right)$$

where d_i are desirability functions corresponding to molecular descriptors.

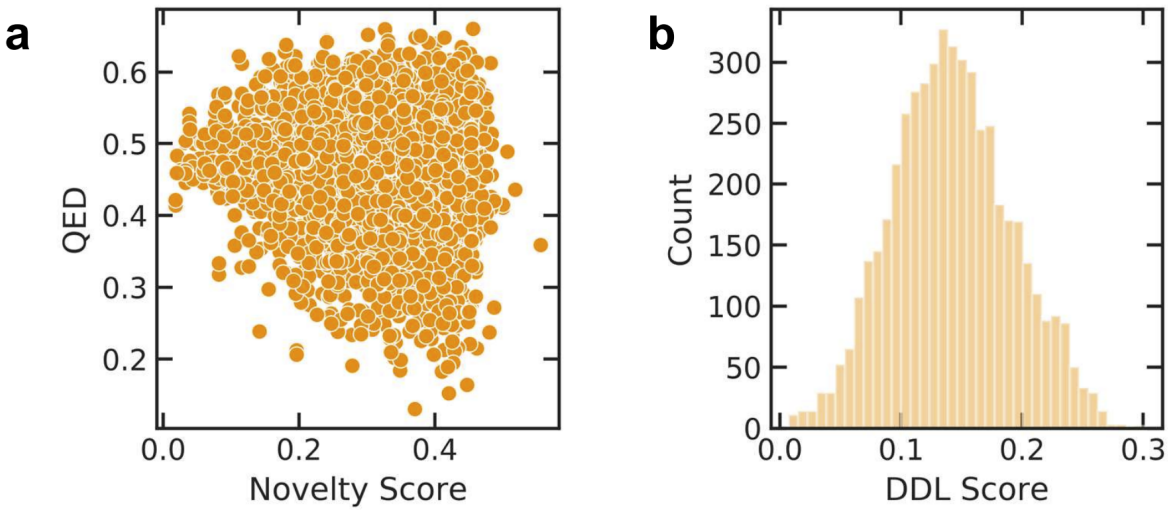
Joint metric – DDL

$$\text{DDL} = (1 - \max(\{T_C\})) \times \text{QED}$$

jointly maximizes drug likeness QED along with **novelty score** $1 - \max(\{T_C\})$ where $\{T_C\}$ is the set of all Tanimoto coefficients comparing that molecule to the training data. It seems to encourage dissimilarity.

Experiment

Question: But the experiment is on QM9 that is **not** a drug dataset.



Architecture	% Valid, Unique, and Novel Molecules	Reference
GraphNVP	47.97	Madhawa et al. [2019]
GRF	32.68	Honda et al. [2019]
GraphAF	83.95	Shi et al. [2020]
MoFlow	97.24 ± 0.21	Zang and Wang [2020]
This work	77.81 ± 0.06	

Note: This work seems to be similar to GraphAF (graph auto-regressive flow) but taking input as SELFIES instead of adjacency.

Paper 52

Conditional generation of molecules from disentangled representations (paper 52)

Amina Mollaysa, Brooks Paige, Alexandros Kalousis

`https://ml4molecules.github.io/papers2020/ML4Molecules_2020_paper_52.pdf`

Overview

Conditional generation:

$$p_\theta(\boldsymbol{x}|\boldsymbol{y}) = \int p_\theta(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}$$

$$\mathcal{L}_{\text{ELBO}}(\theta, \phi) = \sum_{i=1}^{N} \left\{ \mathbb{E}_{q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i)}[\log p_\theta(\boldsymbol{x}_i|\boldsymbol{y}_i, \boldsymbol{z}_i)] - \mathcal{D}_{\text{KL}}(q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i)||p(\boldsymbol{z}_i)) \right\}$$

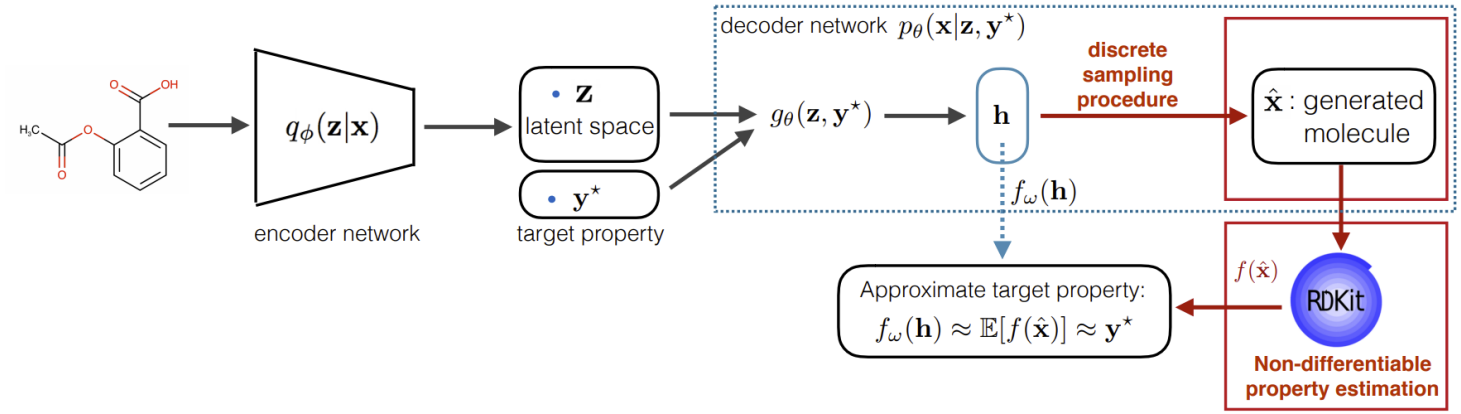


Figure 1: Setting and modeling pipeline for conditional generation of molecules, with supervision provided via an external property prediction oracle. Red lines correspond to non-differentiable components, the blue dashed line corresponds to the approximate property predictor.

Conditional generation (1)

Constrained ELBO

Assume we have access to some oracle function f (non-differentiable) which for any given $\boldsymbol{x}$ outputs a property estimate $\boldsymbol{y}$, i.e. RDKit. We want to:

$$\max_{\theta,\phi} \mathcal{L}_{\text{ELBO}}(\theta,\phi) \qquad \text{s.t.} \qquad \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{y}_i)}[\mathbb{I}[f(\boldsymbol{x}) = \boldsymbol{y}_i]] = 1$$

Relaxed constraint with soft penalty:

$$\mathcal{L}(\theta,\phi) = \mathcal{L}_{\text{ELBO}}(\theta,\phi) - \frac{\lambda}{2}\sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{x}\sim p_\theta(\boldsymbol{x}|\boldsymbol{y}_i)} ||f(\boldsymbol{x}) - \boldsymbol{y}_i||^2$$

Conditional generation (2)

Because f is non-differentiable, we need to approximate the property predictor:

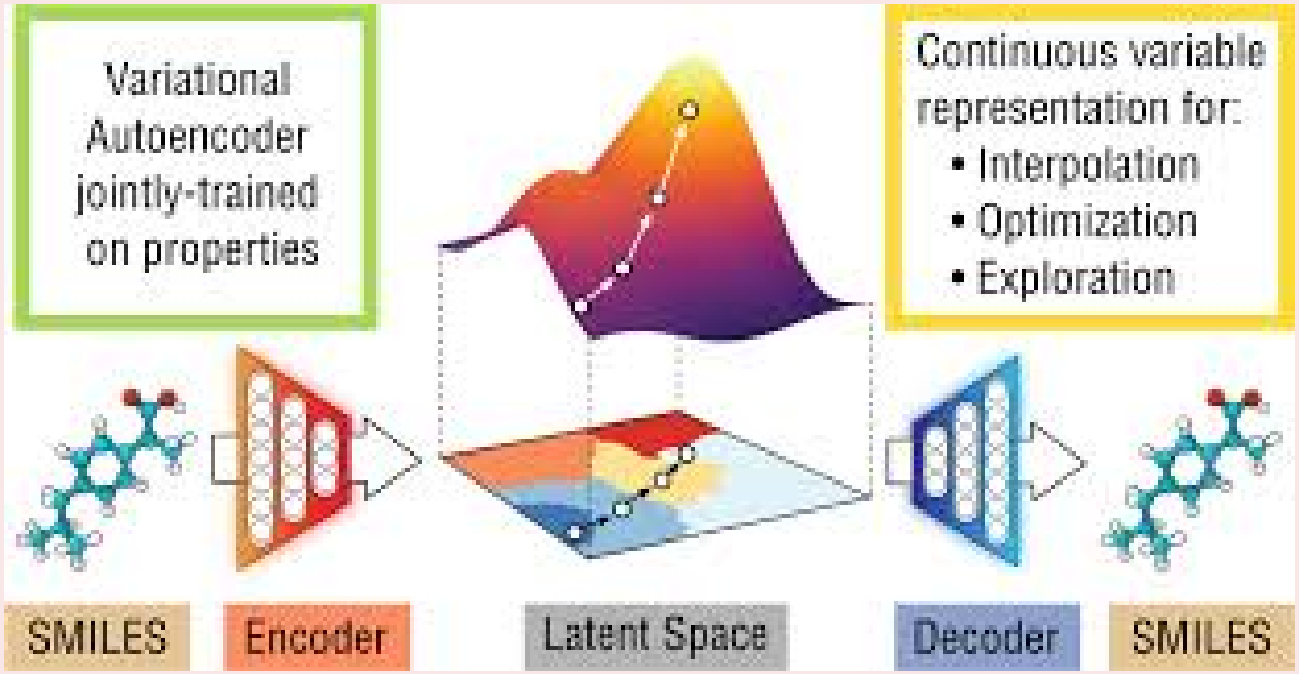
$$f_w(g_\theta(\boldsymbol{z}, \boldsymbol{y}_0)) \approx \mathbb{E}_{p_\theta(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{y}_0)}[f(\boldsymbol{x})]$$

where f_w is a neural network taking input as the last hidden layer of the decoder $g_\theta(\boldsymbol{z}, \boldsymbol{y})$. Then we jointly train the property estimator and the generative model.

Conditional generation (3)

Previous work done

- **Constrained Graph Variational Autoencoders for Molecule Design** (NeurIPS 2018)
- **Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules** (ACS Cent. Sci. 2018)



Experiment & Result

Model	QM9 Reconstruction %	QM9 Valid%	QM9 Unique %	QM9 Novel %	ZINC Reconstruction %	ZINC Valid%	ZINC Unique %	ZINC Novel %
CVAE [4]	3.61	10.3	-	90.0	44.6	0.70	-	100
GVAE [12]	96.00	60.20	-	80.90	53.70	7.20	-	100
SD-VAE [3]	97.84	**98.40**	99.28	91.97	76.20	**43.50**	-	-
Sup-VAE-1-GRU	97.53	93.66	91.30	92.05	74.12	32.84	94.61	100
CGD-VAE-1-GRU	99.27	95.61	93.65	87.87	**88.64**	29.00	**99.24**	100
Sup-VAE-3-GRU	97.81	97.90	95.09	89.47	82.40	36.16	86.26	100
CGD-VAE-3-GRU	**99.31**	97,80	**98.77**	**96.21**	81.80	37.78	98.75	100

Table 1: Reconstruction performance and generation quality (Valid, Unique, Novel).

	Model	$\mathbf{z} \sim \hat{q}_\sigma(\mathbf{z})$	$\mathbf{z} \sim q(\mathbf{z}\|\mathbf{x})$
QM9	Sup-VAE-1-GRU	0.5420	0.2526
	CGD-VAE-1-GRU	0.7185	0.5005
	Sup-VAE-3-GRU	0.6958	0.4204
	CGD-VAE-3-GRU	0.7414	0.4715
ZINC	Sup-VAE-1-GRU	0.2301	0.0481
	CGD-VAE-1-GRU	0.3877	0.0880
	Sup-VAE-3-GRU	0.3514	0.1808
	CGD-VAE-3-GRU	0.3966	0.1559

Table 2: Correlation between the desired input property and the obtained property .

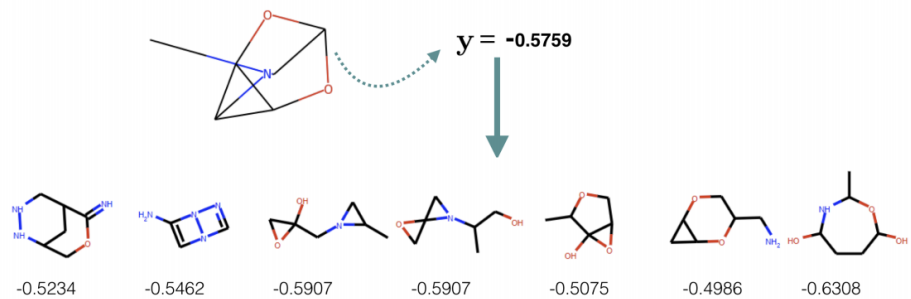


Table 3: Conditional generation given the desired logP=-0.5759, row molecules have a logP within a 15% range of the desired one.

Note: Their result is poor!

Paper 13

Masked Graph Modeling for Molecule Generation (paper 13)
Omar Mahmood, Elman Mansimov, Richard Bonneau, Kyunghyun Cho
`https://ml4molecules.github.io/papers2020/ML4Molecules_2020_paper_13.pdf`

Proposal

Proposal

Masked graph model that learns a distribution over graphs by capturing all possible **conditional distributions over unobserved** nodes and edges **given observed** ones.

Note

I find this work **similar** to the setting of the imitation learning from Eric Jonas' NeurIPS 2019 paper – *Deep imitation learning for molecular inverse problems*.

Main idea

Assume that the missing components η of the conditional distribution $p(\eta|G_{-\eta})$ are conditionally independent of each other given $G_{-\eta}$:

$$p(\eta|G_{-\eta}) = \prod_{v \in \mathcal{V}} p(v|G_{-\eta}) \prod_{e \in \mathcal{E}} p(e|G_{-\eta})$$

Optimization problem (maximizing the likelihood):

$$\arg\max_{\theta} \mathbb{E}_{G \sim D} \mathbb{E}_{G_{-\eta} \sim C(G_{-\eta}|G)} \log p_\theta(\eta|G_{-\eta})$$

During training, we randomly replace a fraction $\alpha_{\text{train}} \in [0, 0.2]$ of features of each node and edge with the symbol MASK. My opinion: It is somewhat similar to self-supervised.

Generation process

To start generation, the authors **initialize** the molecule in either:

1. **Training initialization (TI):** uses a random training set graph as an initial graph:
   - This is cheating!
   - Iteratively mask out each component, and sample a new one to replace it (like Gibbs sampling).
2. **Marginal initialization (MI):**
   - Initializes each graph component according to a categorical distribution over the components from the training set.

Experiments & Results

	Model	Valid	Uniq	Novel	KL Div	Fréchet Dist
SMILES	CharacterVAE [Gómez-Bombarelli et al., 2016]	0.103	0.675	0.900	N/A	N/A
	GrammarVAE [Kusner et al., 2017]	0.602	0.093	0.809	N/A	N/A
	LSTM [Hochreiter and Schmidhuber, 1997] (ours)	0.980	0.962	0.138	0.998	0.984
	Transformer Sml [Vaswani et al., 2017] (ours)	0.947	0.963	0.203	0.987	0.927
	Transformer Reg [Vaswani et al., 2017] (ours)	0.965	0.957	0.183	0.994	0.958
Graph	GraphVAE [Simonovsky and Komodakis, 2018]	0.557	0.760	0.616	N/A	N/A
	MolGAN [Cao and Kipf, 2018]	0.981	0.104	0.942	N/A	N/A
	NAT GraphVAE [Kwon et al., 2019]	0.945	0.343	0.806	N/A	N/A
	MGM (ours proposed)	0.886	0.978	0.518	0.966	0.842

Table 3: QM9 distributional results. Baseline results are taken from [Cao and Kipf, 2018] and [Kwon et al., 2019].

	Model	Valid	Uniq	Novel	KL Div	Fréchet Dist
SMILES	AAE [Polykovskiy et al., 2018]	0.822	1.000	0.998	0.886	0.529
	ORGAN [Guimaraes et al., 2017]	0.379	0.841	0.687	0.267	0.000
	VAE [Gómez-Bombarelli et al., 2016]	0.870	0.999	0.974	0.982	0.863
	LSTM [Hochreiter and Schmidhuber, 1997]	0.959	1.000	0.912	0.991	0.913
	Transformer Sml [Vaswani et al., 2017] (ours)	0.920	0.999	0.939	0.968	0.859
	Transformer Reg [Vaswani et al., 2017] (ours)	0.961	1.000	0.846	0.977	0.883
Graph	Graph MCTS [Jensen, 2018]	1.000	1.000	0.994	0.522	0.015
	NAT GraphVAE [Kwon et al., 2019]	0.830	0.944	1.000	0.554	0.016
	MGM (ours proposed)	0.849	1.000	0.722	0.987	0.845

Table 4: ChEMBL distributional results. Baseline results are taken from [Brown et al., 2018] and [Kwon et al., 2019].

The result is questionable. The mask rate is 10-20% in QM9, and 1-5% in ChEMBL. It is possible that their model overfits and correctly predicts every missing components of the training data, but unable to generate anything new!!

Next time!

1. **Natural Graph Networks** (NeurIPS 2020)
   `https://arxiv.org/abs/2007.08349`
2. **Finite Exchangeable Sequences** (Diaconis & Freedman)
3. **Partial Exchangeability and Sufficiency** (Diaconis & Freedman)