Masked Graph Modeling for Molecule Generation

Omar Mahmood, Elman Mansimov, Richard Bonneau, and Kyunghyun Cho
New York University
kyunghyun.cho@nyu.edu

Abstract

De novo, in-silico design of molecules is a challenging problem. Here, we introduce a masked graph model, which learns a distribution over graphs by capturing all possible conditional distributions over unobserved nodes and edges given observed ones. We train our model on existing molecular graphs and then sample novel molecular graphs from it by iteratively masking and replacing different parts of initialized graphs. We evaluate our approach on the QM9 and ChEMBL datasets using the distribution-learning benchmark from the GuacaMol framework. The benchmark contains five metrics: the validity, uniqueness, novelty, KL-divergence and Fréchet ChemNet Distance scores, the last two of which are measures of the similarity of the generated samples to the dataset distributions. We find that KL-divergence and Fréchet scores are anti-correlated with novelty scores. By varying generation initialization and the fraction of the graph masked and replaced at each generation step, we can increase the Fréchet score at the cost of novelty. In this way, we show that our model offers transparent and tunable control of the trade-off between these metrics. Our model outperforms previously proposed graph-based approaches and is competitive with SMILES-based approaches.

1 Introduction

We frame graph generation as sampling a graph G from a distribution $p^\star(G)$ defined over all possible graphs. As we do not have access to $p^\star(G)$, it is typically modeled by a distribution $p_\theta(G)$. Once we have trained our model on this distribution, we carry out generation by sampling from the trained model. In this paper, we explore an alternative to modeling the joint distribution $p_\theta(G)$. Our approach, *masked graph modeling*, parameterizes and learns conditional distributions $p_\theta(\eta|G_{\setminus\eta})$ where η is a subset of the components (nodes and edges) of G and $G_{\setminus\eta}$ is a graph without those components. With these conditional distributions estimated from data, we sample a graph by iteratively updating its components. At each generation iteration, this involves masking a subset of components and sampling new values for them according to the corresponding conditional distribution. There are two advantages to this approach. First, we do not specify an arbitrary order of graph components, unlike autoregressive (AR) models. Second, learning is exact, unlike in ELBO-based latent variable models.

2 Model

A masked graph model (MGM) operates on a graph G, which consists of a set of N vertices $\mathcal{V} = \{v_i\}_{i=1}^N$ and a set of edges $\mathcal{E} = \{e_{i,j}\}_{i,j=1}^N$. A vertex is denoted by $v_i = (i, t_i)$, where i is the unique index assigned to it, and $t_i \in C_v = \{1, ..., T\}$ is its type, with T the number of node types. An edge is denoted by $e_{i,j} = (i, j, r_{i,j})$, where i, j are the indices to the incidental vertices of this edge and $r_{i,j} \in C_e = \{1, ..., R\}$ is the type of this edge, with R the number of edge types. We use a single graph neural network to parameterize any conditional distribution induced by a given graph. We assume that the missing components η of the conditional distribution $p(\eta|G_{\setminus\eta})$ are conditionally

Machine Learning for Molecules Workshop at NeurIPS 2020. https://ml4molecules.github.io

independent of each other given $G_{\setminus\eta}$, with $\mathcal{V}$ and $\mathcal{E}$ the sets of all vertices and edges in η respectively:

$$p(\eta|G_{\setminus\eta}) = \prod_{v\in\mathcal{V}} p(v|G_{\setminus\eta}) \prod_{e\in\mathcal{E}} p(e|G_{\setminus\eta}), \tag{1}$$

We start by embedding the vertices and edges in the graph $G_{\setminus\eta}$ to get continuous representations $h_{v_i} \in \mathbb{R}^{d_0}$ and $h_{e_{i,j}} \in \mathbb{R}^{d_0}$ respectively, where d_0 is the representation space dimensionality [Bengio et al., 2003]. We pass these representations to a message passing neural network [Gilmer et al., 2017] with L layers that share parameters. At each layer l, we first update the hidden state of each node v_i by computing its accumulated message $u_{v_i}^{(l)}$ using an aggregation function J_v and a spatial residual connection R between neighboring nodes:

$$\begin{aligned}
u_{v_i}^{(l)} &= J_v(h_{v_i}^{(l-1)}, \{h_{v_j}^{(l-1)}\}_{j\in N(i)}, \{h_{e_{i,j}}^{(l-1)}\}_{j\in N(i)}) + R(\{h_{v_j}^{(l-1)}\}_{j\in N(i)}), \\
J_v(h_{v_i}^{(l-1)}, \{h_{v_j}^{(l-1)}\}_{j\in N(i)}, \{h_{e_{i,j}}^{(l-1)}\}_{j\in N(i)}) &= \sum_{j\in N(i)} h_{e_{i,j}}^{(l-1)} \cdot h_{v_j}^{(l-1)}, \\
R(\{h_{v_j}^{(l-1)}\}_{j\in N(i)}) &= \sum_{j\in N(i)} h_{v_j}^{(l-1)}, \\
h_{v_i}^{(l)} &= \text{LayerNorm}(\text{GRU}(h_{v_i}^{(l-1)}, u_{v_i}^{(l)})),
\end{aligned}$$

where $N(i)$ is the set of indices corresponding to nodes that are in the one-hop neighbourhood of node v_i. GRU [Cho et al., 2014] refers to a gated recurrent unit which updates the representation of each node using its previous representation and accumulated message. LayerNorm [Ba et al., 2016] refers to layer normalization.

Similarly, the hidden states of each edge $h_{e_{i,j}}$ are updated using the following rule for all $j \in N(i)$:

$$h_{e_{i,j}}^{(l)} = J_e(h_{v_i}^{(l-1)} + h_{v_j}^{(l-1)}),$$

where J_e is a two-layer fully connected network with ReLU activation between the two layers [Nair and Hinton, 2010, Glorot et al., 2011], to yield a new hidden edge representation. The node and edge representations from the final layer are then processed by a node projection layer $A_v : \mathbb{R}^{d_0} \to \Lambda^T$ and an edge projection layer $A_e : \mathbb{R}^{d_0} \to \Lambda^R$, where Λ^T and Λ^R are probability simplices over node and edge types respectively. The result is the distributions $p(v|G_{\setminus\eta})$ and $p(e|G_{\setminus\eta})$ for all $v \in \mathcal{V}$ and all $e \in \mathcal{E}$.

Learning We corrupt each graph G from a training dataset D with a corruption process $C(G_{\setminus\eta}|G)$. Following work for language models [Devlin et al., 2019], we randomly replace a fraction α_{train} of features of each node and edge with the symbol MASK. We vary α_{train} randomly between 0 and 0.2 throughout training. After passing $G_{\setminus\eta}$ through our model we obtain the conditional distribution $p(\eta|G_{\setminus\eta})$. We maximize the log probability $\log p(\eta|G_{\setminus\eta})$ of the masked components η given the rest of the graph $G_{\setminus\eta}$. This results in the optimization problem: $\arg\max_\theta \mathbb{E}_{G\sim D}\mathbb{E}_{G_{\setminus\eta}\sim C(G_{\setminus\eta}|G)} \log p_\theta(\eta|G_{\setminus\eta})$.

Generation To begin generation, we initialize a molecule in one of two ways.The first way, training initialization (TI), uses a random training set graph as an initial graph. The second way, marginal initialization (MI), initializes each graph component according to a categorical distribution over the values the component takes in our training set. For example, the probability of an edge having type $r \in C_e$ is equal to the fraction of edges in the training set of type r. We then use an approach motivated by Gibbs sampling to update graph components iteratively from the learned conditional distributions. At each generation step, we sample uniformly at random a fraction α_{gen} of components η in the graph and replace the values of these components with the MASK symbol. We compute the conditional distribution $p(\eta|G_{\setminus\eta})$ using the model, sample new values for the masked components, and place these values in the graph. We repeat this procedure for K steps, where K is a hyperparameter.

3 Experiments

Datasets and Evaluation We use two widely used [Gómez-Bombarelli et al., 2016, Simonovsky and Komodakis, 2018, Li et al., 2018] small-molecule datasets: QM9 [Ruddigkeit et al., 2012,

Ramakrishnan et al., 2014] and ChEMBL [Mendez et al., 2018]. QM9 has approximately 132,000 molecules with a median and maximum of 9 heavy atoms each and $T = 5$ atom types. ChEMBL contains approximately 1,591,000 molecules with a median of 27 and a maximum of 88 heavy atoms each, with $T = 12$. For both datasets, each bond is either a no-bond, single, double, triple or aromatic bond ($R = 5$). To evaluate our approach, we use distribution-learning metrics from the GuacaMol benchmark [Brown et al., 2018]: validity, uniqueness, novelty, KL-divergence (KLD) [Kullback and Leibler, 1951] and Fréchet ChemNet Distance (FCD) [Preuer et al., 2018]. These are all scores between 0 and 1. Validity, uniqueness and novelty measure the proportion of generated molecules that are valid, that remain after removing duplicated molecules and that are not dataset molecules respectively. The KLD and FCD scores measure the similarity of generated samples to the dataset distribution.

Baselines We train an LSTM [Hochreiter and Schmidhuber, 1997] on QM9 and get a pretrained LSTM for ChEMBL using the GuacaMol baselines implementation [gua]. We train two Transformers [Vaswani et al., 2017] of different size on each dataset. Other QM9 results (CharacterVAE [Gómez-Bombarelli et al., 2016], GrammarVAE [Kusner et al., 2017], GraphVAE [Simonovsky and Komodakis, 2018], MolGAN [Cao and Kipf, 2018]) are from Cao and Kipf [2018]. Other ChEMBL results (LSTM, Graph MCTS [Jensen, 2018], AAE [Polykovskiy et al., 2018], ORGAN [Guimaraes et al., 2017], VAE [Simonovsky and Komodakis, 2018] (bidirectional GRU [Cho et al., 2014] encoder, AR GRU decoder) are from Brown et al. [2018].

Mutual Dependence of GuacaMol Metrics If dependence exists between GuacaMol metrics, comparing models using a straightforward measure such as sum of metrics may not be reasonable. We calculate pairwise the Spearman correlation between all metrics using MGM on QM9 (Table 1a), while varying the masking rate α_{gen}, initialization and number of sampling iterations. We carry out a similar run for the Transformer Small, Transformer Regular, and LSTM baselines by varying softmax sampling temperatures at generation (Table 1b). Validity, KLD and FCD scores correlate highly with each other and negatively with novelty. Uniqueness does not correlate strongly with any metric. This suggests we can gauge generation quality using a subset of metrics, such as FCD and novelty scores.

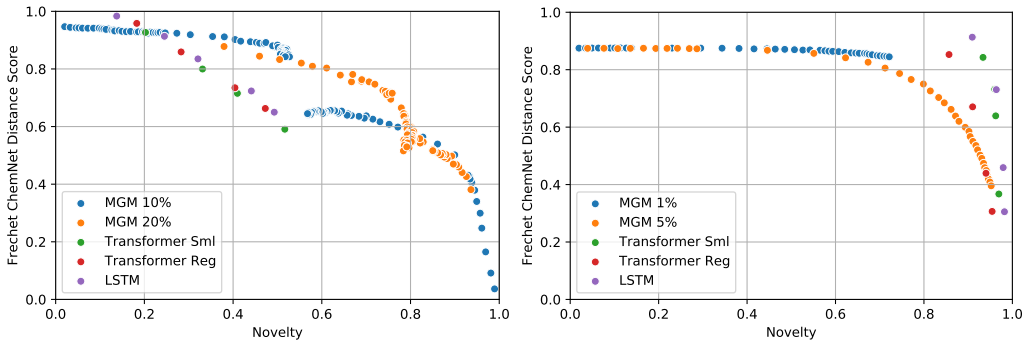
	Valid	Uniq	Novel	KLD	FCD
Valid	1.00	-0.56	-0.83	0.73	0.75
Uniq	-0.56	1.00	0.50	-0.32	-0.37
Novel	-0.83	0.50	1.00	-0.94	-0.95
KL	0.73	-0.32	-0.94	1.00	0.99
Fréchet	0.75	-0.37	-0.95	0.99	1.00

(a) MGM

	Valid	Uniq	Novel	KLD	FCD
Valid	1.00	0.03	-0.99	0.98	0.98
Uniq	0.03	1.00	0.00	0.03	0.03
Novel	-0.99	0.00	1.00	-0.99	-0.99
KL	0.98	0.03	-0.99	1.00	1.00
Fréchet	0.98	0.03	-0.99	1.00	1.00

(b) LSTM, Transformer Small and Transformer Regular

Table 1: Spearman correlation between metrics from QM9 results using MGM and AR models.



(a) QM9 (b) ChEMBL

Figure 1: Plots of the Fréchet ChemNet Distance score against novelty.

Analysis of Representative Metrics We plot the FCD and novelty scores against each other in Figure 1. On both QM9 and ChEMBL, as novelty increases, the FCD score decreases for the MGMs

as well as for the AR LSTM and Transformer models. We also see that the line's slope has a lower magnitude for the MGMs than for the AR models. This shows that our model trades off novelty for similarity to the dataset distributions more effectively than the baseline models. This gives us a higher degree of controllability in generating samples that are optimized towards either metric. On the QM9 plot, we see that several MGM points are beyond each baseline model's Pareto frontier. On ChEMBL, the AR models generally achieve a higher combination of novelty and FCD score than do the MGMs.

Effect of Generation Hyperparameters on Generation Quality Table 2 shows how masking rate and initialization affect generation, using scores at the final generation step. On QM9, using MI as compared to TI leads to slightly higher validity and novelty but lower KLD and FCD scores. On ChEMBL, MI results in validity scores close to 0, hence we only consider TI in Table 2. On both datasets, novelty increases significantly when increasing the masking rate while the validity, KLD and FCD scores drop. We can trade off between metrics by adjusting initialization and masking rate.

Dataset	Mask Rate	Graph Init	Valid	Uniq	Novel	KL Div	Fréchet Dist
QM9	10%	train	0.886	0.978	0.518	0.966	0.842
	10%	marginal	0.922	0.972	0.568	0.930	0.645
	20%	train	0.678	0.988	0.789	0.901	0.544
	20%	marginal	0.719	0.982	0.792	0.893	0.529
ChEMBL	1%	train	0.849	1.000	0.722	0.987	0.845
	5%	train	0.558	1.000	0.952	0.869	0.396

Table 2: Effect of varying masking rate and graph initialization on MGM benchmark 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].

Comparison with Baseline Models We select MGM results from Table 2 for each dataset corresponding to the highest geometric mean among all five metrics. On QM9 (Table 3), our model performs comparably to existing methods. Our approach has higher validity and uniqueness and lower novelty scores compared to CharacterVAE [Gómez-Bombarelli et al., 2016], GrammarVAE [Kusner et al., 2017], GraphVAE [Simonovsky and Komodakis, 2018] and MolGAN [Cao and Kipf, 2018]. Our model has lower validity and novelty but significantly higher uniqueness scores than

non-AR graph VAE [Kwon et al., 2019]. Compared to the LSTM and Transformer models, our model has lower validity, KLD and FCD scores but higher uniqueness and novelty scores.

On ChEMBL (Table 4), our approach outperforms existing graph-based methods. Compared to graph MCTS [Jensen, 2018] and non-AR graph VAE [Kwon et al., 2019], our approach shows lower novelty but significantly higher KLD and FCD scores. The baseline graph-based models do not capture the properties of the dataset distributions, as shown by their low KLD and almost-zero FCD scores. Our approach is competitive with SMILES-based models. It outperforms the GAN-based model (ORGAN) across all metrics and outperforms the adversarial autoencoder (AAE) on all but the uniqueness (both are 1.00) and novelty scores. It performs comparably to the VAE with AR GRU [Cho et al., 2014] decoder on all metrics except novelty. Our approach lags behind the SMILES-based LSTM and Transformer models. It outperforms both Transformer models on KLD score but underperforms them on validity, novelty and FCD score. Our approach also results in lower scores across most metrics when compared to the LSTM model.

References

Guacamol baselines. `https://github.com/BenevolentAI/guacamol_baselines`.

J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization, 2016.

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, Mar. 2003. ISSN 1532-4435.

N. Brown, M. Fiscato, M. H. Segler, and A. C. Vaucher. Guacamol: Benchmarking models for de novo molecular design. *arXiv preprint 1811.09621*, 2018.

N. D. Cao and T. Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint 1805.11973*, 2018.

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014.

J. Devlin, M.-W. Chang, and K. T. Kenton Lee. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1263–1272, 2017.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. JMLR Workshop and Conference Proceedings. URL `http://proceedings.mlr.press/v15/glorot11a.html`.

R. Gómez-Bombarelli, D. Duvenaud, J. M. Hernández-Lobato, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *arXiv preprint 1610.02415*, 2016.

G. L. Guimaraes, B. Sanchez-Lengeling, P. L. C. Farias, and A. Aspuru-Guzik. Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. *CoRR*, abs/1705.10843, 2017. URL `http://arxiv.org/abs/1705.10843`.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

J. H. Jensen. Graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. 2018.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22: 79–86, 1951.

M. J. Kusner, B. Paige, and J. M. Hernández-Lobato. Grammar variational autoencoder. In *ICML*, 2017.

Y. Kwon, J. Yoo, Y. Choi, W. joon Son, D. Lee, and S. Kang. Efficient learning of non-autoregressive graph variational autoencoders for molecular graph generation. *Journal of Cheminformatics*, 11, 2019.

Y. Li, L. Zhang, and Z. Liu. Multi-objective de novo drug design with conditional graph generative model. *Journal of Cheminformatics*, 10(1), Jul 2018. ISSN 1758-2946. doi: 10.1186/s13321-018-0287-6. URL http://dx.doi.org/10.1186/s13321-018-0287-6.

D. Mendez, A. Gaulton, A. P. Bento, J. Chambers, M. De Veij, E. Félix, M. Magariños, J. Mosquera, P. Mutowo, M. Nowotka, M. Gordillo-Marañón, F. Hunter, L. Junco, G. Mugumbate, M. Rodriguez-Lopez, F. Atkinson, N. Bosc, C. Radoux, A. Segura-Cabrera, A. Hersey, and A. Leach. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Research*, 2018.

V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010. URL https://icml.cc/Conferences/2010/papers/432.pdf.

D. Polykovskiy, A. Zhebrak, D. Vetrov, Y. Ivanenkov, V. Aladinskiy, P. Mamoshina, M. Bozdaganyan, A. Aliper, A. Zhavoronkov, and A. Kadurin. Entangled conditional adversarial autoencoder for de novo drug discovery. *Molecular Pharmaceutics*, 2018.

K. Preuer, P. Renz, T. Unterthiner, S. Hochreiter, and G. Klambauer. Fréchet chemnet distance: A metric for generative models for molecules in drug discovery. *Journal of chemical information and modeling*, 2018.

R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data*, 1:140022:1–7, 2014.

L. Ruddigkeit, R. Van Deursen, L. C. Blum, and J.-L. Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *J. Chem. Inf. Model.*, 52(11): 2864–2875, 2012.

M. Simonovsky and N. Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. *arXiv preprint 1802.03480*, 2018.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017.