

---

# MOLUCINATE: A Generative Model for Molecules in 3D Space

---

**Michael Arcidiacono**  
mixarcidiacono@gmail.com

**David Ryan Koes**  
Comp. & Systems Biology  
University of Pittsburgh  
Pittsburgh, PA 15213  
dkoes@pitt.edu

## Abstract

Recent advances in machine learning have enabled generative models for both optimization and *de novo* generation of drug candidates with desired properties. Previous generative models have focused on producing SMILES strings or 2D molecular graphs, while attempts at producing molecules in 3D have focused on reinforcement learning (RL), distance matrices, and pure atom density grids. Here we present MOLUCINATE (MOLEcUlar ConvolutioNal generATive modEl), a novel architecture that simultaneously generates topological and 3D atom position information. We demonstrate the utility of this method by using it to optimize molecules for desired radius of gyration. In the future, this model can be used for more useful optimization such as binding affinity for a protein target.

## 1 Introduction

Computational drug discovery aims to generate molecules with optimized activity against protein targets. Given the success of machine learning (ML) techniques in image and text processing, there has been interest in applying similar techniques to *de novo* drug design. Recent successes at generation of SMILES strings [31, 7, 14, 9, 18] and graphs [3, 32, 27, 12] are promising, but most do not generate 3D conformer information. Since a molecule’s binding affinity against a protein target is dependant on its 3D conformation within the binding site, it is desirable to generate this information as well. [30, 21, 11, 6].

Current work in this area poses the problem of 3D molecular generation as a reinforcement learning (RL) problem [5, 26, 16, 19], which show promise for *de novo* molecule generation. However, RL-based methods do not share certain advantages with generative methods that produce a latent space of molecules, allowing interpolation between molecules and optimization of properties directly via gradient descent. Among non-RL-based methods, there has been some work on generating distance matrices [4, 8, 20], but this research is preliminary and so far restricted to very small molecules and constitutional isomers. Additionally, Samanta et al. [25] generate 3D coordinates directly using graph convolutions, but this approach fails to encode inductive biases about 3D geometry and thus generates unrealistic coordinates. Lastly, Ragoza et al. [21] generate atomic density grids, which has the advantage of being the same format as the input to convolutional networks predicting binding affinity [30, 21, 11, 6], but requires an extra overhead for decoding the molecule topology from the grid. In this work, we build upon this representation to also incorporate information about the molecular graph. By simultaneously generating the molecular graph and its 3D structure, we can create a latent space of molecules that enables easy optimization of 3D properties while maintaining topological similarity to the original molecule. In the future, this model can be used for tasks such as *de novo* drug discovery and optimization of candidate molecules for potency against a protein target.

In this work, we use a novel representation of 3D molecules that use per-atom voxel grids to encode atom position information. By combining spatial and graph convolutions into a novel architecture, we create a variational auto-encoder (VAE) [13] to generate a latent space of 3D molecules. Additionally, we demonstrate the usefulness of this auto-encoder by using it to optimize molecules for radius of gyration (Rg). This is an inherently 3D property of a molecule and can be readily and unambiguously computed for any molecule. In the future, we hope to use this architecture to optimize molecules for more interesting metrics (such as binding affinity against protein targets) and generate molecules conditional on a protein binding pocket.

## 2 Methods

### 2.1 Representation of Molecules

In order to represent molecules in 3D, we store each molecule as a graph  $\mathcal{G}$  with a list of nodes  $\mathcal{A}$  and edges  $\mathcal{E}$ . The atoms are ordered according to the canonical SMILES representation (as returned by RD-Kit [1]). Each atom  $a_i \in \mathcal{A}$  contains information about its type  $t_i \in \{\text{C, N, O, P, S, F, Cl, Br, I, END}\}$  and valence  $v_i \in \mathbb{N}^3$ . The atom valences tensor specifies the number of total bonds of each type (single, double triple) that the atom has to other (heavy) atoms. For instance, a carbon in a benzene ring would have its valence encoded as  $[1, 1, 0]$ . A final, virtual, atom of type END is also appended to the graph.

To store 3D information, each atom has a position tensor with shape  $(G, G, G)$ , where  $G$  is the grid size. For this application, an 8 Å grid with 0.5 Å resolution was used, so  $G = 16$ . We used two separate density encodings for the input and the output of the VAE. For the input molecules, the density function is taken from Ragoza et al. [21] and is based on 3D exponential decay defined by the Van Der Waals radius. For the output, we one-hot encode the atom positions: we set the voxel that contains the atom center to 1 and set all other voxels to zero. While the one-hot representation is too sparse for the encoder’s 3D convolutions [15], it is useful for the output because it enables us to softmax the output “densities” over all space, allowing us to interpret the output as a 3D probability distribution.

For simplicity, we store bond types as per-atom information. Each atom  $a_i$  has a bond type list  $b_{ij} \in \{\text{SINGLE, DOUBLE, TRIPLE, NO\_BOND}\}$  defined as the list of bond types to previous atoms  $a_j$ , where  $j < i$ . Each atom can have a maximum of 4 bonds to previous atoms and lists with less than 4 elements are padded with the extra bond type NO\_BOND. Note that, since double and triple bonds are encoded as individual “bonds” with different types, the 4 bond maximum still allows relevant hypervalent moieties such as phosphate groups.

### 2.2 Combining Graph and Spatial Convolutions

To encode and decode the atom position grids, graph and spatial convolutions were combined. For layer  $l$  in the network, for each atom  $i$ , we define a voxel hidden state  $h_i^{(l)}$  with shape  $(C_l, G_l, G_l, G_l)$ , where  $C_l$  is the number of channels (the input atom positions have  $C_0 = 1$ ) and  $G_l$  is the grid size for that layer. To perform the graph-spatial convolution, the summed channels of each atom’s neighbors are concatenated with the atom’s channels, and then a 3x3x3 spatial convolution is performed.

### 2.3 Bond prediction

To predict bonded atoms, we take inspiration from the attention mechanism. For each atom  $a_i$ , we separately predict 4 bonds to atoms before in the range  $[0, i)$ . Once the VAE has decoded a hidden state  $h_i$ , for each bond type  $b \in b_i$  we use a linear layer (followed by a leaky ReLU) to compute keys  $k_j^b$ . We compute values  $v_j^b$  in the same way. The score for atom  $a_i$  bonded to atom  $a_{j < i}$  is simply  $k_i^{b\top} v_j^b$ . When decoding the final molecule, a bond is created between  $a_i$  and the top scoring atom  $a_j$  if  $b \neq \text{NO\_BOND}$ .

### 2.4 VAE Architecture

The overall auto-encoder architecture is shown in Figure 1. To encode a molecule into the latent space, we first encode the atom density grids into single per-atom vectors by alternating spatial-graph

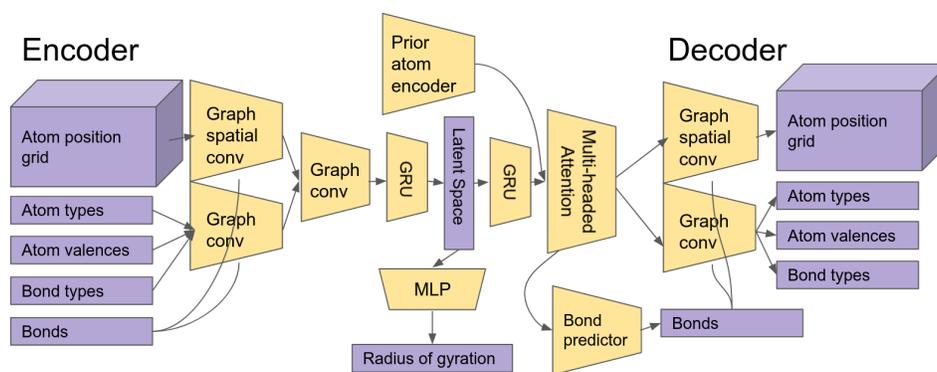


Figure 1: Model architecture.

convolutional layers and  $2 \times 2 \times 2$  max pooling. This hidden state is then concatenated with the hidden state obtained by running graph convolutions on the remaining atom features (atom type, valences, and bonds types). After concatenation, further graph convolutions are used to produce the final per-atom hidden state. To turn this into a single latent code, a gated recurrent unit (GRU) [2] is used. The latent space size was 512.

To decode, we generate new atoms autoregressively. To generate atom  $a_i$ , we encode atoms  $a_0, \dots, a_{i-1}$  via an encoder with a similar architecture to the one described above – however, all bonds to atoms  $i$  or greater are masked out of the graph convolutions to prevent cheating. This encoded representation is concatenated with the latent code and a GRU is run to produce a new hidden representation. This is then fed into a multi-headed attention module [29] with 8 heads. To run graph convolutions, we need bonding information, so the bond predictor is immediately run from this hidden state. Using the predicted bonds, we run graph convolutions to produce the atom types, atom valences, and bond types. We additionally use spatial-graph convolutions to produce the predicted atom positions. To generate new molecules, we iteratively run the decoder to produce atoms one by one until an atom is reached with type END.

Additionally, we desire to predict Rg from the latent code. This is accomplished with a simple two-layer multi-layer perceptron (MLP). This property predictor is trained concurrently with the auto-encoder.

## 2.5 Training

The VAE was trained separately on the QM9 [24, 22] and ZINC 15 datasets [28, 10]. The datasets were filtered down to molecules with 16 or fewer heavy atoms that fit into an  $8 \text{ \AA}$  grid (60,888 files for ZINC, 121,836 for QM9). The QM9 dataset was divided according to a 80:10:10 train:validation:test split, while ZINC was divided with a 90:10 train:validation split.<sup>1</sup> During training, molecules were loaded, kekulized, rotated randomly, and then fed into the VAE. The model was trained to simultaneously reconstruct the input and predict Rg. To do this, we utilized seven loss functions: the cross-entropy for the atom types, atom valences, atom positions, bond types, and bonded atoms, mean-squared-error (MSE) for Rg, and Kullback–Leibler (KL) divergence of the latent code to a multivariate normal distribution. The loss weights were, respectively, 1.0, 1.0, 0.6, 2.0, 2.0, 1.0, and 0.1. The VAE was optimized using AdamW [17] with a learning rate of 0.001 and batch size of 64 for 168 epochs.

## 3 Results

The model was evaluated according to its ability to reconstruct, generate, and optimize molecules. To test reconstruction accuracy, molecules were sampled from the test dataset, encoded, and then decoded. The topological accuracy is the fraction of molecules whose decoded molecular graphs were

<sup>1</sup>Due to an error early in training, a proper test set for ZINC was not used. No such mistake was made for QM9.

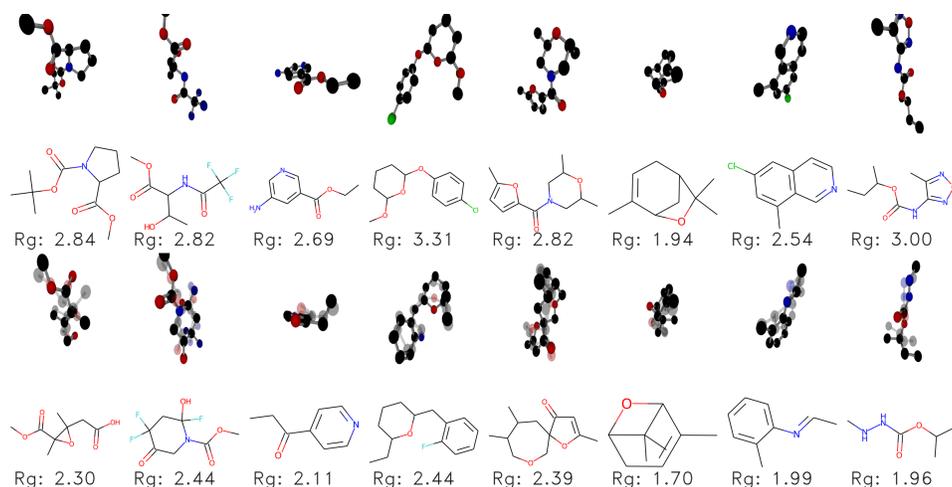


Figure 2: Results of minimizing Rg of molecules sampled from the ZINC validation set. The top row shows the original molecules, and the bottom row shows the molecules after optimization. The optimized molecules are shown after UFF relaxation; the raw output of the model is overlaid transparently.

exactly the same as the input molecules. Of the molecules that were correctly decoded, the RMSD of the conformers was also measured. On the QM9 test set, our model achieved a reconstruction accuracy of 94.7% and the mean RMSD was 0.15 Å. On the ZINC validation set, the Reconstruction accuracy was 50.6% and the mean RMSD was 0.62 Å.

To test generation, 16,000 latent points were randomly sampled from a multivariate normal distribution. The topological validity was defined to be the fraction of generated molecules that did not throw an exception when using RDKit’s `SanitizeMol`. Additionally, the stability of the generated conformers was tested by running Unified Force Field (UFF) optimization [23]. The geometric validity was the fraction of topologically valid molecules that did not throw an exception when running UFF optimization. The RMSD between the (aligned) optimized conformer and the generated conformer was also measured for each geometrically valid molecule. Lastly, the distribution of computed properties of the generated molecules was compared to the distribution of the filtered original dataset. Of the molecules generated from the QM9 model, 84.0% were topologically valid. Of those, 75.0% were geometrically valid. Of the geometrically valid molecules, the mean UFF RMSD was 1.23 Å. For the ZINC model, the topological validity was 60.3%, the geometric validity was 75.0%, and the UFF RMSD was 1.05 Å.

To test optimization, molecules were chosen from the ZINC validation set and encoded into latent vectors. Rg was predicted using the prediction network on the latent vector, and the derivative of the predicted radius with respect to the latent vector was computed. Gradient descent was then run on the latent vector for 500 iterations (learning rate 0.2) to minimize the radius. The results of Rg minimization are shown in Figure 2. After optimizing 1,000 molecules, 75.0% of the resulting molecules were valid, 99.5% of the valid molecules had reduced Rg, and the average difference in Rg was -0.53 Å. Qualitatively, the results are consistent with the algorithm “shrinking” the molecules while preserving overall shape and features.

## 4 Conclusions

By using a novel graph representation of molecules that incorporates voxelized atom position information, we have created a generative model that allows encoding, decoding, interpolation, and optimization of both the geometric and topological properties of molecules. Our model produces topologically valid molecules 60.3% of the time on the ZINC validation set, and the UFF RMSD of 1.0 Å demonstrates that the molecules produced are often close to low energy conformations. However, the fact that the UFF optimization routine failed to converge for a quarter of valid molecules

implies that the actual mean RMSD to a low-energy structure might be larger. More tweaking of the UFF optimization parameters needs to be done so that it does not fail so often.

We have also demonstrated that this model can optimize molecules for characteristics defined by the molecule’s 3D geometry – here, Rg. While this is admittedly a toy problem, the optimization serves as a useful proof of concept. In the future, we hope to use this model to optimize molecules for more useful characteristics such as binding affinity for protein targets.

Perhaps the biggest current limitation of the model was the small size of molecules used: the model was only trained and tested on molecules with 16 or fewer heavy atoms that fit into an 8 Å cube. These parameters were chosen for ease of iteration; since the model run time scales as  $O(ag^3)$ , where  $a$  is the number of atoms and  $g$  is the grid size, supporting larger molecules takes more time to train. Now that we have narrowed in on an architecture that works for these small molecules, we hope to scale up the model to support larger and more useful molecules.

The full source code for training and evaluating this model is available at <https://github.com/mixarcid/molucinate>.

## 5 Acknowledgements

The authors thank Dr. Alexander Tropsha, Matthew Ragoza, and Jack Lynch for their insightful comments while discussing this work. Additionally, this work is supported by R01GM108340 from the National Institute of General Medical Sciences.

## References

- [1] RDKit: Open-source cheminformatics. URL <https://www.rdkit.org/>.
- [2] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. URL <http://arxiv.org/abs/1406.1078>.
- [3] Nicola De Cao and Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. URL <http://arxiv.org/abs/1805.11973>.
- [4] Niklas W. A. Gebauer, Michael Gastegger, and Kristof T. Schütt. Generating equilibrium molecules with deep neural networks. . URL <http://arxiv.org/abs/1810.11347>.
- [5] Niklas W. A. Gebauer, Michael Gastegger, and Kristof T. Schütt. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. . URL <http://arxiv.org/abs/1906.00957>.
- [6] Joseph Gomes, Bharath Ramsundar, Evan N. Feinberg, and Vijay S. Pande. Atomic convolutional networks for predicting protein-ligand binding affinity. URL <http://arxiv.org/abs/1703.10603>.
- [7] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. 4(2):268–276. ISSN 2374-7943, 2374-7951. doi: 10.1021/acscentsci.7b00572. URL <http://arxiv.org/abs/1610.02415>.
- [8] Moritz Hoffmann and Frank Noé. Generating valid euclidean distance matrices. URL <http://arxiv.org/abs/1910.03131>.
- [9] Seung Hwan Hong, Jaechang Lim, Seongok Ryu, and Woo Youn Kim. Molecular generative model based on adversarially regularized autoencoder. URL <http://arxiv.org/abs/1912.05617>.
- [10] John J. Irwin and Brian K. Shoichet. ZINC – a free database of commercially available compounds for virtual screening. 45(1):177–182. ISSN 1549-9596. doi: 10.1021/ci049714. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1360656/>.
- [11] José Jiménez, Miha Škalič, Gerard Martínez-Rosell, and Gianni De Fabritiis. KDEEP: Protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. 58(2): 287–296. ISSN 1549-9596. doi: 10.1021/acs.jcim.7b00650. URL <https://doi.org/10.1021/acs.jcim.7b00650>. Publisher: American Chemical Society.

- [12] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. URL <http://arxiv.org/abs/1802.04364>.
- [13] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. URL <http://arxiv.org/abs/1312.6114>.
- [14] Matt J. Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. URL <http://arxiv.org/abs/1703.01925>.
- [15] Denis Kuzminykh, Daniil Polykovskiy, Artur Kadurin, Alexander Zhebrak, Ivan Baskov, Sergey Nikolenko, Rim Shayakhmetov, and Alex Zhavoronkov. 3d molecular representations based on the wave transform for convolutional neural networks. 15(10):4378–4385. ISSN 1543-8384. doi: 10.1021/acs.molpharmaceut.7b01134. URL <https://doi.org/10.1021/acs.molpharmaceut.7b01134>. Publisher: American Chemical Society.
- [16] Yibo Li, Jianfeng Pei, and Luhua Lai. Learning to design drug-like molecules in three-dimensional space using deep generative models. URL <http://arxiv.org/abs/2104.08474>.
- [17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. URL <http://arxiv.org/abs/1711.05101>.
- [18] Łukasz Maziarka, Agnieszka Pocha, Jan Kaczmarczyk, Krzysztof Rataj, Tomasz Danel, and Michał Warchoń. Mol-CycleGAN: a generative model for molecular optimization. 12(1):2. ISSN 1758-2946. doi: 10.1186/s13321-019-0404-1. URL <https://doi.org/10.1186/s13321-019-0404-1>.
- [19] Søren Ager Meldgaard, Jonas Köhler, Henrik Lund Mortensen, Mads-Peter V. Christiansen, Frank Noé, and Bjørk Hammer. Generating stable molecules using imitation and reinforcement learning. URL <http://arxiv.org/abs/2107.05007>.
- [20] Vitali Nesterov, Mario Wieser, and Volker Roth. 3dmolnet: A generative network for molecular structures. page 5.
- [21] Matthew Ragoza, Joshua Hochuli, Elisa Idrobo, Jocelyn Sunseri, and David Ryan Koes. Protein–ligand scoring with convolutional neural networks. 57(4):942–957. ISSN 1549-9596. doi: 10.1021/acs.jcim.6b00740. URL <https://doi.org/10.1021/acs.jcim.6b00740>. Publisher: American Chemical Society.
- [22] Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. 1(1):140022. ISSN 2052-4463. doi: 10.1038/sdata.2014.22. URL <https://www.nature.com/articles/sdata201422>. Bandiera\_abtest: a Cg\_type: Nature Research Journals Number: 1 Primary\_atype: Research Publisher: Nature Publishing Group Subject\_term: Computational chemistry;Density functional theory;Quantum chemistry Subject\_term\_id: computational-chemistry;density-functional-theory;quantum-chemistry.
- [23] A. K. Rappe, C. J. Casewit, K. S. Colwell, W. A. Goddard, and W. M. Skiff. UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations. 114 (25):10024–10035. ISSN 0002-7863. doi: 10.1021/ja00051a040. URL <https://doi.org/10.1021/ja00051a040>. Publisher: American Chemical Society.
- [24] Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. 52(11): 2864–2875. ISSN 1549-9596. doi: 10.1021/ci300415d. URL <https://doi.org/10.1021/ci300415d>. Publisher: American Chemical Society.
- [25] Bidisha Samanta, Abir De, Gourhari Jana, Pratim Kumar Chattaraj, Niloy Ganguly, and Manuel Gomez-Rodriguez. NeVAE: A deep generative model for molecular graphs. URL <http://arxiv.org/abs/1802.05283>.
- [26] Gregor N C Simm, Robert Pinsler, Gábor Csányi, and José Miguel Hernández-Lobato. 3d molecular design with covariant neural networks. page 6.
- [27] Martin Simonovsky and Nikos Komodakis. GraphVAE: Towards generation of small graphs using variational autoencoders. URL <http://arxiv.org/abs/1802.03480>.
- [28] Teague Sterling and John J. Irwin. ZINC 15 – ligand discovery for everyone. 55(11):2324–2337. ISSN 1549-9596. doi: 10.1021/acs.jcim.5b00559. URL <https://doi.org/10.1021/acs.jcim.5b00559>. Publisher: American Chemical Society.

- [29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. URL <http://arxiv.org/abs/1706.03762>.
- [30] Izhar Wallach, Michael Dzamba, and Abraham Heifets. AtomNet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. URL <http://arxiv.org/abs/1510.02855>.
- [31] David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. 28(1):31–36. ISSN 0095-2338. doi: 10.1021/ci00057a005. URL <https://pubs.acs.org/doi/abs/10.1021/ci00057a005>. Publisher: American Chemical Society.
- [32] Chengxi Zang and Fei Wang. MoFlow: An invertible flow model for generating molecular graphs. pages 617–626. doi: 10.1145/3394486.3403104. URL <http://arxiv.org/abs/2006.10137>.