Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules

Niklas W. A. Gebauer
Machine Learning Group
Technische Universität Berlin
10587 Berlin, Germany
n.wa.gebauer@gmail.com

Michael Gastegger
Machine Learning Group
Technische Universität Berlin
10587 Berlin, Germany
michael.gastegger@tu-berlin.de

Kristof T. Schütt
Machine Learning Group
Technische Universität Berlin
10587 Berlin, Germany
kristof.schuett@tu-berlin.de

Abstract

Deep learning has proven to yield fast and accurate predictions of quantum-chemical properties to accelerate the discovery of novel molecules and materials. As an exhaustive exploration of the vast chemical space is still infeasible, we require generative models that guide our search towards systems with desired properties. While graph-based models have previously been proposed, they are restricted by a lack of spatial information such that they are unable to recognize spatial isomerism and non-bonded interactions. Here, we introduce a generative neural network for 3d point sets that respects the rotational invariance of the targeted structures. We apply it to the generation of molecules and demonstrate its ability to approximate the distribution of equilibrium structures using spatial metrics as well as established measures from chemoinformatics. As our model is able to capture the complex relationship between 3d geometry and electronic properties, we bias the distribution of the generator towards molecules with a small HOMO-LUMO gap – an important property for the design of organic solar cells.

1   Introduction

In recent years, machine learning enabled a significant acceleration of molecular dynamics simulations [1–6] as well as the prediction of chemical properties across chemical compound space [7–12]. Still, the discovery of novel molecules and materials with desired properties remains a major challenge in chemistry and materials science. Accurate geometry-based models require the atom types and positions of candidate molecules in *equilibrium*, i.e. local minima of the potential energy surface. These have to be obtained by geometry optimization, either using computationally expensive quantum chemistry calculations or a neural network trained on both compositional (chemical) and configurational (structural) degrees of freedom [12–15]. On the other hand, bond-based methods that predict chemical properties using only the molecular graph [16–19] do not reach the accuracy of geometry-based methods [10, 20]. In particular, they lack crucial information to distinguish spatial isomers, i.e. molecules that exhibit the same bonding graph, but different 3d structures and chemical properties. As an exhaustive search of chemical space is infeasible, several methods have been proposed to generate molecular graphs [21, 22] and to bias the distribution of the generative model

towards desired chemical properties for a guided search [23, 24]. Naturally, these models suffer from the same drawbacks as bond-based predictions of chemical properties.

In this work, we propose *G-SchNet (Generative SchNet)* for the generation of rotational invariant 3d point sets. Points are placed sequentially according to a learned conditional probability distribution that reflects the symmetries of the previously placed points by design. Therefore, G-SchNet conserves rotational invariance as well as local symmetries in the predicted probabilities. We build upon the SchNet architecture [13] using continuous-filter convolutional layers to model the interactions of already placed points. This allows local structures such as functional groups but also non-bonded interactions to be captured. As we directly obtain 3d equilibrium structures of novel molecules, our approach enables further analysis using atomistic simulations after generation. In particular, this work provides the following main contributions:

- We propose the autoregressive neural network *G-SchNet* for the generation of 3d point sets incorporating the constraints of Euclidean space and rotational invariance of the atom distribution as prior knowledge.

- We apply the introduced method to the generation of organic molecules with arbitrary composition based on the QM9 dataset [25–27]. We show that our model yields novel and accurate equilibrium molecules while capturing spatial and structural properties of the training data distribution.

- We demonstrate that the generator can be biased towards complex electronic properties such as the HOMO-LUMO gap – an important property for the design of organic solar cells.

- We introduce datasets containing novel, generated molecules not present in the QM9 dataset. These molecules are verified and optimized at the same level of theory used for QM9[1].

2   Related work

Existing generative models for 3d structures are usually concerned with volumetric objects represented as point cloud data [28, 29]. Here, the density of points characterizes shapes of the target structures. This involves a large amount of points where the exact placement of individual points is less important. Furthermore, these architectures are not designed to capture rotational invariances of the modeled objects. However, the accurate relative placement of points is the main requirement for the generation of molecular structures.

Other generative models for molecules resort to bond-based representations instead of spatial structures. They use text-processing architectures on SMILES [30] strings (e.g. [18, 31–39]) or graph deep neural networks on molecular graphs (e.g. [21–24, 40, 41]). Another approach is to translate generated graphs to 3d structures [42]. Although these models allow to generate novel molecules, ignoring the spatial component of molecules during generation discards valuable information connected to possible target properties.

Gebauer et al. [43] have previously proposed a similar factorization to generate 3d isomers given a fixed composition of atoms. The G-SchNet architecture is an extension of this idea that can deal with arbitrary compositions of atoms and introduces auxiliary tokens which drastically increase the scalability of the model and the robustness of the approximated distribution.

3   Symmetry-adapted factorization of point set distributions

We aim to factorize a distribution over point sets that enables autoregressive generation where new points are sampled step by step. Here, the conditional distributions for each step should be symmetry-adapted to the previously sampled points. This includes invariance to permutation, equivariance to rotation and translation[2] as well as the ability to capture local symmetries, e.g. local rotations of functional groups.

A 3d point set consists of a variable number n of positions $\mathbf{R}_{\leq n} = (\mathbf{r}_1, ..., \mathbf{r}_n) \in \mathbb{R}^{n \times 3}$. Each of these is associated with a type $\mathbf{Z}_{\leq n} = (Z_1, ..., Z_n)$. When generating molecules, these types

---

[1]The datasets are publicly available at `www.quantum-machine.org`.

[2]If we consider a point set *invariant* to rotation and translation, the conditional distributions have to be *equivariant* to fulfill this property.

correspond to chemical elements. We define the distribution over a collection of point sets as the joint distribution of positions and types, $p(\mathbf{R}_{\leq n}, \mathbf{Z}_{\leq n})$. Instead of learning the usually intractable joint distribution directly, we factorize it using conditional probabilities

$$p(\mathbf{R}_{\leq n}, \mathbf{Z}_{\leq n}) = \prod_{i=1}^{n-1} \left[p\left(\mathbf{r}_{i+1}, Z_{i+1} | \mathbf{R}_{\leq i}, \mathbf{Z}_{\leq i}, c_i\right) \right] \cdot p(Z_1) \cdot p(stop | \mathbf{R}_{\leq n}, \mathbf{Z}_{\leq n}) \quad (1)$$

analog to autoregressive graph generative networks [21, 23] or PixelCNN [44]. Here $p(Z_1)$ is the distribution over the type associated with the first point in the set. Note that the first position can be determined arbitrarily due to translational invariance. In order to allow for differently sized point sets, $p(stop | \mathbf{R}_{\leq n}, \mathbf{Z}_{\leq n})$ determines the probability of stopping the generation process.

To make our approach scalable, we condition each step on a focus point $c_i \in \{(\mathbf{r}_1, Z_1), ..., (\mathbf{r}_i, Z_i)\}$ which can be selected randomly and restricts the potential position and type of the next point. We require that the new point is in the neighborhood of c_i which may be defined by a fixed radial cutoff. This effectively limits the potential space of new positions to a region of fixed size around c_i that does not grow with the number of preceding points. Beyond that, we rewrite the joint distribution of type and position such that the probability of the next position $\mathbf{r}_{i+1}$ depends on its associated type Z_{i+1},

$$p\left(\mathbf{r}_{i+1}, Z_{i+1} | \mathbf{R}_{\leq i}, \mathbf{Z}_{\leq i}, c_i\right) = p(\mathbf{r}_{i+1} | Z_{i+1}, \mathbf{R}_{\leq i}, \mathbf{Z}_{\leq i}, c_i)\, p(Z_{i+1} | \mathbf{R}_{\leq i}, \mathbf{Z}_{\leq i}, c_i) \quad (2)$$

which allows us to sample the type first and the position afterwards at each generation step.

While the discrete distribution of types is rotationally invariant, the continuous distribution over positions transforms with already placed points. Instead of learning the 3d distribution of absolute positions directly, we construct it from pairwise distances to preceding points which guarantees the equivariance properties:

$$p(\mathbf{r}_{i+1} | \mathbf{R}_{\leq i}, \mathbf{Z}_{\leq i+1}, c_i) = \frac{1}{\alpha} \prod_{j=1}^{i} p(d_{(i+1)j} | \mathbf{D}_{\leq i}, \mathbf{Z}_{\leq i+1}, c_i). \quad (3)$$

Here α is the normalization constant, $d_{(i+1)j} = ||\mathbf{r}_{i+1} - \mathbf{r}_j||_2$ is the distance between the new position and a preceding point and $\mathbf{D}_{\leq i}$ are pairwise distances between positions $\mathbf{R}_{\leq i}$. As the probability of $\mathbf{r}_{i+1}$ is given in these symmetry-adapted coordinates, the predicted probability is independent of translation and rotation. Transformed to Cartesian coordinates, this leads to rotationally equivariant probabilities of positions in 3d space according to Eq. 3. Since we employ a focus point, the space to be considered at each step remains small regardless of the size of the generated point set.

4 G-SchNet

4.1 Neural network architecture

In order to apply the proposed generative framework to molecules, we adopt the SchNet architecture originally developed to predict quantum-chemical properties of atomistic systems [5, 13]. It allows to extract atom-wise features that are invariant to rotation and translation as well as the order of atoms. The main building blocks are an embedding layer, mapping atom types to initial feature vectors, and interaction blocks that update the features according to the atomic environments using continuous-filter convolutions. We use the atom representation as a starting point to predict the necessary probability distributions outlined above. Fig. 1 gives an overview of the proposed G-SchNet architecture and illustrates one step of the previously described generation process for a molecule. The first column shows how SchNet extracts atom-wise features from previously placed atoms. For our experiments, we use nine interaction blocks and 128 atom features.

In order to predict the probability of the next type (Eq. 2), we reuse the embedding layer of the feature extraction to embed the chemical elements as well as the stop token, which is treated as a separate atom type. The feature vectors are multiplied entry-wise with each of the embeddings before being processed by five atom-wise dense layers with shifted softplus non-linearity[3] and a softmax. We obtain the final distribution as follows:

$$p(Z_{i+1} | \mathbf{R}_{\leq i}, \mathbf{Z}_{\leq i}, c_i) = \frac{1}{\beta} \prod_{j=1}^{i} p\left(Z_{i+1} | \mathbf{x}_j\right). \quad (4)$$

---

[3] as used in SchNet [13]: $\mathrm{ssp}(x) = \ln\left(0.5e^x + 0.5\right)$
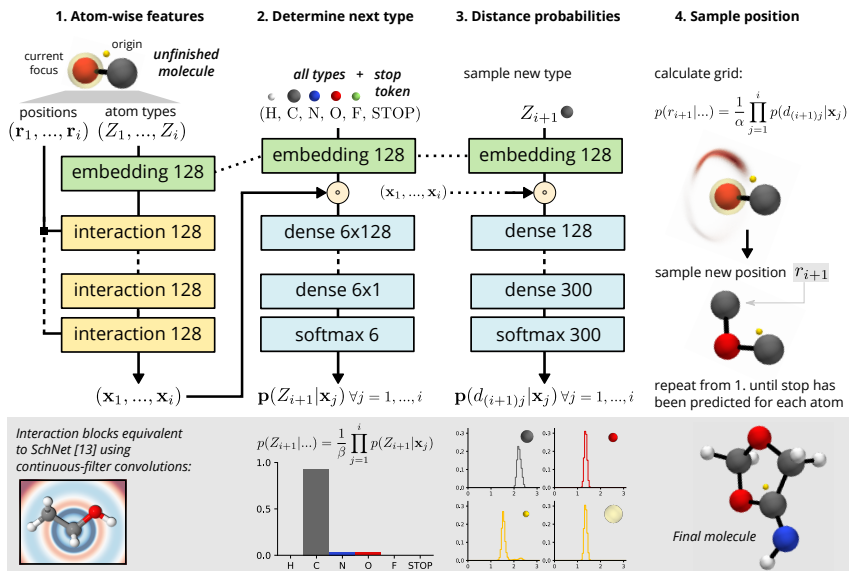
Figure 1: Scheme of G-SchNet architecture with an exemplary generation step where an atom is sampled given two already placed atoms and two auxiliary tokens: the focus and the origin. The complete molecule obtained by repeating the depicted procedure is shown in the lower right. Network layers are presented as blocks where the shape of the output is given. The embedding layers share weights and the ⊙-operator represents element-wise multiplication of feature vectors. All molecules shown in figures have been rendered with the 3d visualization package Mayavi [45].

The third column of Fig. 1 shows how the distance predictions are obtained in a similar fashion where the atom features of the sampled type from column 2 are reused. A fully-connected network yields a distribution over distances for every input atom, which is discretized on a 1d grid. For our experiments, we covered distances up to 15 Å with a bin size of approximately 0.05 Å. The final probability distribution of positions is obtained as described in Eq. 3. We use an additional temperature parameter that allows controlling the randomness during generation. The exact equation and more details on all layers of the architecture are provided in the supplement.

4.2 Auxiliary tokens

We introduce two auxiliary tokens which are fed into the network and yield predictions like regular atoms of a molecule. This is illustrated in Fig. 2, where we show the distribution over positions of a new carbon atom given two already placed carbon atoms with and without the placement of additional tokens. Suppose the most probable position is next to either one of the already placed atoms, but *not* between them. Then, without tokens, both carbon atoms predict the same distance distribution with two peaks: at a distance of one bond length and at the distance of two bond lengths to cover both likely positions of the next atom. This leads to high placement probabilities between the atoms as an artifact of the molecular symmetry. This issue can be resolved by the earlier introduced focus point c_i. Beyond localizing the prediction for scalability, the focus token breaks the symmetry as the new atom is supposed to be placed in its neighborhood. Thus, the generation process is guided to predict only one of the two options and the grid distribution is free of phantom positions (Fig. 2, middle). A second auxiliary token marks the origin around which the molecule grows. The origin token encodes global geometry information, as it is located at the center of mass of a molecule during training. In contrast to the focus token, which is placed on the currently focused atom for each placement step, the position of the origin token remains fixed during generation. The illustration on the right-hand side shows how the distribution over positions is further narrowed by the origin token.

4.3 Training

For training, we split each molecule in the training set into a trajectory of single atom placement steps. We sample a random atom placement trajectory in each training epoch. This procedure is described
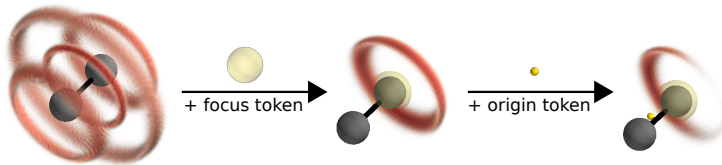
4

Figure 2: Effect of the introduced tokens on the distribution over positions of a new carbon atom.

in detail in the supplement. The next type at each step i can then be one-hot encoded by $\mathbf{q}_{i+1}^{\text{type}}$. To obtain the distance labels $\mathbf{q}_{(i+1)j}^{\text{dist}}$., the distances $d_{(i+1)j}$ between the preceding atoms and the one to be placed are expanded using Gaussians located at the respective bins of 1d distance grid described above. The loss at step i is given by the cross-entropy between labels $\mathbf{Q}_i$ and predictions $\mathbf{P}_i$

$$H\left(\mathbf{Q}_i, \mathbf{P}_i\right) = \underbrace{-\sum_{k=1}^{n_{\text{types}}} \left[\mathbf{q}_{i+1}^{\text{type}}\right]_k \cdot \log\left[\mathbf{p}_{i+1}^{\text{type}}\right]_k}_{\text{cross-entropy of types}} \underbrace{-\frac{1}{i}\sum_{j=1}^{i}\sum_{l=1}^{n_{\text{bins}}} \left[\mathbf{q}_{(i+1)j}^{\text{dist}}\right]_l \cdot \log\left[\mathbf{p}_{(i+1)j}^{\text{dist}}\right]_l}_{\text{average cross-entropy of distances}},$$

where $\mathbf{p}_{i+1}^{\text{type}} = \frac{1}{\beta}\prod_{j=1}^{i}\mathbf{p}(Z_{i+1}|\mathbf{x}_j)$ with normalizing constant β is the predicted type distribution and $\mathbf{p}_{(i+1)j}^{\text{dist}} = \mathbf{p}(d_{(i+1)j}|\mathbf{x}_j)$ is the predicted distance distribution. For further details on the training procedure, please refer to the supplement.

5 Experiments and results

We train G-SchNet on a randomly selected subset of 50k molecules from the QM9 benchmark dataset [25–27] consisting of ~134k organic molecules with up to nine heavy atoms from carbon, nitrogen, oxygen, and fluorine. We use 5k molecules for validation and early stopping while the remaining data is used as a test set. All models are trained with the ADAM optimizer [46].

After training, we generate 20k molecules. In our analysis, we filter generated molecules for *validity*, i.e. we only consider structures that have no disconnected parts and where the number of bonds is equal to the valence of the respective atom type for all atoms. We use Open Babel [47] to obtain (kekulized) bond orders of generated structures for the valency check. In our experiments, about 77% of generated molecules are valid. For the statistics presented in the following, we also filter non-unique structures removing approximately 10%. A step-by-step description of the generation process and details on the subsequent matching of molecules are given in the supplement.

5.1 Accuracy of generated molecules

As G-SchNet is trained on equilibrium structures, we assess how close the generated molecules are to true equilibrium configurations. To this end, we relax the valid and unique molecules at the same level of theory that was used for the QM9 dataset (B3LYP/6-31G(2df,p) [48–51] with the Gaussian version of B3LYP) using the Orca quantum chemistry package [52]. Afterwards, we calculate the root-mean-square deviation (RMSD) of atomic positions between generated and ground-truth geometries. The relaxation process is shown in Fig. 3a for an example molecule.

Fig. 3b shows examples of *unseen* generated molecules that are not in the training set with low, average, and high RMSD (left to right). The molecules in the middle column with an RMSD around 0.2 Å closely match the equilibrium configuration. Even the examples with high RMSD do not change excessively. In fact, most of the relaxed molecules are close to equilibrium as the error distribution shown in Fig. 3c reveals. It shows boxplots for molecules which resemble training data (*in train*), unseen test data (*in test*), and *novel* molecules that cannot be found in QM9. We observe that unseen test structures are generated as accurately as training examples, however the RMSDs are slightly higher for the novel geometries. This is most likely caused by large molecules with more than nine heavy atoms, which make up almost one-third of the novel structures. However, the RMSDs are still sufficiently small to conclude that novel molecules are generated close to equilibrium. The last boxplot shows RMSDs considering only heavy atoms of all generated molecules. It is significantly
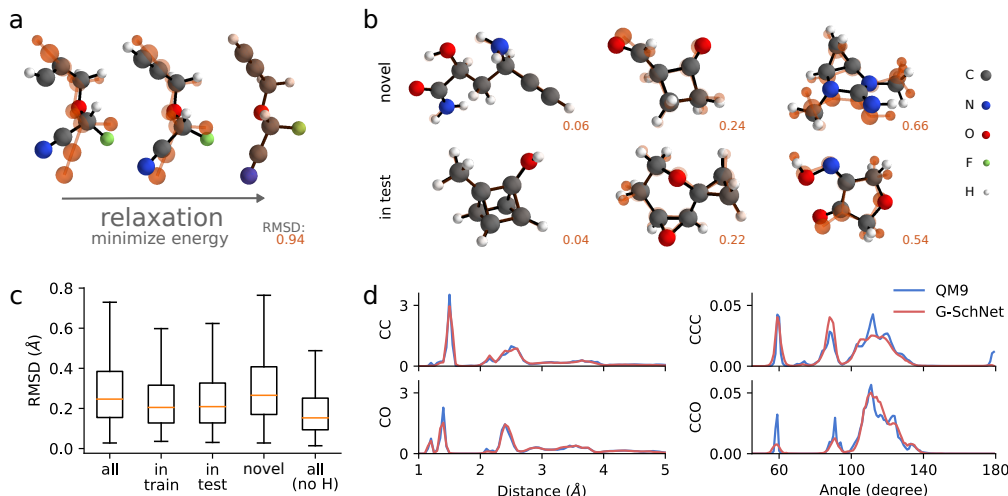
Figure 3: (a) Scheme showing the relaxation of a molecule along with the resulting root-mean-square deviation (RMSD) between its atomic positions before and after relaxation. The equilibrium structure is indicated by orange shadows. (b) Examples of generated molecules which are novel (top) or resemble test data (bottom). The RMSD increases from left to right. As before, the orange shadows indicate the corresponding equilibrium configurations. (c) Boxplots of the RMSD for generated molecules divided into different sets. The boxes extend from the lower to the upper quartile and the whiskers reach up to 1.5 times the interquartile range. (d) Radial distribution functions for carbon-carbon and carbon-oxygen atom pairs (left) and angular distribution functions for bonded carbon-carbon-carbon and carbon-carbon-oxygen chains (right) in the training data and in generated molecules.

lower than the RMSDs of all atoms, which suggests that the necessary relaxation is to a large part concerned with the rearrangement of hydrogens.

To judge how well our model fits the distribution of QM9, we compare the radial distribution functions of the two most common heavy atom pairs (carbon-carbon and carbon-oxygen) in Fig. 3d. They align well as our training objective directly penalizes deviations in the prediction of distances. Beyond that we plotted angular distribution functions which are not directly related to the optimization objective. They show the distribution over angles of bonded carbon-carbon-carbon and carbon-carbon-oxygen chains. While our model produces smoother peaks here, they also align well indicating that the angles between atoms are accurately reproduced. This is remarkable insofar that only distances have been used to calculate the probabilities of positions.

5.2 Structural properties of generated molecules

As proposed by Liu et al. [21], we calculate the atom, bond, and ring counts of the generated molecules to see if G-SchNet captures these structural statistics of the training data. We use the same sets of valid and unique molecules as for the spatial analysis above. The atom and bond counts are determined with Open Babel. For ring counts, the canonical SMILES representation of generated molecules is read with RDKit [53] and the symmetrized smallest set of smallest rings is computed.

We compute the statistics for QM9, molecules generated with G-SchNet as well as molecules obtained by the CGVAE model of Liu et al. [21], a graph-based generative model for molecules (structures provided by the authors). The CGVAE model incorporates valency constraints in its generation process, allowing it to sample only valid molecules. This cannot easily be transferred to 3d models since there is no direct correspondence to bond order for some spatial arrangements. This is because the associated bond order may depend on the not yet sampled atomic environment. However, the valency check is a comparably fast post-processing step that allows us to remove invalid molecules at low cost. Furthermore, in contrast to our approach, CGVAE does not actively place hydrogen atoms and uses all available structures from QM9 for training.
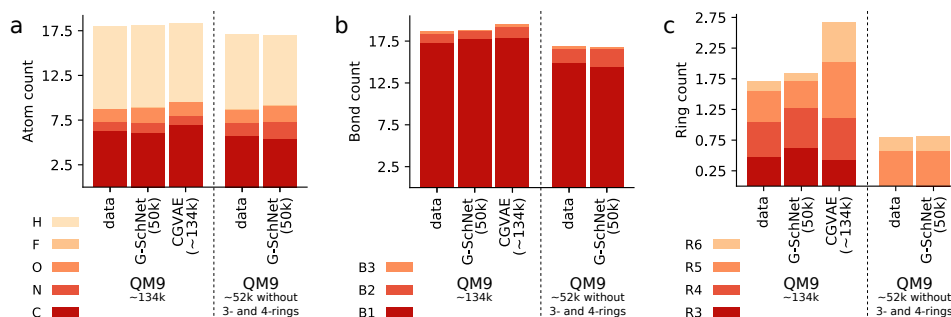
Figure 4: Bar plots showing the average numbers of atoms, bonds, and rings per molecule. Left-hand side of plots: QM9 dataset, generated molecules of G-SchNet, and generated molecules of CGVAE [21]. Right-hand side of plots: QM9 subset without 3- and 4-rings and generated molecules of G-SchNet trained on that subset. Numbers below model names represent the amount of training data used. B1, B2, and B3 correspond to single, double, and triple bonds. R3, R4, R5, R6 are rings of size 3 to 6.

Fig. 4 shows the statistics for the QM9 dataset (first bars of each plot), G-SchNet (second bars), as well as CGVAE (third bars). We observe that G-SchNet captures the atom and bond count of the dataset accurately. CGVAE shows comparable results with a slightly increased heavy atom and bond count. The ring count of CGVAE is significantly higher than in the training data distribution, especially concerning five- and six-membered rings. The average number of rings per molecule is better reproduced by G-SchNet, but here we observe an increase in three- and four-membered rings.

As illustrated in Fig. 2, approximating positions in terms of distances can lead to symmetry artifacts at wrong bond angles if no auxiliary tokens are used. In order to rule out that the increase of three- and four-membered rings is an artifact of our model, we filter all molecules containing such rings from QM9. We train G-SchNet on the resulting subset of approximately 52k structures, where we use 50k molecules for training and the remaining 2k for validation. If our model was biased towards small rings, we expect to find them in the generated molecules regardless. We generate 20k molecules with this model and keep structures that are both valid (77% of all generated) and unique (89% of all valid). The right-hand side of each bar plot in Fig. 4 shows the corresponding results, comparing the statistics of the QM9 subset (fourth bars) with those of the generated molecules (fifth bars). Again, the atom and bond count statistics are accurately matched. More importantly, the generated molecules almost perfectly resemble the ring count of the training data, exhibiting no three- or four-membered rings. This hints at the possibility to bias the predicted distribution by constraining the training data.

5.3 Targeted discovery of molecules with small HOMO-LUMO gaps

Finally, we aim to guide the generator towards a desired value range of a complex electronic property, namely a small HOMO-LUMO gap. This property describes the energy difference between the highest occupied and lowest unoccupied molecular orbital and is an important measure for designing organic semiconductors, an essential component of e.g. organic solar cells and OLED displays. To this end, we filter all ~3.8k equilibrium structures from QM9 with a HOMO-LUMO gap smaller than 4.5 eV. Then we fine-tune the G-SchNet model previously trained on all 50k molecules of the training set on this small subset. We use 3.3k structures for training and the remaining 0.5k for validation and early stopping. The optimizer is reset to its initial learning rate and the optimization protocol for biasing is the same as for regular training.

We generate 20k molecules with this fine-tuned model. The validity decreases to 69% of which 74% are unique. This drop in validity and uniqueness is expected as the amount of generated molecules is significantly higher than the amount of available training structures with the target property. We relax the valid and unique molecules using the same level of theory as described in section 5.1.

Fig. 5a shows histograms of calculated HOMO-LUMO gaps for generated molecules before and after relaxation. There is a significant shift to smaller values compared to the histograms of HOMO-LUMO gaps of the full QM9 dataset. Before biasing, only 7% of all validly generated molecules had a HOMO-LUMO gap smaller than 5.0 eV. The fine-tuning raises this number to 43%. We observe in
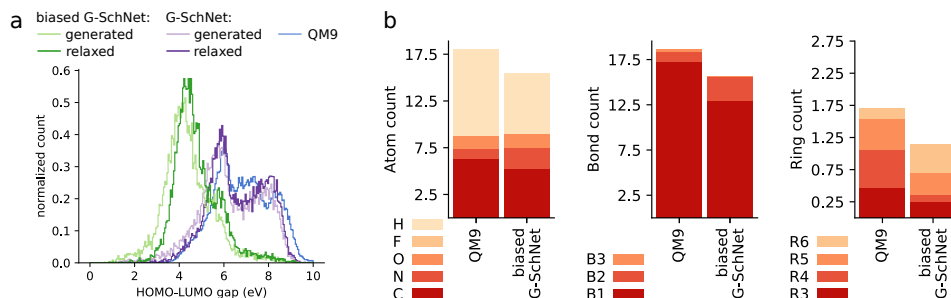
Figure 5: (a) Histograms of calculated HOMO-LUMO gaps for molecules generated with the biased G-SchNet (green curves), G-SchNet before biasing (purple curves), and for the QM9 dataset. b): Bar plots showing the average numbers of atoms, bonds, and rings per molecule for QM9 and for molecules generated with the biased G-SchNet. B1, B2, and B3 correspond to single, double, and triple bonds. R3, R4, R5, R6 are rings of size 3 to 6.

Fig. 5b how the structural statistics of molecules generated with the biased G-SchNet deviate from the statistics of all molecules in QM9. Compared to the original dataset, the new molecules exhibit an increased number of double bonds in addition to a tendency towards forming six-membered cycles. These features indicate the presence of conjugated systems of alternating single and double bonds, as well as aromatic rings, both of which are important motifs in organic semiconductors.

6 Conclusions

We have developed the autoregressive G-SchNet for the symmetry-adapted generation of rotationally invariant 3d point sets. In contrast to graph generative networks and previous work on point cloud generation, our model both incorporates the constraints of euclidean space and the spatial invariances of the targeted geometries. This is achieved by determining the next position using distances to previously placed points, resulting in an equivariant conditional probability distribution. Beyond these invariances, local symmetries are captured by means of point-wise features from the SchNet architecture for atomistic predictions.

We have shown that G-SchNet is able to generate accurate 3d equilibrium structures of organic molecules. The generated molecules closely resemble both spatial and structural distributions of the training data while our model yields 79% unseen molecules. On this basis, we have introduced a dataset of >9k novel organic molecules that are not contained in QM9. Finally, we fine-tuned G-SchNet on a small subset of molecules to bias the distribution of the generator towards molecules with small HOMO-LUMO gap – an important electronic property for the design of organic solar cells. We have generated a dataset of >3.6k novel structures with the desired property, demonstrating a promising strategy for targeted discovery of molecules.

Directions for future work include scaling to larger systems, a direct conditioning on chemical properties as well as including periodic boundary conditions. We expect G-SchNet to be widely applicable, for instance, to the guided exploration of chemical space or crystal structure prediction.

Acknowledgments

References

[1] J. Behler and M. Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98(14):146401, 2007.

[2] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.*, 104(13):136403, 2010.

[3] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller. Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv.*, 3(5):e1603015, 2017.

[4] M. Gastegger, J. Behler, and P. Marquetand. Machine learning molecular dynamics for the simulation of infrared spectra. *Chem. Sci.*, 8(10):6924–6935, 2017.

[5] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. SchNet – A deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148 (24):241722, 2018.

[6] S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko. Towards exact molecular dynamics simulations with machine-learned force fields. *Nature communications*, 9(1):3887, 2018.

[7] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. Von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.*, 108(5):058301, 2012.

[8] M. Eickenberg, G. Exarchakis, M. Hirn, and S. Mallat. Solid harmonic wavelet scattering: Predicting quantum molecular energy from invariant descriptors of 3d electronic densities. In *Advances in Neural Information Processing Systems 30*, pages 6543–6552. Curran Associates, Inc., 2017.

[9] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature Communications*, 8:13890, 2017.

[10] 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.

[11] D. Jha, L. Ward, A. Paul, W.-k. Liao, A. Choudhary, C. Wolverton, and A. Agrawal. Elemnet: Deep learning the chemistry of materials from only elemental composition. *Scientific reports*, 8 (1):17593, 2018.

[12] J. S. Smith, O. Isayev, and A. E. Roitberg. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chem. Sci.*, 8(4):3192–3203, 2017.

[13] K. Schütt, P.-J. Kindermans, H. E. S. Felix, S. Chmiela, A. Tkatchenko, and K.-R. Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems*, pages 992–1002, 2017.

[14] J. S. Smith, O. Isayev, and A. E. Roitberg. Ani-1, a data set of 20 million calculated off-equilibrium conformations for organic molecules. *Scientific data*, 4:170193, 2017.

[15] E. V. Podryabinkin, E. V. Tikhonov, A. V. Shapeev, and A. R. Oganov. Accelerating crystal structure prediction by machine-learning interatomic potentials with active learning. *Physical Review B*, 99(6):064114, 2019.

[16] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *NIPS*, pages 2224–2232, 2015.

[17] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande. Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072*, 2015.

[18] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.*, 2016.

[19] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley. Molecular graph convolutions: moving beyond fingerprints. *J. Comput. Aided Mol. Des.*, 30(8):595–608, 2016.

[20] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2): 513–530, 2018.

[21] Q. Liu, M. Allamanis, M. Brockschmidt, and A. Gaunt. Constrained graph variational autoencoders for molecule design. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7795–7804. Curran Associates, Inc., 2018.

[22] W. Jin, R. Barzilay, and T. Jaakkola. Junction tree variational autoencoder for molecular graph generation. *arXiv preprint arXiv:1802.04364*, 2018.

[23] J. You, B. Liu, Z. Ying, V. Pande, and J. Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6410–6421. Curran Associates, Inc., 2018.

[24] Y. Li, L. Zhang, and Z. Liu. Multi-objective de novo drug design with conditional graph generative model. *Journal of cheminformatics*, 10(1):33, 2018.

[25] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data*, 1, 2014.

[26] J.-L. Reymond. The chemical space project. *Acc. Chem. Res.*, 48(3):722–730, 2015.

[27] 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. *Journal of chemical information and modeling*, 52(11):2864–2875, 2012.

[28] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 82–90. Curran Associates, Inc., 2016.

[29] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas. Learning representations and generative models for 3d point clouds. In *International Conference on Machine Learning*, pages 40–49, 2018.

[30] D. Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28 (1):31–36, 1988.

[31] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato. Grammar variational autoencoder. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1945–1954. JMLR. org, 2017.

[32] G. L. Guimaraes, B. Sanchez-Lengeling, P. L. C. Farias, and A. Aspuru-Guzik. Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. *arXiv preprint arXiv:1705.10843*, 2017.

[33] H. Dai, Y. Tian, B. Dai, S. Skiena, and L. Song. Syntax-directed variational autoencoder for structured data. In *International Conference on Learning Representations*, 2018.

[34] D. Janz, J. van der Westhuizen, B. Paige, M. Kusner, and J. M. H. Lobato. Learning a generative model for validity in complex discrete structures. In *International Conference on Learning Representations*, 2018.

[35] M. H. S. Segler, T. Kogej, C. Tyrchan, and M. P. Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4(1):120–131, 2018.

[36] M. Popova, O. Isayev, and A. Tropsha. Deep reinforcement learning for de novo drug design. *Science Advances*, 4(7), 2018.

[37] J. Lim, S. Ryu, J. W. Kim, and W. Y. Kim. Molecular generative model based on conditional variational autoencoder for de novo molecular design. In *J. Cheminformatics*, 2018.

[38] T. Blaschke, M. Olivecrona, O. Engkvist, J. Bajorath, and H. Chen. Application of generative autoencoder in de novo molecular design. *Molecular Informatics*, 37(1-2):1700123, 2017.

[39] A. Gupta, A. T. Müller, B. J. H. Huisman, J. A. Fuchs, P. Schneider, and G. Schneider. Generative recurrent networks for de novo drug design. *Molecular Informatics*, 37(1-2):1700111, 2017.

[40] Y. Li, O. Vinyals, C. Dyer, R. Pascanu, and P. Battaglia. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018.

[41] W. Jin, K. Yang, R. Barzilay, and T. Jaakkola. Learning multimodal graph-to-graph translation for molecular optimization. In *International Conference on Learning Representations*, 2019.

[42] E. Mansimov, O. Mahmood, S. Kang, and K. Cho. Molecular geometry prediction using a deep generative graph neural network. *arXiv preprint arXiv:1904.00314*, 2019.

[43] N. W. Gebauer, M. Gastegger, and K. T. Schütt. Generating equilibrium molecules with deep neural networks. *arXiv preprint arXiv:1810.11347*, 2018.

[44] A. van den Oord, N. Kalchbrenner, L. Espeholt, K. Kavukcuoglu, O. Vinyals, and A. Graves. Conditional image generation with PixelCNN decoders. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4790–4798. Curran Associates, Inc., 2016.

[45] P. Ramachandran and G. Varoquaux. Mayavi: 3D Visualization of Scientific Data. *Computing in Science & Engineering*, 13(2):40–51, 2011. ISSN 1521-9615.

[46] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[47] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison. Open babel: An open chemical toolbox. *Journal of Cheminformatics*, 3(1):33, Oct 2011.

[48] A. D. Becke. Density-functional thermochemistry. III. the role of exact exchange. *J. Chem. Phys.*, 98:5648–5652, 1993.

[49] C. Lee, W. Yang, and R. G. Parr. LYP correlation: Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B*, 37(2):785, 1988.

[50] S. H. Vosko, L. Wilk, and M. Nusair. Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis. *Can. J. Phys.*, 58(8):1200–1211, 1980.

[51] P. J. Stephens, F. J. Devlin, C. F. Chabalowski, and M. J. Frisch. Ab initio calculation of vibrational absorption and circular dichroism spectra using density functional force fields. *J. Phys. Chem.*, 98(45):11623–11627, 1994.

[52] F. Neese. The ORCA program system. *WIREs Comput. Mol. Sci.*, 2(1):73–78, 2012.

[53] RDKit, online. RDKit: Open-source cheminformatics. `http://www.rdkit.org`. [Online; accessed 23-May-2019].

[54] J. P. Perdew, K. Burke, and M. Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865–3868, 1996.

[55] F. Weigend and R. Ahlrichs. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.*, 7(18):3297–3305, 2005.

[56] K. Eichkorn, O. Treutler, H. Öhm, M. Häser, and R. Ahlrichs. Auxiliary basis sets to approximate Coulomb potentials. *Chem. Phys. Lett.*, 240(4):283–290, 1995.

[57] O. Vahtras, J. Almlöf, and M. W. Feyereisen. Integral approximations for LCAO-SCF calculations. *Chem. Phys. Lett.*, 213(5-6):514–518, 1993.

[58] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg. A consistent and accurate *ab initio* parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu. *J. Chem. Phys.*, 132(15):154104, 2010.

Supplementary material: Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules

Architecture

Here we summarize the exact settings used in all layers of our neural network architecture. The structure of the interaction blocks can be found in Schütt et al. [13]. The number of atom features was set to 128 and used in all atom-wise dense layers of the interaction block and filter-generating layers. Distances are expanded in the filter-generating layers using 25 Gaussians with equally spaced centers $0\text{Å} \leq \mu \leq 10\text{Å}$. Overall, we use nine interaction blocks for feature extraction. We re-use one embedding layer with 128 features at all steps depicted in Fig. 1 of the paper. The output network for type predictions consists of five atom-wise dense layers with shifted-softplus non-linearity and 128, 96, 64, 32, and 1 atom features, respectively. The output network for distance predictions also consists of five atom-wise dense layers with shifted-softplus non-linearity and 128, 171, 214, 257, and 300 atom features. Both output networks contain a final softmax layer.

Training details

The neural networks were trained with stochastic gradient descent using the ADAM optimizer [46]. We start with a learning rate of 10^{-4} which is reduced using a decay factor of 0.5 after 10 epochs without improvement of the validation loss. The training is stopped at lr $\leq 10^{-6}$. Afterwards, the model with lowest validation error is selected for generation.

While the atom type labels $\mathbf{q}^{\text{type}}_{(i+1)j}$ can be directly obtained from the training data, the labels for distance distributions $\mathbf{q}^{\text{dist}}_{(i+1)j}$ are obtained using:

$$\left[\mathbf{q}^{\text{dist}}_{(i+1)j}\right]_l = \frac{\exp(-\frac{1}{\gamma}(d_{(i+1)j} - \mu_l)^2)}{\sum_{l=1}^{300} \exp(-\frac{1}{\gamma}(d_{(i+1)j} - \mu_l)^2)}.$$

The width of the Gaussians can be tuned with the γ parameter, which we set to 10% of the bin size in our experiments, resulting in very peaky, uni-modal label vectors.

Controlling randomness with the temperature parameter

In order to control the randomness during generation, we do not directly implement Eq. 3 but include a temperature parameter T:

$$p(\mathbf{r}_{i+1}|\mathbf{R}_{\leq i}, \mathbf{Z}_{\leq i+1}, c_i) = \frac{1}{\alpha} \exp\left(\frac{\sum_{j=1}^{i} \log p(d_{(i+1)j}|\mathbf{x}_j)}{T}\right). \tag{5}$$

Increasing T will smoothen the grid distribution, effectively increasing randomness, whereas small values lead to a peaky distribution and thus less randomness. For all experiments, we chose a fixed temperature of $T = 0.1$ according to the following procedure.

We used the G-SchNet model trained on 50k equilibrium structures from QM9 [25–27] and generated 20k molecules for each $T \in \{2, 1, 0.1, 0.01, 0.001\}$. From each set, 1k valid and unique molecules were randomly chosen, where 800 resembled test structures, 100 resembled training structures, and the remaining 100 were novel structures with more than 9 heavy atoms. We relaxed the five sets of 1k molecules at the PBE/def2-SVP level of theory[54, 55] using the Orca program, employing the resolution of identity (RI) approximation[56, 57] and Grimme D3 dispersion correction with Becke-Johnson damping.[58] The root-mean-square deviation (RMSD) between atomic positions before and after relaxation was measured. A smaller RMSD means that the generated structures are closer to a true equilibrium configuration.

In Fig. 6 we show boxplots of the RMSD for the five different temperatures T. We see that the smallest median values and interquartile ranges are observed for values of T smaller than 1. Since increasing T corresponds to increasing randomness during sampling, the increase in the RMSD for $T = 1$ and $T = 2$ is expected. However, decreasing T to smaller values than 0.1 does not lead to smaller RMSDs. Generally, we expect the number of unique sampless to decrease as T gets too small. Therefore, for the other experiments, we chose the highest value for T that still produces structures close to equilibrium, namely $T = 0.1$.
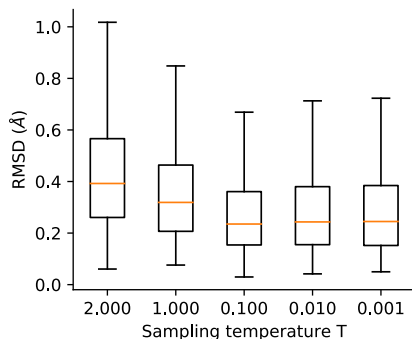
Figure 6: The boxplots show the RMSD of atomic positions between generated and relaxed structures for different values of T, where the boxes extend from the lower to the upper quartile values, the red lines indicate the medians, and the whiskers reach up to 1.5 times the interquartile ranges. Outliers are not shown.

Sampling generation traces for training

The generation traces start with the focus and origin tokens set to the same position, which is the center of mass of the respective training molecule. The first new type and position are taken from the atom closest to the center of mass. At each subsequent step, we randomly select one of the already placed non-token atoms as focus point (which is only a single choice for the second step). The new position and type are then taken from the neighbor which is closest to the origin token, where neighbors are all unplaced atoms of the training molecule that are connected to the focus by a bond. If no unplaced neighbors are left, the new type is set to the stop token and the focused atom is marked as finished, i.e. it cannot be chosen as focus anymore. For the next step, another already placed (unfinished) atom is randomly chosen as focus and the procedure is repeated until all atoms have been placed and marked as finished. After training, the generation process of a molecule will stop as soon as all placed atoms have been predicted to be finished.

Generating molecules

For the first step of molecule generation, the start and origin token are set to the origin of a 3d grid. The grid extends up to 1.7Å into all dimensions with a step-size of 0.05Å. The token are processed by G-SchNet to sample the type of the first non-token atom and to obtain the two predicted distance distributions. The probabilities of the grid cell positions are calculated according to Eq. 5 in order to sample the position of the first atom. Then, at each subsequent step, a random unfinished non-token atom is selected as focus point (which is only a single choice for the second step). The grid is centered on the focus point and G-SchNet predictions are obtained to sample the type and position of the next atom (see Fig. 1 in the paper for an exemplary generation step). If the stop token is predicted as next type, the currently focused atom is marked as finished and no position is sampled. Instead the next generation step is initialized by randomly selecting one of the remaining unfinished atoms as focus point. The generation process stops if no unfinished atoms are left. For our experiments, we also stopped the process if molecules were not finished after placing 35 atoms and discarded these structures as invalid (usually ~1% of generated molecules).

Matching molecules

To remove non-unique structures and identify moleclues that resemble training or test data, we calculate the Tanimoto similarity of path-based fingerprints (FP2) [47] for pairs of molecules with Open Babel. If the similarity is one, we compare the canonical SMILES representations in a second step. If they match, the two molecules are treated as equal. Note that this is a conservative approach as it filters out some spatial isomers which cannot be distinguished with SMILES.

Table 1: Statistics for all of our models and CGVAE [21] (molecules provided by the authors). CGVAE was trained on all molecules in QM9. Our models were trained on 50k randomly selected molecules from QM9. For the second model molecules with three- and four-membered rings were excluded. The third model was fine-tuned on 3k molecules with a HOMO-LUMO gap $\leq$ 4.5 eV. The numbers are percent of all validly generated molecules (which are 77%, 78%, 69% and 100% of 20k generated for the models from left to right). The molecules are categorized according to the checkmarks in the first five columns. Unseen refers to molecules not found in the training data and novel stands for molecules not found in QM9. Gap refers to the HOMO-LUMO gap and was calculated for relaxed structures.

unique	unseen	novel	$\leq$9 heavy	gap $\leq$5eV	G-SchNet 50k	G-SchNet 50k no R3/R4	G-SchNet 50k + 3k gap$\leq$4.5eV	CGVAE ~134k
✓					91.6%	89.4%	73.8%	98.4%
✓	✓				79.4%	68.3%	66.8%	87.4%
✓		✓			63.8%	68.3%	57.5%	87.4%
✓			✓		70.2%	58.2%	57.8%	30.0%
✓	✓		✓		58.0%	37.1%	50.8%	18.9%
✓		✓	✓		42.4%	37.1%	41.5%	18.9%
✓	✓			✓	6.8%	21.2%	43.4%	–