GENERATING REALISTIC 3D MOLECULES WITH AN EQUIVARIANT CONDITIONAL LIKELIHOOD MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

The number of drug-like molecules that could potentially exist is thought to be above 10^{33}, precluding exhaustive computational or experimental screens for molecules with desirable pharmaceutical properties. Machine learning models that can propose novel molecules with specific characteristics are powerful new tools to break through the intractability of searching chemical space. Most of these models generate molecular graphs—representations that describe the topology of covalently bonded atoms in a molecule—because the bonding information in the graphs is required for many downstream applications, such as virtual screening and molecular dynamics simulation. These models, however, do not themselves generate 3D coordinates for the atoms within a molecule (which are also required for these applications), and thus they cannot easily incorporate information about 3D geometry when optimizing molecular properties. In this paper, we present GEN3D, a model that concurrently generates molecular graphs and 3D geometries, and is equivariant to rotations, translations, and atom permutations. The model extends a partially generated molecule by computing a conditional distribution over atom types, bonds, and spatial locations, and then sampling from that distribution to update the molecular graph and geometries, one atom at a time. We found that GEN3D proposes molecules that have much higher rates of chemical validity, and much better atom-distance distributions, than those generated with previous models. In addition, we validated our model's geometric accuracy by forcing it to predict geometries for benchmark molecular graph inputs, and found that it also advances the state of the art on this test. We believe that the advantages that GEN3D provides over other models will enable it to contribute substantially to structure-based drug discovery efforts.

1 INTRODUCTION

Identifying molecules with desirable characteristics is of fundamental importance in many fields, including drug discovery. The astronomically large number of possible drug-like compounds, however, makes an exhaustive experimental or virtual screening intractable (Polishchuk et al., 2013). As a result, a large body of work has used machine learning to explore chemical space and propose molecules with specific characteristics. Most commonly, these works have trained machine learning models to generate molecules as molecular graphs—representations that describe the covalent bonds between atoms—because this chemical bonding information is required for many downstream applications such as virtual screening and molecular dynamics simulation. Many works generate a text-based molecular graph representation called SMILES (Weininger, 1988), which uses a string of characters to describe the connectivity of atoms (Gómez-Bombarelli et al., 2018; Maragakis et al., 2020). Using SMILES representations is convenient because they can be easily generated by the best architectures from the field of natural language processing.

Despite its convenience, limitations arise when proposing molecules as SMILES strings. The SMILES encoding of a given molecule is often not unique, and very similar molecules can have very different SMILES strings; both of these factors make training a model challenging. In addition, some models are also capable of generating invalid SMILES strings (although this can be addressed by imposing constraints on the generative process, as in Maragakis et al. (2020) and Kusner et al. (2017), or by using a representation in which fewer strings are invalid, as in Krenn et al. (2020)).

In response to these challenges, graph neural network (GNN) models have been built that generate atoms and bonds without utilizing a string representation (Kingma & Welling, 2013; Liu et al., 2018; Shi et al., 2020; Jin et al., 2018; 2020).

Whereas the models above are capable of generating a wide range of molecules whose atoms have the correct number and type of bonds, their outputs do not contain any geometric information, and may include molecular graphs for which stable 3D geometries (i.e., low-energy conformations with physically realistic inter-atomic distances) do not exist. Although it is possible that a model could learn to assess the geometric feasibility of a molecular graph without training on any geometric data, a model that is trained explicitly on 3D molecules would likely be better at generating molecular graphs with corresponding low-energy 3D geometries. The lack of geometric information is also problematic because many downstream applications require 3D information. For instance, many drug discovery efforts screen for molecules with high predicted affinity for a target protein pocket using a computational docking process to score molecule poses. It would be extremely valuable if a machine learning model could directly generate such molecules, as this would expedite the screening process. A model that is explicitly trained to recreate the geometric poses of bound molecules, rather than just molecular graphs, has the potential to produce molecules and geometries that are more conducive to binding.

To address the above shortcomings, we created GEN3D: a graph generative model that proposes molecules with 3D coordinates. The model is rotationally and translationally equivariant, providing an inductive bias that exploits the symmetries of chemical space. GEN3D creates molecular graphs through a sequential sampling process, like other graph-based generative models (Shi et al., 2020; Liu et al., 2018), but at each step GEN3D also calculates a likelihood function over positions in 3D space, and uses that function to sample coordinates for each new atom. We show that GEN3D generates novel, chemically valid molecular graphs that have realistic low-energy geometries. In particular, we found that our model outperforms existing graph generators in its ability to create chemically valid molecular graphs, and outperforms existing 3D-generative models in the ability to create realistic geometries. (In Appendix E, we also show that our pretrained model can be tuned to generate novel molecules in geometries that score well in a virtual screening tool.) Finally, we demonstrate the quality of our model by using it to sample geometries for fixed molecular graphs, and show that it achieves state-of-the-art results on the previously established benchmark of Xu et al. (2021b).

2 Related Work

Different models construct molecular graphs using GNNs in a number of different ways. CGVAE, for example, uses a GNN to create VAE latent representations for each node in a molecular graph (Liu et al., 2018). The nodes and edges of the graph are then reconstructed in a sequential decision process guided by another GNN, in which atoms are connected one at a time to a growing molecular graph. Another model, GraphAF, also generates molecular graphs one atom at a time, but it samples molecules using a Gaussian Autoregressive Flow (Shi et al., 2020). Other models, like JT-VAE (Jin et al., 2018) and HierVAE (Jin et al., 2020), generate molecules by sequentially adding multi-atom motifs rather than individual atoms.

Because of their importance in classification tasks on point clouds and 3D molecules, there has been a great deal of interest in machine learning problems involving 3D structured data. When working with such data, it is often desirable that models be invariant or equivariant to rotations or translations of the input data, as these transformations are only an arbitrary change of coordinate systems. Approaches like SchNet (Schütt et al., 2017) and EGNN (Satorras et al., 2021b) have achieved equivariance by using invariant features like pairwise distances as inputs to the model. These approaches have proven effective in many domains, but they are reflection-invariant, so they cannot distinguish between mirror-image isomers. Other approaches, like DimeNet (Klicpera et al., 2020) and SphereNet (Liu et al., 2021), achieve equivariance through message-passing schemes that encode distance and angular information, while Tensor Field Networks (Thomas et al., 2018) and SE(3)-Transformers (Fuchs et al., 2020) process data using a basis of equivariant spherical harmonic functions.

There are several prior works on generative models for 3D molecular data. Two papers from the same group explore generating 3D molecules, both independently and conditionally on protein pock-

ets of interest (Ragoza et al., 2020; Masuda et al., 2020). In these works, a generative model outputs a voxelized grid of atomic densities, which is then converted to a set of atoms and bonds in a secondary optimization step. While efficient to implement using convolutions, the voxelized grid approach imposes a practical limit on the size of generated molecules, and is not equivariant. The process of reconstructing a molecule from a density field also suffers from relatively high rates of chemical and geometric invalidity, and the one-shot nature of the generative process does not allow for valence constraints to be enforced during the generative process. Another model, ASLA, also generates 3D molecules in a voxelized space, using reinforcement learning to identify molecular graphs that are stable and have low-energy geometries (Meldgaard et al., 2021). ASLA, however, can only generate molecules conditioned on specific stoichiometries.

Two other 3D-generative models are closely related to GEN3D: ENF (Satorras et al., 2021a) and G-Schnet (Gebauer et al., 2019). ENF uses an equivariant, EGNN-based normalizing flow to convert random initial atom positions into realistic molecular geometries. This model, however, has only been demonstrated on relatively small molecules, is very expensive to train, and produces molecules with low rates of chemical validity. G-SchNet is an equivariant model that autoregressively generates 3D molecules. It exclusively generates atomic positions, however, and does not generate molecular graphs that provide the chemical bonding information required by many downstream applications. In addition, without explicitly generating bonds, one cannot use simple rules to constrain chemical properties, like atomic valences and ring sizes, during the generative process.

There are also machine learning methods that predict molecular geometries for a given molecular graph. A number of works have approached this problem by predicting inter-atomic distances from the molecular graph, and then using the predictions to generate a 3D geometry in a secondary optimization step (Xu et al., 2021a; Simm & Hernández-Lobato, 2019). Most recently, Xu et al. (2021b) proposed the ConfVAE architecture, which estimates inter-atomic distances and then optimizes molecular geometry using an end-to-end differentiable optimization procedure.

3 Architecture and Training

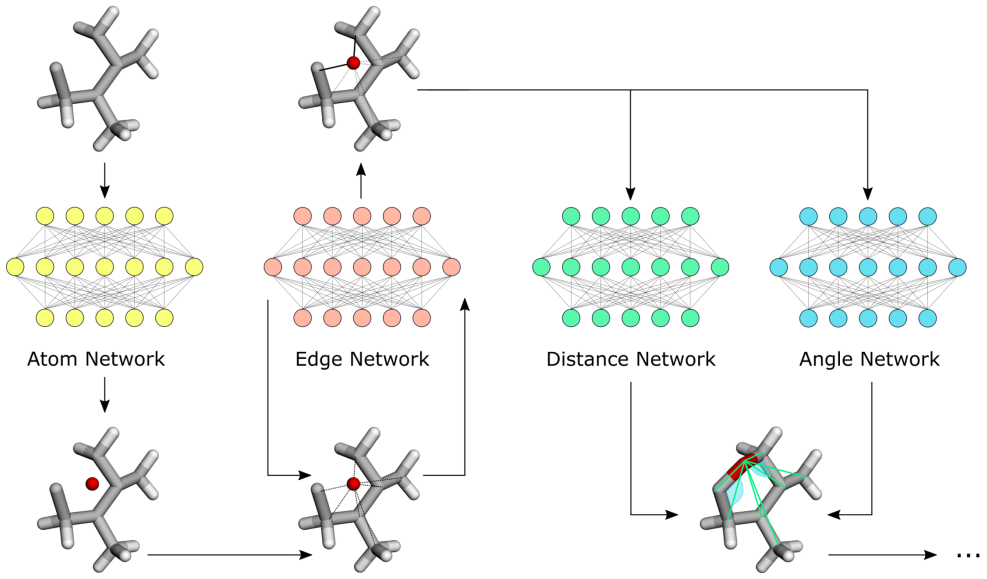

Figure 1: The process by which GEN3D samples a 3D molecule.

GEN3D generates 3D molecules by adding atoms to a partially complete molecular graph, attaching them with new edges, and localizing them in 3D space. This process can be seeded with a single randomly-sampled atom to generate molecules from scratch, or used to extend existing molecular fragments. The architecture of GEN3D consists of four 3D GNNs: the atom network (denoted F_A), the edge network (denoted F_E), the distance network (denoted F_D), and the angle network

(denoted F_θ). Each of these networks is implemented as a 7-layer EGNN with a hidden dimension of 128, as described in Satorras et al. (2021b). The equivariance of the EGNN subnetworks makes the entire generative process equivariant to translations and rotations. We found that adding batch normalization to the atom, distance, and angle networks improved stability (Ioffe & Szegedy, 2015). The EGNNs produce embeddings for each point in the input graph, and the embeddings can be aggregated into a global graph representation using sum-pooling.

GEN3D is an autoregressive model that augments a partially completed molecular graph. In the remainder of this paper, we will denote a partially completed graph with n atoms as $G_n = (V_n, A_n, X_n)$. $V_n \in \mathbb{R}^{n \times d}$ is a list of one-hot encoded atom types (i.e., the different chemical elements appearing in the dataset), and d is the number of possible atom types. $A_n \in \mathbb{R}^{n \times n \times b}$ is an adjacency matrix recording the one-hot encoded bond type between each pair of atoms, with b representing the number of bond types. $X_n \in \mathbb{R}^{n \times 3}$ is a list of atom positions. For the adjacency matrix A_n, we include an extra bond type indicating that the atoms are not chemically bonded (unbonded atoms are still connected in the sense that information can propagate between them during the EGNN computation).

The addition of a new atom proceeds in three steps. First, a new atom type is selected as follows:

$$H_n = \text{SumPool}(F_A(G_n))$$
$$a_{n+1} \sim \text{Categorical}(\text{Softmax}(D_A(H_n))),$$

where a_{n+1} is the type of the new atom, and D_A is a neural network that decodes the EGNN graph embedding into a set of softmax probabilities. We implement D_A as a 3-layer MLP. Note that, in addition to all of the atom species in the training set, we allow a_{n+1} to take on an extra "stop token" value. If this value is generated, the molecule is complete, and generation terminates.

The next step in the generation procedure is to connect the new atom to the existing graph with edges. We do this in a similar manner to GraphAF, and query every atom sequentially to determine its new bond type, updating the adjacency list as needed (Shi et al., 2020). More formally, this procedure works as follows:

- Initialize $E_n \in \mathbb{R}^{n \times b}$ as a matrix containing each atom's edge type to the new atom a_{n+1}. At initialization, let E_n contain all unbonded edge types.
- for i in $1..n$ do:
  - $V'_n = \text{Concat}(V_n, E_n, \text{OneHot}(a_{n+1}))$. V'_n is a $n \times (2d + b)$ matrix of modified atom features. Row i contains the one-hot encoded type of atom i, the one-hot encoded type of atom i's current edge to atom $n + 1$, and the one-hot encoded type of atom $n + 1$.
  - $G'_n = (V'_n, A_n, X_n)$
  - $h_i = F_E(G'_n)_i$
  - $b_i \sim \text{Categorical}(\text{Softmax}(D_E(h_i)))$. b_i is a sampled bond type between atom i and atom $n + 1$. D_E is another MLP decoder which acts on the node-specific embedding of atom i.
  - $E_n[i, :] \leftarrow \text{OneHot}(b_i)$

Through this procedure, a set of bonds is sampled for the new atom.

In the final step, the new atom is given a 3D position. We accomplish this by predicting a discrete distribution of distances from each atom in the graph to the new atom, and a discrete distribution of bond angles between edges that contain the new atom and all adjacent edges. These predictions induce a distribution over 3D coordinates. In a secondary step, we approximately sample from this spatial distribution by drawing points from a fine, stochastic 3D grid using the likelihood function given by the distance and angle predictions. More formally, the positions of the atoms are predicted as follows:

$$V'_n = \text{Concat}(V_n, E_n, \text{OneHot}(a_{n+1}))$$
$$G'_n = (V'_n, A_n, X_n)$$
$$h_1, .., h_n = F_D(G'_n)$$
$$h'_1, .., h'_n = F_\theta(G'_n)$$
$$p_i = \text{Softmax}(D_D(h_i)) \text{ for } i = 1..n$$

$$q_{ij} = \text{Softmax}(D_\theta(h'_i, h'_j)) \text{ for } i, j \text{ s.t. } argmax(A_n[i,j]) > 0 \text{ and } argmax(E_n[j]) > 0,$$

where D_D and D_θ are MLP decoders as before. Note that we are re-using the matrix E_n from the edge prediction step, which has accumulated all of the new edges to atom $n+1$. The probability vectors $p_1, ..., p_n$ now define discrete distributions over the distances between each atom in the graph and the new atom, and the vectors q_{ij} define distributions over bond angles. If we treat these distributions as being independent, we can simply use the product rule to compute the likelihood of any point in 3D space:

$$L_D(x_{n+1}) = \prod_{i=1}^{n} p_i(||x_{n+1} - x_i||)$$

$$L_\theta(x_{n+1}) = \prod_{i,j \in I} q_{ij}(\text{Angle}(x_{n+1} - x_j, x_i - x_j))$$

$$L(x_{n+1}) = L_D(x_{n+1}) \times L_\theta(x_{n+1}),$$

where x_i is the location of atom i, I is the set of incident edges to the neighbors of a_{n+1}, and "Angle" denotes the angle between two vectors. To sample a point from the likelihood $L(x_{n+1})$, we simply assign a likelihood to every point in a fine, stochastic grid surrounding the atoms that are bonded to a_{n+1}, and sample from it as a categorical distribution to produce a new spatial location.

By repeating this procedure until termination, GEN3D produces a 3D molecule from a single starting atom. Note that, because the generation process is sequential, it is simple to mask out atom or edge selections that would violate valence constraints, thereby guaranteeing that generated molecules follow basic chemical rules. It is also possible for the model to predict a non-terminating atom, but then predict that no edges connect to that atom. In this case, we rerun the edge sampling procedure until at least one edge to the new atom is generated. If no edge to the new atom is produced after 10 resampling attempts, the new atom is discarded and the generation process is said to have terminated. The GEN3D generation procedure is illustrated in Figure 1.

The model is trained to autoregressively predict the next atom type, next edge types, next atom distances, and next atom angles for sequentially presented subgraphs of training molecules. The training subgraphs are presented in a random breadth-first order, following previous work on molecular graph generation (Liu et al., 2018; Shi et al., 2020). The model is trained with a cross-entropy loss and a small amount of weight decay. Our experiments use the Adam optimizer with a base learning rate of 0.001 (Kingma & Ba, 2014). While the batch sizes and training times varied across our different experiments, all models were able to train in approximately one day on a single NVIDIA A100 GPU. The model is trained using teacher forcing, so it only learns to make accurate predictions when given well-formed, ground-truth structures as input. This could make the model more brittle at generation time because it will struggle to recover from its own mistakes in previous iterations. To increase geometric robustness and mitigate this issue, we add uniform random noise of up to .05 Å to the atomic coordinates during training.

4 RESULTS

We trained GEN3D to generate 3D molecules from three datasets: QM9, GEOM-QM9 and GEOM-Drugs (Ramakrishnan et al., 2014; Axelrod & Gomez-Bombarelli, 2020). QM9 contains 134,000 small molecules with up to nine heavy atoms (i.e., not including hydrogen) of the chemical elements C, N, O, and F. Each molecule has a single set of 3D coordinates obtained via Density Functional Theory calculations, which approximately compute the quantum mechanical energy of a set of atoms in 3D space. GEOM-QM9 contains the same set of compounds as QM9, but with multiple geometries for each molecule. GEOM-Drugs also has multiple geometries for each molecule, and contains over 300,000 drug-like compounds with more heavy atoms and atomic species than QM9.

On QM9, we trained one version of the model with heavy atoms only, and one version with hydrogens. To ensure the quality of our geometric data,we converted the coordinates from the QM9 source files into SDF files using OpenBabel (O'Boyle et al., 2011), which contain both coordinates and connectivity information inferred based on inter-atomic distances. We discarded all molecules where the inferred connectivity did not match the intended SMILES string from the QM9 source data. This left a set of approximately 124,000 molecules with SDF-formatted bonding information inferred by OpenBabel. We used 100,000 of these molecules for training, with the additional 24,000 molecules

held out for validation. For GEOM-QM9 we trained on 200,000 molecule-geometry pairs, and excluded all SMILES strings from the test set of Xu et al. (2021b). For GEOM-Drugs we trained with heavy atoms only, using 50,000 randomly chosen molecule-geometry pairs for training. We found that, after 60 epochs of training, GEN3D was able to generate highly realistic 3D molecules from all of these datasets. Visualizations of GEN3D samples from QM9 and GEOM-Drugs are shown in Figure 2.

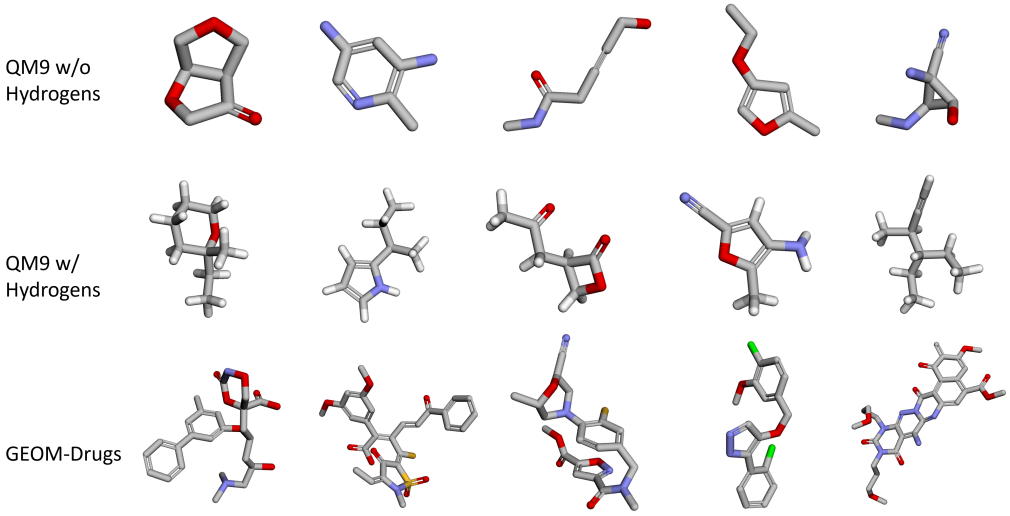

Figure 2: Samples of molecules produced by GEN3D when trained on various training sets listed in the first column. These examples have been curated to showcase the qualitative diversity of molecules created by GEN3D. For a larger number of raw samples, see Appendix A.

4 .1 Characteristics of Generated Molecular Graphs

To assess the quality of our generated molecules, we began by analyzing the characteristics of our generated molecular graphs on QM9. As is standard for molecular generative models, we assessed the percentages of novel and unique molecular graphs generated by the heavy atom QM9 model in a sample of 10,000 molecules. A novel molecular graph is defined as a graph not present in the training data. The uniqueness rate is defined by the number of distinct molecular graphs generated, divided by the total number of molecules generated. We focused on the heavy atom model for these metrics because most existing molecular graph generators use only heavy atoms. As mentioned previously, we can mask-out atom and edge selections at each step of the generative process such that 100% of generated molecular graphs have the correct number of bonds for each atom. We can also assess the percentage of molecules that are chemically valid even when no masking is performed, and doing so gives an indication of the model's understanding of fundamental chemical constraints. We compare our novelty, validity, and uniqueness metrics against GraphAF and CGVAE, which are two recently published molecular graph generators that add one atom at a time. We also compare against GEN3D−, a geometry-unaware baseline that was created by removing the geometric networks from GEN3D and setting all positional inputs to 0. These results are reported in Table 1.

Table 1: Properties of QM9 Molecular Graphs (Heavy Atoms)

Model	Validity	Validity (w/o check)	Uniqueness	Novelty	Strain
CGVAE	100%	49.55%	97.09%	88.18%	70.84%
GraphAF	100%	67%	94.51%	88.93 %	-
GEN3D−	100%	**99.79%**	95.59%	25.93%	90.97%
GEN3D (ours)	100%	**98.80%**	94.33%	33.18%	**93.36**%
QM9 (truth)	100%	100%	-	-	92.41%

We found that, even without imposing checks at generation-time, GEN3D produces molecules that obey valence constraints 98.8% of the time after training on QM9. This far exceeds the unchecked validity rate of 67% achieved by GraphAF, suggesting that GEN3D has a better understanding of basic chemical rules. Interestingly, our geometry-free baseline achieves 99.8% validity, suggesting that our improvements in chemical validity come from architectural differences that are unrelated to the generation of 3D geometries. GEN3D achieves a uniqueness rate of 94.3%, which is similar to the rates for GraphAF and CGVAE. We also assessed the geometric feasibility of generated graphs by converting them into 3D coordinates using CORINA (Sadowski & Gasteiger, 1993), and then computing the volume of the tetrahedron enclosed by each sp^3 tetrahedral center, with vertices located 1 Å along each tetrahedral bond. Graphs that could not be converted with CORINA, or contained tetrahedral centers with volumes less than 0.345 cubic Å, were classified as being overly strained. This test was adapted from Ruddigkeit et al. (2012), where it was used to filter overly strained graphs from the GDB-17 dataset. We found that GEN3D produced fewer overly strained molecules than other models, including GEN3D−, suggesting that explicitly generating molecular geometries helps bias the model towards stable compounds.

Interestingly, GEN3D showed a novelty score of only 33.18%, which is far lower than reported for previously published models. We are confident, however, that our model is not overfit to its training data. Of molecules created by GEN3D that matched a molecule in our QM9 dataset, 18.6% match a molecule in the held-out validation set. Based on the relative sizes of our validation and training sets, an unbiased model would generate models matching the validation set 19.5% of the time. This indicates that our model suffers from minimal overfitting despite its relatively low novelty score.

When considering the composition of the QM9 dataset, it is not particularly surprising that GEN3D achieved a relatively low novelty score. QM9 is a subset of the massive GDB-17 dataset, which aims to enumerate all possible compounds with up to 17 heavy atoms, and with only very basic structural and chemical constraints limiting the set of compounds in the database (Ruddigkeit et al., 2012). QM9 is simply an enumeration of all GDB-17 compounds with up to nine heavy atoms (Ramakrishnan et al., 2014). For a model to generate a molecular graph outside of the QM9 dataset, that molecule must thus have more than nine heavy atoms or violate a basic chemical constraint that is satisfied by all compounds in the GDB-17 databases. Models like CGVAE and GraphAF report $>$90% novelty rates when trained on the majority of compounds in QM9, meaning that their generated molecules violate the constraints of the QM9 data distribution very frequently.

To further explain the discrepancy in novelty scores between GEN3D and other models, we analyzed why molecules generated by CGVAE and GEN3D are present or absent from QM9, and present the results in Appendix B. We found that CGVAE produces far more molecules with over nine heavy atoms, and this accounts for most of the difference in novelty. This is despite the fact that CGVAE is constrained to create at most 10 atoms, while GEN3D is completely unconstrained in the number of atoms it generates, suggesting that GEN3D better learns the distribution of atom counts in QM9.

4 .2 Accuracy of Molecular Geometries

After assessing the quality of our generated graphs by comparing them to other molecular graph generators, we sought to verify the quality of the 3D geometries produced by GEN3D. Here, we compared our model to ENF and G-SchNet, which are the only other published models that generate samples from the distribution of 3D QM9 molecules. Both ENF and G-SchNet produce the positions of heavy atoms and hydrogens as the output of their generative process. In order to facilitate a direct comparison, we compared these models to our all-atom QM9 model. The ENF paper reports atomic stability as the percentage of atoms that have a correct number of bonds, and molecular stability as the fraction of all molecules with the correct number of bonds for every atom. We report these metrics in Table 2, and compare GEN3D to ENF, G-SchNet, and related baselines. GEN3D outperformed all other models, achieving 97.5% molecular stability without any valence masking, compared to 77% for G-SchNet and 4.3% for ENF. In order to assess the geometric realism of the generated molecules, the authors of ENF computed the Jensen-Shannon divergence between a normalized histogram of inter-atomic distances and the true distribution of pairwise distances from the QM9 dataset. We also computed this metric, and found that GEN3D advances the state of the art, reducing the JS divergence by a factor of two over G-SchNet and a factor of four over ENF. Unlike G-SchNet and ENF, GEN3D also generates bonding information, which may help increase chemical validity and geometric realism. In addition, GEN3D is much cheaper to train than ENF's

flow-based generative process, and is applicable to larger drug-like molecules. These comparisons are reported in Table 2, and the true and learned histograms of pairwise distances are plotted in Appendix C. In order to be consistent with the ENF paper, the Jensen-Shannon divergence was only computed between generated and QM9 molecules with exactly 19 total atoms.

Table 2: Properties of QM9 Molecules (3D models with hydrogens)

Model	Atom Stability	Mol Stability	Distance JS
GNF-attention	72%	.3%	.007
E-NF	84%	4.2%	.006
G-SchNet	98.7%	77%	.0031
GEN3D (w/o check)	**99.7%**	**97.5%**	**.0014**
GEN3D (w/ check)	**99.87%**	**99.1%**	**.0014**
QM9 (truth)	99.99%	99.9%	0

The Jensen-Shannon divergence metric introduced by ENF provides confidence that GEN3D is, on average, generating accurate molecular geometries. This metric, however, is relatively insensitive to the correctness of individual molecular geometries because it only compares the aggregate distributions of distances. In order to further validate the accuracy of GEN3D's generated geometries, we used GEN3D to predict the geometries of specific molecular graphs, and compared its accuracy with purpose-built tools designed for molecular geometry prediction, such as the model described in Xu et al. (2021b). Details and limitations about how geometries of a fixed molecular graph were sampled by GEN3D are presented in Appendix D.

To evaluate the ability of GEN3D to predict molecular geometries, we trained GEN3D to generate molecules form GEOM-QM9 (Axelrod & Gomez-Bombarelli, 2020). We then followed the evaluation protocol described in Xu et al. (2021a) and Xu et al. (2021b) with the same set of 150 molecular graphs, excluding these graphs from our training set. As in these prior works, we predicted an ensemble of geometries, and then computed COV and MAT scores with respect to the test set. The COV score measures what fraction of reference geometries have a "close" neighbor in the set of generated geometries, where closeness is measured with an aligned RMSD threshold. We used a threshold of 0.5 Å, following Xu et al. (2021b). The MAT score summarizes the aligned RMSD of each reference geometry to its closest neighbor in the set of generated geometries. For additional detail on the evaluation protocol, see Xu et al. (2021a). GEN3D achieves results that are among the best for published models on both metrics. In particular, its MAT scores outperform all prior methods that do not refine geometries using a rules-based force field. We compare GEN3D with previous machine learning models for molecular geometry prediction, as well as the ETKDG algorithm implemented in RDKit (which predicts molecular geometries using a database of preferred torsional angles and bond lengths (Riniker & Landrum, 2015)). Table 3 shows the results of the evaluation, and Figure 3 visualizes representative geometry predictions.

Table 3: Geometry Prediction on GEOM-QM9

Metric	**COV (%)**		**MAT (Å)**	
	Mean	Median	Mean	Median
GraphDG	55.09 %	56.47 %	0.4649	0.4298
CGCF	69.60 %	70.64 %	0.3915	0.3986
ConfVAE	77.98 %	82.82 %	0.3778	0.3770
RDKit	80.68 %	87.50 %	0.3349	0.3245
GEN3D (ours)	73.62 %	77.14 %	**0.3168**	**0.3049**

The results in Table 3 indicate that GEN3D is accurately sampling from the joint distribution of molecular graphs and molecular geometries.

5 Conclusions

In this paper, we have introduced GEN3D: an autoregressive model for generating molecules in 3D space that is rotationally and translationally equivariant. We have demonstrated that GEN3D

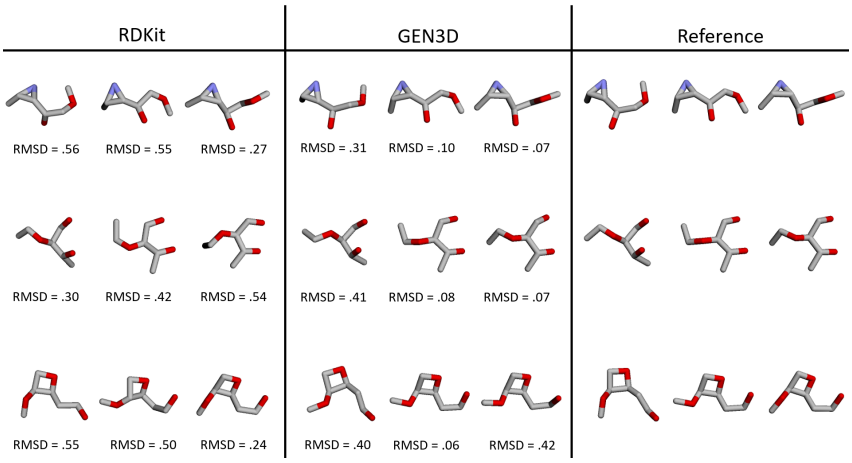

Figure 3: Predicted geometries for GEOM-QM9. The right columns contains reference geometries, and the left two columns show the nearest neighbor to the reference geometries among the geometries generated by RDKit and GEN3D.

produces 3D molecules that are chemically stable and geometrically realistic, and achieves state-of-the-art results on multiple benchmarks in molecular machine learning.

We believe models like GEN3D will have many practical applications in chemistry and drug discovery, as well as in 3D machine learning more broadly. A natural extension of GEN3D is a model that generates molecules in a 3D space occupied by features of a protein pocket, such as pharmacophore features or amino acid side chain atoms. Such a model could learn to generate potential binding molecules directly inside a protein pocket. The generated molecules and geometries could then move directly into downstream stability screenings using molecular dynamics simulations or docking programs. This approach could greatly decrease the amount of computation needed to select and verify promising drug molecules. Quantum mechanical energy calculations, docking scores, and molecular dynamics simulation results from generated poses could also be used to train the model via reinforcement learning.

Conditional likelihood models like GEN3D can also work together with other drug discovery methods. For instance, if a virtual screening campaign identifies several small molecules that each bind tightly to a different region in a protein pocket, the conditional likelihood function of a model like GEN3D could be used to propose large molecules that retain the respective poses of the small molecules, and join them together in a chemically and geometrically feasible way. Alternatively, if a screening campaign identifies a family of small molecules that each bind to the same region of a protein pocket and share a moiety in the context of their bound poses, GEN3D could also be used to extend this common molecular scaffold in new ways. Such applications of GEN3D could help with the enumeration of possible side chain substitutions and the closure of rings, which are common tasks is drug design.

While the current architecture of GEN3D already demonstrates impressive performance, it only stands to improve with architectural developments in the broader world of 3D machine learning. For example, GEN3D is currently reflection invariant due its use of EGNN subunits, but this could be fixed by substituting the EGNN with reflection-sensitive networks like DimeNet or SphereNet (Klicpera et al., 2020; Liu et al., 2021).

Beyond the fields of chemistry and drug discovery, GEN3D provides a simple equivariant approach for generating 3D graphs and point clouds, and is much cheaper to train than flow-based generative models used in previous works. We hope GEN3D and similar subsequent models will find diverse uses in generating and optimizing structures in 3D space.

REPRODUCIBILITY STATEMENT

In order to facilitate reproducibility, we plan to release the source code for the GEN3D model prior to publication, together with pre-trained GEN3D model weights and our curated version of the QM9 dataset.

REFERENCES

Simon Axelrod and Rafael Gomez-Bombarelli. Geom: Energy-annotated molecular conformations for property prediction and molecular generation. *arXiv preprint arXiv:2006.05531*, 2020.

Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1 (1):269–271, 1959.

Fabian B Fuchs, Daniel E Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3D roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503*, 2020.

Niklas WA Gebauer, Michael Gastegger, and Kristof T Schütt. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. *arXiv preprint arXiv:1906.00957*, 2019.

J Andrew Grant, MA Gallardo, and Barry T Pickup. A fast method of molecular shape comparison: A simple application of a gaussian description of molecular shape. *Journal of Computational Chemistry*, 17(14):1653–1666, 1996.

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. *ACS Central Science*, 4(2):268–276, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448–456. PMLR, 2015.

Paul Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*, pp. 2323–2332. PMLR, 2018.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In *International Conference on Machine Learning*, pp. 4839–4848. PMLR, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020.

Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *International Conference on Machine Learning*, pp. 1945–1954. PMLR, 2017.

Greg Landrum et al. Rdkit: Open-source cheminformatics software. *http://www.rdkit.org/, https://zenodo.org/record/5242603#.YVtf6EbMJzU*, 2021.

Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander L Gaunt. Constrained graph variational autoencoders for molecule design. *arXiv preprint arXiv:1805.09076*, 2018.

Yi Liu, Limei Wang, Meng Liu, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3D graph networks. *arXiv preprint arXiv:2102.05013*, 2021.

Paul Maragakis, Hunter Nisonoff, Brian Cole, and David E. Shaw. A deep-learning view of chemical space designed to facilitate drug discovery. *Journal of Chemical Information and Modeling*, 60 (10):4487–4496, 2020.

Tomohide Masuda, Matthew Ragoza, and David Ryan Koes. Generating 3D molecular structures conditional on a receptor binding site with deep generative models. *arXiv preprint arXiv:2010.14442*, 2020.

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. *arXiv preprint arXiv:2107.05007*, 2021.

Noel O'Boyle, Michael Banck, Craig James, Chris Morely, Tim Vandermeersch, and Geoffrey Hutchison. Open Babel: An open chemical toolbox. *Journal of Cheminformatics*, 3(33), 2011.

Emanuele Perola and Paul S Charifson. Conformational analysis of drug-like molecules bound to proteins: an extensive study of ligand reorganization upon binding. *Journal of medicinal chemistry*, 47(10):2499–2510, 2004.

PG Polishchuk, TI Madzhidov, and A Varnek. Estimation of the size of drug-like chemical space based on GDB-17 data. *Journal of Computer-Aided Molecular Design*, 27(8):675–679, 2013.

Matthew Ragoza, Tomohide Masuda, and David Ryan Koes. Learning a continuous representation of 3D molecular structures with deep generative models. *arXiv preprint arXiv:2010.08687*, 2020.

Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, von Lilienfeld, and O. Anatole. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(140022), 2014.

Sereina Riniker and Gregory A. Landrum. Better informed distance geometry: Using what we know to improve conformation generation. *Journal of Chemical Information and Modeling*, 55 (12):2562–2574, 2015.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

David J. Rogers and Taffee T. Tanimoto. A computer program for classifying plants. *Science*, 132 (3434):1115–1118, 1960.

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. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, 2012.

Jens Sadowski and Johann Gasteiger. From atoms and bonds to three-dimensional atomic coordinates: automatic model builders. *Chemical Reviews*, 93(7):2567–2581, 1993.

Victor Garcia Satorras, Emiel Hoogeboom, Fabian B Fuchs, Ingmar Posner, and Max Welling. E(n) equivariant normalizing flows for molecule generation in 3D. *arXiv preprint arXiv:2105.09016*, 2021a.

Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. *arXiv preprint arXiv:2102.09844*, 2021b.

Kristof T Schütt, Pieter-Jan Kindermans, Huziel E Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *arXiv preprint arXiv:1706.08566*, 2017.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. GraphAF: a flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382*, 2020.

Gregor NC Simm and José Miguel Hernández-Lobato. A generative model for molecular distance geometry. *arXiv preprint arXiv:1909.11459*, 2019.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3D point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

David 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.

Minkai Xu, Shitong Luo, Yoshua Bengio, Jian Peng, and Jian Tang. Learning neural generative dynamics for molecular conformation generation. *arXiv preprint arXiv:2102.10240*, 2021a.

Minkai Xu, Wujie Wang, Shitong Luo, Chence Shi, Yoshua Bengio, Rafael Gomez-Bombarelli, and Jian Tang. An end-to-end framework for molecular conformation generation via bilevel programming. *arXiv preprint arXiv:2105.07246*, 2021b.

Appendices

A Additional Sample Molecules

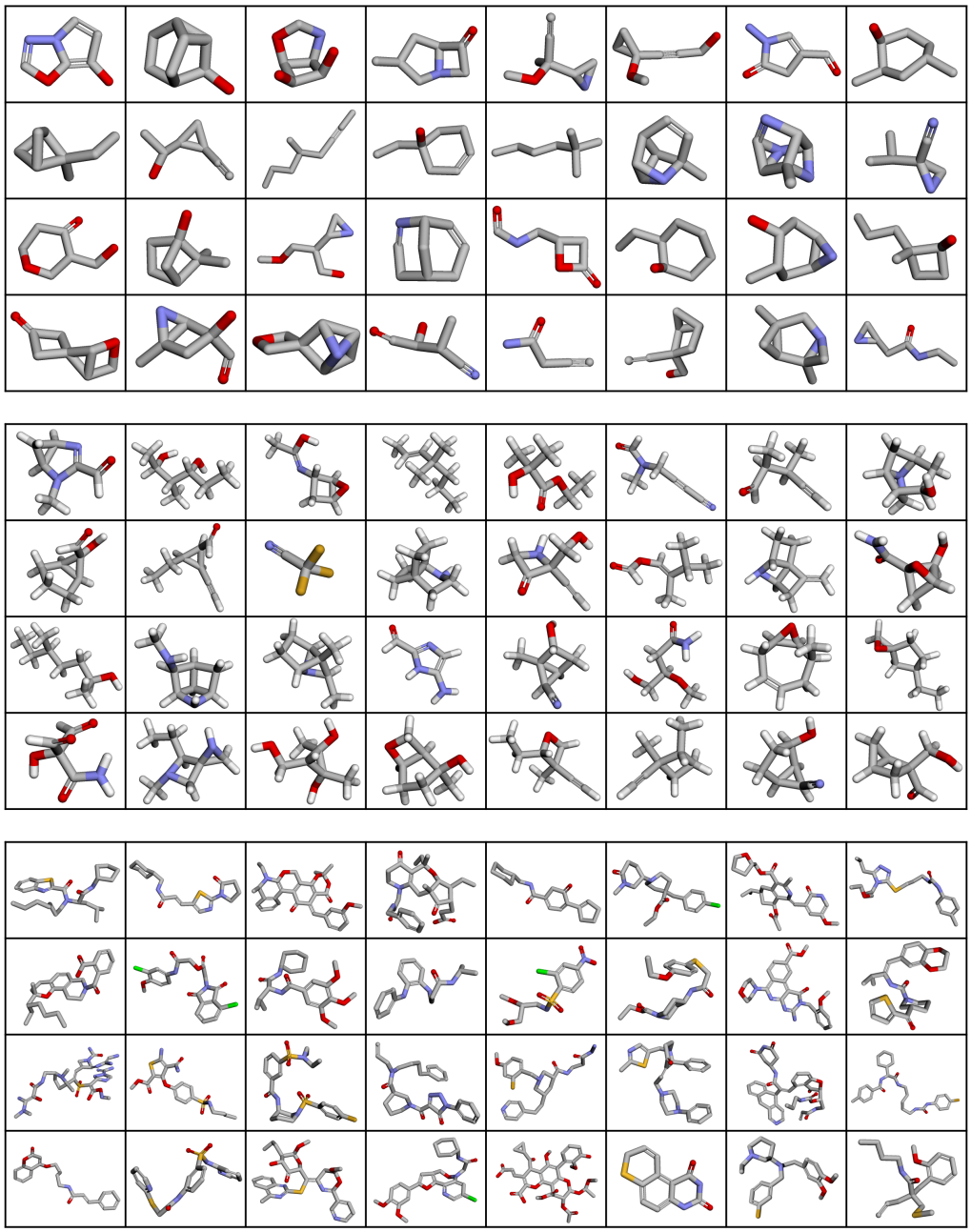

Figure S1: Unfiltered, independent random samples from GEN3D models trained on QM9 without hydrogens (top), QM9 with hydrogens (middle), and GEOM-Drugs (bottom).

B COMPOSITION OF GENERATED MOLECULES

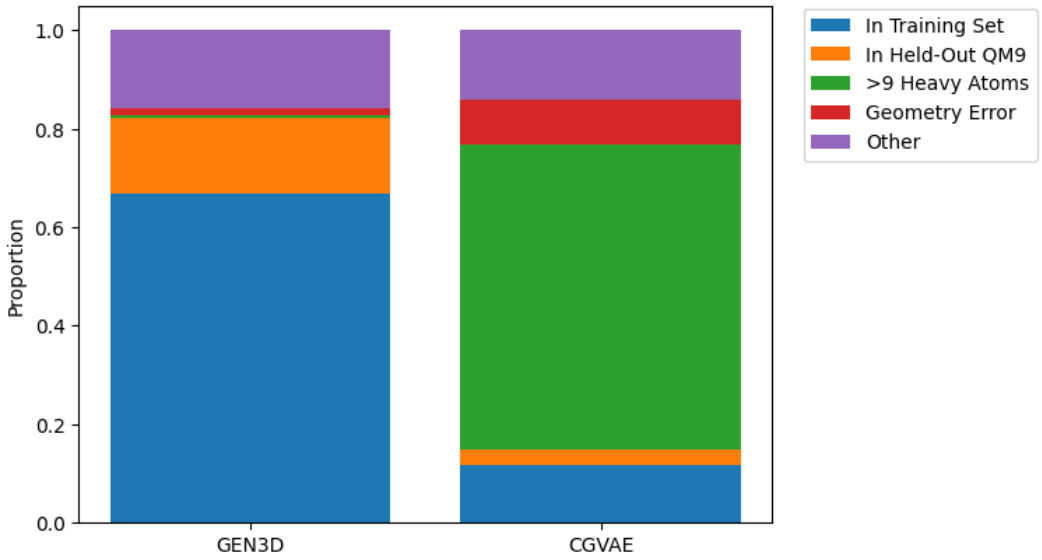

Figure S2: The composition of molecules generated from GEN3D and CGVAE.

C DISTRIBUTION OF INTER-ATOMIC DISTANCES

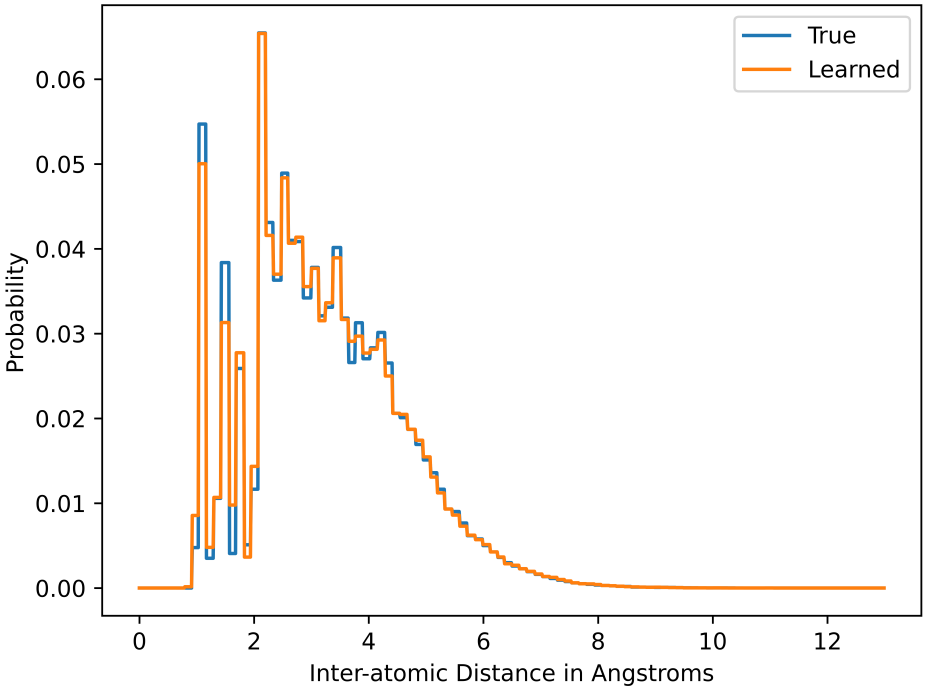

Figure S3: Histograms of inter-atom distances for GEN3D-generated and QM9 molecules with 19 total atoms.

D MOLECULAR GEOMETRY PREDICTION

In order to predict the geometry of a specific molecular graph, we utilized Dijkstra's algorithm to search for geometries of those molecules that are assigned a high likelihood (Dijkstra, 1959). The given molecular graph is unrolled in a breadth-first order, so predicting the molecule's geometry amounts to determining a sequence of positions for each atom during the rollout. If atomic positions are discretized, then the space of possible molecular geometries forms a tree. Each edge in the tree can be assigned a likelihood by GEN3D. Predicting a plausible geometry thus amounts to finding a path where the sum of the log-likelihoods of the edges is large. This can be accomplished using standard graph search algorithms like A* or Dijkstra's algorithm. The geometry prediction algorithm is presented in Algorithm 1 below.

Algorithm 1 Molecular Geometry Search using GEN3D

Input: A molecular graph $G = (V, A)$ and a number of iterations T.
Perform a Breadth-First Search on G, producing a series of vertex lists $V_1, ..., V_N$ and adjacency matrices $A_1, ..., A_N$.
$Q \leftarrow EmptyPriorityQueue()$
$Q.push((0, 1, [0.0, 0.0, 0.0]))$ ▷ Queue entries have the form (Negative Log Likelihood, Number of Atoms, Position Matrix)
$t \leftarrow 0$
$results \leftarrow []$
while $t < T$ **do**
  $nll, i, X \leftarrow Q.dequeue()$
  if $i == N$ **then**
    $results.append((nll, X))$
  else
    for j **in** $\{1..i\}$ **do** ▷ Find an atom which is a neighbor of the next atom in the rollout
      if $argmax(A_{i+1}[i+1, j]) > 0$ **then**
        $neigh \leftarrow j$
      end if
    end for
    $points \leftarrow MakeGrid(X[neigh])$ ▷ Create a fine grid of potential positions around the neighbor atom
    for x **in** $points$ **do**
      $dst_lik = p_D(x|V_{i+1}, A_{i+1}, X_i)$
      $ang_lik = p_\theta(x|V_{i+1}, A_{i+1}, X_i)$
      $atm_lik = p_A(V_{i+2}|V_{i+1}, A_{i+1}, [X_i; x])$
      $edg_lik = p_E(A_{i+2}|V_{i+2}, A_{i+1}, [X_i; x])$
      $nll \leftarrow nll - \log(dst_lik) - \log(ang_lik) - \log(atm_lik) - \log(edg_lik)$
      $Q.push((nll, i+1, [X; x]))$
    end for
  end if
end while
return $results$

We found this procedure to be effective and computational feasible for molecules in GEOM-QM9. The tree of potential geometries expands rapidly with increased numbers of atoms, however, limiting its efficiency for large molecules. Future work could identify search heuristics that make this approach more efficient.

E DEMONSTRATION OF PROPERTY OPTIMIZATION

We evaluated the ability of GEN3D to generate 3D molecules in poses that have favorable predicted interactions with a target protein pocket, as evaluated by the ROCS virtual screening algorithm (Grant et al., 1996). We started from a GEN3D model trained on GEOM-drugs (we call this model GEN3D-gd). We curated a large pre-existing library of 62.9-million compounds, containing up to 250 molecular geometries for each compound generated with OpenEye Omega (Perola & Charifson, 2004), and screened the resulting 13.8-billion conformations against our target pocket using ROCS. We then selected the top 1000 scoring geometries belonging to distinct molecular graphs from the library, and we fine-tuned GEN3D-gd on these 1000 3D molecules for 100 epochs (we call the resulting model GEN3D-ft). To evaluate the ability of GEN3D to learn chemical and geometric features that are conducive to binding the pocket, we generated 10,000 molecules with 3D coordinates from GEN3D-gd and GEN3D-ft. In addition, we took the molecular graphs generated by GEN3D-ft and recalculated molecular geometries for them using OpenEye Omega. We excluded molecules generated by GEN3D-ft if the molecular graph overlapped with the fine-tuning set (2.07% of the total), and scored the remainder using ROCS. We found that fine-tuning significantly increased the scores of generated compounds. Because GEN3D-ft was fine-tuned on high-scoring molecular geometries, the molecular geometries it generated implicitly include information about the target geometry that were unavailable to GEN3D-gd and OpenEye Omega. As a result, the scores for GEN3D-ft geometries were, on average, better than those generated by other methods. These results are shown in Figures S4.

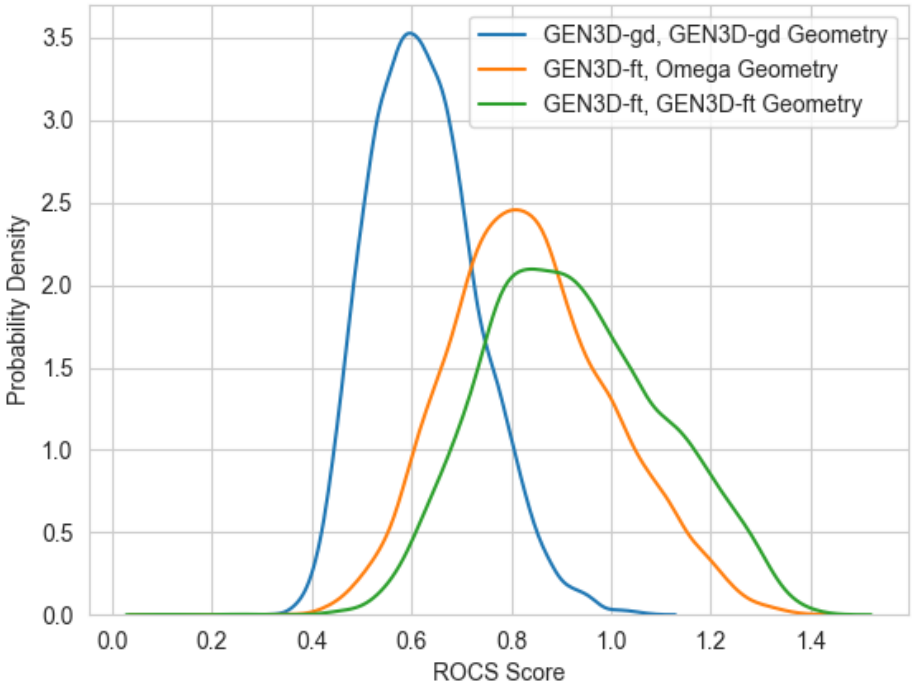

Figure S4: The probability densities of ROCS scores for molecular graphs and geometries generated by GEN3D-gd (blue), molecular graphs generated by GEN3D-ft with the Omega geometries (orange), and molecular graphs and geometries generated by GEN3D-ft (green). Higher ROCS scores indicate that a model is better able to produce desirable molecules.

Ideally, this training procedure would allow our models to generate strong binders that are significantly different from those in the fine-tuning set. To compare each model's ability to produce both high-quality and novel compounds, we picked the top 2% of molecules (by ROCS score) generated by each model, and plotted their ROCS scores against their maximum Tanimoto similarity coeffi-

cient to an element of the set used for fine-tuning. A Tanimoto similarity coefficient (also called a Jaccard coefficient of community) ranges from fully dissimilar at 0.0 to identical at 1.0, and is a measure of the structural closeness of two molecular graphs (Rogers & Tanimoto, 1960; Jaccard, 1912). It is computed by representing two molecules with Extended-Connectivity Fingerprints, which are essentially lists of activated bits corresponding to substructures present in each molecule (Rogers & Hahn, 2010). (Here, we used RDKit's implementation of Morgan fingerprints with 2048 bit, radius 2, and without chirality (Landrum et al., 2021).) We found that GEN3D-ft generated molecules with high ROCS scores across a wide range of Tanimoto similarities to the fine-tuning set. Molecules generated by GEN3D-ft had significantly higher scores than those generated by GEN3D-gd, even when comparing molecules from each model with comparable similarities to the fine-tuning set. In this particular instance, the highest ROCS scoring molecule generated by GEN3D-ft had a Tanimoto similarity to the fine-tuning set of about 0.4. These results are shown in Figure S5.

These experiments indicate that GEN3D is able to shift its generative distribution into specific regions of chemical and geometric space.

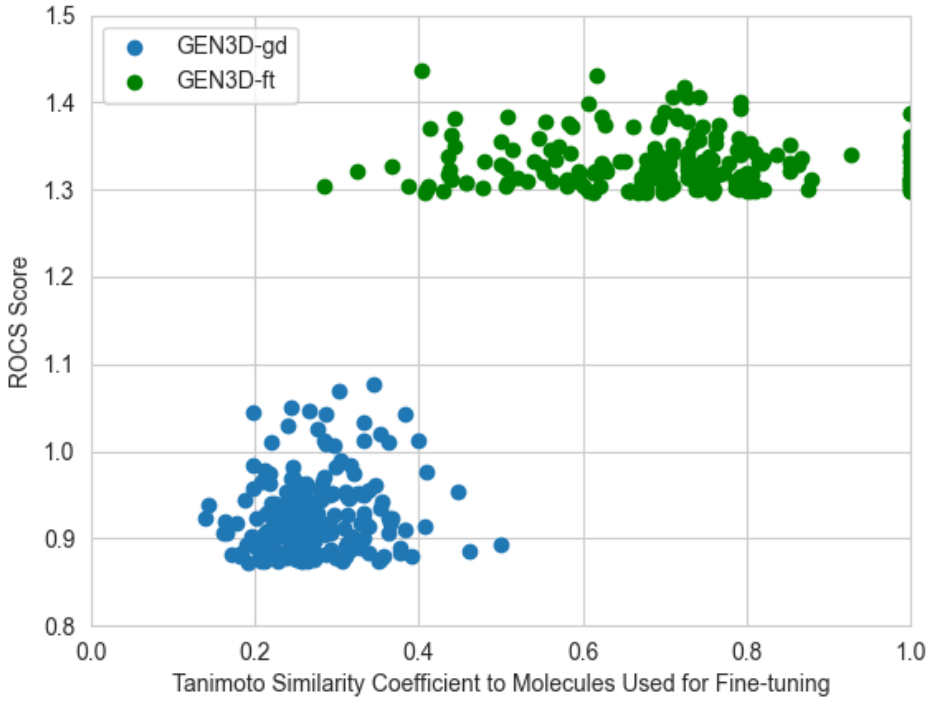

Figure S5: The similarity to the fine-tuning dataset of the molecules with the top 2% of ROCS scores. For clarity, we include those molecules that exactly match a member of the fine-tuning training set (i.e., they have a Tanimoto similarity coefficient of 1.0).