Shape-conditioned 3D Molecule Generation via Equivariant Diffusion Models

Ziqi Chen
The Ohio State University
chen.8484@osu.edu

Bo Peng
The Ohio State University
peng.707@osu.edu

Srinivasan Parthasarathy
The Ohio State University
srini@cse.ohio-state.edu

Xia Ning*
The Ohio State University
ning.104@osu.edu

Abstract

Ligand-based drug design aims to identify novel drug candidates of similar shapes with known active molecules. In this paper, we formulated an *in silico* shape-conditioned molecule generation problem to generate 3D molecule structures conditioned on the shape of a given molecule. To address this problem, we developed an equivariant shape-conditioned generative model ShapeMol. ShapeMol consists of an equivariant shape encoder that maps molecular surface shapes into latent embeddings, and an equivariant diffusion model that generates 3D molecules based on these embeddings. Experimental results show that ShapeMol can generate novel, diverse, drug-like molecules that retain 3D molecular shapes similar to the given shape condition. These results demonstrate the potential of ShapeMol in designing drug candidates of desired 3D shapes binding to protein target pockets.

1 Introduction

Generating novel drug candidates is a critical step in drug discovery to identify possible therapeutic solutions. Recently, several models [1–3] have been designed to generate 3D molecules conditioned on the protein targets, aiming to facilitate structured-based drug design (SBDD) [4], given that molecules exist in 3D space and the efficacy of drug molecules depends on their 3D structures fitting into protein pockets. However, SBDD needs high-quality 3D structures of protein binding pockets, which are often unavailable [5]. Different from SBDD, ligand-based drug design (LBDD) [6] utilizes ligands known to interact with a protein target, and does not require knowledge of protein structures. In LBDD, shape-based virtual screening tools such as ROCS [7] have been widely used to identify molecules with similar shapes to known ligands by enumerating molecules in chemical libraries. However, virtual screen tools cannot probe the novel chemical space. Therefore, it is highly needed to develop generative methods to generate novel molecules with desired 3D shapes.

In this paper, we present a novel generative model for 3D molecule generation conditioned on given 3D shapes. Our method, denoted as ShapeMol, employs an equivariant shape embedding module to map 3D molecule surface shapes into shape latent embeddings. It then uses an equivariant diffusion generative model to generate molecules conditioned on these embeddings, by iteratively denoising atom positions and features. During molecule generation, ShapeMol can utilize additional shape guidance by pushing the predicted atoms far from the condition shapes to those shapes. ShapeMol with shape guidance achieves the highest average 3D shape similarity between the generated molecules and condition molecules compared to the state-of-the-art baseline. A comprehensive review of existing molecule generation methods is available in Supplementary Section S1.

*Contact Author

NeurIPS 2023 Generative AI and Biology Workshop.

2 Method

We represent a 3D molecule $\mathtt{M}$ as a set of atoms $\mathtt{M} = \{a_1, a_2, \cdots, a_{|\mathtt{M}|}\}$, where $|\mathtt{M}|$ is the number of atoms in $\mathtt{M}$; each atom a_i has a 3D coordinate $\mathbf{x}_i \in \mathbb{R}^3$ and a one-hot feature vector $\mathbf{v}_i \in \mathbb{R}^K$ indicating the atom type and its aromaticity. We represent the 3D surface shape $\mathtt{s}$ of a molecule $\mathtt{M}$ as a point cloud $\mathcal{P}$ constructed by sampling points over the molecular surface. We develop ShapeMol to generate a new molecule $\mathtt{M}_y$, conditioned on the 3D shape $\mathcal{P}$ of a given molecule $\mathtt{M}_x$. ShapeMol consists of an equivariant shape embedding module SE that maps $\mathcal{P}$ to latent embeddings $\mathbf{H}^{\mathtt{s}}$, and an equivariant diffusion model DIFF that generates 3D molecules conditioned on $\mathbf{H}^{\mathtt{s}}$. Figure 1 presents the overall architecture of ShapeMol.

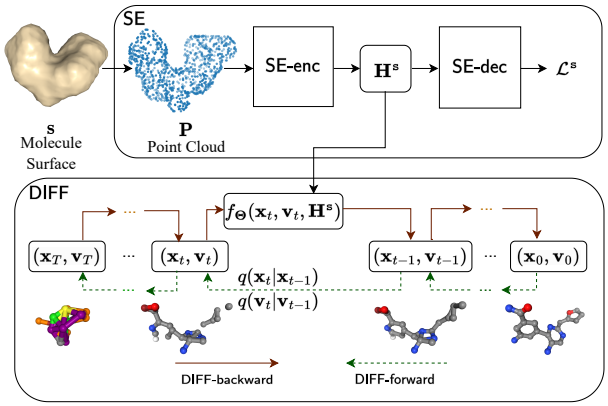


Figure 1: **Model Architecture of** ShapeMol

Equivariant Shape Embedding (SE) ShapeMol pre-trains a shape embedding module SE to generate surface shape embeddings $\mathbf{H}^{\mathtt{s}}$. SE uses an encoder SE-enc to map $\mathcal{P}$ to the equivariant latent embedding $\mathbf{H}^{\mathtt{s}}$. Following the previous work [3, 8, 9], to ensure translation equivariance, SE-enc shifts the center of each $\mathcal{P}$ to zero to eliminate all translations. To ensure rotation equivariance, SE-enc leverages Vector Neurons (VNs) [10] and Dynamic Graph Convolutional Neural Networks (DGCNNs) [11] to encode $\mathcal{P}$ into $\mathbf{H}^{\mathtt{s}}$. SE employs a decoder SE-dec to optimize $\mathbf{H}^{\mathtt{s}}$ by recovering the signed distances [12] of sampled query points in 3D space to the molecule surface based on $\mathbf{H}^{\mathtt{s}}$. The details of SE is available in Supplementary Section S2.

Shape-Conditioned Molecule Generation In ShapeMol, a shape-conditioned molecule diffusion model, referred to as DIFF, is used to generate a 3D molecule structure (i.e., atom coordinates and features) conditioned on a given 3D surface shape that is represented by the shape latent embedding $\mathbf{H}^{\mathtt{s}}$. Following the denoising diffusion probabilistic models [13], DIFF includes a forward diffusion process based on a Markov chain, denoted as DIFF-forward, which gradually adds noises step by step to the training molecules. Following Guan *et al.* [3], at step $t \in [1, T]$, a small Gaussian noise and a small categorical noise are added to transform the continuous atom positions and discrete atom features from $\{(\mathbf{x}_{t-1}, \mathbf{v}_{t-1})\}$ into $\{(\mathbf{x}_t, \mathbf{v}_t)\}$, respectively. At the final step T, $\{(\mathbf{x}_T, \mathbf{v}_T)\}$ resemble a simple distribution like a standard normal distribution and a uniform categorical distribution, respectively. During training, DIFF is learned to reverse the forward diffusion process via another Markov chain, referred to as the backward generative process and denoted as DIFF-backward, to remove the noises in the noisy molecules. The details of DIFF-forward and DIFF-backward are available in Supplementary Section S3.1 and S3.2, respectively.

Equivariant Shape-Conditioned Molecule Predictor In DIFF-backward, the predictor $f_{\Theta}(\mathbf{x}_t, \mathbf{v}_t, \mathbf{H}^{\mathtt{s}})$ predicts the atom positions and features $(\tilde{\mathbf{x}}_{0,t}, \tilde{\mathbf{v}}_{0,t})$ given the noisy data $(\mathbf{x}_t, \mathbf{v}_t)$ conditioned on $\mathbf{H}^{\mathtt{s}}$. For brevity, in this subsection, we eliminate the subscript t in the notations when no ambiguity arises. $f_{\Theta}(\cdot)$ leverages two multi-layer graph neural networks: (1) an equivariant graph neural network, denoted as EQ-GNN, that equivariantly predicts atom positions that change under transformations, and (2) an invariant graph neural network, denoted as INV-GNN, that invariantly predicts atom features that remain unchanged under transformations.

In EQ-GNN, the atom position $\mathbf{x}_i^{l+1} \in \mathbb{R}^3$ of a_i at the (l+1)-th layer is calculated in an equivariant way as below,

$$\Delta\mathbf{x}_i^{l+1} = \sum_{j \in N(a_i), i \neq j} (\mathbf{x}_i^l - \mathbf{x}_j^l) \mathrm{MHA}^{\mathtt{x}}(d_{ij}^l, \mathbf{h}_i^{l+1}, \mathbf{h}_j^{l+1}, \text{VN-In}(\mathbf{H}^{\mathtt{s}})),$$
$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + \mathrm{Mean}(\Delta\mathbf{x}_i^{l+1}) + \text{VN-Lin}(\mathbf{x}_i^l, \Delta\mathbf{x}_i^{l+1}, \mathbf{H}^{\mathtt{s}}), \quad (1)$$

where $N(a_i)$ is the set of N-nearest neighbors of a_i based on atomic distances; $\Delta\mathbf{x}_i^{l+1} \in \mathbb{R}^{n_h \times 3}$ aggregates the neighborhood information of a_i; $\mathrm{MHA}^{\mathtt{x}}(\cdot)$ denotes the multi-head attention layer in

EQ-GNN with n_h heads; d_{ij}^l is the distance between i-th and j-th atom positions $\mathbf{x}_i^l$ and $\mathbf{x}_j^l$ at the l-th layer; Mean$(\Delta\mathbf{x}_i^{l+1})$ converts $\Delta\mathbf{x}_i^{l+1}$ into a 3D vector via meaning pooling to adjust the atom position; VN-Lin$(\cdot) \in \mathbb{R}^3$ denotes the equivariant VN-based linear layer [10]. VN-Lin$(\cdot)$ adjusts the atom positions to fit the shape condition represented by $\mathbf{H}^{\mathtt{s}}$, by considering the current atom positions $\mathbf{x}_i^l$ and the neighborhood information $\Delta\mathbf{x}_i^{l+1}$. The learned atom position $\mathbf{x}_i^L$ at the last layer L of EQ-GNN is used as the prediction of $\tilde{\mathbf{x}}_{i,0}$.

In INV-GNN, inspired by the previous work [3] and VN-Layer [10], the atom feature embedding $\mathbf{h}_i^{l+1} \in \mathbb{R}^{d_h}$ of the i-th atom a_i at the (l+1)-th layer is updated in an invariant way as follows,

$$\mathbf{h}_i^{l+1} = \mathbf{h}_i^l + \sum_{j \in N(a_i), i \neq j} \text{MHA}^{\mathtt{h}}(d_{ij}^l, \mathbf{h}_i^l, \mathbf{h}_j^l, \text{VN-In}(\mathbf{H}^{\mathtt{s}})), \quad \mathbf{h}_i^0 = \mathbf{v}_i, \tag{2}$$

where $\text{MHA}^{\mathtt{h}}(\cdot) \in \mathbb{R}^{d_h}$ denotes the multi-head attention layer in INV-GNN. The learned atom feature embedding $\mathbf{h}_i^L$ at the last layer L encodes the neighborhood information of a_i and the conditioned molecular shape, and is used to predict the atom features $\tilde{\mathbf{v}}_{i,0}$ via an MLP layer. The proof of equivariance in Eq. 1 and invariance in Eq. 2 is available in Supplementary Section S3.5 and S3.6.

Model Training ShapeMol optimizes DIFF by minimizing the squared errors between the predicted positions ($\tilde{\mathbf{x}}_{0,t}$) and the ground-truth positions ($\mathbf{x}_0$) of atoms in molecules as follows:

$$\mathcal{L}_t^{\mathtt{x}}(\mathtt{M}) = w_t^{\mathtt{x}} \sum\nolimits_{\forall a \in \mathtt{M}} \|\tilde{\mathbf{x}}_{0,t} - \mathbf{x}_0\|^2, \quad \text{where } w_t^{\mathtt{x}} = \min(\lambda_t, \delta),\ \lambda_t = \bar{\alpha}_t^{\mathtt{x}}/(1-\bar{\alpha}_t^{\mathtt{x}}), \tag{3}$$

where $w_t^{\mathtt{x}}$ is a weight at step t, and is calculated by clipping the signal-to-noise ratio $\lambda_t > 0$ with a threshold $\delta > 0$; $\bar{\alpha}_t^{\mathtt{x}} \in [0,1]$ is a signal ratio at step t. As $\bar{\alpha}_t^{\mathtt{x}}$ decreases monotonically as t increases from 1 to T, $w_t^{\mathtt{x}}$ decreases as well when λ_t is smaller then δ. Thus, $w_t^{\mathtt{x}}$ imposes lower weights on the loss when the noise level in $\mathtt{x}_t$ is higher. This encourages the model training to focus more on accurately recovering molecule structures when there are sufficient signals in the data, rather than being potentially confused by major noises in the data. Following the literature [14], ShapeMol also minimizes the KL divergence $\mathcal{L}_t^{\mathtt{v}}(\mathtt{M})$ between the ground-truth posterior $p(\mathbf{v}_{t-1}|\mathbf{v}_t, \mathbf{v}_0)$ and its approximate $p_\theta(\mathbf{v}_{t-1}|\mathbf{v}_t, \tilde{\mathbf{v}}_{0,t})$ for atom features. The overall ShapeMol loss function is a weighted sum of atom position loss and atom feature loss: $\mathcal{L}_t(\mathtt{M}) = \mathcal{L}_t^{\mathtt{x}}(\mathtt{M}) + \xi\mathcal{L}_t^{\mathtt{v}}(\mathtt{M})$, in which ξ is a hyperparameter. The derivation of the loss functions is available in Supplementary Section S3.7.

Molecule Generation and Shape Guidance During inference, ShapeMol generates novel molecules by gradually denoising $(\mathbf{x}_T, \mathbf{v}_T)$ to $(\mathbf{x}_0, \mathbf{v}_0)$ using the equivariant shape-conditioned molecule predictor. Specifically, ShapeMol samples $\mathbf{x}_T$ and $\mathbf{v}_T$ from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathcal{C}(\mathbf{1}/K)$, respectively. After that, ShapeMol samples $\mathbf{x}_{t-1}$ from $\mathbf{x}_t$ using $p_{\boldsymbol{\Theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \tilde{\mathbf{x}}_{0,t})$ and $\mathbf{v}_{t-1}$ from $\mathbf{v}_t$ using $p_{\boldsymbol{\Theta}}(\mathbf{v}_{t-1}|\mathbf{v}_t, \tilde{\mathbf{v}}_{0,t})$ until t reaches 1.

During molecule generation, ShapeMol can also utilize additional shape guidance by pushing the predicted atoms to the shape of the given molecule $\mathtt{M}_x$. Following Adams and Coley *et al.* [17], the shape used for guidance is defined as a set of points $\mathcal{Q}$ sampled according to atom positions in $\mathtt{M}_x$. Particularly, for each atom a_i in $\mathtt{M}_x$, 20 points are randomly sampled into $\mathcal{Q}$ from a Gaussian distribution centered at $\mathbf{x}_i$ with variance ϕ. Given the predicted atom position $\tilde{\mathbf{x}}_{0,t}$ at step t, ShapeMol applies the shape guidance by adjusting the predicted positions to $\mathtt{M}_x$ as follows:

$$\mathbf{x}_{0,t}^* = (1-\sigma)\tilde{\mathbf{x}}_{0,t} + \sigma \sum_{\mathbf{z} \in n(\tilde{\mathbf{x}}_{0,t}; \mathcal{Q})} \mathbf{z}/n, \text{when} \sum_{\mathbf{z} \in n(\tilde{\mathbf{x}}_{0,t}; \mathcal{Q})} d(\tilde{\mathbf{x}}_{0,t}, \mathbf{z})/n > \gamma, \tag{4}$$

where $\sigma > 0$ is the weight used to balance the prediction $\tilde{\mathbf{x}}_{0,t}$ and the adjustment; $d(\tilde{\mathbf{x}}_{0,t}, \mathbf{z})$ is the Euclidean distance between $\tilde{\mathbf{x}}_{0,t}$ and $\mathbf{z}$; $n(\tilde{\mathbf{x}}_{0,t}; \mathcal{Q})$ is the set of n-nearest neighbors of $\tilde{\mathbf{x}}_{0,t}$ in $\mathcal{Q}$ based on $d(\cdot)$; $\gamma > 0$ is a distance threshold. By doing the above adjustment, the predicted atom positions will be pushed to those of $\mathtt{M}_x$ if they are sufficiently far away. Note that the shape guidance is applied exclusively for steps

$$t = T, T-1, \cdots, S, \text{where } S > 1, \tag{5}$$

not for all the steps, and thus it only adjusts predicted atom positions when there are a lot of noises and the prediction needs more guidance. ShapeMol with the shape guidance is referred to as ShapeMol+g.

Table 1: Overall Comparison on Shape-Conditioned Molecule Generation

method	#v%	#s%	#u%	QED	avgSim$_s$(std)	avgSim$_g$(std)	maxSim$_s$(std)	maxSim$_g$(std)	div
VS	100.0	100.0	100.0	0.795	0.729 (0.039)	0.226 (0.038)	0.807 (0.042)	0.241 (0.087)	0.759
SQUID (λ=0.3)	100.0	100.0	94.2	0.766	0.717 (0.083)	0.349 (0.088)	0.904 (0.070)	0.549 (0.243)	0.677
SQUID (λ=1.0)	100.0	100.0	95.0	0.760	0.670 (0.069)	0.235 (0.045)	0.842 (0.061)	0.271 (0.096)	0.744
ShapeMol	99.6	98.8	100.0	0.748	0.689 (0.044)	0.239 (0.049)	0.803 (0.042)	0.243 (0.068)	0.712
ShapeMol+g	99.6	98.7	100.0	0.749	0.746 (0.036)	0.241 (0.050)	0.852 (0.034)	0.247 (0.068)	0.703

Columns represent: "#v%": the percentage of valid molecules; "#s%": the percentage of valid and complete molecules; "#u%": the percentage of unique molecules; "QED": the average drug-likeness of generated molecules; "avgSim$_s$/avgSim$_g$": the average of shape or graph similarities between condition molecules and generated molecules; "std": the standard deviation; "maxSim$_s$": the maximum of shape similarities between condition molecules and generated molecules; "maxSim$_g$": the graph similarities between condition molecules and generated molecules with highest shape similarities; "div": the diversity among generated molecules.

3 Experiments

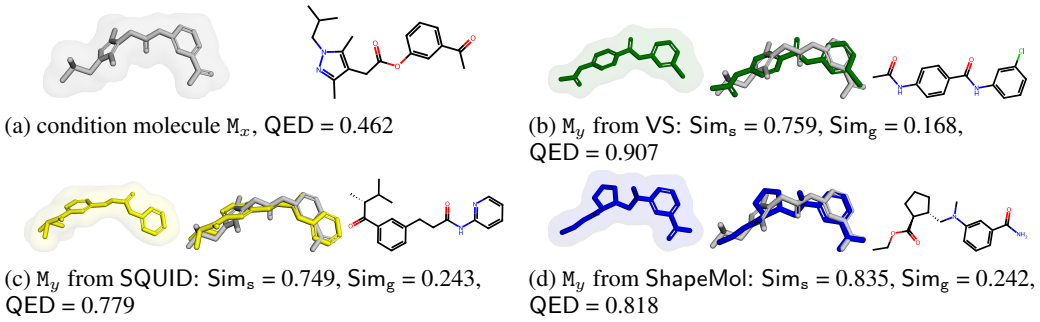
Experimental Setup We used molecules in the MOSES dataset [15], with their 3D conformers calculated by RDKit [16]. We used the same training and test split as in the previous work [17]. We compared ShapeMol and ShapeMol+g with the state-of-the-art baseline SQUID and a virtual screening method denoted as VS. The details about the dataset, baselines, and evaluation metrics are available in Supplementary Section S4. Detailed parameters in all the experiments, code, and data are reported in Supplementary Section S5. Additional experimental results are available in Supplementary Section S6, including a comparison between ShapeMol with the diffusion weighting scheme w_t^{x} (in Eq. 3) and the version without the scheme, a parameter study of shape guidance on distance threshold γ and step threshold S and an ablation study of ShapeMol with and without shape condition.

Experimental Results As shown in Table 1, ShapeMol+g achieves the highest average shape similarity 0.746$\pm$0.036, with 2.3% improvement from the best baseline VS (0.729$\pm$0.039), although at the cost of a slightly higher graph similarity. This indicates that ShapeMol+g could generate molecules that align more closely with the shape conditions than those in the dataset. Furthermore, ShapeMol+g achieves the second-best performance in maximum shape similarity maxSim$_s$ at 0.852$\pm$0.034 among all the methods. While it underperforms the best baseline (0.904$\pm$0.070 for SQUID with λ=0.3) on this metric, ShapeMol+g achieves substantially lower maximum graph similarity maxSim$_g$ of 0.247$\pm$0.068 compared with the best baseline (0.549$\pm$0.243). Please note that unlike SQUID, which neglects distorted bonding geometries in real molecules and limits itself to generating molecules with fixed bond lengths and angles and based on a pre-defined fragment library, both ShapeMol and ShapeMol+g are able to generate molecules without such limitations.

Figure 2 presents three generated molecules from three methods given the same condition molecule. As shown in Figure 2, the molecule generated by ShapeMol has higher shape similarity with the condition molecule than those from the baselines. Particularly, the molecule from ShapeMol has the surface shape (represented as blue shade in Figure 2d) most similar to that of the condition molecule. All three molecules have low graph similarities with the condition molecule and higher QED scores than the condition molecule. This example shows the ability of ShapeMol to generate novel molecules that are more similar in 3D shape to condition molecules than those from baselines.

4 Discussions and Conclusions

In this paper, we develop a novel generative model ShapeMol, which generates 3D molecules conditioned on the 3D shape of given molecules. ShapeMol utilizes a pre-trained equivariant shape encoder to generate equivariant embeddings for 3D shapes of given molecules. Conditioned on the embeddings, ShapeMol learns an equivariant diffusion model to generate novel molecules. To improve the shape similarities between the given molecule and the generated ones, we develop ShapeMol+g, which incorporates shape guidance to push the generated atom positions to the shape of the given molecule. We compare ShapeMol and ShapeMol+g against state-of-the-art baseline methods. Our experimental results demonstrate that ShapeMol and ShapeMol+g could generate molecules with higher shape similarities, and competitive qualities compared to the baseline methods.



(a) condition molecule $\mathtt{M}_x$, QED = 0.462

(b) $\mathtt{M}_y$ from VS: $\mathrm{Sim_s}$ = 0.759, $\mathrm{Sim_g}$ = 0.168, QED = 0.907

(c) $\mathtt{M}_y$ from SQUID: $\mathrm{Sim_s}$ = 0.749, $\mathrm{Sim_g}$ = 0.243, QED = 0.779

(d) $\mathtt{M}_y$ from ShapeMol: $\mathrm{Sim_s}$ = 0.835, $\mathrm{Sim_g}$ = 0.242, QED = 0.818

Figure 2: **Generated 3D Molecules and Their 2D Molecular Graphs from Different Methods.** Molecule shapes are in shades; generated molecules are superpositioned with the condition molecule.

References

[1] Shitong Luo, Jiaqi Guan, Jianzhu Ma, and Jian Peng. A 3d generative model for structure-based drug design. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[2] Xingang Peng, Shitong Luo, Jiaqi Guan, Qi Xie, Jian Peng, and Jianzhu Ma. Pocket2Mol: Efficient molecular sampling based on 3D protein pockets. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 17644–17655. PMLR, 17–23 Jul 2022.

[3] Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. In *The Eleventh International Conference on Learning Representations*, 2023.

[4] Maria Batool, Bilal Ahmad, and Sangdun Choi. A structure-based drug discovery paradigm. *International Journal of Molecular Sciences*, 20(11):2783, Jun 2019.

[5] Heping Zheng, Jing Hou, Matthew D Zimmerman, Alexander Wlodawer, and Wladek Minor. The future of crystallography in drug discovery. *Expert Opinion on Drug Discovery*, 9(2): 125–137, Dec 2013.

[6] Chayan Acharya, Andrew Coop, James E. Polli, and Alexander D. MacKerell. Recent advances in ligand-based drug design: Relevance and utility of the conformationally sampled pharmacophore approach. *Current Computer Aided-Drug Design*, 7(1):10–22, Mar 2011.

[7] Paul C. D. Hawkins, A. Geoffrey Skillman, and Anthony Nicholls. Comparison of shape-matching and docking as virtual screening tools. *Journal of Medicinal Chemistry*, 50(1):74–82, Dec 2006.

[8] Emiel Hoogeboom, Víctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3D. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8867–8887. PMLR, 17–23 Jul 2022.

[9] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022.

[10] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J. Guibas. Vector neurons: A general framework for so(3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12200–12209, Oct 2021.

[11] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.*, 38(5), Oct 2019. ISSN 0730-0301.

[12] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[14] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12454–12465. Curran Associates, Inc., 2021.

[15] Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alán Aspuru-Guzik, and Alex Zhavoronkov. Molecular sets (moses): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11, 2020. ISSN 1663-9812.

[16] Greg Landrum, Paolo Tosco, Brian Kelley, Ric, David Cosgrove, Sriniker, Gedeck, Riccardo Vianello, NadineSchneider, Eisuke Kawashima, Dan N, Gareth Jones, Andrew Dalke, Brian Cole, Matt Swain, Samo Turk, AlexanderSavelyev, Alain Vaucher, Maciej Wójcikowski, Ichiru Take, Daniel Probst, Kazuya Ujihara, Vincent F. Scalfani, Guillaume Godin, Juuso Lehtivarjo, Axel Pahl, Rachel Walker, Francois Berenger, Jasondbiggs, and Strets123. rdkit/rdkit: 2023_03_2 (q1 2023) release, 2023.

[17] Keir Adams and Connor W. Coley. Equivariant shape-conditioned generation of 3d molecules for ligand-based drug design. In *The Eleventh International Conference on Learning Representations*, 2023.

Shape-conditioned 3D Molecule Generation via Equivariant Diffusion Models (Supplementary Materials)

S1 Related Work

S1.1 Molecule Generation

A variety of deep generative models have been developed to generate molecules using various molecule representations, incliuding generating SMILES string representations [1], or 2D molecular graph representations [2, 3]. Recent efforts have been dedicated to the generation of 3D molecules. These 3D molecule generative models can be divided into two categories: autoregressive models and non-autoregressive models. Autoregressive models generate 3D molecules by sequentially adding atoms into the 3D space [4, 5]. While these models ensure the validity and connectivity of generated molecules, any errors made in sequential predictions could accumulate in subsequent predictions. Non-autoregressive models generate 3D molecules using flow-based methods [6] or diffusion methods [7–9]. In these models, atoms are generated or adjusted all together. For example, Hoogeboom *et al.* [8] developed an equivariant diffusion model, in which an equivariant network is employed to jointly predict both the positions and features of all atoms.

S1.2 Shape-Conditioned Molecule Generation

Following the idea of ligand-based drug design (LBDD) [10], previous work has been focused on generating molecules with similar 3D shapes to those of efficacy, based on the observation that structurally similar molecules tend to have similar properties. Papadopoulos *et al.* [11] developed a reinforcement learning method to generate SMILES strings of molecules that are similar to known antagonists of DRD2 receptors in 3D shapes and pharmacophores. Imrie *et al.* [12] generated 2D molecular graphs conditioned on 3D pharmacophores using a graph-based autoencoder. However, there is limited work that generates 3D molecule structures conditioned on 3D shapes. Adams and Coley [13] developed a shape-conditioned generative framework SQUID for 3D molecule generation. SQUID learns a variational autoencoder to generate fragments conditioned on given 3D shapes, and decodes molecules by sequentially attaching fragments with fixed bond lengths and angles. Long *et al.* [14] developed a two-stage framework to generate molecules binding to protein targets by first sketching the desired molecular shapes and then generating molecules corresponding to those shapes. To generate molecules, they pre-trained an autoencoder, referred to as Shape2Mol, which encodes the voxel grids of given 3D shapes and decodes the sequence of molecular fragments. While LBDD plays a vital role in drug discovery, the problem of generating 3D molecule structures conditioned on 3D shapes is still under-addressed.

S2 Equivariant Shape Embedding (SE)

S2.1 Shape Encoder (SE-enc)

SE-enc generates equivariant shape embeddings $\mathbf{H}^{\mathsf{s}}$ from the 3D surface shape $\mathcal{P}$ of molecules, such that $\mathbf{H}^{\mathsf{s}}$ is equivariant to both translation and rotation of $\mathcal{P}$. That is, any translation and rotation applied to $\mathcal{P}$ is reflected in $\mathbf{H}^{\mathsf{s}}$ accordingly. To ensure translation equivariance, SE-enc shifts the center of each $\mathcal{P}$ to zero to eliminate all translations. To ensure rotation equivariance, SE-enc leverages Vector Neurons (VNs) [15] and Dynamic Graph Convolutional Neural Networks (DGCNNs) [16] as follows:

$$\{\mathbf{H}^{\mathsf{p}}_1, \mathbf{H}^{\mathsf{p}}_2, \cdots, \mathbf{H}^{\mathsf{p}}_{|\mathcal{P}|}\} = \text{VN-DGCNN}(\{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_{|\mathcal{P}|}\}),$$

NeurIPS 2023 Generative AI and Biology Workshop.

$$\mathbf{H}^{\mathsf{s}} = \sum\nolimits_j \mathbf{H}^{\mathsf{p}}_j / |\mathcal{P}|,$$

where VN-DGCNN$(\cdot)$ is a VN-based DGCNN network to generate equivariant embedding $\mathbf{H}^{\mathsf{p}}_j \in \mathbb{R}^{d_p \times 3}$ for each point z_j in $\mathcal{P}$; and $\mathbf{H}^{\mathsf{s}} \in \mathbb{R}^{d_p \times 3}$ is the embedding of $\mathcal{P}$ generated via a mean-pooling over the embeddings of all the points. Note that VN-DGCNN$(\cdot)$ generates a matrix as the embedding of each point (i.e., $\mathbf{H}^{\mathsf{p}}_j$) to guarantee the equivariance.

S2.2 Shape Decoder (SE-dec)

To optimize $\mathbf{H}^{\mathsf{s}}$, following Deng *et al.* [15], SE learns a decoder SE-dec to predict the signed distance of a query point z_q sampled from 3D space using Multilayer Perceptrons (MLPs) as follows:

$$\tilde{o}_q = \text{MLP}(\text{concat}(\langle \mathbf{z}_q, \mathbf{H}^{\mathsf{s}} \rangle, \|\mathbf{z}_q\|^2, \text{VN-In}(\mathbf{H}^{\mathsf{s}}))), \tag{S1}$$

where $\tilde{o}_q$ is the predicted signed distance of z_q, with positive and negative values indicating z_q is inside or outside the surface shape, respectively; $\langle \cdot, \cdot \rangle$ is the dot-product operator; $\|\mathbf{z}_q\|^2$ is the Euclidean norm of the coordinates of z_q; VN-In$(\cdot)$ is an invariant VN network [15] that converts the equivariant shape embedding $\mathbf{H}^{\mathsf{s}} \in \mathbb{R}^{d_p \times 3}$ into an invariant shape embedding. Thus, SE-dec predicts the signed distance between the query point and 3D surface by jointly considering the position of the query point ($\|\mathbf{z}_q\|^2$), the molecular surface shape (VN-In$(\cdot)$) and the interaction between the point and surface $\langle \cdot, \cdot \rangle$. The predicted signed distance $\tilde{o}_q$ is used to calculate the loss for the optimization of $\mathbf{H}^{\mathsf{s}}$ (discussed below). As shown in the literature [15], $\tilde{o}_q$ remains invariant to the rotation of the 3D molecule surface shapes (i.e., $\mathcal{P}$). We present the sampling process of z_q in the Supplementary Section S2.5.

S2.3 SE Pre-training

ShapeMol pre-trains SE by minimizing the squared-errors loss between the predicted and the ground-truth signed distances of query points as follows:

$$\mathcal{L}^{\mathsf{s}} = \sum\nolimits_{z_q \in \mathcal{Z}} \|o_q - \tilde{o}_q\|^2, \tag{S2}$$

where $\mathcal{Z}$ is the set of sampled query points and o_q is the ground-truth signed distance of query point z_q. By pretraining SE, ShapeMol learns $\mathbf{H}^{\mathsf{s}}$ that will be used as the condition in the following 3D molecule generation.

S2.4 Point Cloud Construction

In ShapeMol, we represented molecular surface shapes using point clouds ($\mathcal{P}$). $\mathcal{P}$ serves as input to ShapeMol-enc, from which we derive shape latent embeddings. To generate $\mathcal{P}$, we initially generated a molecular surface mesh using the algorithm from the Open Drug Discovery Toolkit [17]. Following this, we uniformly sampled points on the mesh surface with probability proportional to the face area, using the algorithm from PyTorch3D [18]. This point cloud $\mathcal{P}$ is then centralized by setting the center of its points to zero.

S2.5 Query Point Sampling

As described in Supplementary Section S2.2, the signed distances of query points z_q to molecule surface shape $\mathcal{P}$ are used to optimize SE. In this section, we present how to sample these points z_q in 3D space. Particularly, we first determined the bounding box around the molecular surface shape, using the maximum and minimum (x, y, z)-axis coordinates for points in our point cloud $\mathcal{P}$, denoted as $(x_{\text{min}}, y_{\text{min}}, z_{\text{min}})$ and $(x_{\text{max}}, y_{\text{max}}, z_{\text{max}})$. We extended this box slightly by defining its corners as $(x_{\text{min}} - 1, y_{\text{min}} - 1, z_{\text{min}} - 1)$ and $(x_{\text{max}} + 1, y_{\text{max}} + 1, z_{\text{max}} + 1)$. For sampling $|\mathcal{Z}|$ query points, we wanted an even distribution of points inside and outside the molecule surface shape. When a bounding box is defined around the molecule surface shape, there could be a lot of empty spaces within the box that the molecule does not occupy due to its complex and irregular shape. This could lead to that fewer points within the molecule surface shape could be sampled within the box. Therefore, we started by randomly sampling $3k$ points within our bounding box to ensure that there are sufficient points within the surface. We then determined whether each point lies within the molecular surface, using an algorithm from Trimesh [1] based on the molecule surface mesh. If there are n_w points found

[1] https://trimsh.org/

within the surface, we selected $n = \min(n_w, k/2)$ points from these points, and randomly choose the remaining $k - n$ points from those outside the surface. For each query point, we determined its signed distance to the molecule surface by its closest distance to points in $\mathcal{P}$ with a sign indicating whether it is inside the surface.

S3 Shape-Conditioned Molecule Generation

S3.1 Forward Diffusion Process (DIFF-forward)

Following the previous work [9], at step $t \in [1, T]$, a small Gaussian noise and a small categorical noise are added to the continuous atom positions and discrete atom features $\{(\mathbf{x}_{i,t-1}, \mathbf{v}_{i,t-1})\}$, respectively. When no ambiguity arises, we will eliminate subscript i in the notations and use $(\mathbf{x}_{t-1}, \mathbf{v}_{t-1})$ for brevity. The noise levels of the Gaussian and categorical noises are determined by two predefined variance schedules $(\beta_t^{\mathtt{x}}, \beta_t^{\mathtt{v}}) \in (0, 1)$, where $\beta_t^{\mathtt{x}}$ and $\beta_t^{\mathtt{v}}$ are selected to be sufficiently small to ensure the smoothness of DIFF-forward. The details about variance schedules are available in Supplementary Section S3.3. Formally, for atom positions, the probability of $\mathbf{x}_t$ sampled given $\mathbf{x}_{t-1}$, denoted as $q(\mathbf{x}_t|\mathbf{x}_{t-1})$, is defined as follows,

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t|\sqrt{1-\beta_t^{\mathtt{x}}}\mathbf{x}_{t-1}, \beta_t^{\mathtt{x}}\mathbf{I}), \tag{S3}$$

where $\mathcal{N}(\cdot)$ is a Gaussian distribution of $\mathbf{x}_t$ with mean $\sqrt{1-\beta_t^{\mathtt{x}}}\mathbf{x}_{t-1}$ and covariance $\beta_t^{\mathtt{x}}\mathbf{I}$. Following Hoogeboom *et al.* Hoogeboom et al. [19], for atom features, the probability of $\mathbf{v}_t$ across K classes given $\mathbf{v}_{t-1}$ is defined as follows,

$$q(\mathbf{v}_t|\mathbf{v}_{t-1}) = \mathcal{C}(\mathbf{v}_t|(1-\beta_t^{\mathtt{v}})\mathbf{v}_{t-1} + \beta_t^{\mathtt{v}}\mathbf{1}/K), \tag{S4}$$

where $\mathcal{C}$ is a categorical distribution of $\mathbf{v}_t$ derived by noising $\mathbf{v}_{t-1}$ with a uniform noise $\beta_t^{\mathtt{v}}\mathbf{1}/K$ across K classes.

Since the above distributions form Markov chains, the probability of any $\mathbf{x}_t$ or $\mathbf{v}_t$ can be derived from $\mathbf{x}_0$ or $\mathbf{v}_0$:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t|\sqrt{\bar{\alpha}_t^{\mathtt{x}}}\mathbf{x}_0, (1-\bar{\alpha}_t^{\mathtt{x}})\mathbf{I}), \tag{S5}$$

$$q(\mathbf{v}_t|\mathbf{v}_0) = \mathcal{C}(\mathbf{v}_t|\bar{\alpha}_t^{\mathtt{v}}\mathbf{v}_0 + (1-\bar{\alpha}_t^{\mathtt{v}})\mathbf{1}/K), \tag{S6}$$

$$\text{where } \bar{\alpha}_t^{\mathtt{u}} = \prod_{\tau=1}^{t} \alpha_\tau^{\mathtt{u}},\ \alpha_\tau^{\mathtt{u}} = 1 - \beta_\tau^{\mathtt{u}},\ \mathtt{u} = \mathtt{x} \text{ or } \mathtt{v}. \tag{S7}$$

Note that $\bar{\alpha}_t^{\mathtt{u}}$ ($\mathtt{u} = \mathtt{x}$ or $\mathtt{v}$) is monotonically decreasing from 1 to 0 over $t = [1, T]$. As $t \to 1$, $\bar{\alpha}_t^{\mathtt{x}}$ and $\bar{\alpha}_t^{\mathtt{v}}$ are close to 1, leading to that $\mathbf{x}_t$ or $\mathbf{v}_t$ approximates $\mathbf{x}_0$ or $\mathbf{v}_0$. Conversely, as $t \to T$, $\bar{\alpha}_t^{\mathtt{x}}$ and $\bar{\alpha}_t^{\mathtt{v}}$ are close to 0, leading to that $q(\mathbf{x}_T|\mathbf{x}_0)$ resembles $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and $q(\mathbf{v}_T|\mathbf{v}_0)$ resembles $\mathcal{C}(\mathbf{1}/K)$.

Using Bayes theorem, the ground-truth Normal posterior of atom positions $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ can be calculated in a closed-form [20] as below,

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}|\mu(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t^{\mathtt{x}}\mathbf{I}), \tag{S8}$$

$$\mu(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}^{\mathtt{x}}}\beta_t^{\mathtt{x}}}{1-\bar{\alpha}_t^{\mathtt{x}}}\mathbf{x}_0 + \frac{\sqrt{\alpha_t^{\mathtt{x}}}(1-\bar{\alpha}_{t-1}^{\mathtt{x}})}{1-\bar{\alpha}_t^{\mathtt{x}}}\mathbf{x}_t,\ \tilde{\beta}_t^{\mathtt{x}} = \frac{1-\bar{\alpha}_{t-1}^{\mathtt{x}}}{1-\bar{\alpha}_t^{\mathtt{x}}}\beta_t^{\mathtt{x}}. \tag{S9}$$

Similarly, the ground-truth categorical posterior of atom features $p(\mathbf{v}_{t-1}|\mathbf{v}_t, \mathbf{v}_0)$ can be calculated [19] as below,

$$p(\mathbf{v}_{t-1}|\mathbf{v}_t, \mathbf{v}_0) = \mathcal{C}(\mathbf{v}_{t-1}|\mathbf{c}(\mathbf{v}_t, \mathbf{v}_0)), \tag{S10}$$

$$\mathbf{c}(\mathbf{v}_t, \mathbf{v}_0) = \tilde{\mathbf{c}}/\textstyle\sum_{k=1}^{K}\tilde{c}_k, \tag{S11}$$

$$\tilde{\mathbf{c}} = [\alpha_t^{\mathtt{v}}\mathbf{v}_t + \tfrac{1-\alpha_t^{\mathtt{v}}}{K}] \odot [\bar{\alpha}_{t-1}^{\mathtt{v}}\mathbf{v}_0 + \tfrac{1-\bar{\alpha}_{t-1}^{\mathtt{v}}}{K}], \tag{S12}$$

where $\tilde{c}_k$ denotes the likelihood of k-th class across K classes in $\tilde{\mathbf{c}}$; $\odot$ denotes the element-wise product operation; $\tilde{\mathbf{c}}$ is calculated using $\mathbf{v}_t$ and $\mathbf{v}_0$ and normalized so as to represent probabilities. The proof of the above equations is available in Supplementary Section S3.4.

S3.2 Backward Generative Process (DIFF-backward)

DIFF learns to reverse DIFF-forward by denoising from $(\mathbf{x}_t, \mathbf{v}_t)$ to $(\mathbf{x}_{t-1}, \mathbf{v}_{t-1})$ at $t \in [1, T]$, conditioned on the shape latent embedding $\mathbf{H}^{\mathtt{s}}$. Specifically, the probabilities of $(\mathbf{x}_{t-1}, \mathbf{v}_{t-1})$ denoised from $(\mathbf{x}_t, \mathbf{v}_t)$ are estimated by the approximates of the ground-truth posteriors $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$

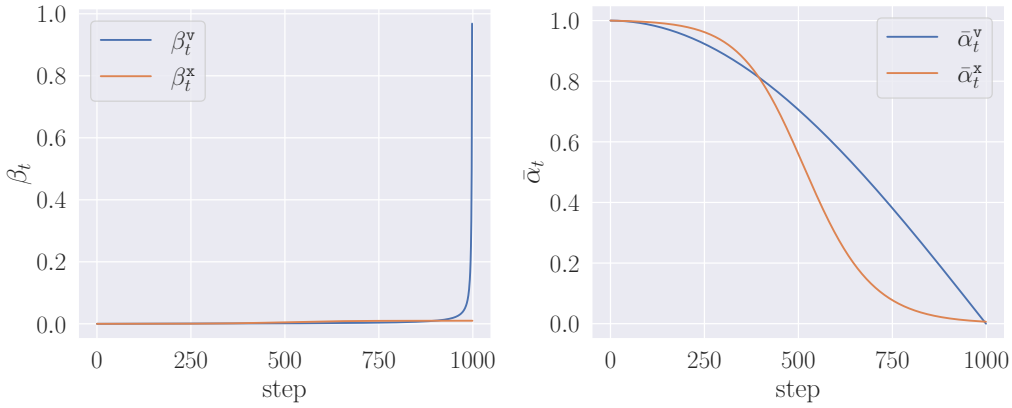


Figure S1: Schedule

(Eq. S8) and $p(\mathbf{v}_{t-1}|\mathbf{v}_t, \mathbf{v}_0)$ (Eq. S10). Given that $(\mathbf{x}_0, \mathbf{v}_0)$ is unknown in the generative process, a predictor $f_{\boldsymbol{\Theta}}(\mathbf{x}_t, \mathbf{v}_t, \mathbf{H}^{\mathrm{s}})$ is employed to predict at t the atom position and feature $(\mathbf{x}_0, \mathbf{v}_0)$ as below,

$$(\tilde{\mathbf{x}}_{0,t}, \tilde{\mathbf{v}}_{0,t}) = f_{\boldsymbol{\Theta}}(\mathbf{x}_t, \mathbf{v}_t, \mathbf{H}^{\mathrm{s}}), \tag{S13}$$

where $\tilde{\mathbf{x}}_{0,t}$ and $\tilde{\mathbf{v}}_{0,t}$ are the predictions of $\mathbf{x}_0$ and $\mathbf{v}_0$ at t; $\boldsymbol{\Theta}$ is the learnable parameter. Following Ho *et al.* [20], with $\tilde{\mathbf{x}}_{0,t}$, the probability of $\mathbf{x}_{t-1}$ denoised from $\mathbf{x}_t$, denoted as $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$, can be estimated by the approximated posterior $p_{\boldsymbol{\Theta}}((\mathbf{x}_{t-1}|\mathbf{x}_t, \tilde{\mathbf{x}}_{0,t})$ as below,

$$\begin{aligned} p(\mathbf{x}_{t-1}|\mathbf{x}_t) &\approx p_{\boldsymbol{\Theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \tilde{\mathbf{x}}_{0,t}) \\ &= \mathcal{N}(\mathbf{x}_{t-1}|\mu_{\boldsymbol{\Theta}}(\mathbf{x}_t, \tilde{\mathbf{x}}_{0,t}), \tilde{\beta}_t^{\mathrm{x}}\mathbf{I}), \end{aligned} \tag{S14}$$

where $\mu_{\boldsymbol{\Theta}}(\mathbf{x}_t, \tilde{\mathbf{x}}_{0,t})$ is an estimate of $\mu(\mathbf{x}_t, \mathbf{x}_0)$ by replacing $\mathbf{x}_0$ with its estimate $\tilde{\mathbf{x}}_{0,t}$ in Eq. S8. Similarly, with $\tilde{\mathbf{v}}_{0,t}$, the probability of $\mathbf{v}_{t-1}$ denoised from $\mathbf{v}_t$, denoted as $p(\mathbf{v}_{t-1}|\mathbf{v}_t)$, can be estimated by the approximated posterior $p_{\boldsymbol{\Theta}}(\mathbf{v}_{t-1}|\mathbf{v}_t, \tilde{\mathbf{v}}_{0,t})$ as below,

$$p(\mathbf{v}_{t-1}|\mathbf{v}_t) \approx p_{\boldsymbol{\Theta}}(\mathbf{v}_{t-1}|\mathbf{v}_t, \tilde{\mathbf{v}}_{0,t}) = \mathcal{C}(\mathbf{v}_{t-1}|\mathbf{c}_{\boldsymbol{\Theta}}(\mathbf{v}_t, \tilde{\mathbf{v}}_{0,t})), \tag{S15}$$

where $\mathbf{c}_{\boldsymbol{\Theta}}(\mathbf{v}_t, \tilde{\mathbf{v}}_{0,t})$ is an estimate of $\mathbf{c}(\mathbf{v}_t, \mathbf{v}_0)$ by replacing $\mathbf{v}_0$ with its estimate $\tilde{\mathbf{v}}_{0,t}$ in Eq. S10.

S3.3 Variance Scheduling in DIFF-forward

Following Guan *et al.* [9], we used a sigmoid β schedule for the variance schedule β_t^{x} of atom coordinates as below,

$$\beta_t^{\mathrm{x}} = \mathrm{sigmoid}(w_1(2t/T - 1))(w_2 - w_3) + w_3, \tag{S16}$$

in which w_i(i=1,2, or 3) are hyperparameters; T is the maximum step. We set $w_1 = 6$, $w_2 = 1.e-7$ and $w_3 = 0.01$. For atom types, we used a cosine β schedule [21] for β_t^{v} as below,

$$\begin{aligned} \bar{\alpha}_t^{\mathrm{v}} &= \frac{f(t)}{f(0)}, f(t) = \cos(\frac{t/T + s}{1+s} \cdot \frac{\pi}{2})^2, \\ \beta_t^{\mathrm{v}} &= 1 - \alpha_t^{\mathrm{v}} = 1 - \frac{\bar{\alpha}_t^{\mathrm{v}}}{\bar{\alpha}_{t-1}^{\mathrm{v}}}, \end{aligned} \tag{S17}$$

in which s is a hyperparameter and set as 0.01.

As shown in Supplementary Section S3.1, the values of β_t^{x} and β_t^{v} should be sufficiently small to ensure the smoothness of forward diffusion process. In the meanwhile, their corresponding $\bar{\alpha}_t$ values should decrease from 1 to 0 over $t = [1, T]$. Figure S1 shows the values of β_t and $\bar{\alpha}_t$ for atom coordinates and atom types with our hyperparameters. Please note that the value of β_t^{x} is less than 0.1 for 990 out of 1,000 steps. This guarantees the smoothness of the forward diffusion process.

S3.4 Derivation of Forward Diffusion Process

In ShapeMol, a Gaussian noise and a categorical noise are added to continuous atom position and discrete atom features, respectively. Here, we briefly describe the derivation of posterior equations (i.e., Eq. S8, and S10) for atom positions and atom types in our work. We refer readers to Ho *et al.* [20] and Kong *et al.* [22] for a detailed description of diffusion process for continuous variables and Hoogeboom *et al.* [19] for the description of diffusion process for discrete variables.

For continuous atom positions, as shown in Kong *et al.* [22], according to Bayes theorem, given $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ defined in Eq. S3 and $q(\mathbf{x}_t|\mathbf{x}_0)$ defined in Eq. S5, the posterior $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ is derived as below (superscript $\mathtt{x}$ is omitted for brevity),

$$
\begin{aligned}
q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) &= \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\
&= \frac{\mathcal{N}(\mathbf{x}_t|\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})\mathcal{N}(\mathbf{x}_{t-1}|\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I})}{\mathcal{N}(\mathbf{x}_t|\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})} \\
&= (2\pi\beta_t)^{-\frac{3}{2}}(2\pi(1-\bar{\alpha}_{t-1}))^{-\frac{3}{2}}(2\pi(1-\bar{\alpha}_t))^{\frac{3}{2}} \\
&\times \exp(-\frac{\|\mathbf{x}_t - \sqrt{\alpha}_t\mathbf{x}_{t-1}\|^2}{2\beta_t} - \frac{\|\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}}_{t-1}\mathbf{x}_0\|^2}{2(1-\bar{\alpha}_{t-1})} + \frac{\|\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0\|^2}{2(1-\bar{\alpha}_t)}) \\
&= (2\pi\tilde{\beta}_t)^{-\frac{3}{2}}\exp(-\frac{1}{2\tilde{\beta}_t}\|\mathbf{x}_{t-1} - \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 - \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t\|^2), \\
&\text{where } \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t.
\end{aligned}
\tag{S18}
$$

Therefore, the posterior of atom positions is derived as below,

$$
q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}|\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t, \tilde{\beta}_t\mathbf{I}). \tag{S19}
$$

For discrete atom features, as shown in Hoogeboom *et al.* [19] and Guan *et al.* [9], according to Bayes theorem, the posterior $q(\mathbf{v}_{t-1}|\mathbf{v}_t, \mathbf{v}_0)$ is derived as below (supperscript $\mathtt{v}$ is omitted for brevity),

$$
q(\mathbf{v}_{t-1}|\mathbf{v}_t, \mathbf{v}_0) = \frac{q(\mathbf{v}_t|\mathbf{v}_{t-1}, \mathbf{v}_0)q(\mathbf{v}_{t-1}|\mathbf{v}_0)}{\sum_{\mathbf{v}_{t-1}} q(\mathbf{v}_t|\mathbf{v}_{t-1}, \mathbf{v}_0)q(\mathbf{v}_{t-1}|\mathbf{v}_0)}. \tag{S20}
$$

For $q(\mathbf{v}_t|\mathbf{v}_{t-1}, \mathbf{v}_0)$, we have

$$
\begin{aligned}
q(\mathbf{v}_t|\mathbf{v}_{t-1}, \mathbf{v}_0) &= \mathcal{C}(\mathbf{v}_t|(1-\beta_t)\mathbf{v}_{t-1} + \beta_t/K) \\
&= \begin{cases} 1-\beta_t+\beta_t/K, & \text{when } \mathbf{v}_t = \mathbf{v}_{t-1}, \\ \beta_t/K, & \text{when } \mathbf{v}_t \neq \mathbf{v}_{t-1}, \end{cases} \\
&= \mathcal{C}(\mathbf{v}_{t-1}|(1-\beta_t)\mathbf{v}_t + \beta_t/K).
\end{aligned}
\tag{S21}
$$

Therefore, we have

$$
\begin{aligned}
&q(\mathbf{v}_t|\mathbf{v}_{t-1}, \mathbf{v}_0)q(\mathbf{v}_{t-1}|\mathbf{v}_0) \\
&= \mathcal{C}(\mathbf{v}_{t-1}|(1-\beta_t)\mathbf{v}_t + \beta_t\frac{\mathbf{1}}{K})\mathcal{C}(\mathbf{v}_{t-1}|\bar{\alpha}_{t-1}\mathbf{v}_0 + (1-\bar{\alpha}_{t-1})\frac{\mathbf{1}}{K}) \\
&= [\alpha_t\mathbf{v}_t + \frac{1-\alpha_t}{K}] \odot [\bar{\alpha}_{t-1}\mathbf{v}_0 + \frac{1-\bar{\alpha}_{t-1}}{K}].
\end{aligned}
\tag{S22}
$$

Therefore, with $q(\mathbf{v}_t|\mathbf{v}_{t-1}, \mathbf{v}_0)q(\mathbf{v}_{t-1}|\mathbf{v}_0) = \tilde{\mathbf{c}}$, the posterior is as below,

$$
q(\mathbf{v}_{t-1}|\mathbf{v}_t, \mathbf{v}_0) = \mathcal{C}(\mathbf{v}_{t-1}|\mathbf{c}(\mathbf{v}_t, \mathbf{v}_0)) = \frac{\tilde{\mathbf{c}}}{\sum_k^K \tilde{c}_k}. \tag{S23}
$$

S3.5 Proof of Equivariance in Atom Coordinate Prediction

In this section, we prove that our design in Eq. 1 in the main manuscript updates atom coordinates in an equivariant way. A function $f(\mathbf{x})$ is equivariant with respect to rotation, if given any rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$, it satisfies,

$$f(\mathbf{R}\mathbf{x}) = \mathbf{R}f(\mathbf{x}). \tag{S24}$$

In Eq. 1, the embedding $\Delta\mathbf{x}_i^{l+1} \in \mathbb{R}^{n_h\times3}$, which aggregates the neighborhood information of a_i, is updated by atom coordinates $\mathbf{x}_i^l$ from previous layer and embeddings from multi-head attention layer $\text{MHA}^{\mathtt{x}}(\cdot)$ as below,

$$\begin{aligned}
\Delta\mathbf{x}_i^{l+1} &= \sum_{j\in N(a_i), i\neq j}(\mathbf{x}_i^l - \mathbf{x}_j^l)\text{MHA}^{\mathtt{x}}(d_{ij}^l, \mathbf{h}_i^{l+1}, \mathbf{h}_j^{l+1}, \text{VN-In}(\mathbf{H}^{\mathtt{s}})),\\
\mathbf{x}_i^{l+1} &= \mathbf{x}_i^l + \text{Mean}(\Delta\mathbf{x}_i^{l+1}) + \text{VN-Lin}(\mathbf{x}_i^l, \Delta\mathbf{x}_i^{l+1}, \mathbf{H}^{\mathtt{s}}).
\end{aligned}$$

The embeddings from $\text{MHA}^{\mathtt{x}}(\cdot)$ and atom feature embeddings $\mathbf{h}_i^l$ will be proved to be invariant in Supplementary Section S3.6 later. Therefore, $\text{MHA}^{\mathtt{x}}(\cdot)$ and $\mathbf{h}_i^{l+1}$ remain the same with rotation $\mathbf{R}$. Then, we can prove that $\Delta\mathbf{x}_i^{l+1}$ is updated in an equivariant way as below,

$$\begin{aligned}
&\sum_{j\in N(a_i), i\neq j}(\mathbf{R}\mathbf{x}_i^l - \mathbf{R}\mathbf{x}_j^l)\text{MHA}^{\mathtt{x}}(d_{ij}^l, \mathbf{h}_i^{l+1}, \mathbf{h}_j^{l+1}, \text{VN-In}(\mathbf{H}^{\mathtt{s}}))\\
&= \mathbf{R}\sum_{j\in N(a_i), i\neq j}(\mathbf{x}_i^l - \mathbf{x}_j^l)\text{MHA}^{\mathtt{x}}(d_{ij}^l, \mathbf{h}_i^{l+1}, \mathbf{h}_j^{l+1}, \text{VN-In}(\mathbf{H}^{\mathtt{s}}))\\
&= \mathbf{R}\Delta\mathbf{x}_i^{l+1}.
\end{aligned}$$

Given that mean pooling is a linear operation that respects rotation equivariance, $\text{Mean}(\Delta\mathbf{x}_i^{l+1})$ is equivariant to rotations. The vector-neuron-based linear layer $\text{VN-Lin}(\cdot)$ with leaky-ReLu activation function also produces rotational equivariant embeddings according to Deng *et al.* [15]. Therefore, we prove that atom coordinates $\mathbf{x}_i^{l+1}$ are updated in an equivariant way as below,

$$\begin{aligned}
&\mathbf{R}\mathbf{x}_i^l + \text{Mean}(\mathbf{R}\Delta\mathbf{x}_i^{l+1}) + \text{VN-Lin}(\mathbf{R}\mathbf{x}_i^l, \mathbf{R}\Delta\mathbf{x}_i^{l+1}, \mathbf{R}\mathbf{H}^{\mathtt{s}}))\\
&= \mathbf{R}\mathbf{x}_i^l + \mathbf{R}\text{Mean}(\Delta\mathbf{x}_i^{l+1}) + \mathbf{R}\text{VN-Lin}(\mathbf{x}_i^l, \Delta\mathbf{x}_i^{l+1}, \mathbf{H}^{\mathtt{s}})\\
&= \mathbf{R}(\mathbf{x}_i^l + \text{Mean}(\Delta\mathbf{x}_i^{l+1}) + \text{VN-Lin}(\mathbf{x}_i^l, \Delta\mathbf{x}_i^{l+1}, \mathbf{H}^{\mathtt{s}}))\\
&= \mathbf{R}\mathbf{x}_i^{l+1}
\end{aligned} \tag{S25}$$

S3.6 Proof of Invariance in Atom Feature Prediction

In this section, we prove that our design in Eq. 2 in the main manuscript updates atom features in an invariant way. A function $f(\mathbf{x})$ is invariant with respect to rotation, if it satisfies,

$$f(\mathbf{R}\mathbf{x}) = f(\mathbf{x}). \tag{S26}$$

Recall that in Eq. 2, atom feature embeddings $\mathbf{h}_i^{l+1}$ is updated according to atomic distances d_{ij}^l, embeddings $\mathbf{h}_i$ from previous layer and invariant shape embeddings $\text{VN-In}(\mathbf{H}^{\mathtt{s}})$ from the vector-neuron based invariant layer adapted by [23] as follows,

$$\mathbf{h}_i^{l+1} = \mathbf{h}_i^l + \sum_{j\in N(a_i), i\neq j}\text{MHA}^{\mathtt{h}}(d_{ij}^l, \mathbf{h}_i^l, \mathbf{h}_j^l, \text{VN-In}(\mathbf{H}^{\mathtt{s}})), \mathbf{h}_i^0 = \mathbf{v}_i.$$

Atomic distances are invariant to transformations. Atom feature embeddings are initialized with atom feature vectors, which are also invariant to transformations. The $\text{VN-In}(.)$ layer maps the equivariant shape embedding $\mathbf{H}^{\mathtt{s}} \in \mathbb{R}^{d_p\times3}$ into the invariant shape embedding $\mathbf{h}^{\mathtt{s}} \in \mathbb{R}^{d_p}$ as below,

$$\mathbf{h}_c^{\mathtt{s}} = \text{MLP}(\langle\mathbf{H}_c^{\mathtt{s}}, \frac{\bar{\mathbf{H}}^{\mathtt{s}}}{\|\bar{\mathbf{H}}^{\mathtt{s}}\|}\rangle).$$

where $\bar{\mathbf{H}}^{\mathtt{s}} = \frac{1}{d_p}\sum_{c=1}^{d_p}\mathbf{H}_c^{\mathtt{s}}$ and $\mathbf{H}_c^{\mathtt{s}}$ is the c-th channel of $\mathbf{H}^{\mathtt{s}}$; $\mathbf{h}_c^{\mathtt{s}}$ denotes the c-th channel of $\mathbf{h}^{\mathtt{s}}$; $\langle\cdot\rangle$ is the inner-product operation which the results will not be changed by any rotation vector. Given

that all the variables in the right-hand side of Eq. 2 are invariant to rotations, we prove that the atom feature embeddings are updated in an invariant way. Therefore, the invariant atom feature embeddings lead to invariant atom feature predictions. Similarly, we can prove that the embeddings from MHA^{x} layer (Eq. 1) are invariant to rotations.

S3.7 ShapeMol Loss Function Derivation

In this section, we demonstrate that a step weight w_t^{x} based on the signal-to-noise ratio λ_t should be included into the atom position loss (Eq. 3). In the diffusion process for continuous variables, the optimization problem is defined as below [20],

$$\begin{aligned}
&\arg\min_{\boldsymbol{\Theta}} KL(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)|p_{\boldsymbol{\Theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \tilde{\mathbf{x}}_{0,t})) \\
&= \arg\min_{\boldsymbol{\Theta}} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)}{2(1-\bar{\alpha}_{t-1})(1-\bar{\alpha}_t)} \|\tilde{\mathbf{x}}_{0,t} - \mathbf{x}_0\|^2 \\
&= \arg\min_{\boldsymbol{\Theta}} \frac{1-\alpha_t}{2(1-\bar{\alpha}_{t-1})\alpha_t} \|\tilde{\boldsymbol{\epsilon}}_{0,t} - \boldsymbol{\epsilon}_0\|^2,
\end{aligned}$$

where $\boldsymbol{\epsilon}_0 = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}$ is the ground-truth noise variable sampled from $\mathcal{N}(\mathbf{0}, \mathbf{1})$ and is used to sample $\mathbf{x}_t$ from $\mathcal{N}(\mathbf{x}_t|\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$ in Eq. S6; $\tilde{\boldsymbol{\epsilon}}_0 = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\tilde{\mathbf{x}}_{0,t}}{\sqrt{1-\bar{\alpha}_t}}$ is the predicted noise variable.

Ho *et al.* [20] further simplified the above objective as below and demonstrated that the simplified one can achieve better performance:

$$\arg\min_{\boldsymbol{\Theta}} \|\tilde{\boldsymbol{\epsilon}}_{0,t} - \boldsymbol{\epsilon}_0\|^2 = \arg\min_{\boldsymbol{\Theta}} \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t} \|\tilde{\mathbf{x}}_{0,t} - \mathbf{x}_0\|^2, \tag{S27}$$

where $\lambda_t = \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}$ is the signal-to-noise ratio. While previous work [9] applies uniform step weights across different steps, we demonstrate that a step weight should be included into the atom position loss according to Eq. S27. However, the value of λ_t could be very large when $\bar{\alpha}_t$ is close to 1 as t approaches 1. Therefore, we clip the value of λ_t with threshold δ in Eq. 3.

S4 Experimental Setup

S4.1 Data

Following SQUID [13], we used molecules in the MOSES dataset [24], with their 3D conformers calculated by RDKit [25]. We used the same training and test split as in SQUID. Please note that SQUID further modifies the generated conformers into artificial ones, by adjusting acyclic bond distances to their empirical means and fixing acyclic bond angles using heuristic rules. Unlike SQUID, we did not make any additional adjustments to the calculated 3D conformers, as ShapeMol is designed with sufficient flexibility to accept any 3D conformers as input and generate 3D molecules without restrictions on fixed bond lengths or angles. Limited by the predefined fragment library, SQUID also removes molecules with fragments not present in its fragment library. In contrast, we kept all the molecules, as ShapeMol is not based on fragments. Our final training dataset contains 1,593,653 molecules, out of which a random set of 1,000 molecules was selected for validation. Both the SE and DIFF models are trained using this training set. 1,000 test molecules (i.e., conditions) as used in SQUID are used to test ShapeMol.

S4.2 Baselines

We compared ShapeMol and ShapeMol+g with the state-of-the-art baseline SQUID and a virtual screening method over the training dataset, denoted as VS. SQUID consists of a fragment-based generative model based on variational autoencoder that sequentially decodes fragments from molecule latent embeddings and shape embeddings, and a rotatable bond scoring framework that adjusts the angles of rotatable bonds between fragments to maximize the 3D shape similarity with the condition molecule. VS aims to sift through the training set to identify molecules with high shape similarities with the condition molecule. For SQUID, we assessed two interpolation levels, $\lambda = 0.3$ and 1.0 (prior), following the original SQUID paper [13]. For SQUID, ShapeMol and ShapeMol+g, we

generated 50 molecules for each testing molecule (i.e., condition) as the candidates for evaluation. For VS, we randomly sampled 500 training molecules for each testing molecule, and considered the top-50 molecules with the highest shape similarities as candidates for evaluation. Note that we did not consider Shape2Mol [14] as our baseline. The code they provided is closely tied to Bytedance infrastructure [2]. This infrastructure is not publicly available, which makes it difficult to run their code on other infrastructures. Moreover, the training of Shape2Mol requires the use of 32 Tesla V100 GPUs for 2 weeks as reported in their paper. Both of these factors make it infeasible for us to train and test Shape2Mol on our dataset with our infrastructure and hardware resources.

S4.3 Evaluation Metrics

We calculated the shape similarity $\mathsf{Sim}_\mathtt{s}(\mathtt{s}_x, \mathtt{s}_y)$ via the overlap volumes between two aligned molecules as in the literature [13]. Each molecule candidate $\mathtt{M}_y$ for evaluation is aligned with the condition molecule $\mathtt{M}_x$ by the ShaEP tool [26]. For the molecular graph similarity $\mathsf{Sim}_\mathtt{g}(\mathtt{M}_x, \mathtt{M}_y)$, we used the Tanimoto similarity over Morgan fingerprints between $\mathtt{M}_x$ and $\mathtt{M}_y$ calculated by RDKit [25].

S5 Parameters for Reproducibility

We implemented both SE and DIFF using Python-3.7.16, PyTorch-1.11.0, PyTorch-scatter-2.0.9, Numpy-1.21.5, Scikit-learn-1.0.2. We trained the models using a Tesla V100 GPU with 32GB memory and a CPU with 80GB memory on Red Hat Enterprise 7.7. We released the code, data, and the trained model at Google Drive [3].

S5.1 Parameters of SE

In SE, we tuned the dimension of all the hidden layers including VN-DGCNN layers (Eq. S2.1), MLP layers (Eq. S1) and VN-In layer (Eq. S1), and the dimension d_p of generated shape latent embeddings $\mathbf{H}^\mathbf{s}$ with the grid-search algorithm in the parameter space presented in Table S1. We determined the optimal hyper-parameters according to the mean squared errors of the predictions of signed distances for 1,000 validation molecules that are selected as described in Supplementary Section S4.1. The optimal dimension of all the hidden layers is 128, and the optimal dimension d_p of shape latent embedding $\mathbf{H}^\mathbf{s}$ is 32. The optimal number of points $|\mathcal{P}|$ in the point cloud $\mathcal{P}$ is 512. We sampled 1,024 query points in $\mathcal{Z}$ for each molecule shape. We constructed graphs from point clouds, which are employed to learn $\mathbf{H}^\mathbf{s}$ with VN-DGCNN layer (Eq. S2.1), using the k-nearest neighbors based on Euclidean distance with $k = 20$. We set the number of VN-DGCNN layers as 4. We set the number of MLP layers in the decoder (Eq. S1) as 4. We set the number of VN-In layers as 1.

We optimized the SE model via Adam [27] with its parameters (0.950, 0.999), learning rate 0.001, and batch size 16. We evaluated the validation loss every 2,000 training steps. We scheduled to decay the learning rate with a factor of 0.6 and a minimum learning rate of 1e-6 if the validation loss does not decrease in 5 consecutive evaluations. The optimal SE model has 27.3K learnable parameters. We trained the SE model with ∼172,600 training steps. The training took 80 hours with our GPUs. The trained SE model achieved the minimum validation loss at 162,000 steps.

Table S1: Hyper-Parameter Space for SE Optimization

Hyper-parameters	Space
hidden layer dimension	{64, 128}
dimension d_p of $\mathbf{H}^\mathbf{s}$	{32, 64}
# points in $\mathcal{P}$	{512, 1024}
# query points in $\mathcal{Z}$	1024
# nearest neighbors	20
# VN-DGCNN layers (Eq S2.1)	4
# MLP layers in Eq S1	4

[2] https://github.com/longlongman/DESERT/tree/830562e13a0089e9bb3d77956ab70e606316ae78

[3] `https://drive.google.com/drive/folders/146cpjuwenKGTd6Zh4sYBy-Wv6BMfGwe4?usp=sharing`

Table S2: Hyper-Parameter Space for DIFF Optimization

Hyper-parameters	Space
hidden layer dimension	128
weight of atom type loss ξ	100
threshold of step weight δ (Eq. 3)	10
# atom features K	15
# EQ-GNN/INV-GNN layers	8
# heads n_h in $\text{MHA}^{\mathbf{x}}/\text{MHA}^{\mathbf{v}}$	16
# nearest neighbors N (Eq. 1 and 2)	8
# diffusion steps T	1,000

S5.2 Parameters of DIFF

Table S2 presents the parameters used to train DIFF. In DIFF, we set the dimension of all the hidden layers including $\text{MHA}^{\mathbf{x}}$ and VN-Lin layer (Eq. 1), $\text{MHA}^{\mathbf{h}}$ and VN-In layer (Eq. 2) and MLP layer as 128. We set the number of layers L in EQ-GNN and INV-GNN as 8. Each layer is a multi-head attention layer ($\text{MHA}^{\mathbf{x}}$ or $\text{MHA}^{\mathbf{h}}$) with 16 heads. We set the number of VN-Lin and VN-In layer as 1, and the number of MLP layer as 2.

We constructed graphs from atoms in molecules, which are employed to predict atom coordinates and features (Eq. 1 and 2), using the N-nearest neighbors based on Euclidean distance with $N = 8$. We used $K = 15$ atom features in total, indicating the atom types and its aromaticity. These atom features include 10 non-aromatic atoms (i.e., "H", "C", "N", "O", "F", "P", "S", "Cl", "Br", "I"), and 5 aromatic atoms (i.e., "C", "N", "O", "P", "S"). We set the number of diffusion steps T as 1,000. We set the weight ξ of atom type loss as 100 to balance the values of atom type loss and atom coordinate loss. We set the threshold δ (Eq. 3) as 10. The parameters $\beta_t^{\mathbf{x}}$ and $\beta_t^{\mathbf{v}}$ of variance scheduling in the forward diffusion process of DIFF are discussed in Supplementary Section S3.3. Following SQUID, we did not perform extensive hyperparameter tunning for DIFF given that the used hyperparameters have enabled good performance.

We optimized the DIFF model via Adam [27] with its parameters (0.950, 0.999), learning rate 0.001, and batch size 32. We evaluated the validation loss every 2,000 training steps. We scheduled to decay the learning rate with a factor of 0.6 and a minimum learning rate of 1e-5 if the validation loss does not decrease in 10 consecutive evaluations. The DIFF model has 2.7M learnable parameters. We trained the DIFF model with ∼900,000 training steps. The training took 60 hours with our GPUs. The trained DIFF achieved the minimum validation loss at 746,000 steps.

During inference, given a condition molecule, we determined the number of atoms for the molecule to be generated, according to the shape size of the condition molecule. We assumed that molecules with similar shape sizes should have similar numbers of atoms. Therefore, we identified the training molecules with similar shape sizes to the condition molecule. From these molecules, we built an atom number distribution and sampled the number of atoms in the molecule to be generated from it. Particularly, to calculate the shape size of each molecule, we first converted it into an atomic density grid of side length 22 Å with 0.5 Å resolution and then counted the non-empty voxels in this grid. For a condition molecule with shape size n_s, we selected training molecules with shape sizes in the range $[n_s - 200, n_s + 200]$ to build the atom number distribution. Following Adams and Coley [13], we set the variance ϕ of atom-centered Gaussians as 0.049, which is used to build a set of points for shape guidance in Section "Molecule Generation and Shape Guidance" in the main manuscript. The optimal distance threshold γ is 0.2, and the optimal stop step S for shape guidance is 300. With shape guidance, each time we updated the atom position (Eq. 4), we randomly sampled the weight σ from $[0.2, 0.8]$. For each condition molecule, it took around 40 seconds on average to generate 50 molecule candidates with our GPUs.

S6 Additional Experimental Results

Table S3: Comparison of Diffusion Weighting Schemes

method	weights	#v%	#s%	#u%	QED	JS divergence	
						bond	C-C
ShapeMol	w_t^{x}	99.6	98.8	100.0	0.748	0.095	0.321
	uniform	99.2	89.4	100.0	0.660	0.115	0.393
ShapeMol+g	w_t^{x}	99.6	98.7	100.0	0.749	0.093	0.317
	uniform	99.1	90.1	100.0	0.671	0.112	0.384

Columns represent: "weights": different weighting schemes; "#v%": the percentage of valid molecules; "#s%": the percentage of valid and complete molecules; "#u%": the percentage of unique molecules; "QED": the average drug-likeness of generated molecules; "JS distance of bond/C-C": the Jensen-Shannon (JS) divergence of bond length among all the bond types ("bond")/carbon-carbon single bonds ("C-C") between real molecules and generated molecules;

S6.1 Comparison of Diffusion Weighting Schemes

While previous work [28, 9] applied uniform weights on different diffusion steps, ShapeMol uses different weights (i.e., w_t^{x} in Eq. 3). We conducted an ablation study to demonstrate the effectiveness of this new weighting scheme. Particularly, we trained two DIFF modules with the varying step weights w_t^{x} (with $\delta = 10$ in Eq. 3) and uniform weights, respectively, while fixing all the other hyper-parameters in ShapeMol and ShapeMol+g. Table S3 presents their performance comparison.

The results in Table S3 show that the different weights on different steps substantially improve the quality of the generated molecules. Specifically, ShapeMol with different weights ensures more valid and complete molecules and higher drug-likeness than that with uniform weights (98.8% vs 89.4% for valid and complete molecules; 0.748 vs 0.660 for QED). ShapeMol with different weights also produces molecules with bond length distributions closer to those of real molecules (i.e., lower Jensen-Shannon divergence), for example, the Jensen-Shannon (JS) divergence of bond lengths between real and generated molecules decreases from 0.115 to 0.095 when different weights are applied. The same trend can be observed for ShapeMol+g, for which the different weights also improve the generated molecule qualities. Since w_t^{x} increases as the noise level in the data decreases (See discussions earlier in "Model Training"), the results in Table S3 demonstrate the effectiveness of the new weighting scheme in promoting new molecules generated more similarly to real ones when the noise level in data is small.

S6.2 Parameter Study in Shape Guidance

We conducted a parameter study to evaluate the impact of the distance threshold γ (Eq. 4) and the step threshold S in the shape guidance. Particularly, using the same trained DIFF module, we sampled molecules with different values of γ and S and present the results in Table S4. As shown in Table S4, the average shape similarities $\mathsf{avgSim_s}$ and maximum shape similarities $\mathsf{maxSim_s}$ consistently decrease as γ and S increase. For example, when $S = 50$, $\mathsf{avgSim_s}$ and $\mathsf{maxSim_s}$ decreases from 0.794 to 0.763 and 0.890 to 0.861, respectively, as γ increases from 0.2 to 0.6. Similarly, when $\gamma = 0.2$, $\mathsf{avgSim_s}$ and $\mathsf{maxSim_s}$ decreases from 0.794 to 0.746 and 0.890 to 0.852, respectively, as S increases from 50 to 300. As presented in "ShapeMol with Shape Guidance", larger γ and S indicate stronger shape guidance in ShapeMol+g. These results demonstrate that stronger shape guidance in ShapeMol+g could effectively induce higher shape similarities between the given molecule and generated molecules.

It is also noticed that as shown in Table S4, incorporating shape guidance enables a trade-off between the quality of the generated molecules (QED), and the shape similarities ($\mathsf{avgSim_s}$ and $\mathsf{maxSim_s}$) between the given molecule and the generated ones. For example, when $\gamma = 0.2$, QED increases from 0.630 to 0.749 and $\mathsf{avgSim_s}$ decreases from 0.794 to 0.746 as S increases from 50 to 300. These results indicate the effects of γ and S in guiding molecule generation conditioned on given shapes.

Table S4: Parameter Study of Shape Guidance and Comparison between ShapeMol with and without Shape Condition

γ	S	with shape condition $\mathbf{H}^{\mathbf{s}}$					without shape condition $\mathbf{H}^{\mathbf{s}}$				
		QED	avgSim$_s$	avgSim$_g$	maxSim$_s$	maxSim$_g$	QED	avgSim$_s$	avgSim$_g$	maxSim$_s$	maxSim$_g$
-	-	0.748	0.689	0.239	0.803	0.243	0.752	0.584	0.239	0.751	0.241
0.2	50	0.630	0.794	0.236	0.890	0.244	0.474	0.697	0.233	0.859	0.241
0.2	100	0.666	0.786	0.238	0.883	0.245	0.541	0.691	0.238	0.851	0.242
0.2	300	0.749	0.746	0.241	0.852	0.247	0.732	0.650	0.248	0.809	0.246
0.4	50	0.678	0.779	0.240	0.875	0.245	0.562	0.676	0.242	0.834	0.243
0.4	100	0.700	0.772	0.241	0.870	0.247	0.614	0.670	0.245	0.828	0.246
0.4	300	0.752	0.738	0.242	0.845	0.247	0.737	0.636	0.248	0.796	0.250
0.6	50	0.706	0.763	0.242	0.861	0.246	0.617	0.656	0.246	0.813	0.246
0.6	100	0.720	0.758	0.242	0.857	0.247	0.655	0.651	0.247	0.808	0.247
0.6	300	0.753	0.731	0.242	0.838	0.247	0.741	0.626	0.251	0.791	0.258

Columns represent: "γ"/"S": distance threshold/step threshold in shape guidance; "QED": the average drug-likeness of generated molecules; "avgSim$_s$/avgSim$_g$": the average of shape or graph similarities between the condition molecules and generated molecules; "std": the standard deviation; "maxSim$_s$": the maximum of shape similarities between the condition molecules and generated molecules; "maxSim$_g$": the graph similarities between the condition molecules and the molecules with the maximum shape similarities;

S6.3 Ablation Study of Shape Condition

We conducted an ablation study to demonstrate the effectiveness of shape condition $\mathbf{H}^{\mathbf{s}}$ by comparing the performance of ShapeMol with and without this shape condition. Particularly, to evaluate the performance of ShapeMol without $\mathbf{H}^{\mathbf{s}}$, we removed the shape embeddings $\mathbf{H}^{\mathbf{s}}$ from the predictor $f_{\boldsymbol{\Theta}}(\mathbf{x}_t, \mathbf{v}_t, \mathbf{H}^{\mathbf{s}})$ and trained an unconditional diffusion model for molecule generation. As in Supplementary Section S6.2, we sampled molecules using the unconditional diffusion model with different values of γ and S and present the results in Table S4. As shown in Table S4, ShapeMol with $\mathbf{H}^{\mathbf{s}}$ consistently outperforms that without it, in terms of both avgSim$_s$ and maxSim$_s$. For example, when $\gamma = 0.2$ and $S = 300$, incorporating shape condition into ShapeMol boosts avgSim$_s$ from 0.650 to 0.746 and maxSim$_s$ from 0.809 to 0.852. Moreover, the increase in QED from 0.732 to 0.749 also indicates that shape condition could mitigate distortion in the generated molecule structures due to shape guidance. This might be because shape condition can move atom positions in the same direction as shape guidance and thus reduce the discrepancy between $\tilde{\mathbf{x}}_{0,t}$ and $\mathbf{x}^*_{0,t}$ (in Eq. 4). These results demonstrate that incorporating the shape condition in ShapeMol and ShapeMol+g can effectively improve the shape similarities between the given molecule and generated molecules and maintain the drug-likeness of generated molecules under shape guidance.

References

[1] 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, Jan 2018.

[2] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2323–2332. PMLR, 10–15 Jul 2018.

[3] Ziqi Chen, Martin Renqiang Min, Srinivasan Parthasarathy, and Xia Ning. A deep generative model for molecule optimization via one fragment modification. *Nature Machine Intelligence*, 3(12):1040–1049, Dec 2021.

[4] Shitong Luo, Jiaqi Guan, Jianzhu Ma, and Jian Peng. A 3d generative model for structure-based drug design. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[5] Xingang Peng, Shitong Luo, Jiaqi Guan, Qi Xie, Jian Peng, and Jianzhu Ma. Pocket2Mol: Efficient molecular sampling based on 3D protein pockets. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 17644–17655. PMLR, 17–23 Jul 2022.

[6] Victor Garcia Satorras, Emiel Hoogeboom, Fabian Fuchs, Ingmar Posner, and Max Welling. E(n) equivariant normalizing flows. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 4181–4192. Curran Associates, Inc., 2021.

[7] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022.

[8] Emiel Hoogeboom, Víctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3D. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8867–8887. PMLR, 17–23 Jul 2022.

[9] Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. In *The Eleventh International Conference on Learning Representations*, 2023.

[10] Chayan Acharya, Andrew Coop, James E. Polli, and Alexander D. MacKerell. Recent advances in ligand-based drug design: Relevance and utility of the conformationally sampled pharmacophore approach. *Current Computer Aided-Drug Design*, 7(1):10–22, Mar 2011.

[11] Kostas Papadopoulos, Kathryn A. Giblin, Jon Paul Janet, Atanas Patronov, and Ola Engkvist. De novo design with deep generative models based on 3d similarity scoring. *Bioorganic &amp: Medicinal Chemistry*, 44:116308, Aug 2021.

[12] Fergus Imrie, Thomas E. Hadfield, Anthony R. Bradley, and Charlotte M. Deane. Deep generative design with 3d pharmacophoric constraints. *Chemical Science*, 12(43):14577–14589, 2021.

[13] Keir Adams and Connor W. Coley. Equivariant shape-conditioned generation of 3d molecules for ligand-based drug design. In *The Eleventh International Conference on Learning Representations*, 2023.

[14] Siyu Long, Yi Zhou, Xinyu Dai, and Hao Zhou. Zero-shot 3d drug design by sketching and generating. *Advances in Neural Information Processing Systems*, 35:23894–23907, 2022.

[15] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J. Guibas. Vector neurons: A general framework for so(3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12200–12209, Oct 2021.

[16] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.*, 38(5), Oct 2019. ISSN 0730-0301.

[17] Maciej Wójcikowski, Piotr Zielenkiewicz, and Pawel Siedlecki. Open drug discovery toolkit (ODDT): a new open-source player in the drug discovery field. *Journal of Cheminformatics*, 7 (1), jun 2015.

[18] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.

[19] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12454–12465. Curran Associates, Inc., 2021.

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[21] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR, 18–24 Jul 2021.

[22] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

[23] Yunlu Chen, Basura Fernando, Hakan Bilen, Matthias Nießner, and Efstratios Gavves. 3d equivariant graph implicit functions. In *Lecture Notes in Computer Science*, pages 485–502. Springer Nature Switzerland, 2022.

[24] Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alán Aspuru-Guzik, and Alex Zhavoronkov. Molecular sets (moses): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11, 2020. ISSN 1663-9812.

[25] Greg Landrum, Paolo Tosco, Brian Kelley, Ric, David Cosgrove, Sriniker, Gedeck, Riccardo Vianello, NadineSchneider, Eisuke Kawashima, Dan N, Gareth Jones, Andrew Dalke, Brian Cole, Matt Swain, Samo Turk, AlexanderSavelyev, Alain Vaucher, Maciej Wójcikowski, Ichiru Take, Daniel Probst, Kazuya Ujihara, Vincent F. Scalfani, Guillaume Godin, Juuso Lehtivarjo, Axel Pahl, Rachel Walker, Francois Berenger, Jasondbiggs, and Strets123. rdkit/rdkit: 2023_03_2 (q1 2023) release, 2023.

[26] Mikko J. Vainio, J. Santeri Puranen, and Mark S. Johnson. ShaEP: Molecular overlay based on shape and electrostatic potential. *Journal of Chemical Information and Modeling*, 49(2): 492–502, Feb 2009.

[27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 2015*, 2015.

[28] Xingang Peng, Jiaqi Guan, Qiang Liu, and Jianzhu Ma. MolDiff: Addressing the atom-bond inconsistency problem in 3D molecule diffusion generation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 27611–27629. PMLR, 23–29 Jul 2023.