Manifold-Constrained Nucleus-Level Denoising Diffusion Model for Structure-Based Drug Design

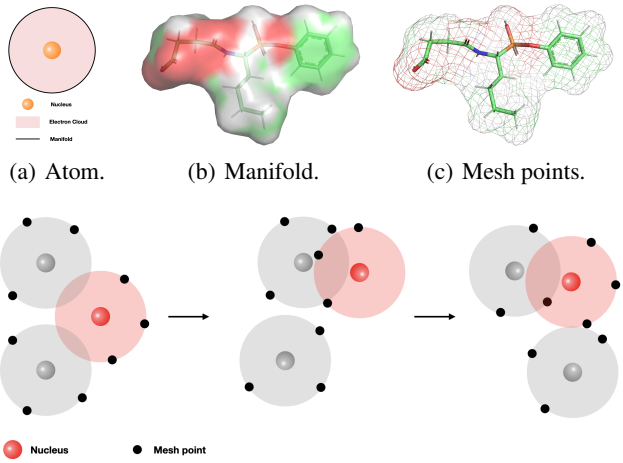
Shengchao Liu [*1,2] **Divin Yan** [*1] **Weitao Du** [3] **Weiyang Liu** [4] **Hongyu Guo** [5,6] **Christian Borgs** [2†] **Jennifer Chayes** [2†] **Anima Anandkumar** [1†]

Abstract

Deep generative models (DGMs) have shown great potential in structured-based drug design (SBDD). However, existing methods overlook a crucial physical constraint during both the learning and inference processes. That is, due to the attractive and repulsive forces, two atoms need to maintain a minimum distance as defined by their atomic radii. We refer to cases that violate this principle as *atomic collisions*. To address this problem, we first introduce three novel metrics to measure the atomic collisions at three granularities. We then demonstrate that existing DGMs for SBDD can generate ligands exhibiting atomic collisions. To mitigate such an issue, we further devise NucleusDiff. It jointly models the distribution of atomic nuclei and surrounding electrons on a manifold, ensuring adherence to physical laws by constraining the distance between the nucleus and the manifold. Empirical findings demonstrate that NucleusDiff not only achieves superior performance on four out of seven metrics for stability and potency but also circumvents collision issues by up to 30% on the three novel metrics, leading to a more efficient and effective drug design pipeline.

1. Introduction

Structure-based drug design (SBDD) is a cornerstone of drug discovery, aiming to design and optimize 3D small molecules (known as ligands) based on the three-dimensional structures of biological targets, often the protein pocket structures. SBDD is challenging because of the large chemical searching space and the intricate geometric complexities between two molecules within Euclidean space (Johnson & Karanicolas, 2013).



(a) Atom. (b) Manifold. (c) Mesh points.

(d) Pipeline of NucleusDiff.

Figure 1. Illustrations on atoms and molecules. Figure 1(a) displays the nucleus, electron cloud, and the manifold around the electron cloud. Figure 1(b) depicts the manifold surrounding molecule, and Figure 1(c) shows the mesh points after discretizing the manifold. Figure 1(d) illustrates the pipeline of NucleusDiff. NucleusDiff conducts denoising diffusion on both nuclei and mesh points, and the mean of the corresponding nuclei and mesh points are close to the van der Waals radii.

With this regard, machine learning (ML) emerges as a potent tool for navigating the design space of small molecules, leveraging geometric information from both ligands and proteins. Current state-of-the-art deep generative models (DGMs) employ SE(3)-equivariant scoring functions (Thomas et al., 2018; Liu et al., 2023) to estimate ligand geometry distributions based on the pocket structures within proteins. This includes methods such as denoising score matching (DSM) (Vincent, 2011; Song & Ermon, 2019) and denoising diffusion probabilistic models (DDPM) (Ho et al., 2020).

In addition to the SE(3)-equivariance, other physical principles must also be adhered to during modeling. One such

[*]Equal contribution ,[†]Joint supervision. [1]Caltech [2]UCB [3]DAMO Academy [4]Max Planck Institute [5]University of Ottawa [6]National Research Council Canada. Correspondence to: Shengchao Liu <shengchao1224@gmail.com>.

Proceedings of the Geometry-grounded Representation Learning and Generative Modeling at 41st International Conference on Machine Learning, Vienna, Austria. PMLR Vol Number, 2024.

property is the *attraction and repulsion*. In reality, each atom occupies a distinct sphere surrounding the *atomic nucleus*. This sphere is known as the *electron cloud*, the region where electrons are most likely found orbiting the nucleus. Within the electron cloud, *attractive forces* on each electron pulls it towards the nucleus and *repulsive forces* on each electron repel it from other electrons. Both the attractive forces and repulsive forces influence the distribution and arrangement of electrons. As a result, the *atomic radii* is derived as the average distance from the nucleus to the outermost electrons in the electron cloud. We illustrate this in Figure 1.

However, existing DGMs for SBDD presume that each atom is a solid point and overlook the physical properties induced by the law of electrostatic attraction and repulsion. Consequently, the *atomic collision* issue occurs: two atoms approach each other too closely during the generation process, violating the principle of electron attraction and repulsion by having atom pairwise distance below the atomic radii threshold. We illustrate this phenomenon in Figure 2.

Our Contributions. In this work, we aim to address the aforementioned challenge. In a nutshell, we first design three metrics for measuring the atomic collision in the existing DGMs for SBDD. Then we propose NucleusDiff, a manifold-constrained denoising diffusion model for the structure-based drug design framework. NucleusDiff consists of two generative components: one for modeling the atomic nuclei and one for modeling the electron cloud around each nucleus, as shown in Figure 1. This enables NucleusDiff to maintain, during learning and generation, the atomic radii between nuclei and their corresponding manifolds. We verify the effectiveness of NucleusDiff using 100K protein-ligand binding complexes from the CrossDocked2020 (Francoeur et al., 2020). Our quantitative analysis demonstrates that NucleusDiff significantly outperforms the state-of-the-art SBDD models. For instance, NucleusDiff exhibits a 22.16% improvement in Vina Score compared to TargetDiff. Moreover, NucleusDiff particularly excels in reducing the collision ratio, as evidenced by enhancements of over 30.00% across all three proposed collision metrics, ultimately achieving an almost negligible collision ratio.

We believe this work will pioneer a new direction in integrating physical laws into generative models for structure-based drug discovery. It underscores the importance of ensuring that learning models comply with these essential physical principles.

2. Method

2.1. Backgrounds

Small Molecules and Proteins. In our work, we consider small molecule ligands, which are sets of atoms in the 3D Euclidean space, $\{\boldsymbol{v}^L, \boldsymbol{x}^L\}$, where $\boldsymbol{v}^L$ represents the atomic number and $\boldsymbol{x}^L$ represents the atomic nucleus coordinate, respectively. Proteins are macromolecules, *i.e.*, chains of residues (or amino acids). In nature, there are 20 types of residues. Each residue is a small molecule, with a fixed backbone structure: a basic amino group, an acidic carboxyl group, a side chain that is unique to each amino acid, and a carbon C_α connecting three components. In this work, we consider modeling proteins in the backbone-level information, *i.e.*, the backbone atomic number and backbone atomic nucleus coordinates, $\{\boldsymbol{v}^P, \boldsymbol{x}^P\}$.

Nucleus and Electron Cloud. Each atom constitutes a nucleus and an electron cloud surrounds each nucleus, as shown in Figure 1. Recent works employ manifold learning on such an electron cloud for molecule property prediction (Zhang et al., 2023; Wang et al., 2022) and protein modeling in structure-based drug design (Mallet et al., 2023). In this work, we consider modeling the manifold around each nucleus with the atomic radius in the ligands. Then we discretize the manifold into triangle mesh points, a form suitable for computational analysis. This is implemented using the Python package PyMesh (Zhou, 2019). For notation, for each ligand atom $(\boldsymbol{v}^L, \boldsymbol{x}^L)$, the coordinate and nuclei type are the same as atom-level features, *i.e.*, $\boldsymbol{v}^N = \boldsymbol{v}^L$ and $\boldsymbol{x}^N = \boldsymbol{x}^L$. The coordinate of the discretized point on the manifold is marked as $\boldsymbol{x}^M$. Notice that we use a special token to delegate the electron points on the manifold.

Structure-based Drug Design. The structure-based drug design (SBDD) task utilizes the geometric structures of proteins to design and optimize ligands, like small molecules. This can be formulated as a conditional distribution modeling problem, $p(\boldsymbol{x}^L, \boldsymbol{v}^L | \boldsymbol{v}^P, \boldsymbol{x}^P)$. Notice that NucleusDiff improves this objective function by introducing nucleus-level modeling combined with manifold-sensitive constraints of small molecules, so the problem formulation becomes $p(\boldsymbol{x}^N, \boldsymbol{v}^N, \boldsymbol{x}^M | \boldsymbol{v}^P, \boldsymbol{x}^P)$. More details will be discussed in the Method Section.

2.2. Atomic Collision and Measurement

The atomic collision occurs when two atoms come into proximity such that their electron clouds overlap, violating the principle of electron attraction and repulsion. We introduce using van der Waals radii d for measuring. Suppose we have one ligand atom coordinate $\boldsymbol{x}_i$, one protein atom coordinate $\boldsymbol{x}_j$, and the corresponding van der Waals radii are d_i and d_j. Then during the sampling process for ligand generation, if two atoms get too close to each other, *i.e.*, $\|\boldsymbol{x}_j - \boldsymbol{x}_j\| \leq D_{ij} = d_i + d_j$, we refer to this as the *atomic collision* issue. To quantify this phenomenon, we propose three measurable metrics from three assessment granularities.

Pairwise-Level Collision Ratio (PLCR). The first metric is the atom pairwise-level ratio (PLCR), which quantifies

the collision ratio between all the ligand atoms and protein atoms. For each ligand atom ($\boldsymbol{x}_i^L$), we extract its K nearest protein atoms within the binding site. Then the PLCR is defined as

$$\text{PLCR} = \frac{\sum_{k\in N_{\text{mol}}, i\in N_{\text{atom}}^k, j\in N_{\text{nearest}}^i} \mathbb{1}(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| < D_{ij})}{K \cdot \sum_{k\in N_{\text{mol}}} N_{\text{atom}}^k}, \quad (1)$$

where N_{mol} is the number of ligand molecules, N_{atom}^k is the number of atoms in the k-th molecule, N_{nearest}^i is the number of the nearest protein atoms of the t-th ligand atom, and $\mathbb{1}(\cdot)$ is the indicator function.

Atom-Level Collision Ratio (ALCR). The second metric is the atom-level collision ratio (ALCR). It measures the collision ratio for each ligand atom by aggregating the pairwise-level collisions. An atom is marked as collision if at least one of its k nearest protein atoms is within a distance smaller than the sum of their van der Waals radii. More rigorously, it is defined as

$$\text{ALCR} = \frac{\sum_{k\in N_{\text{mol}}, i\in N_{\text{atom}}^k} \mathbb{1}(\sum_{j\in N_{\text{nearest}}^i} \mathbb{1}(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| < D_{ij}))}{\sum_{k\in N_{\text{mol}}} N_{\text{atom}}^k}. \quad (2)$$

Molecule-Level Collision Ratio (MLCR). The last metric is the molecule-level collision ratio (MLCR). It measures the collision ratio for each molecule by aggregating the atom-level collisions. A molecule is marked as collision if at least one of its atoms raises the atom-level collision. More rigorously, it is defined as

$$\text{MLCR} = \frac{\sum_{k\in N_{\text{mol}}} \mathbb{1}(\sum_{i\in N_{\text{atom}}^k, j\in N_{\text{nearest}}^i} \mathbb{1}(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| < D_{ij}))}{N_{\text{mol}}}. \quad (3)$$

The three defined metrics gauge atomic collisions between pockets and generated ligands, aiding in understanding the ML inference process for SBDD tasks. We initially tested them on a widely used ML method for SBDD (Guan et al., 2022). As illustrated in Figure 2, existing works exhibit atomic collision issues. Subsequently, we introduce NucleusDiff to address this challenge in the following sections.

2.3. Manifold-Constrained Nucleus-Level DDPM: NucleusDiff

We propose NucleusDiff to reduce the atomic collision in ML for the SBDD task. The main idea here is to jointly model the atomic nucleus and manifold over the electron cloud using the denoising diffusion model. In this section, we provide a brief introduction to NucleusDiff, and more detailed descriptions can be found in the Method section.

Diffusion Model for Geometry Generation. We first introduce the denoising diffusion model for density estimation on general geometries, $\boldsymbol{x}$. The denoising diffusion probabilistic model (DDPM) (Ho et al., 2020) consists of two stages: a forward and a backward process. The forward process gradually adds noise to the input geometric data $\boldsymbol{x}_0 = \boldsymbol{x}$ to a prior Gaussian distribution $\boldsymbol{x}_T$,

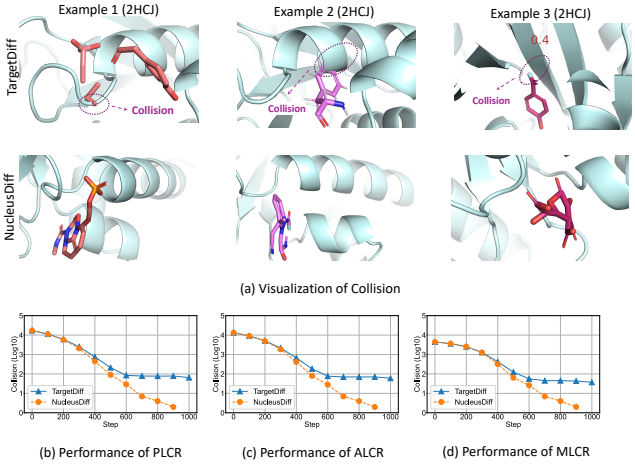


Figure 2. Illustration on atomic collision. (a): Visualization of Collision problem. (b-d): Visualization of the atomic collision ratio in TargetDiff and NucleusDiff. The x-axis is the number of diffusion steps and the y-axis is the atomic collision ratio.

and the backward process is the denoising process from the prior distribution to the data distribution. In concrete, suppose the data distribution is $\boldsymbol{x} \sim q(\boldsymbol{x})$, and we have T forward and backward steps with the scheduled variance $\{\beta_t\}_{t=1}^T$. Then each forward step can be represented as $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t I)$, which gives $q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)I)$, where $\alpha_t = 1-\beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$. Following the Bayes theorem, the posterior $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ can also be expressed as a Gaussian distribution:

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_{t-1}; \tilde{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\beta}_t I), \quad (4)$$

where $\tilde{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t$ and $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}$. The goal is to maximize the log-likelihood of data distribution $p(\boldsymbol{x})$, and after reparameterization, we aim to directly predict the ground-truth coordinates $\boldsymbol{x}_0$ with a parameterized network $\hat{\boldsymbol{x}}_0 = \phi_\theta(\boldsymbol{x}_t, t)$. The training objective is

$$\mathcal{L}_{t-1}(\boldsymbol{x}) = \mathbb{E}_q\left[\|\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0\|^2\right]. \quad (5)$$

Please refer to (Ho et al., 2020) for detailed derivations. We note that Equation (5) holds for arbitrary density estimation, and in the following paragraphs, we will discuss how we adapt this for our proposed NucleusDiff for structure-based drug design.

Nucleus Diffusion for Ligand Generation. Recall that in SBDD, the goal is to model the atomic types and coordinates in ligands given the pocket structure: $p(\boldsymbol{v}^L, \boldsymbol{x}^L|\boldsymbol{v}^P, \boldsymbol{x}^P)$. To adapt this into Equation (5), we can have the training objective over atomic coordinates as

$$\mathcal{L}_{t-1}(\boldsymbol{x}^N) = \mathbb{E}_q\left[\|\boldsymbol{x}_0^N - \hat{\boldsymbol{x}}_0^N(\boldsymbol{x}_t^N, t, \boldsymbol{v}^P, \boldsymbol{x}^P)\|^2\right]. \quad (6)$$

Notice the training objective for SBDD includes a categorical term $\mathcal{L}_{t-1}(\boldsymbol{v}^L)$ on atomic types and a continuous term $\mathcal{L}_{t-1}(\boldsymbol{x}^L)$ on atomic coordinates (Guan et al., 2022). In this section, we are mainly discussing the continuous part, while the discrete objective function $\mathcal{L}_{t-1}(\boldsymbol{v}^N)$ is described in the Method section.

Manifold Diffusion as Soft Constraint. Meanwhile, to reduce the atomic collision issue, we introduce an extra soft constraint: we should keep the distance between the nucleus and the manifold over the electron cloud as atomic radii, R. This constraint aligns with chemical principles, as electron clouds exert both attraction and repulsion forces, preventing atomic collisions. To adopt this into modeling, for each nucleus, we obtain its K closest mesh points in the manifold, marked as $\boldsymbol{x}^M$. Thus, the goal becomes the joint distribution of nuclei and mesh points conditioned on the pocket, as $p(\boldsymbol{v}^N, \boldsymbol{x}^N, \boldsymbol{x}^M | \boldsymbol{v}^P, \boldsymbol{x}^P)$. To adapt this into Equation (5), the objective of the manifold is

$$\mathcal{L}_{t-1}(\boldsymbol{x}^M) = \mathbb{E}_q \left[\|\boldsymbol{x}_0^M - \hat{\boldsymbol{x}}_0^M(\boldsymbol{x}_t^M, t, \boldsymbol{v}^P, \boldsymbol{x}^P)\|^2 \right]. \quad (7)$$

On the other hand, recall that the mesh points are scattered around the nuclei with a fixed atomic radius R. Thus we add a regularization term by forcing the distance between each mesh point $\boldsymbol{x}_j^N$ and nuclei $\boldsymbol{x}_i^M$ to be close to R:

$$\mathcal{L}_{t-1}(\boldsymbol{x}^N, \boldsymbol{x}^M, R) = \sum_i \sum_j \| \|\hat{\boldsymbol{x}}_0^N(\boldsymbol{x}_t^N, t, \boldsymbol{v}^P, \boldsymbol{x}^P) - \hat{\boldsymbol{x}}_0^M(\boldsymbol{x}_t^M, t, \boldsymbol{v}^P, \boldsymbol{x}^P)\| - R_{ij} \|. \quad (8)$$

The complete objective function becomes

$$\mathcal{L} = \mathbb{E}_t[\mathcal{L}_{t-1}(\boldsymbol{x}^N) + \mathcal{L}_{t-1}(\boldsymbol{v}^N) + \mathcal{L}_{t-1}(\boldsymbol{x}^M) + \mathcal{L}_{t-1}(\boldsymbol{x}^N, \boldsymbol{x}^M, R)]. \quad (9)$$

3. Experiment

3.1. Experimental Setup

Datasets. We utilize CrossDocked2020 (Francoeur et al., 2020) to train and evaluate our model. Similar to (Luo et al., 2021), we further refined the 22.5 million docked protein binding complexes by only selecting the poses with a low (< 1Å) and sequence identity less than 30%. In the end, we have 100,000 complexes for training and 100 complexes for testing.

Construction of Mesh Datasets for CrossDock. We utilize `MSMS` (Ewing & Hermisson, 2010) to compute the solvent-excluded surface of the molecule, employing a probe radius of 1.5 Å and a sampling density of 3.0 for small molecules, generating a triangular mesh representation. To further refine the surface mesh, we employ `PyMesh` (Zhou, 2019), which helps in reducing the number of vertices and correcting poorly meshed regions. Addressing degenerate vertices or disconnected surfaces is crucial, as these issues can lead to an improper distribution of mesh points when training the models. Finally, we selected the n mesh points that are closest to the Van der Waals radii distance from the nucleus to construct a mesh point dataset for the ligand. This dataset predominantly includes the 3D coordinates of the mesh points.

3.2. Evaluation of NucleusDiff for the Collision Problems

The Collision Metrics. For analysis, we compare NucleusDiff and TargetDiff, one of the most recent works for SBDD. Both methods are diffusion-based, which allows us to gain a comprehensive understanding of how the *atomic collision* issue evolves during the inference process of DDPM for SBDD. For both DDPM methods, we set the same timesteps (1000 steps) for learning and inference, and we analyze three metrics for atomic collision at 11 timesteps, every 100 steps from Step-0 to Step-1000. The main results are presented in Appendix D. For brevity, we present the results from Step-700 to Step-1000 in Table 1.

Analysis. In Table 1, it is evident that TargetDiff exhibits stable performance from Step-700 to Step-1000. Similarly, NucleusDiff shows a consistently low collision ratio during the same inference steps. However, it is noteworthy that NucleusDiff significantly outperforms TargetDiff by nearly one order of magnitude on all three collision metrics (PCLR, ACLR, and MLCR). Remarkably, in the last sampling steps, NucleusDiff ultimately achieves an almost negligible collision ratio, further highlighting its superior performance. Referring back to Figure 2, we observe a stark contrast in the convergence trends between NucleusDiff and TargetDiff across the three metrics, with NucleusDiff exhibiting a markedly more pronounced convergence trend than TargetDiff. Additional results and analyses of NucleusDiff's collision performance are provided in Appendix D.

Table 1. Atomic collision results among pocket-ligand pair for structure-based drug design.

Metrics	Targetdiff			NucleusDiff		
	PCLR	ACLR	MLCR	PCLR	ACLR	MLCR
Step-700	78/2300930	70/230093	45/10000	7/2300930	7/230093	7/10000
Step-800	77/2300930	70/230093	45/10000	4/2300930	4/230093	4/10000
Step-900	78/2300930	70/230093	43/10000	2/2300930	2/230093	2/10000
Step-1000	65/2300930	60/230093	37/10000	0/2300930	0/230093	0/10000

4. Conclusion

In this work, we first propose three novel metrics and scrutinize the atomic collision issue in the current ML for ligand design in the structure-based drug design tasks. Then we devise NucleusDiff, which models the manifold over the electron cloud to alleviate such a collision issue. Empirical results reveal that NucleusDiff not only reaches better performance on the existing metrics on stability and potency but also avoids the atomic issue and converges faster to the target geometric distribution.

References

Alhossary, A., Handoko, S. D., Mu, Y., and Kwoh, C.-K. Fast, accurate, and reliable molecular docking with quickvina 2. *Bioinformatics*, 31(13):2214–2216, 2015.

Anand, N. and Achim, T. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019*, 2022.

Boscaini, D., Bronstein, M., and Correia, B. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2): 184–192, 2020.

Ewing, G. and Hermisson, J. Msms: a coalescent simulation program including recombination, demographic structure and selection at a single locus. *Bioinformatics*, 26(16): 2064–2065, 2010.

Francoeur, P. G., Masuda, T., Sunseri, J., Jia, A., Iovanisci, R. B., Snyder, I., and Koes, D. R. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *Journal of Chemical Information and Modeling*, 60(9):4200–4215, 2020.

Gebauer, N., Gastegger, M., and Schütt, K. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. *Advances in Neural Information Processing Systems*, 32, 2019.

Guan, J., Qian, W. W., Peng, X., Su, Y., Peng, J., and Ma, J. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. In *The Eleventh International Conference on Learning Representations*, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34, 2021.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

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

Johnson, D. K. and Karanicolas, J. Druggable protein interaction sites are more predisposed to surface pocket formation than the rest of the protein surface. *PLoS computational biology*, 9(3):e1002951, 2013.

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

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

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, Y., Pei, J., and Lai, L. Structure-based de novo drug design using 3d deep generative models. *Chemical science*, 12(41):13664–13675, 2021.

Lin, H., Huang, Y., Liu, M., Li, X., Ji, S., and Li, S. Z. Diffbp: Generative diffusion of 3d molecules for target protein binding. *arXiv preprint arXiv:2211.11214*, 2022.

Liu, M., Luo, Y., Uchino, K., Maruhashi, K., and Ji, S. Generating 3d molecules for target protein binding. In *International Conference on Machine Learning*, 2022.

Liu, Q., Allamanis, M., Brockschmidt, M., and Gaunt, A. Constrained graph variational autoencoders for molecule design. *Advances in neural information processing systems*, 31, 2018.

Liu, S., Du, W., Li, Y., Li, Z., Zheng, Z., Duan, C., Ma, Z.-M., Yaghi, O. M., Anandkumar, A., Borgs, C., Chayes, J. T., Guo, H., and Tang, J. Symmetry-informed geometric representation for molecules, proteins, and crystalline materials. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=ygXSNrIU1p.

Luo, S., Guan, J., Ma, J., and Peng, J. A 3d generative model for structure-based drug design. *Advances in Neural Information Processing Systems*, 34, 2021.

Luo, S., Su, Y., Peng, X., Wang, S., Peng, J., and Ma, J. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *Advances in Neural Information Processing Systems*, 35: 9754–9767, 2022.

Luo, Y. and Ji, S. An autoregressive flow model for 3d molecular geometry generation from scratch. In *International Conference on Learning Representations (ICLR)*, 2022.

Mallet, V., Attaiki, S., and Ovsjanikov, M. Atomsurf: Surface representation for learning on protein structures. *arXiv preprint arXiv:2309.16519*, 2023.

Morris, G. M., Huey, R., Lindstrom, W., Sanner, M. F., Belew, R. K., Goodsell, D. S., and Olson, A. J. Autodock4

and autodocktools4: Automated docking with selective receptor flexibility. *Journal of computational chemistry*, 30(16):2785–2791, 2009.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Peng, X., Luo, S., Guan, J., Xie, Q., Peng, J., and Ma, J. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. *arXiv preprint arXiv:2205.07249*, 2022.

Powers, A. S., Yu, H. H., Suriana, P., and Dror, R. O. Fragment-based ligand generation guided by geometric deep learning on protein-ligand structure. *bioRxiv*, pp. 2022–03, 2022.

Ragoza, M., Masuda, T., and Koes, D. R. Generating 3D molecules conditional on receptor binding sites with deep generative models. *Chem Sci*, 13:2701–2713, Feb 2022. doi: 10.1039/D1SC05976A.

Shi, C., Xu, M., Zhu, Z., Zhang, W., Zhang, M., and Tang, J. Graphaf: a flow-based autoregressive model for molecular graph generation. In *International Conference on Learning Representations*, 2019.

Skalic, M., Sabbadin, D., Sattarov, B., Sciabola, S., and De Fabritiis, G. From target to drug: generative modeling for the multimodal structure-based ligand design. *Molecular pharmaceutics*, 16(10):4282–4291, 2019.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Steinegger, M. and Söding, J. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Tan, C., Gao, Z., and Li, S. Z. Target-aware molecular graph generation. *arXiv preprint arXiv:2202.04829*, 2022.

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Trippe, B. L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R., and Jaakkola, T. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Wang, Y., Shen, Y., Chen, S., Wang, L., Fei, Y., and Zhou, H. Learning harmonic molecular representations on riemannian manifold. In *The Eleventh International Conference on Learning Representations*, 2022.

Wu, L., Gong, C., Liu, X., Ye, M., and Liu, Q. Diffusion-based molecule generation with informative prior bridges. *Advances in Neural Information Processing Systems*, 35: 36533–36545, 2022.

Xu, M., Ran, T., and Chen, H. De novo molecule design through the molecular generative model conditioned by 3d information of protein binding sites. *Journal of Chemical Information and Modeling*, 61(7):3240–3254, 2021.

Zhang, O., Wang, T., Weng, G., Jiang, D., Wang, N., Wang, X., Zhao, H., Wu, J., Wang, E., Chen, G., et al. Learning on topological surface and geometric structure for 3d molecular generation. *Nature Computational Science*, pp. 1–11, 2023.

Zhou, Q. Pymesh—geometry processing library for python. *Software available for download at https://github.com/PyMesh/PyMesh*, 2019.

A. More Related Work

Structure-based Drug Design. In recent years, the availability of structural data has catalyzed the development of numerous generative models for the *target-aware* molecule generation task. Such models include those by (Skalic et al., 2019; Xu et al., 2021), which generate SMILES representations based on protein contexts, and the flow-based model proposed by (Tan et al., 2022) for generating molecular graphs conditional on protein target sequence embeddings. (Ragoza et al., 2022) have explored the generation of 3D molecules through the voxelization of molecules in atomic density grids within a conditional VAE framework. Further, (Li et al., 2021) have employed Monte-Carlo Tree Search coupled with a policy network for optimizing molecules in 3D space. Notably, (Luo et al., 2021; Liu et al., 2022; Peng et al., 2022) have developed autoregressive models for atom-by-atom 3D molecule generation using Graph Neural Networks (GNNs). Despite these advancements, current models still grapple with several challenges. These include the separate encoding of small molecules and protein pockets (Skalic et al., 2019; Xu et al., 2021; Tan et al., 2022; Ragoza et al., 2022), reliance on voxelization techniques and non-equivariance networks (Skalic et al., 2019; Xu et al., 2021; Ragoza et al., 2022), and the limitations inherent to autoregressive sampling methods (Luo et al., 2021; Liu et al., 2022; Peng et al., 2022). Contrasting with these approaches, our proposed equivariant model uniquely addresses the interactions between proteins and molecules in a 3D context and facilitates non-autoregressive sampling, thus ensuring a closer alignment between training and sampling methodologies.

Molecular & Protein Manifold Learning. Recent advancements in manifold learning for Molecular&Protein have garnered widespread attention in the scientific community. Several studies (Boscaini et al., 2020; Zhang et al., 2023; Wang et al., 2022; Mallet et al., 2023) have embarked on an innovative trajectory by employing or integrating sophisticated representational learning techniques pertaining to molecular and protein surfaces. This approach facilitates a precise articulation and comprehension of the intricate complexities inherent in molecular structures. (Boscaini et al., 2020) introduces MaSIF, a method using deep learning to identify and predict how proteins interact with other molecules by analyzing patterns on their surfaces. SurfGen (Zhang et al., 2023), introducing its two neural networks, Geodesic-GNN and Geoatom-GNN, effectively analyzes topological interactions on pocket surfaces and spatial interactions between ligand atoms and surface nodes for advanced molecular prediction. HMR (Wang et al., 2022) employs Laplace-Beltrami eigenfunctions for representing molecules on 2D Riemannian manifolds, enhancing molecular encoding through harmonic message passing. Atomsurf (Mallet et al., 2023) explores the use of 3D mesh surfaces for representing proteins, revealing that while promising, this method alone is less effective than 3D grids, and proposes a novel framework that synergistically combines surface representations with graph-based methods for improved protein representation learning.

3D Molecular Generation. Recent research has mainly focused on 2D molecule (Jin et al., 2018; Liu et al., 2018; Shi et al., 2019), but interest in 3D molecule generation has grown. Techniques like G-Schnet and G-SphereNet (Gebauer et al., 2019; Luo & Ji, 2022) use autoregressive methods to build molecules by adding atoms or fragments sequentially, also applying these to drug design (Li et al., 2021; Peng et al., 2022; Powers et al., 2022). Yet, this requires carefully defining action spaces and sequences. Some methods use atomic density grids for one-step molecule creation in 3D space (Kingma & Welling, 2013), but lack equivariance and need additional fitting algorithms. Recently, focus has shifted to using diffusion models for 3D molecule generation (Hoogeboom et al., 2022; Wu et al., 2022), showing success in drug generation (Lin et al., 2022; Guan et al., 2022), antibody (Luo et al., 2022), and protein design (Anand & Achim, 2022; Trippe et al., 2022). However, our work has the distinct honor of pioneering the exploration of atomic collision phenomena within the ambit of diffusion models for 3D Molecule Generation, an inquiry that positions us at the vanguard of this particular avenue of scientific inquiry.

B. More Details of Our Method: NucleusDiff

Our main goal is to jointly model the nucleus and the manifoldover the electron cloud surrounding each nucleus, to reduce the atomic collision issue in the diffusion model sampling process for structure-based drug design. We first explain how the DDPM model was applied to the existing structure-based drug design modeling. Following this, we introduce how we adopt the manifold-constrained modeling in NucleusDiff. Last but not least, we provide more details on the training objective and inference, along with insights into the architecture specifics.

B.1. Nucleus Diffusion for Atomic Nuclei Generation

Here our main goal is to model the nuclei types $\boldsymbol{v}^N$ and nuclei coordinates $\boldsymbol{x}^N$ given the protein pocket $(\boldsymbol{v}^P, \boldsymbol{x}^P)$: $p(\boldsymbol{v}^N, \boldsymbol{x}^N | \boldsymbol{v}^P, \boldsymbol{x}^P)$. We follow the existing DDPM for the SBDD pipeline by estimating this conditional with a categorical diffusion model on atomic types and a continuous diffusion model on atomic coordinates (Guan et al., 2022; Hoogeboom et al., 2021).

In the main manuscript, we define the variance scheduler β_t and α_t, and how to derive the prior $q(\boldsymbol{x}_t^N | \boldsymbol{x}_0^N)$ and posterior $q(\boldsymbol{x}_{t-1}^N | \boldsymbol{x}_t^N, \boldsymbol{x}_0^N)$ for nuclei coordinates at time t. Similarly, for the nuclei types, we use categorical distribution C, and suppose we have K nuclei types in total. The prior distribution and posterior distribution of nuclei types at time t are

$$\begin{aligned} q(\boldsymbol{v}_t^N | \boldsymbol{v}_0^N) &= C(\boldsymbol{v}_t^N | \bar{\alpha}_t \boldsymbol{v}_0^N + (1 - \bar{\alpha}_t)/K), \\ q(\boldsymbol{v}_t^N | \boldsymbol{v}_t^N, \boldsymbol{v}_0^N) &= C(\boldsymbol{v}_t^N | \theta_c(\boldsymbol{v}_t^N, \boldsymbol{v}_0^N)), \end{aligned} \tag{10}$$

where $\theta_c(\boldsymbol{v}_t^N, \boldsymbol{v}_0^N) = \theta^* / \sum_k \theta_k^*$, and $\theta^* = [\alpha_t \boldsymbol{v}_t^N + (1 - \alpha_t)/K] \odot [\bar{\alpha}_{t-1} \boldsymbol{v}_0^N + (1 - \bar{\alpha}_{t-1})/K]$, where $\odot$ is element-wise product. Then after reparameterization, we predict $\hat{\boldsymbol{v}}_0^N$ from $\boldsymbol{v}_t^N$, *i.e.*, $\tilde{\boldsymbol{v}}_0^N = \mu_\theta(\boldsymbol{v}_t^N, t)$. Injecting this into the posterior, then the objective functions for the discrete types and continuous coordinates are

$$\begin{aligned} \mathcal{L}_{t-1}(\boldsymbol{v}^N) &= \mathrm{KL}(q(\boldsymbol{v}_t^N | \boldsymbol{v}_t^N, \boldsymbol{v}_0^N) || q(\boldsymbol{v}^N | \tilde{\boldsymbol{v}}_0^N)) = \sum_k \theta_c(\boldsymbol{v}_t^N, \boldsymbol{v}_0^N)_k \cdot \frac{\theta_c(\boldsymbol{v}_t^N, \boldsymbol{v}_0^N)_k}{\theta_c(\boldsymbol{v}_t^N, \tilde{\boldsymbol{v}}_0^N)_k}, \\ \mathcal{L}_{t-1}(\boldsymbol{x}^N) &= \mathbb{E}_q \big[\| \boldsymbol{x}_0^N - \hat{\boldsymbol{x}}_0^N(\boldsymbol{x}_t^N, t, \boldsymbol{v}^P, \boldsymbol{x}^P) \|^2 \big]. \end{aligned} \tag{11}$$

B.2. Manifold Diffusion as Soft Constraint

Meanwhile, as illustrated above, the generated nuclei coordinates should follow the chemical properties: there exist attraction forces between electrons and nuclei and repulsion forces between electrons, otherwise, the atomic collision can occur. To alleviate this issue, we jointly model the manifold and nuclei for structure-based drug design.

To be more concrete, for each nucleus, we construct a discrete manifold, where the radius is the atomic radii R. Then for each nucleus, we obtain its c closest mesh points in the manifold, marked as $\boldsymbol{x}^M$. Thus, instead of $p(\boldsymbol{v}^N, \boldsymbol{x}^N | \boldsymbol{v}^P, \boldsymbol{x}^P)$, the objective is to maximize the following likelihood $p(\boldsymbol{v}^N, \boldsymbol{x}^N, \boldsymbol{x}^M | \boldsymbol{v}^P, \boldsymbol{x}^P)$. The objective on the manifold at time t is

$$\mathcal{L}_{t-1}(\boldsymbol{x}^M) = \mathbb{E}_q \big[\| \boldsymbol{x}_0^M - \hat{\boldsymbol{x}}_0^M(\boldsymbol{x}_t^M, t, \boldsymbol{v}^P, \boldsymbol{x}^P) \|^2 \big]. \tag{12}$$

On the other hand, recall that the mesh points are scattered around the nuclei with atomic radii R. Motivated by this, we add a regularization term by forcing the distance between each mesh point and nuclei to be close to R as:

$$\mathcal{L}_{t-1}(\boldsymbol{x}^N, \boldsymbol{x}^M, R) = \sum_i \sum_j \| \| \hat{\boldsymbol{x}}_0^N(\boldsymbol{x}_t^N, t, \boldsymbol{v}^P, \boldsymbol{x}^P) - \hat{\boldsymbol{x}}_0^M(\boldsymbol{x}_t^M, t, \boldsymbol{v}^P, \boldsymbol{x}^P) \| - R_{ij} \|. \tag{13}$$

B.3. Learning and Inference

To sum up, the training objective function is composed of three parts:

$$\mathcal{L} = \mathbb{E}_t [\mathcal{L}_{t-1}(\boldsymbol{x}^N) + \mathcal{L}_{t-1}(\boldsymbol{v}^N) + \mathcal{L}_{t-1}(\boldsymbol{x}^M) + \mathcal{L}_{t-1}(\boldsymbol{x}^N, \boldsymbol{x}^M, R)]. \tag{14}$$

For inference, because the mesh points from manifold modeling are only used as auxilarate of the physics-guided nuclei modeling, they can be ignored, while only the nuclei coordinates are required for SBDD.

B.4. Computational Resources

All algorithms and models have been developed using Python 3.8.13, with PyTorch version 1.12.1 and PyTorch Geometric version 2.5.2, under CUDA 11.0. Experiments are conducted on a server with 8 NVIDIA V100 GPUs (32 GB memory) and Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz. We employ a single V100 GPU for training while leveraging eight GPUs to accelerate the sampling procedure. The models typically converge after approximately 48 hours of training and sampling 10k ligands using eight GPUs takes about 12 hours.

C. Evaluation of NucleusDiff on the General Metrics

General Metrics for Structure-Based Drug Design. The *Vina Score*, *Vina Min*, and *Vina Dock* metrics are employed to assess the binding affinity and potential biological efficacy of small molecule drug candidates in interaction with target proteins, via the computation of docking efficiency scores. Following the methodologies outlined in (Luo et al., 2021), (Peng et al., 2022), and (Guan et al., 2022), we utilize the open-source AutoDockTools software (Morris et al., 2009) for these calculations. The *High Affinity* metric gauges the strength of the ligand-target protein interaction. *QED* provides a numerical assessment of a compound's drug-like characteristics, with higher values indicating a greater propensity for a compound to embody successful drug attributes. *SA* quantifies the ease with which a compound can be synthesized. Lastly, *Diversity* measures the range and heterogeneity of molecular structures and properties across a set of compounds.

Baselines. For benchmarking, we compare with various baselines: liGAN (Ragoza et al., 2022), AR (Luo et al., 2021), Pocket2Mol (Peng et al., 2022), GraphBP (Liu et al., 2022) and Targetdiff (Guan et al., 2022). liGAN (Ragoza et al., 2022) is a 3D CNN-based method that generates 3D voxelized molecular images following a conditional VAE scheme. AR (Luo et al., 2021), Pocket2Mol (Peng et al., 2022), and GraphBP (Liu et al., 2022) are all GNN-based methods that generate 3D molecules by *sequentially* placing atoms into a protein binding pocket. We choose AR-SBDD (Luo et al., 2021) and Pocket2Mol (Peng et al., 2022) as representative baselines with autoregressive sampling scheme because of their good empirical performance. Targetdiff (Guan et al., 2022) employs a diffusion-based technique for generating atom coordinates and types.

Analysis Under General Metrics. We generate 100 ligand molecules for each target protein in the test set, resulting in a total of 10,000 molecules. The size of each generated molecule, *i.e.*, the number of atoms in each molecule, is determined by sampling from the size distribution observed in the training set. The comprehensive results for NucleusDiff and the baseline models are displayed in Table 2.

We note that NucleusDiff surpasses all baseline models in nearly every evaluated metric, with the exceptions of QED, SA, and Diversity. In our evaluation, NucleusDiff is only surpassed by GraphBP (Liu et al., 2022) in terms of Diversity, yet it exhibits superior performance compared to another diffusion model, TargetDiff (Guan et al., 2022). According to the Vina Score, NucleusDiff is able to generate the molecules with high affinity to the pocket (-7.90), which is 6.43% better than the best autoregressive model baseline, AR-SBDD (Luo et al., 2021) and 22.16% than another diffusion model baseline Targetdiff (Guan et al., 2022). Besides, NucleusDiff surpasses AR-SBDD (Luo et al., 2021) and TargetDiff (Guan et al., 2022) on High Affinity (60.1%) by 58.6% and 6.7%, and Diversity (0.74) by 6.71% and 4.23%. On the other hand, the SA

Table 2. A summary of various properties of reference molecules and those generated by our model and other baselines is presented. The symbols (↑) and (↓) indicate whether a higher or lower value is preferable for each property. Due to incompatibility between certain atom types produced by liGAN (Ragoza et al., 2022) and GraphBP (Liu et al., 2022) and the parsing capabilities of AutoDock Vina, we employ QVina (Alhossary et al., 2015) to conduct the docking simulations for these two methods.

Metrics	**Vina Score (↓)**		**Vina Min (↓)**		**Vina Dock (↓)**		**High Affinity (↑)**		**QED (↑)**		**SA (↑)**		**Diversity (↑)**	
	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.
Reference	-6.36	-6.46	-6.71	-6.49	-7.45	-7.26	-	-	0.48	0.47	0.73	0.74	-	-
liGAN *	-	-	-	-	-6.33	-6.20	21.1%	11.1%	0.39	0.39	0.59	0.57	0.66	0.67
GraphBP *	-	-	-	-	-4.80	-4.70	14.2%	6.7%	0.43	0.45	0.49	0.48	**0.79**	**0.78**
AR-SBDD	-5.75	-5.64	-6.18	-5.88	-6.75	-6.62	37.9%	31.0%	0.51	0.50	0.63	0.63	0.70	0.70
Pocket2Mol	-5.14	-4.70	-6.42	-5.82	-7.15	-6.79	48.4%	51.0%	**0.56**	**0.57**	**0.74**	**0.75**	0.69	0.71
Targetdiff	-5.01	-5.69	-6.33	-6.47	-7.62	-7.64	56.3%	57.3%	0.48	0.48	0.59	0.58	0.71	0.71
NucleusDiff	**-6.12**	**-6.80**	**-6.93**	**-6.85**	**-7.90**	**-7.76**	**60.1%**	**63.0%**	0.39	0.39	0.53	0.53	0.74	0.73

of generated molecules should fall within a reasonable range so that the ability to explore the molecular space confined by protein pockets is high enough to discover potential molecules. As in Table 2, the QED of NucleusDiff is a little bit lower than that of AR-SBDD (Luo et al., 2021) and Targetdiff (Guan et al., 2022) but compared to that of liGAN (Ragoza et al., 2022), implying that our model satisfies this desired property. Notably, the molecules generated by NucleusDiff perform even better than those in the test set on Vina Score, Vina Min, and Vina Dock, suggesting that NucleusDiff has great potential to generate more drug-like molecules with higher affinity outside the distribution of the dataset. NucleusDiff concurrently learns the distribution of electron clouds and the spatial arrangement of atomic nuclei, thereby deepening its comprehension of physical constraints. This enhanced understanding is critically important for the generation of high-affinity and realistically viable pharmaceuticals. Besides, although Targetdiff (Guan et al., 2022) also generates molecules in a one-shot manner. However, the TargetDiff (Guan et al., 2022) model, when learning the distribution of atoms, solely considers the positional information of atomic nuclei. This approach, which fails to incorporate physical information, is problematic. Consequently, it is reasonable to assert that the NucleusDiff model, a geometric diffusion generative model that incorporates spatial physical constraints, provides critical insights for the synthesis of molecules and pharmaceuticals with high affinity.

C.1. Performance Analysis of NucleusDiff Through Visualization

Figure 3 illustrates the visual representation of the ligands generated by NucleusDiff and TargetDiff, given the specific binding pockets. We select several pocket proteins to visualize as representative samples for structure analysis. As depicted in Figure 3, we choose 4U5S, 2GNS, 5MMA, 4RV4, and 2HCJ as the targeted pocket proteins.

We can observe that both TargetDiff and NucleusDiff have the potential to generate relatively stable structures. From the perspective of the Vina Score, NucleusDiff has the potential to generate ligands with higher affinity compared to those produced by TargetDiff and the Test Set. This can be specifically observed in the ligands generated for the pockets 4U5S, 2GNS, 5MMA, and 4RV4.

However, when considering the presence of atomic collisions in the generated molecules, a more detailed comparison between TargetDiff and NucleusDiff can be made. For instance, with the pocket proteins 5MMA and 2HCJ, it is evident that the ligands generated by TargetDiff have a less clear relative positioning with the pockets, posing a risk of atomic collisions. In contrast, the ligands generated by NucleusDiff have a much clearer boundary with the protein pockets. This indicates that NucleusDiff has not only learned the relative positional relationship between the ligands and the protein pockets but also the physical rules concerning the relative distribution of atomic nuclei and electron clouds within the ligands.

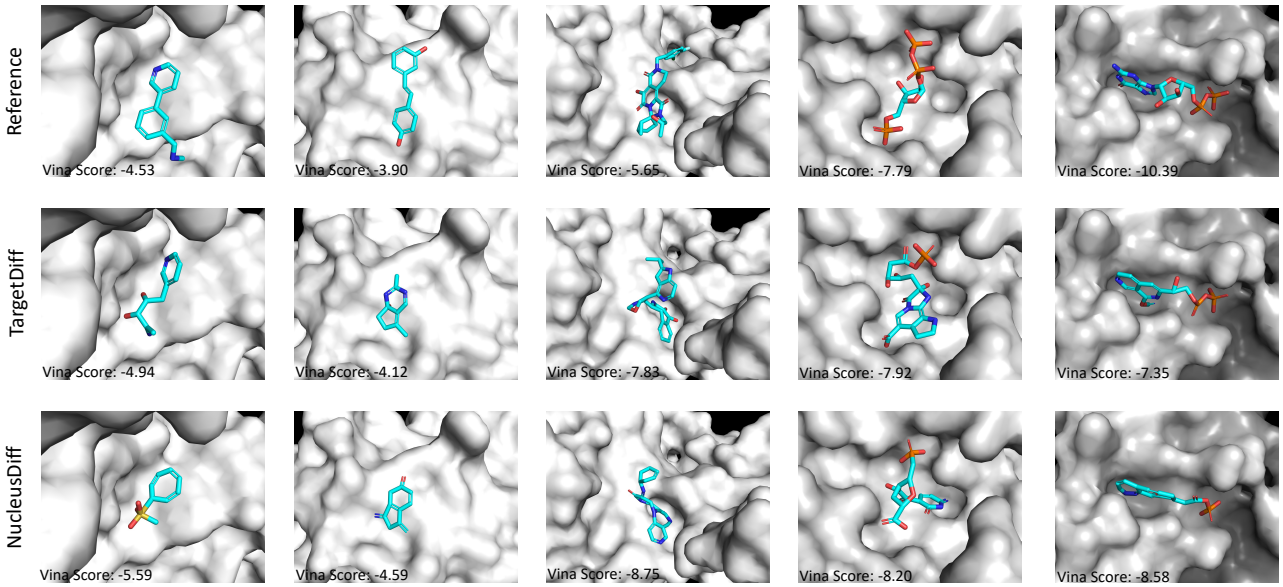


Figure 3. Visualization of the generated molecules by TargetDiff and NucleusDiff.

D. More Experiments

In this section, we conducted additional experiments to further ascertain the effectiveness of our proposed model, particularly in addressing the *atomic collision issue*.

D.1. More Results on Evaluating Collision Issues

Table 3 presents a comprehensive evaluation of *collision issues* (PLCR, ALCR, MLCR) from Time Step 0 to Time Step 1000 during the inference phase. The experimental results clearly indicate that NucleusDiff and TargetDiff exhibit nearly identical performance regarding collision issues in the initial phase of inference (Step 0 to Step 300). However, from Step 400 onward, NucleusDiff demonstrates a significantly faster convergence rate in addressing collision problems. By approximately Step 700, NucleusDiff appears to have almost completely resolved the collision issues. In contrast, TargetDiff shows rapid convergence in addressing collision problems from Step 400 to Step 600, after which its performance related to collision issues shows little to no change. Table 3 provides a complete depiction of the performance of these two diffusion-based models concerning collision problems during the inference phase, allowing us to gain a more intuitive understanding of how the trained NucleusDiff model mitigates collision issues.

Table 3. Atomic collision results among pocket-ligand pair for structure-based drug design. **A lower value is better.**

Metrics	**Targetdiff**			**NucleusDiff**		
	PLCR	ALCR	MLCR	PLCR	ALCR	MLCR
Step-0	17103/2300930	13241/230093	4425/10000	17120/2300930	13257/230093	4428/10000
Step-100	11293/2300930	9083/230093	3613/10000	11185/2300930	9040/230093	3627/10000
Step-200	6019/2300930	5079/230093	2543/10000	5779/2300930	4918/230093	2540/10000
Step-300	2482/2300930	2142/230093	1285/10000	2081/2300930	1847/230093	1254/10000
Step-400	751/2300930	671/230093	426/10000	444/2300930	414/230093	321/10000
Step-500	211/2300930	183/230093	124/10000	90/2300930	80/230093	64/10000
Step-600	84/2300930	77/230093	56/10000	29/2300930	28/230093	26/10000
Step-700	78/2300930	70/230093	45/10000	7/2300930	7/230093	7/10000
Step-800	77/2300930	70/230093	45/10000	4/2300930	4/230093	4/10000
Step-900	78/2300930	70/230093	43/10000	2/2300930	2/230093	2/10000
Step-1000	65/2300930	60/230093	37/10000	0/2300930	0/230093	0/10000

D.2. The Ring Size Distribution.

Many previous studies (Peng et al., 2022; Luo & Ji, 2022; Guan et al., 2022) have suggested that if the ligands generated by a trained model exhibit high structural consistency with the ground truth ligands in the test set, the generative model can be considered highly successful. However, we argue that this perspective is flawed. A generative model should not only learn a distribution but also possess strong generalization capabilities. Our goal is to generate a diverse array of ligands beyond the training distribution, which holds significant practical implications for drug design and discovery. In this section, we present a detailed analysis of the substructures of the molecules generated by NucleusDiff, specifically focusing on the distribution of ring sizes in the generated molecules. Table 4 illustrates the distribution of different ring sizes present in the test set, as well as in the 10,000 molecules generated by the baselines and NucleusDiff. The results reveal that, in comparison to the test set and another diffusion-based model, TargetDiff, the ring sizes in the molecules generated by these models are primarily concentrated around 5 and 6. In contrast, the ring sizes in the molecules generated by NucleusDiff are mainly distributed among 5, 6, 7, 8, and 9. Notably, the proportion of ring sizes 7, 8, and 9 is significantly higher in NucleusDiff compared to TargetDiff and the test set. This observation leads to two key insights: (1) NucleusDiff has the potential to generate more complex structures, such as intricate ring structures, compared to TargetDiff. (2) The structures generated by NucleusDiff are relatively more novel, as evidenced by the discrepancy between the substructure distribution of the ground-truth ligands in the test set and that of the molecules generated by NucleusDiff.

Table 4. Percentage of different ring sizes for reference and model generated molecules.

Ring Size	Ref.	liGAN	AR	Pocket2Mol	TargetDiff	**NucleusDiff**
3	1.7%	28.1%	29.9%	0.1%	0.0%	0.0%
4	0.0%	15.7%	0.0%	0.0%	2.5%	8.8%
5	30.2%	29.8%	16.0%	16.4%	30.6%	21.4%
6	67.4%	22.7%	51.2%	80.4%	51.8%	37.3%
7	0.7%	2.6%	1.7%	2.6%	11.8%	21.2%
8	0.0%	0.8%	0.7%	0.3%	2.5%	7.6%
9	0.0%	0.3%	0.5%	0.1%	0.8%	3.7%

D.3. The Bond Distribution.

In our study, we evaluated the performance of various generative models in terms of their ability to reproduce the bond distributions observed in reference molecules. Our primary focus was on the NucleusDiff model, which demonstrated a unique capability in generating diverse molecular substructures. Table 5 presents a comparative analysis of the bond distributions for different models, including liGAN, GraphBP, AR, Pocket2Mol, TargetDiff, and NucleusDiff. The NucleusDiff model consistently shows a balanced distribution across various bond types (C-C, C=C, C-N, C=N, C-O, C=O, C:C, C:N), indicating its proficiency in capturing the structural diversity inherent in the reference dataset. Table 5 provides quantitative insights through the Jensen-Shannon divergence between the bond distance distributions of reference molecules and those generated by each model. The NucleusDiff model exhibits competitive divergence values across all bond types, highlighting its effectiveness in mimicking the bond length distributions of real molecules. Notably, for bonds like C=N and C=O, NucleusDiff achieves divergence values of 0.649 and 0.464, respectively, which are relatively low and suggest a high degree of similarity to the reference distributions. The superior performance of NucleusDiff in generating diverse substructures can be attributed to its advanced architectural design, which allows for fine-grained control over molecular features. This is evident from its ability to generate molecules with a wide range of bond types, maintaining a high degree of structural fidelity to natural molecules. Consequently, NucleusDiff not only ensures the generation of chemically valid molecules but also enhances the exploration of the chemical space by producing a variety of substructures, which is crucial for applications in drug discovery and materials science. In summary, the NucleusDiff model stands out in its ability to generate molecular substructures with considerable diversity, closely mirroring the bond distributions of reference molecules. This capability underscores its potential as a powerful tool for the generation of novel and diverse molecular entities.

Table 5. The Jensen-Shannon divergence between the distributions of bond distances for reference versus generated molecules is analyzed. In this context, "-", "=", and ":" denote single, double, and aromatic bonds, respectively.

Bond	liGAN	GraphBP	AR	Pocket2Mol	TargetDiff	**NucleusDiff**
C−C	0.601	0.368	0.609	0.496	0.367	0.544
C=C	0.665	0.530	0.620	0.561	0.507	0.599
C−N	0.634	0.456	0.474	0.416	0.361	0.478
C=N	0.749	0.693	0.635	0.629	0.551	0.649
C−O	0.656	0.467	0.492	0.454	0.424	0.559
C=O	0.661	0.471	0.558	0.516	0.467	0.464
C:C	0.497	0.407	0.451	0.416	0.264	0.517
C:N	0.638	0.689	0.552	0.487	0.234	0.464

D.4. Visualization of the Training Process.

The visualizations in Figure 4 provide insights into the training dynamics of the model. The Total Training Loss Curve (a) shows overall loss with noticeable fluctuations but a general downward trend, indicating that the model is learning over time. The training loss curves for molecular and mesh modeling (b-f) exhibit significant variability, suggesting challenges in these specific tasks. In contrast, the Training Loss Curve for Mesh Feature (g) displays lower values and less fluctuation, indicating that the model finds it easier to learn mesh features. The Constrained Loss curve (h) shows fewer oscillations, implying that constraints help stabilize the training process. The Learning Rate Curve (i) remains relatively constant, suggesting a stable learning rate policy. The Gradient Normalization Curve (j) with occasional spikes indicates moments of large gradient changes, while the Atom Type Accuracy Curve (k) shows stable accuracy, reflecting consistent performance in predicting atom types. The Iteration Curve (l) linearly increases, reflecting the progression of training steps.

The validation loss curves (m-o) mirror the training loss curves, showing high variability but general downward trends, which suggests that the model is generalizing well to the validation data. Despite the fluctuations, the general improvement over time indicates effective learning. The stable learning rate and gradient norms indicate a controlled training environment. However, the persistent oscillations in the loss curves suggest that further optimization might be necessary to achieve smoother convergence and more stable training dynamics.

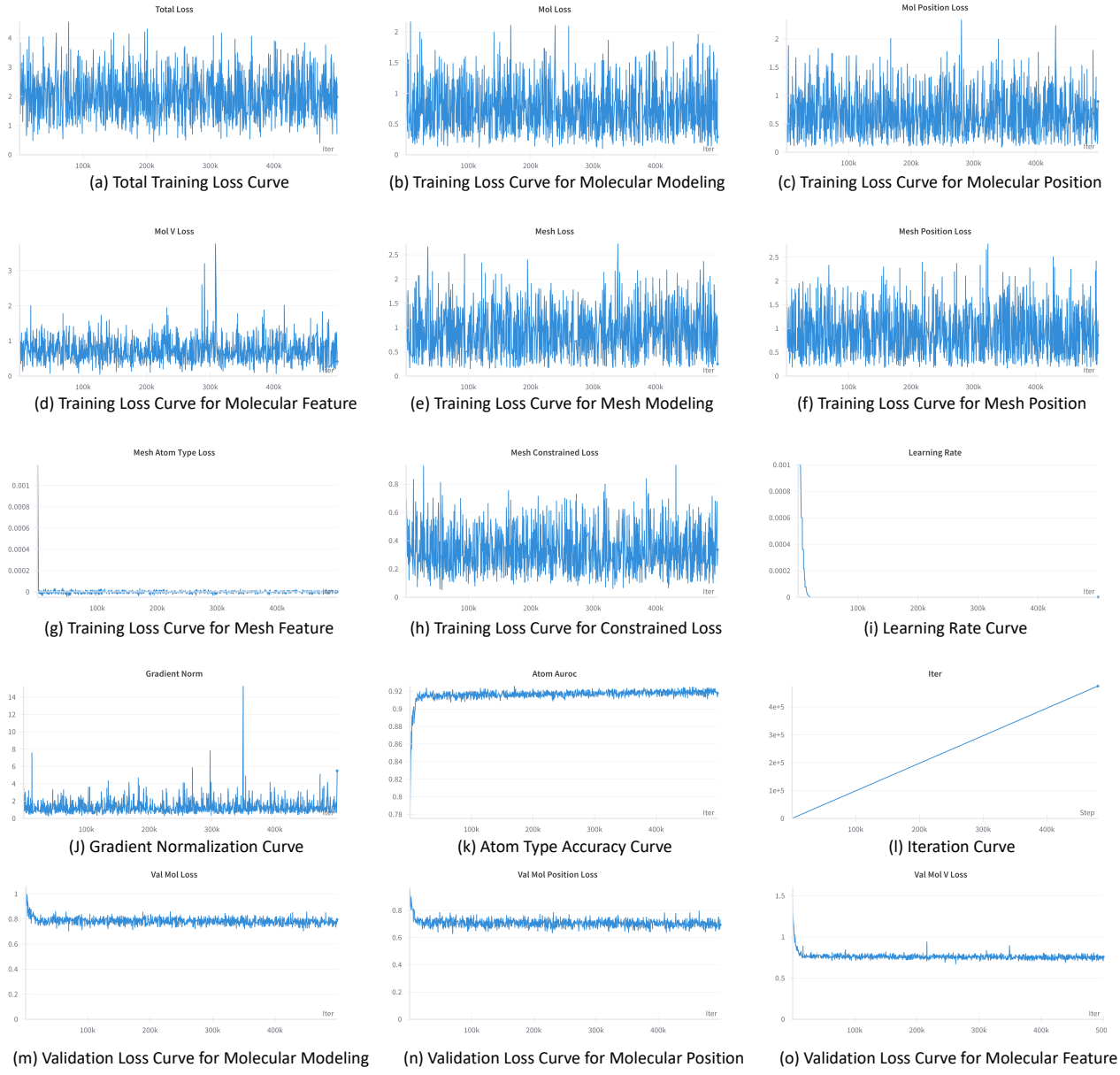


Figure 4. The visualization of the training process of NucleusDiff.

E. Details of the Experimental Setup

E.1. CrossDock Datasets

We conducted experiments to evaluate the generative performance of NucleusDiff on the CrossDocked dataset (Francoeur et al., 2020). This dataset comprises 22.5 million docked protein-ligand pairs, with each pair exhibiting various poses across multiple pockets within the Protein Data Bank. The ligands associated with specific pockets were docked with each receptor assigned to those pockets using smina through Pocketome. Binding data (pK) for the CrossDocked2020 set were sourced from PDBbind v2017, revealing that 41.9% of the complexes have available binding affinity data. For a fair comparison, we followed previous works (Luo & Ji, 2022; Guan et al., 2022) by selecting only binding pose data with root-mean-squared deviations (RMSD) of less than 1 ÅṪhe dataset was further refined through clustering at 30% sequence identity using MMseqs2 (Steinegger & Söding, 2017). This process yielded 100,000 pairs for training and 100 pairs for evaluation.

We utilize `MSMS` (Ewing & Hermisson, 2010) to compute the solvent-excluded surface of the molecule, employing a probe radius of 1.5 Å and a sampling density of 3.0 for small molecules, generating a triangular mesh representation. To further refine the surface mesh, we employ `PyMesh` (Zhou, 2019), which helps in reducing the number of vertices and correcting poorly meshed regions. Addressing degenerate vertices or disconnected surfaces is crucial, as these issues can lead to an improper distribution of mesh points when training the models. Finally, we selected the n mesh points that are closest to the Van der Waals radii distance from the nucleus to construct a mesh point dataset for the ligand. This dataset predominantly includes the 3D coordinates of the mesh points.

E.2. Training Details

The model is trained using the gradient descent method Adam (Kingma & Ba, 2014) with `init_learning_rate=0.001`, `betas=(0.95, 0.999)`, `batch_size=4`, and `clip_gradient_norm=8`. To balance the scales of the two losses, we apply a factor of $\alpha = 100$ to the atom type loss. During the training phase, we add small Gaussian noise with a standard deviation of 0.1 to protein atom coordinates as data augmentation. We also schedule to decay the learning rate exponentially with `a factor of 0.6` and `a minimum learning rate of 1e-6`. The learning rate is decayed if there is no improvement in the validation loss over `10 consecutive evaluations`. The evaluation is performed for `every 100 training steps`.

E.3. Implementation Details

Our NucleusDiff comprises two score models, each consisting of 9 equivariant layers. Each layer is a Transformer (Vaswani et al., 2017) with `hidden_dim=128` and `n_heads=16`. The key/value embeddings and attention scores are generated through a 2-layer MLP with LayerNorm and ReLU activation. For atom coordinates, we use a sigmoid β schedule with $\beta_1 = $ `1e-7` and $\beta_T = $ `2e-3`. For atom types, we adopt a cosine β schedule as suggested by (Nichol & Dhariwal, 2021), with `s=0.01`. We set the number of diffusion steps to `1000`.

E.4. Baselines

We conducted a comparative evaluation of NucleusDiff with leading generative models for structure-based drug design, including liGAN [*], GraphBP [†], AR-SBDD [‡], Pocket2Mol [§], and TargetDiff [¶]. For each comparison model, we utilized the source code obtained from the respective repositories.

E.5. Configuration

All algorithms and models have been developed using Python 3.8.13, with PyTorch version 1.12.1 and PyTorch Geometric version 2.5.2, under CUDA 11.0. Experiments are conducted on a server with 8 NVIDIA V100 GPUs (32 GB memory) and Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz. We employ a single V100 GPU for model training, while leveraging eight GPUs to accelerate the sampling procedure.

[*]LiGAN (GPL-2.0 license): https://github.com/mattragoza/LiGAN.
[†]GraphBP (GPL-3.0 license): https://github.com/divelab/GraphBP.
[‡]AR-SBDD (MIT license): https://github.com/luost26/3D-Generative-SBDD.
[§]Pocket2Mol (MIT license): https://github.com/pengxingang/Pocket2Mol.
[¶]TargetDiff (MIT license): https://github.com/guanjq/targetdiff.