OXFORD

Briefings in Bioinformatics, 2025, **26(2)**, bbaf094

https://doi.org/10.1093/bib/bbaf094

Problem Solving Protocol

MolEM: a unified generative framework for molecular graphs and sequential orders

Hanwen Zhang[1,2], Deng Xiong[3], Xianggen Liu[1,2,*], Jiancheng Lv[1,2,*]

[1]College of Computer Science, Sichuan University, No.24 South Section 1, Yihuan Road, Chengdu 610065, China
[2]Engineering Research Center of Machine Learning and Industry Intelligence, Ministry of Education, No.24 South Section 1, Yihuan Road, Chengdu 610065, China
[3]Department of Mechanical Engineering, Stevens Institute of Technology, 1 Castle Point Terrace, Hoboken, NJ 07030, United States

*Corresponding authors. Xianggen Liu, College of Computer Science, Sichuan University, Chengdu 610065, China. Engineering Research Center of Machine Learning and Industry Intelligence, Ministry of Education, China. E-mail: liuxianggen@scu.edu.cn; Jiancheng Lv, College of Computer Science, Sichuan University, Chengdu 610065, China. Engineering Research Center of Machine Learning and Industry Intelligence, Ministry of Education, China. E-mail: lvjiancheng@scu.edu.cn

Abstract

Structure-based drug design aims to generate molecules that fill the cavity of the protein pocket with a high binding affinity. Many contemporary studies employ sequential generative models. Their standard training method is to sequentialize molecular graphs into ordered sequences and then maximize the likelihood of the resulting sequences. However, the exact likelihood is computationally intractable, which involves a sum over all possible sequential orders. Molecular graphs lack an inherent order and the number of orders is factorial in the graph size. To avoid the intractable full space of factorially-many orders, existing works pre-define a fixed node ordering scheme such as depth-first search to sequentialize the 3D molecular graphs. In these cases, the training objectives are loose lower bounds of the exact likelihoods which are suboptimal for generation. To address the challenges, we propose a unified generative framework named MolEM to learn the 3D molecular graphs and corresponding sequential orders jointly. We derive a tight lower bound of the likelihood and maximize it via variational expectation-maximization algorithm, opening a new line of research in learning-based ordering schemes for 3D molecular graph generation. Besides, we first incorporate the molecular docking method QuickVina 2 to manipulate the binding poses, leading to accurate and flexible ligand conformations. Experimental results demonstrate that MolEM significantly outperforms baseline models in generating molecules with high binding affinities and realistic structures. Our approach efficiently approximates the true marginal graph likelihood and identifies reasonable orderings for 3D molecular graphs, aligning well with relevant chemical priors.

Keywords: structure-based drug design; sequential generative model; molecular graph generation; sequential order; variational expectation-maximization

Introduction

Generation of protein binding molecules represents the core of structure-based drug design (SBDD). In general, proteins often act as targets and regulate important biological functions through drug–target interactions. Understanding the interactions is the key to guiding the design of molecules [1–3]. Contemporary deep generative methods formulate structure-based molecular design as a conditional graph generation task, which aims to train a model to generate 3D molecular graphs with high affinity and specificity conditioned on protein pockets [4–7].

One branch of work uses one-shot generation strategies to generate graphs, which learn continuous mappings from latent distributions to predefined output spaces [8–15]. For example, liGAN [8, 9] utilizes conditional VAEs and GANs to generate 3D ligands; diffusion-based graph generative models apply Gaussian diffusion in dequantized adjacency matrix space [11–13, 15]. These models suffer limitations such as the inability to generate graphs with arbitrarily large numbers of nodes, the impracticality of hundreds or even thousands of forward passes for downstream applications, and the inevitability of generating molecules with disconnected parts [14, 16, 17]. Another branch of work involves sequential generation methods, such as reinforcement-learning models and autoregressive models, which naturally accommodate variable-sized graphs [16, 18, 19]. These methods require an approach to sequentialize molecular graphs into an ordered sequence of tokens (i.e. motifs or fragments), and then maximize the likelihood of the resulting sequences. However, the exact likelihood is computationally intractable. The generation sequence of a 3D graph is arbitrary because it is defined by a non-Euclidean data structure consisting of nodes and edges without an inherent order [14, 20–23]. As shown in Fig. 1, different possible orders can generate the same graph. Computing the likelihood requires marginalizing out all possible sequential orders. The marginalization is intractable in practice due to the number of terms in the sum, as there are n! sequences leading to a graph with n tokens, which makes maximum likelihood estimation challenging. To avoid the intractable full space of factorially many orders, existing works use a single generative order for a graph [24–29]. They predefine a fixed node ordering scheme based on heuristics and graph traversals without fully justifying their choices empirically or theoretically. For example, DeepLigBuilder [24] adopts a policy network combined with Monte Carlo tree search, which is required to imitate an "expert trajectory" using a canonical ranking algorithm in RDKit and a depth-first-search (DFS); DeepICL

Received: October 28, 2024. **Revised:** January 27, 2025. **Accepted:** February 15, 2025
© The Author(s) 2025. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (https:/creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact journals.permissions@oup.com

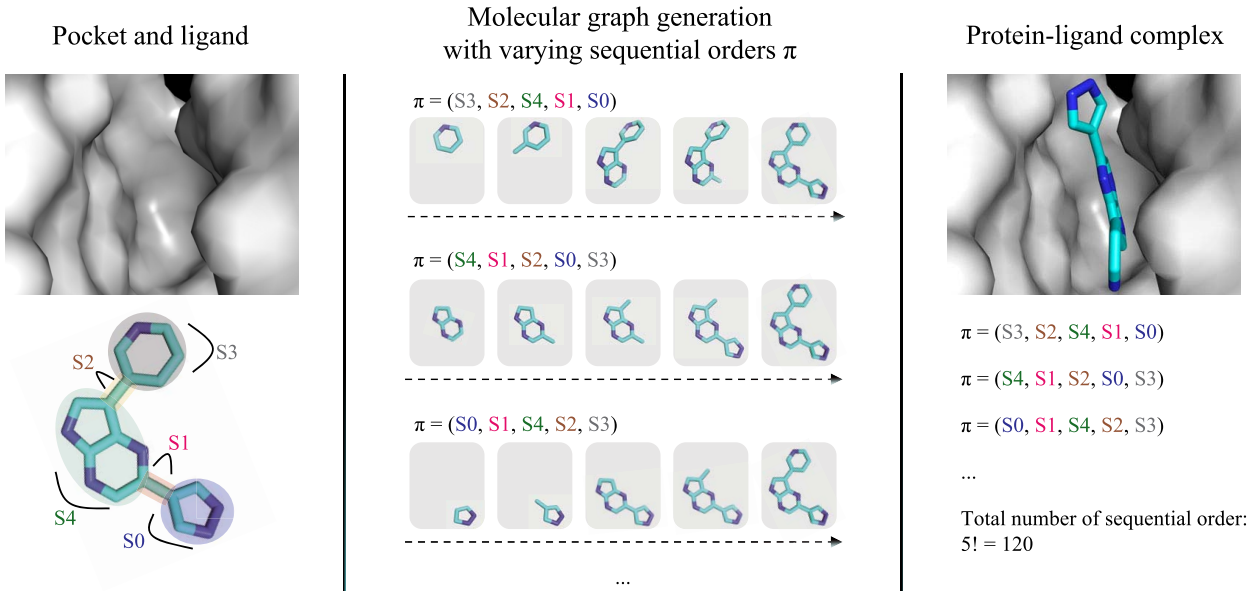

Figure 1. A 3D molecular graph is defined by a non-Euclidean data structure without an inherent order. The PDB ID of the pocket is 1HW8. The ligand has five fragments numbered from S0 to S4. 120 sequential orders can generate it, three of which are depicted.

[28] randomly traverses a trajectory of placing atoms of a ligand; PocketFlow [29] imposes an artificial ordering scheme of graph traversals such as breadth-first-search (BFS). In these cases, the training objective is a loose lower bound of the exact likelihood, which only considers the probability of one sequential order. The training objective does not correspond to rigorous maximum likelihood estimation although it is easy to compute. Therefore, those pre-defined canonical ordering schemes lead to suboptimal performances.

Additionally, previous models generate ligands with unrealistic binding poses and rely on post-fixing [5, 17, 24, 30–32], suggesting that they do not capture true protein–ligand interactions. Research on binding conformation generation can generally classified into two streams: generation simultaneously with a molecular graph (3D ligand generation) and generation for a given pocket–ligand pair (molecular docking). Prior works of the first stream usually adopt distance-based initialization, rotation angle prediction, and rotation matrix to generate conformations [33, 34]. These methods lose flexibility and accuracy since the already-generated molecular motifs are anchored. To fix their generated 3D structures, a notable number of generated ligand conformations rearrange drastically after redocking [6, 7, 26, 35]. The inaccurate binding conformations inhibit the models from capturing true interatomic interactions and harm the credibility of generating molecules directly in 3D space. Another stream of conformation generation is molecular docking, which is a widely-used technique in drug design. Given a ligand and a ligand binding site (called "pocket"), it predicts the binding mode of the protein–ligand complex [36–38]. Traditional docking programs such as AutoDock4 [39], AutoDock Vina [40], QuickVina2 [41], Glide [42], and GOLD [43] leverage heuristic search algorithms to identify one or several near-native docking poses [37, 38]. Recently, deep learning-based docking methods are full of potential [2, 37, 44]. However, they suffer significant limitations [23, 36, 38, 45–48]: (i) deep learning-based models are good at pocket searching, but traditional methods are better than them at molecular docking on given pockets; (ii) the experiments and evaluations are somewhat questionable, and the models have rarely been tested in practical application scenarios; (iii) the deep learning-based models introduce additional dataset bias and some models demonstrate superior scoring power but they perform poorly in docking power. Collectively, traditional molecular docking methods remain adept at predicting accurate binding poses in most scenarios.

To address the challenges, we develop a theoretical framework to learn 3D molecular graphs and generative orders jointly, surpassing the pre-defined canonical ordering schemes. We derive a tight lower bound of the likelihood and maximize it via variational expectation-maximization algorithm [49]. In our framework, the 3D molecular graph is generated by a molecule generator trained under the estimated posterior probability of sequential orders. The distribution of sequential orders is modeled by an ordering generator, approximating the true distribution that maximizes the graph likelihood. Figure 2 illustrates our new framework. Convergence is assured since the algorithm is guaranteed to increase the likelihood at each iteration. To ensure the generation of dependable binding conformations, we first incorporate the molecular docking approach QuickVina 2 (QVina2) [41] into our framework. We do not use the Vina score to guide the molecule generation although it is widely utilized by previous methods [26, 50, 51]. We only make use of the docked binding poses. Experimental results demonstrate that MolEM efficiently approximates the true marginal graph likelihood. Our method can find reasonable orders for 3D molecular graphs, adhering to suitable chemical priors. The generated molecules show higher binding affinities and contain more realistic structures than the state-of-the-art baseline methods. This study opens a new line of research in developing learning-based ordering schemes for 3D molecular graph generation, contributing to the advancement of this field.

Methods

Preliminary and problem formulation

SBDD aims to generate ligand molecules with high affinity and selectivity of a given protein pocket. The 3D geometries of the molecule and pocket can be represented as two sets of atoms $\mathcal{G}^{mol} = (\boldsymbol{a}_i^{mol}, \boldsymbol{r}_i^{mol})$ and $\mathcal{P}^{pro} = (\boldsymbol{a}_j^{pro}, \boldsymbol{r}_j^{pro})$ respectively, where $\boldsymbol{a}_i^{mol}$ and $\boldsymbol{a}_j^{pro}$ are one-hot vectors indicating the atom attributes and

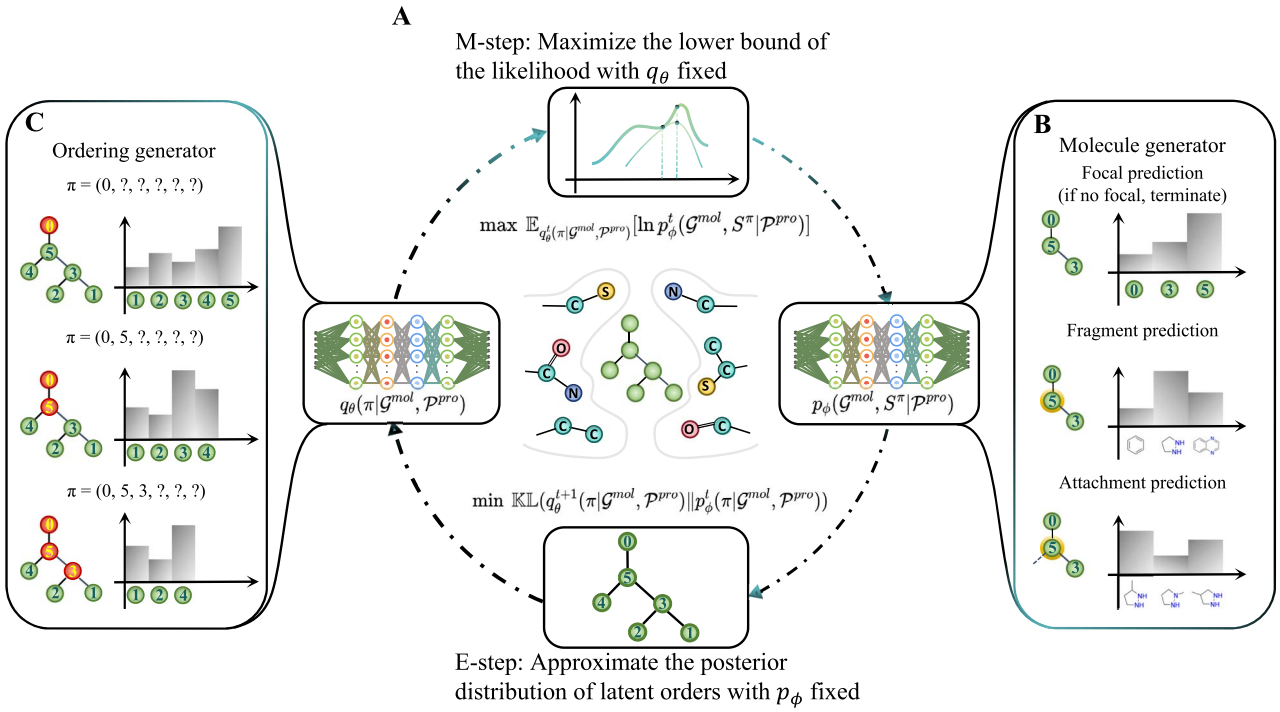

Figure 2. Overview of MolEM framework. (A) The 3D molecular graphs and generative orders are jointly optimized. The E-step and M-step form a closed loop. (B) The generation procedure of the molecule generator. (C) The generation scheme of the ordering generator. The fragments highlighted with red are already predicted.

$\mathbf{r}_i^{mol}, \mathbf{r}_j^{pro} \in \mathbb{R}^3$ denote coordinates. Our objective is to learn a conditional graph generative model $p_\phi(\mathcal{G}^{mol}|\mathcal{P}^{pro})$, where ϕ represents the parameters to be learned. The training dataset of observed protein–ligand complexes is denoted as $\mathcal{G}^{PL}$. A glossary of symbols is provided in Table S1.

We formulate the conditional graph generation task as a sequential generative problem, i.e. modeling graphs as sequences, the tokens of which are molecular fragments. We first define a mapping f_S from graphs to sequences, where for a graph $\mathcal{G}^{mol} \sim p_\phi(\mathcal{G}^{mol}|\mathcal{P}^{pro})$ with n tokens under ordering π, we have

$$S^\pi = f_S(\mathcal{G}^{mol}, \pi) = (\mathcal{G}_1^\pi, ..., \mathcal{G}_n^\pi), \tag{1}$$

where each element $\mathcal{G}_i^\pi$ represents the induced subgraph of $\mathcal{G}^{mol}$ from the first i tokens in the ordering π. Conversely, we define a mapping $f_\mathcal{G}(\cdot)$ from sequences to graphs as S^π determines a unique graph $\mathcal{G}^{mol}$ where $f_G(S^\pi) = \mathcal{G}^{mol}$. Thus, instead of learning $p_\phi(\mathcal{G}^{mol}|\mathcal{P}^{pro})$, whose sample space cannot be easily characterized, we sample the sequential order π to get the observation of S^π and learn $p_\phi(S^\pi|\mathcal{P}^{pro})$, which can be modeled autoregressively with the sequential nature of S^π and be decomposed to Markov chain [52]:

$$p_\phi(S^\pi|\mathcal{P}^{pro}) = \prod_{i=1}^{n} p(\mathcal{G}_i^\pi|\mathcal{G}_{i-1}^\pi, \mathcal{P}^{pro}). \tag{2}$$

Given the above definitions, the joint distribution of a molecular graph and the generative order can be represented as

$$p_\phi(\mathcal{G}^{mol}, S^\pi|\mathcal{P}^{pro}) = p_\phi(S^\pi|\mathcal{P}^{pro})\mathbb{1}[f_G(S^\pi) = \mathcal{G}^{mol}], \tag{3}$$

and we can write $p_\phi(\mathcal{G}^{mol}|\mathcal{P}^{pro})$ as the marginalization of the joint distribution:

$$p_\phi(\mathcal{G}^{mol}|\mathcal{P}^{pro}) = \sum_{S^\pi} p_\phi(\mathcal{G}^{mol}, S^\pi|\mathcal{P}^{pro}). \tag{4}$$

The marginalization is intractable due to the number of terms in sum, i.e. there are $n!$ possible orderings for n graph tokens. Consequently, $p(\mathcal{G}^{mol}|\mathcal{P}^{pro})$ cannot be easily obtained. Previous works propose to train under a chosen canonical node ordering scheme. We remove the requirement of a fixed generative order by developing a unified generative framework.

The generative framework

We learn the model parameters ϕ by maximizing the log-likelihood function of the observed molecular graphs conditioned on protein pockets, i.e. $\ln p_\phi(\mathcal{G}^{mol}|\mathcal{P}^{pro})$. However, directly maximizing the log-likelihood function is impractical. Therefore, we instead optimize the evidence lower bound (ELBO) of the log-likelihood function iteratively.

Theorem 1.1. (ELBO of $\ln p_\phi(\mathcal{G}^{mol}|\mathcal{P}^{pro})$). The exact log-likelihood satisfies the following inequality:

$$\ln p_\phi(\mathcal{G}^{mol}|\mathcal{P}^{pro}) \geq \mathbb{E}_{q_\theta(\pi|\mathcal{G}^{mol},\mathcal{P}^{pro})} [\ln p_\phi(\mathcal{G}^{mol}, S^\pi|\mathcal{P}^{pro}) - \ln q_\theta(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro})]. \tag{5}$$

where $q_\theta(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro})$ denotes the estimated distribution of sequential order π, and the equation holds when $p_\phi(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro}) = q_\theta(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro})$. The equal condition reveals that the approximated distribution of the sequential order should be equal to the true posterior distribution that maximize the graph likelihood. The proof of this theorem is provided in Appendix A.

Such a lower bound can be optimized by alternating between an E-step and an M-step [49, 53–56]. We define t as the iteration step. In the E-step (a.k.a., inference procedure), the goal is to fix p_ϕ^t and update the variational distribution q_θ^{t+1} to approximate the true posterior distribution p_ϕ by minimizing the KL divergence

with the following equation:

$$\ell_E(\theta^{t+1}) = \mathbb{KL}(q_\theta^{t+1}(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro}) \| p_\phi^t(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro})). \quad (6)$$

In the M-step (a.k.a., learning procedure), we fix q_θ^t and update p_ϕ^t to maximize the lower bound with the latent order determined:

$$\ell_M(\phi^t) = \mathbb{E}_{q_\theta^t(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro})}[\ln p_\phi^t(\mathcal{G}^{mol}, S^\pi|\mathcal{P}^{pro})]. \quad (7)$$

See Appendix A for the proof. The distribution of $q_\theta(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro})$ is parameterized using an ordering generator and the distribution of $p_\phi(\mathcal{G}^{mol}, S^\pi|\mathcal{P}^{pro})$ is parameterized using a molecule generator. Next, we detail the two steps respectively.

E-step: inference procedure

In the E-step, we use $q_\theta(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro})$ to approximate the posterior distribution $p_\phi(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro})$, where the p_ϕ is fixed and the q_θ is implemented with an ordering generator.

Consistent with conditional graph generation in Equation (2), we first formulate the distribution of $q_\theta(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro})$ in an autoregressive manner:

$$q_\theta(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro}) = \prod_{j=1}^{n} q_\theta(\pi_j|\pi_{j-1}, \mathcal{G}^{mol}, \mathcal{P}^{pro}), \quad (8)$$

where each element π_j denotes molecular fragments generated before the j^{th} step. To model the distribution, we utilize a graph transformer [57] to learn atom-level representation $\boldsymbol{h}_{\theta,a}^{(\pi_{j-1})}$, residue-level representation $\boldsymbol{h}_{\theta,r}^{(\pi_{j-1})}$, and molecular fragment embedding $\boldsymbol{e}_\theta^{(\pi_{j-1})}$ for a given protein–ligand complex, capturing the context information of binding sites. We use a multilayer perceptron MLP_θ to fuse the representations and predict the next token:

$$\begin{aligned} &q_\theta(\pi_j|\pi_{j-1}, \mathcal{G}^{mol}, \mathcal{P}^{pro}) = \\ &softmax(MLP_\theta(\boldsymbol{h}_{\theta,a}^{(\pi_{j-1})}, \boldsymbol{h}_{\theta,r}^{(\pi_{j-1})}, \boldsymbol{e}_\theta^{(\pi_{j-1})})). \end{aligned} \quad (9)$$

We present the architecture details of the ordering generator in Appendix B.

In practice, problems exist in implementing the E-step: sampling from $p_\phi^t(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro})$ is very inefficient and the $\hat{\pi}_m^{t+1}$ sampled from p_ϕ^t cannot be guaranteed to meet the condition of $f_G(\hat{\pi}_m^{t+1}) \in \mathcal{G}^{PL}$. Therefore, we use a heuristic way when constructing the training dataset $(\mathcal{G}^{PL}, \hat{\pi}_m^{t+1})$: the distribution of $p_\phi^t(\mathcal{G}^{mol}, S^\pi|\mathcal{P}^{pro})$ learned by the molecule generator, is used for scoring, i.e. measuring the probability of orderings in a pre-defined ordering set $\pi_\mathcal{G}$, according to Equation (3). The orderings with maximum probability are chosen:

$$\hat{\pi}_m^{t+1} = \underset{\pi \in \pi_G}{\operatorname{argmax}} p_\phi^t(\mathcal{G}^{mol}, S^\pi|\mathcal{P}^{pro}), \quad (10)$$

Then we train q_θ^{t+1} on $(\mathcal{G}^{PL}, \hat{\pi}_m^{t+1})$ to minimize the KL divergence according to Equation (6):

$$\begin{aligned} \theta^{t+1} =& \underset{\theta}{\operatorname{argmin}} \mathbb{E}_{((\mathcal{G}^{mol}, \mathcal{P}^{pro}), \pi) \sim (\mathcal{G}^{PL}, \hat{\pi}_m^{t+1})} \\ & \mathbb{KL}(q_\theta^{t+1}(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro}) \| p_\phi^t(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro})), \end{aligned} \quad (11)$$

We let q_θ^{t+1} converge under the current molecule generator in each E-step.

Algorithm Training algorithm of MolEM

Require: Training dataset $\mathcal{G}^{PL}$ and pre-defined ordering $\pi_\mathcal{G}$
Ensure: Well trained p_ϕ and q_θ
Let t = 0.
Pre-train q_θ^1 on $(\mathcal{G}^{PL}, \pi_\mathcal{G})$ according to Eq. (11).
while not converged **do**
t = t + 1
⊡ **M-step: Learning Procedure**
Predict orderings for $\mathcal{G}^{PL}$ with q_θ^t according to Eq. (9).
Consturct the dataset $(\mathcal{G}^{PL}, \hat{\pi}_o^t)$ with docking.
Train p_ϕ^t on $(\mathcal{G}^{PL}, \hat{\pi}_o^t)$ according to Eq. (12).
⊡ **E-step: Inference Procedure**
Score orderings for $\mathcal{G}^{PL}$ with p_ϕ^t according to Eq. (10).
Construct the dataset $(\mathcal{G}^{PL}, \hat{\pi}_m^{t+1})$ with docking.
Train q_θ^{t+1} on $(\mathcal{G}^{PL}, \hat{\pi}_m^{t+1})$ according to Eq. (11).
end while

M-step: learning procedure

In the M-step, we maximize the lower bound of the log-likelihood according to Equation (7), where q_θ is fixed and p_ϕ is implemented with a molecule generator.

To model the distribution of $p_\phi(\mathcal{G}^{mol}, S^\pi|\mathcal{P}^{pro})$, we parameterize p_ϕ with a graph transformer, the architecture of which is the same as the one in E-step. The model generates ligands fragment-by-fragment. For each molecular fragment, the generation procedure consists of four major steps, including (i) focal selection, (ii) next fragment prediction, (iii) fragment attachment prediction, and (iv) binding pose prediction. More details of the molecule generator are presented in Appendix B. We sample from $q_\theta^t(\pi|\mathcal{G}^{mol}, \mathcal{P}^{pro})$ to obtain the $\hat{\pi}_o^t$ to construct the training dataset $(\mathcal{G}^{PL}, \hat{\pi}_o^t)$ for the model: given the $\mathcal{G}^{PL}$, the ordering generator predicts generative orderings according to Equations (8) and (9). Then we train p_ϕ^t on $(\mathcal{G}^{PL}, \hat{\pi}_o^t)$ to maximize the lower bound according to Equation (7):

$$\phi^t = \underset{\phi}{\operatorname{argmax}} \mathbb{E}_{((\mathcal{G}^{mol}, \mathcal{P}^{pro}), \pi) \sim (\mathcal{G}^{PL}, \hat{\pi}_o^t)}[\ln p_\phi^t(\mathcal{G}^{mol}, S^\pi|\mathcal{P}^{pro})], \quad (12)$$

We let p_ϕ^t converge under the current ordering generator in each M-step.

Joint optimization

In our framework, the 3D molecular graphs and generative orders are jointly optimized to maximize the tight lower bound of the likelihood. The E-step and M-step form a closed loop. We initially pre-train the ordering generator on pre-defined orderings π_G as q_θ^1. Then we alternatively optimize p_ϕ and q_θ until convergence. The training algorithm is summarized in Algorithm 1.

Results

To evaluate the generation performance of MolEM, we use the CrossDocked dataset [59], which contains 22.5 million protein–ligand complexes, and follow the same data preparation and data splitting as [12]. A detailed description of the dataset is provided in Appendix B. In the end, we have 99940 complexes for training, 100 complexes for validation, and 100 novel complexes as references for testing. We have built an expanded dataset from CrossDocked to further evaluate the performance of MolEM, as shown in Table S2. We compare various baselines: Pocket2Mol [25], DrugGPS [34], TargetDiff [12], FLAG [58], and PMDM [13]. Pocket2Mol, DrugGPS, and FLAG adopt an autoregressive strategy. TargetDiff and PMDM

Table 1. Properties of the test set molecules generated by different methods

Methods	Vina score (kcal/mol, ↓)	High affinity (↑)	QED (↑)	SA (↑)	LogP	Lipinski (↑)	Diversity (↑)
Testset	−7.027 ± 2.19	–	0.476 ± 0.20	0.723 ± 0.14	0.042 ± 2.32	4.090 ± 1.25	–
Pocket2Mol	−7.288 ± 2.53	0.542 ± 0.32	0.563 ± 0.16	**0.765 ± 0.13**	1.586 ± 1.82	4.902 ± 0.42	0.688 ± 0.14
DrugGPS	−7.345 ± 2.42	0.620 ± 0.29	0.592 ± 0.21	0.728 ± 0.23	1.134 ± 2.26	4.923 ± 0.11	0.695 ± 0.17
TargetDiff	−7.839 ± 2.32	0.581 ± 0.29	0.509 ± 0.20	0.559 ± 0.13	1.602 ± 2.41	4.570 ± 0.77	0.738 ± 0.15
FLAG	−7.247 ± 2.25	0.580 ± 0.24	0.495 ± 0.17	0.745 ± 0.16	0.630 ± 2.38	4.943 ± 0.14	0.704 ± 0.15
PMDM	−7.572 ± 2.50	0.628	**0.594 ± 0.12**	0.611 ± 0.16	0.301 ± 1.01	**4.975 ± 0.16**	0.709 ± 0.10
MolEM	**−8.297 ± 1.15**	**0.665 ± 0.32**	0.525 ± 0.07	0.607 ± 0.03	0.349 ± 0.34	4.816 ± 0.06	**0.792 ± 0.10**

Arrows ↑/↓ denote that higher/lower values are preferable. Baseline results are taken from their papers [13, 25, 34, 58]. For each metric, we **bold** the best.

Table 2. Ablation studies for MolEM, including ordering schemes and binding conformations

Ablation	Vina score (kcal/mol, ↓)	QED (↑)	SA (↑)	LogP	Lipinski (↑)
MolEM	**−8.297 ± 1.15**	**0.525 ± 0.07**	0.605 ± 0.05	0.386 ± 0.60	4.832 ± 0.12
Ordering scheme					
BFS	−7.177 ± 2.19	0.458 ± 0.06	0.580 ± 0.04	−0.073 ± 0.60	4.858 ± 0.13
DFS	−7.484 ± 1.34	0.463 ± 0.06	0.583 ± 0.05	0.189 ± 0.70	4.843 ± 0.15
Random	−7.110 ± 1.34	0.449 ± 0.05	0.579 ± 0.04	0.085 ± 0.54	4.812 ± 0.14
Canonical family	−7.102 ± 1.16	0.439 ± 0.06	0.580 ± 0.04	−0.149 ± 0.58	4.835 ± 0.14
Binding conformation					
Rotation angle prediction	−6.608 ± 2.42	0.501 ± 0.21	0.637 ± 2.26	0.866 ± 0.23	**4.929 ± 0.11**
Crystal poses	−7.001 ± 1.46	0.480 ± 0.06	**0.640 ± 0.07**	0.745 ± 0.56	4.918 ± 0.11

We assume the positions directly from the dataset are crystal poses.

are diffusion-based models. The details of hyper-parameters are presented in Appendix B. At each iteration, we fully train generators and choose models with the lowest error in the validation set. MolEM stops the iteration when there is no improvement on the lower bound of the likelihood within five steps. We pick the molecule generator from the convergence iteration to generate molecules.

Evaluation of generated molecules

We adopt widely-used metrics to evaluate the properties of sampled molecules [25]: (i) **Vina score** estimates the binding affinity between the molecules and the pockets [41]. It is obtained after a full re-docking process, involving both global and local optimization in the molecular docking search space; (ii) **High Affinity** measures the percentage of pockets that have generated molecules with higher or equal affinities than the references in the test set; (iii) **QED** is the qualitative estimation of drug-likeness; (iv) **Synthesizability (SA)** represents the difficulty of drug synthesis (normalized between 0 and 1 and higher values indicate easier synthesis); (v) **LogP** is the octanol-water partition coefficient and in general LogP values range from -0.4 to 5.6 [60]; (vi) **Lipinski** calculates how many rules the molecule obeys the Lipinski's rule of five [61, 62]; (vii) **Diversity** measures the diversity of generated molecules for a pocket. It is calculated by averaging pairwise Tanimoto distances [63, 64]. Additionally, we analyze the local geometry to demonstrate the geometric and physical validity of the generated molecules, as shown in Fig. S1 and Tables S3, S4, S5. Furthermore, we measure the strain energy and steric clashes of binding conformations to assess the model's ability to handle intricate chemical structures and spatial configurations, as shown in Table S6.

The mean values of the above metrics are shown in Table 1. In general, our method achieves competitive performance compared with the baseline methods. The generated molecules from MolEM show the best binding affinity. According to the Vina score, MolEM outperforms the baseline methods significantly (-8.297 versus -7.247 to -7.839). In terms of high-affinity metric, MolEM generates molecules with higher affinity than reference molecules over 67% of the protein pocket. The molecules generated by MolEM have achieved the highest diversity. These results demonstrate that order matters to obtain the best performance. Table 2 also empirically shows that different generative ordering schemes result in performance gaps in autoregressive models. Pre-defined orderings may destroy the underlying structure of the molecule and make modeling the joint probability much more difficult. Note that 100% of the molecules are valid because accurate conformations are ensured in each generation step. The metrics MolEM does fall behind are the drug potentials (QED, SA, Lipinski). We put less emphasis on them because in SBDD, QED, SA, and Lipinski are used as rough screening metrics and would be fine as long as they are in a reasonable range [12, 65]. Notably, MolEM has a bias toward generating molecules with lower SA scores than baselines. This is probably because MolEM is trained to generate molecular graphs in learning-based order rather than the canonical order, therefore more likely to generate things that are harder to synthesize.

Ablation studies

Ablation studies are designed to further demonstrate the effectiveness of MolEM. We explore the impact of ordering schemes on 3D molecular graph generation and the influence of conformation poses obtained by different methods. Table 2 shows the

performances. Existing works pre-define a canonical node ordering scheme relying on conventional heuristics or graph traversal methods. The widely adopted methods are breadth-first traversal, depth-first traversal, and uniform random ordering. The canonical family is the mixture of these pre-defined ordering schemes. As the flexibility of molecular structures largely lies in the degree of rotatable bonds, previous methods adopt the binding position prediction and distance matrix-based algorithm to locate the ligand [34, 58]. The results in Table 2 demonstrate that the variational distribution learns reasonable orderings that improve the generative model's performance. Other ordering schemes make modeling the probability suboptimal and may destroy the underlying structure of the molecular graph. Through step-by-step manipulations of conformations to ensure accuracy, we alleviate the accumulated errors of conformations and generate the molecules with guidance of protein–ligand interactions as suggested from the docked poses.

Table 3. Analysis of the convergence for the MolEM framework

	Vina score (kcal/mol, ↓)	Approximated evidence lower bound
t = 0	−8.144	−9.35
t = 1	−7.848	−9.05
t = 2	−7.843	−8.97
t = 3	−7.931	−8.94
t = 4*	**−8.297**	**−8.88**
t = 5	−8.032	−8.89
t = 6	−8.016	−9.02
t = 7	−8.225	−9.00
t = 8	−8.16	−9.00
t = 9	−8.02	−8.99

A lower Vina score indicates higher binding affinity and a higher likelihood reveals more true distribution approximated.

Framework analysis

Next, we analyze the convergence of MolEM empirically. The theoretical analysis is shown in Appendix A. At each iteration, the molecule generator and the ordering generator are trained interactively. In Table 3, we demonstrate the performance of the molecule generator (i.e. Vina score) and the approximated lower bound of the log-likelihood. Specifically, the approximated ELBO is computed according to Equation (7). The computing formula is presented in Appendix A. Intriguingly, the generative performance and the likelihood achieve their best at the same step (i.e. $t = 4^*$). The best Vina score indicates that MolEM generates molecular graphs with reasonable orderings compared with other steps. Besides, MolEM takes only a few iterations to converge. A detailed analysis of convergence efficiency is presented in Appendix C. We summarize the convergence steps and time in Table S8. These results demonstrate that our framework jointly optimizes the 3D molecular graphs and the latent orders through efficient step-by-step elaboration, maximizing the tight lower bound of the likelihood.

Case studies

We test how well our model finds reasonable orders for 3D molecular graphs and generalizes them to unseen protein binding sites. Besides, we test whether the model generates the molecules with accurate binding conformations and accounts for protein–ligand interactions as suggested from the docked poses.

In Fig. 3, we provide an example from the training set for visualization. The learning-based order is generated in the convergence iteration (i.e. $t = 4^*$) by the molecule generator and the ordering generator, which reach an agreement. For the first generation step, the canonical order chooses a carbon–carbon bond, which forms a huge departure from the true position and impacts the following generation. The learned-based order selects Benzopyrazine as the first fragment and occupies a binding site. The choice of the initial fragment demonstrates that the model incorporates structures of the protein pocket into the selection of chemical fragments. At the third generation step, the learning-based order chooses the carbon–oxygen bond, which extends into the cleft noted by the red rectangle to make interactions. The cleft is a "cryptic site"

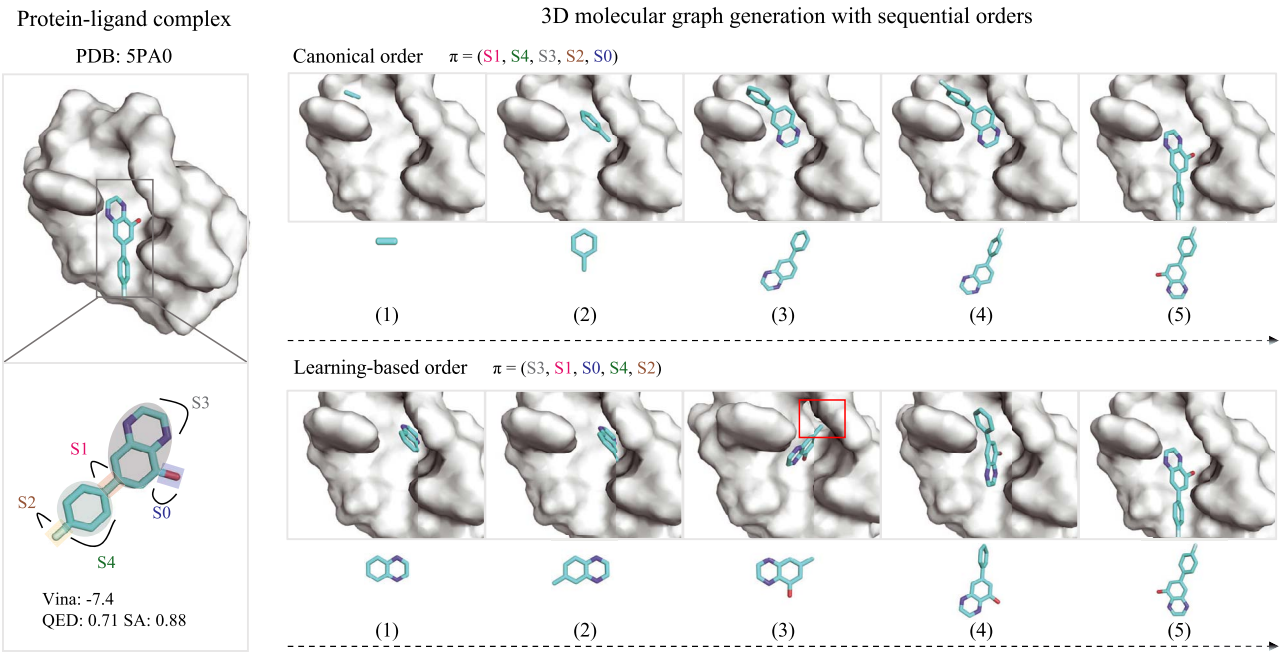

Figure 3. The comparison of generative orders. The ordering above is a canonical order derived from BFS and the ordering below is generated by MolEM. The binding site, indicated by a red rectangle, is a "cryptic site" for the generation of canonical order.

for the generation of canonical order. It identifies unobstructed attachment locations that point toward unfilled areas within the protein pocket and then extends the ligand effectively. At the final generation step, the exact location and orientation of the fragments are adjusted to the true binding position.

In Fig. 4, we conduct a case study on the generative orders of similar protein–ligand environments and compare ligand generations on the unseen pocket with the baseline model. Comparing the two generation trajectories, we can see that the already generated fragments are fixed as anchors in DrugGPS, making the

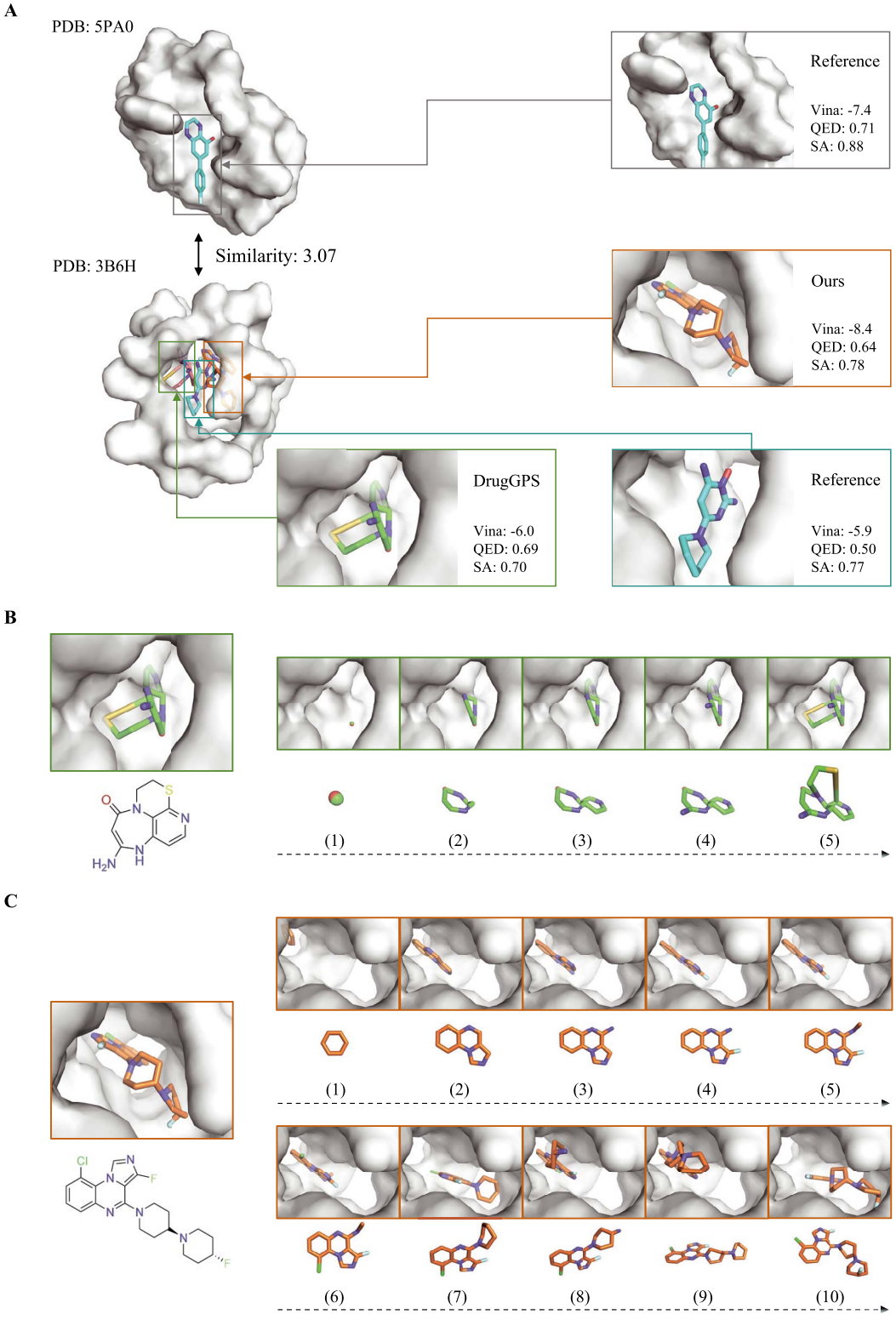

Figure 4. The case study of the generative orders of similar protein–ligand environments and comparison with the baseline model. (A) The similarity between the pockets is 3.07, which is the top 0.1%. For pocket 3B6H, we show the reference ligand, the molecule generated by DrugGPS and MolEM, respectively. (B) The generation procedure of DrugGPS. (C) The generation trajectory of MolEM.

model suffer from deviation accumulations. The invalid structure violates biophysical constraints and produces steric clashes with the protein surface. On the contrary, the conformations generated by MolEM evolve with the addition of fragments. These fragments can flexibly fill the interior of the pocket. Besides, our model recovers the privileged structure patterns, performs comprehensive explorations of protein pockets, and adopts complementarity to the binding sites. Generative orders with similar protein environment information have a clear tendency. Concretely, we compute the Euclidean similarities of pocket representations between the training set and the test set. We depict the most similar pocket to the one in Fig. 3. MolEM generates Benzopyrazine at the first step for PDB 5PA0 and Imidazo[1,5-a]quinoxaline at the prior two steps for PDB 3B6H. The selections demonstrate that generating crucial substructures preferentially has been learned and applied to similar pocket environments, aligning with the well-established chemical prior. In comparison, DrugGPS generates a carbon–oxygen double bond at the first step and a heterocyclic ring at the final step, forming an inefficient generation procedure. Moreover, the protein pocket has a small pit deep inside the protein pocket, which is far from the center of the whole pocket. MolEM has successfully grown fragments within the pit in the first to the sixth step, and the conformations are adjusted to the pit in the eighth and ninth steps. In contrast, it is a "cryptic site" for the baseline model and the standard reference. In addition, MolEM generates protein-interacting fragments at two ends of the uneven protein pocket, which are Imidazo[1,5-a]quinoxaline (i.e. the second step) and 1'-Methyl-1,4'-bipiperidine (i.e. the ninth step). The subpockets have been well occupied, providing good steric complementarity. Comparatively, the reference ligand and the molecule generated by DrugGPS are confined to a corner of the protein pocket.

Additionally, we have provided failure cases of MolEM and the other baselines in Fig. S2 to analyze the limitations and advantages of different computational models. Furthermore, we visualize the generated molecules for a realistic functional protein target in Fig. S3, demonstrating the power of our model in real-world application scenarios.

Discussion

Binding pose generation. The current structure-based AI models are severely hindered by neglecting conformational flexibility (dynamics and induced fit upon binding) [66]. This work incorporates molecular docking to ensure dependable binding conformations and promote favorable protein–ligand interactions. As shown in Table S7, we perturb the conformations of both the generated molecules and protein pockets to assess the robustness and stability of the model. However, it is worth acknowledging the inherent limitations of docking, i.e. proteins are treated as rigid. Considering protein flexibility in the model may help improve the generation performance. In the future, the model may be complemented with Monte Carlo or molecular dynamics (MD) simulation. Besides, physics-based methods are computationally expensive. A detailed analysis of time complexity is presented in Appendix C. For the training efficiency, we summarize the training time in Table S9. For the sampling efficiency, we conduct ablation studies to assess the generation time of each computational model, as shown in Table S10. Frequent calls to docking or MD are time-consuming, which reduces the speed of drug design models. We leave the efficiency improvement to future work.

Evaluation metrics. Evaluating generative models is known to be challenging. The most commonly used metrics for SBDD are docking scores. However, the docking score is an imperfect measure. Molecules with low-ranked scores may also have high binding affinities. Moreover, the docking score cannot accurately measure scenarios such as activity cliffs, where ligands with similar structures are scored similarly no matter whether the binding affinities are approximate. More rigorous metrics are needed for SBDD.

Conclusion

In this study, we propose MolEM, a unified generative framework for molecular graphs and sequential orders. The molecular docking approach QVina2 is incorporated to manipulate the molecular conformations. Results demonstrate that reasonable orders exist for 3D molecular graphs that are optimal for generation. The generative model favors certain orders that provide an efficient approximation of the true marginal graph likelihood and align with suitable chemical prior knowledge. Significantly, MolEM pioneers a novel approach in learning-based ordering schemes for 3D molecular graph generation. We posit that this framework holds substantial promise in molecule design applications. In conclusion, our research presents a new direction for the future design of ligands and opens up fresh challenges in the field.

Key Points

- A novel generative framework that jointly learns the 3D molecular graphs and corresponding sequential orders was developed, opening a new line of research in learning-based ordering schemes for 3D molecular graph generation.
- The molecular docking method was first incorporated into the generative model, ensuring accurate binding conformations and promoting favorable protein–ligand interactions.
- MolEM efficiently approximates the true marginal graph likelihood and identifies reasonable orderings for 3D molecular graphs, aligning well with suitable chemical prior knowledge.
- MolEM outperforms baselines in generating drug candidates with higher binding affinities and more realistic structures.

Acknowledgments

The authors thank anonymous reviewers for valuable suggestions.

Supplementary data

Supplementary data are available at *Briefings in Bioinformatics* online.

Author contributions

H.Z., X.L., and J.L. conceived the research project. X.L. and J.L. supervised the research project. H.Z., X.L., and J.L. designed and implemented the MolEM framework. H.Z., X.L., and J.L. conducted the computational analyses. H.Z., D.X., X.L., and J.L. wrote the manuscript. All the authors discussed the experimental results and commented on the manuscript.

Conflict of interest: None declared.

Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 62206192, in part by the National Major Scientific Instruments and Equipments Development Project of National Natural Science Foundation of China under Grant 62427820, in part by the Natural Science Foundation of Sichuan Province under Grant 2023NSFSC1408 and 2024NSFTD0048, and in part by the Fundamental Research Funds for the Central Universities under Grant 1082204112364. We also acknowledge the support of Sichuan Province Engineering Technology Research Center of Broadband Electronics Intelligent Manufacturing.

Data availability

The code for MolEM is available at https://github.com/Tophoenix-zhw/MolEM.

References

1. Wang K, Zhou R, Li Y. *et al*. DeepDTAF: a deep learning method to predict protein–ligand binding affinity. *Brief Bioinform* 2021;**22**:bbab072. https://doi.org/10.1093/bib/bbab072.
2. Méndez-Lucio O, Ahmad M, del Rio-Chanona EA. *et al*. A geometric deep learning approach to predict binding conformations of bioactive molecules. *Nat Mach Intell* 2021;**3**:1033–9. https://doi.org/10.1038/s42256-021-00409-9.
3. Hongyu H, Tong W, He F. *et al*. The binding mechanism of failed, in processing and succeed inhibitors target SARS-CoV-2 main protease. *J Biomol Struct Dyn* 2023;**42**:10565–76. https://doi.org/10.1080/07391102.2023.2257800.
4. Zhang M, Qamar M, Kang T. *et al*. A survey on graph diffusion models: generative AI in science for molecule, protein and material. arXiv preprint, arXiv:2304.01565, 2023. https://arxiv.org/abs/2304.01565.
5. Hongyan D, Jiang D, Zhang O. *et al*. A flexible data-free framework for structure-based de novo drug design with reinforcement learning. *Chem Sci* 2023;**14**:12166–81. https://doi.org/10.1039/D3SC04091G.
6. Shuqi L, Yao L, Chen X. *et al*. 3D molecular generation via virtual dynamics. *Transact Mach Learn Res* 2024;**2024**:1–20.
7. Yanru Q, Qiu K, Song Y. *et al*. MolCRAFT: structure-based drug design in continuous parameter space. In *Proceedings of the 41st International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research*, pp. 41749–68. New York, NY: ACM, 2024.
8. Masuda T, Ragoza M, Koes DR. Generating 3D molecular structures conditional on a receptor binding site with deep generative models. arXiv preprint, arXiv:2010.14442, 2020. https://arxiv.org/abs/2010.14442.
9. Ragoza M, Masuda T, Koes DR. Learning a continuous representation of 3D molecular structures with deep generative models. arXiv preprint, arXiv:2010.08687, 2020. https://arxiv.org/abs/2010.08687.
10. Ragoza M, Masuda T, Koes DR. Generating 3D molecules conditional on receptor binding sites with deep generative models. *Chem Sci* 2022;**13**:2701–13. https://doi.org/10.1039/D1SC05976A.
11. Schneuing A, Yuanqi D, Harris C. *et al*. Structure-based drug design with equivariant diffusion models. arXiv preprint, arXiv:2210.13695, 2022. https://arxiv.org/abs/2210.13695.
12. Guan J, Qian WW, Peng X. *et al*. 3D equivariant diffusion for target-aware molecule generation and affinity prediction. In: *International Conference on Learning Representations*. San Diego, CA: OpenReview.net, 2023.
13. Huang L, Tingyang X, Yang Y. *et al*. A dual diffusion model enables 3D molecule generation and lead optimization based on target pockets. *Nat Commun* 2024;**15**:2657. https://doi.org/10.1038/s41467-024-46569-1.
14. Cohen-Karlik E, Rozenberg E, Freedman D. Overcoming order in autoregressive graph generation. *Transact Mach Learn Res*, 2024;**2024**:1–16.
15. Igashov I, Stärk H, Vignac C. *et al*. Equivariant 3D-conditional diffusion model for molecular linker design. *Nat Mach Intell* 2024;**6**:417–27. https://doi.org/10.1038/s42256-024-00815-9.
16. Kong L, Cui J, Sun H. *et al*. Autoregressive diffusion model for graph generation. In:*International Conference on Machine Learning*, pp. 17391–408. New York, NY: ACM, 2023.
17. Irwin R, Tibo A, Janet J-P. *et al*. Efficient 3D molecular generation with flow matching and scale optimal transport. arXiv preprint, arXiv:2406.07266v1, 2024. https://arxiv.org/abs/2406.07266v1.
18. Chen X, Xu H, Hu J. *et al*. Order matters: probabilistic modeling of node sequence for graph generation. In: Meila M, Zhang T. (eds.), *International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research*. pp. 1630–9. New York, NY: ACM, 2021.
19. Kelvinius FE, Lindsten F. Autoregressive diffusion models with non-uniform generation order. *International Conference on Machine Learning*. New York, NY: ACM, 2023.
20. Li Y, Vinyals O, Dyer C. *et al*. Learning deep generative models of graphs. arXiv preprint, arXiv:1803.03324, 2018. https://arxiv.org/abs/1803.03324.
21. Goyal N, Jain HV, Ranu S. GraphGen: a scalable approach to domain-agnostic labeled graph generation. In: *Proceedings of the Web Conference 2020*, pp. 1253–63. New York, NY: ACM, 2020.
22. Winter R, Noé F, Clevert D-A. Permutation-invariant variational autoencoder for graph-level representation learning. *Adv Neural Inf Process Syst* 2021;**34**:9559–73.
23. Isert C, Atz K, Schneider G. Structure-based drug design with geometric deep learning. *Curr Opin Struct Biol* 2023;**79**:102548. https://doi.org/10.1016/j.sbi.2023.102548.
24. Li Y, Pei J, Lai L. Structure-based de novo drug design using 3D deep generative models. *Chem Sci* 2021;**12**:13664–75. https://doi.org/10.1039/D1SC04444C.
25. Peng X, Luo S, Guan J. *et al*. Pocket2Mol: efficient molecular sampling based on 3D protein pockets. In:*International Conference on Machine Learning*, pp. 17644–55. New York, NY: ACM, 2022.
26. Zhang O, Wang T, Weng G. *et al*. Learning on topological surface and geometric structure for 3D molecular generation. *Nat Comput Sci* 2023;**3**:849–59. https://doi.org/10.1038/s43588-023-00530-2.
27. Han X, Chen X, Ruiz FJR. *et al*. Fitting autoregressive graph generative models through maximum likelihood estimation. *J Mach Learn Res* 2023;**24**:1–30.
28. Zhung W, Kim H, Kim WY. 3D molecular generative framework for interaction-guided drug design. *Nat Commun* 2024;**15**:2688. https://doi.org/10.1038/s41467-024-47011-2.
29. Jiang Y, Zhang G, You J. *et al*. Pocketflow is a data-and-knowledge-driven structure-based molecular generative model. *Nat Mach Intell* 2024;**6**:326–37. https://doi.org/10.1038/s42256-024-00808-8.
30. Jiang D, Hsieh C-Y, Zhenxing W. *et al*. InteractionGraphNet: a novel and efficient deep graph representation learning framework for accurate protein–ligand interaction predictions. *J Med Chem* 2021;**64**:18209–32. https://doi.org/10.1021/acs.jmedchem.1c01830.

31. Li B, Ran T, Chen H. 3D based generative protac linker design with reinforcement learning. *Brief Bioinform* 2023;**24**:bbad323. https://doi.org/10.1093/bib/bbad323.
32. Feng W, Wang L, Lin Z. *et al.* Generation of 3D molecules in pockets via a language model. *Nat Mach Intell* 2024;**6**:62–73. https://doi.org/10.1038/s42256-023-00775-6.
33. Jing B, Corso G, Chang J. *et al.* Torsional diffusion for molecular conformer generation. *Adv Neural Inf Process Syst* 2022;**35**: 24240–53.
34. Zhang Z, Liu Q. Learning subpocket prototypes for generalizable structure-based drug design. In: *International Conference on Machine Learning*, pp. 41382–98. New York, NY: ACM, 2023.
35. Long S, Zhou Y, Dai X. *et al.* Zero-shot 3D drug design by sketching and generating. *Adv Neural Inf Process Syst* 2022;**35**:23894–907.
36. Yuejiang Y, Shuqi L, Gao Z. *et al.* Do deep learning models really outperform traditional approaches in molecular docking? arXiv preprint, arXiv:2302.07134, 2023. https://arxiv.org/abs/2302.07134.
37. Zhang X, Zhang O, Shen C. *et al.* Efficient and accurate large library ligand docking with KarmaDock. *Nat Comput Sci* 2023;**3**: 789–804. https://doi.org/10.1038/s43588-023-00511-5.
38. Zhu J, Zhonghui G, Pei J. *et al.* DiffBindFR: an SE (3) equivariant network for flexible protein–ligand docking. *Chem Sci* 2024;**15**: 7926–42. https://doi.org/10.1039/D3SC06803J.
39. Morris GM, Huey R, Lindstrom W. *et al.* AutoDock4 and AutoDockTools4: automated docking with selective receptor flexibility. *J Comput Chem* 2009;**30**:2785–91. https://doi.org/10.1002/jcc.21256.
40. Eberhardt J, Santos-Martins D, Tillack AF. *et al.* AutoDock Vina 1.2. 0: new docking methods, expanded force field, and Python bindings. *J Chem Inf Model* 2021;**61**:3891–8. https://doi.org/10.1021/acs.jcim.1c00203.
41. Alhossary A, Handoko SD, Yuguang M. *et al.* Fast, accurate, and reliable molecular docking with QuickVina 2. *Bioinformatics* 2015;**31**:2214–6. https://doi.org/10.1093/bioinformatics/btv082.
42. Friesner RA, Banks JL, Murphy RB. *et al.* Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *J Med Chem* 2004;**47**:1739–49. https://doi.org/10.1021/jm0306430.
43. Jones G, Willett P, Glen RC. *et al.* Development and validation of a genetic algorithm for flexible docking. *J Mol Biol* 1997;**267**:727–48. https://doi.org/10.1006/jmbi.1996.0897.
44. Corso G, Stärk H, Jing B. *et al.* DiffDock: diffusion steps, twists, and turns for molecular docking. In:*International Conference on Learning Representations*. San Diego, CA: OpenReview.net, 2023.
45. Krasoulis A, Antonopoulos N, Pitsikalis V. *et al.* DENVIS: scalable and high-throughput virtual screening using graph neural networks with atomic and surface protein pocket features. *J Chem Inf Model* 2022;**62**:4642–59. https://doi.org/10.1021/acs.jcim.2c01057.
46. Zhou G, Gao Z, Wei Z. *et al.* Do deep learning methods really perform better in molecular conformation generation? arXiv preprint, arXiv:2302.07061, 2023. https://arxiv.org/abs/2302.07061.
47. Wang Z, Zheng L, Wang S. *et al.* A fully differentiable ligand pose optimization framework guided by deep learning and a traditional scoring function. *Brief Bioinform* 2023;**24**:bbac520. https://doi.org/10.1093/bib/bbac520.
48. Bryant P, Kelkar A, Guljas A. *et al.* Structure prediction of protein-ligand complexes from sequence information with Umol. *Nat Commun* 2024;**15**:4536. https://doi.org/10.1038/s41467-024-48837-6.
49. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. *J R Stat Soc B Methodol* 1977;**39**:1–22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.
50. Powers AS, Yu HH, Suriana P. *et al. Geometric Deep Learning for Structure-Based Ligand Design*, Vol. **9**. Washington, DC: ACS Central Science, 2023. 2257–67. https://doi.org/10.1021/acscentsci.3c00572.
51. Danel T, Łeski J, Podlewska S. *et al.* Docking-based generative approaches in the search for new drug candidates. *Drug Discov Today* 2023;**28**:103439. https://doi.org/10.1016/j.drudis.2022.103439.
52. Norris JR. *Markov Chains. Cambridge Series in Statistical and Probabilistic Mathematics*. Shaftesbury Road, Cambridge, CB2 8EA: Cambridge University Press, 1998.
53. Borman S. The expectation maximization algorithm-a short tutorial, vol. 41, 2004, [online] Available: http://www.seanborman.com/publications/EM_algorithm.pdf.
54. Meng Q, Bengio Y, Tang J. GMNN: graph Markov neural networks. In *International Conference on Machine Learning*. New York, NY: ACM, 2019. 5241–50.
55. Sun Z, Yang Y. An EM approach to non-autoregressive conditional sequence generation. In:*International Conference on Machine Learning*, pp. 9249–58. New York, NY: ACM, 2020.
56. Zhang Y, Sharma K, Liu Y. VigDet: knowledge informed neural temporal point process for coordination detection on social media. *Adv Neural Inf Process Syst* 2021;**34**:3218–31.
57. Min E, Chen R, Bian Y. *et al.* Transformer for graphs: an overview from architecture perspective. arXiv preprint, arXiv:2202.08455, 2022. https://arxiv.org/abs/2202.08455.
58. Zhang Z, Min Y, Zheng S. *et al.* Molecule generation for target protein binding with structural motifs. In: *International Conference on Learning Representations*. San Diego, CA: OpenReview.net, 2023.
59. Francoeur PG, Masuda T, Sunseri J. *et al.* Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *J Chem Inf Model* 2020;**60**:4200–15. https://doi.org/10.1021/acs.jcim.0c00411.
60. Ghose AK, Viswanadhan VN, Wendoloski JJ. A knowledge-based approach in designing combinatorial or medicinal chemistry libraries for drug discovery. 1. A qualitative and quantitative characterization of known drug databases. *J Comb Chem* 1999;**1**: 55–68. https://doi.org/10.1021/cc9800071.
61. Lipinski CA, Lombardo F, Dominy BW. *et al.* Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv Drug Deliv Rev* 2012;**64**:4–17. https://doi.org/10.1016/j.addr.2012.09.019.
62. Veber DF, Johnson SR, Cheng H-Y. *et al.* Molecular properties that influence the oral bioavailability of drug candidates. *J Med Chem* 2002;**45**:2615–23. https://doi.org/10.1021/jm020017n.
63. Bajusz D, Rácz A, Héberger K. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *J Chem* 2015;**7**:1–13. https://doi.org/10.1186/s13321-015-0069-3.
64. Tanimoto TT. *Elementary Mathematical Theory of Classification and Prediction*. IBM Internal Report, International Business Machines Corporation, 1958.
65. Li Y, Zhang L, Wang Y. *et al.* Generative deep learning enables the discovery of a potent and selective RIPK1 inhibitor. *Nat Commun* 2022;**13**:6891. https://doi.org/10.1038/s41467-022-34692-w.
66. Siebenmorgen T, Menezes F, Benassou S. *et al.* MISATO: machine learning dataset of protein–ligand complexes for structure-based drug discovery. *Nat Comput Sci* 2024;**4**:367–78.

© The Author(s) 2025. Published by Oxford University Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (https://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact journals.permissions@oup.com
Briefings in Bioinformatics, 2025, **26(2)**, bbaf094
https://doi.org/10.1093/bib/bbaf094
Problem Solving Protocol