
Unified Guidance for Geometry-Conditioned Molecular Generation

Sirine Ayadi*^{1,2}

Leon Hetzel*^{1,2,3}

Johanna Sommer*^{1,2}

Fabian Theis^{1,2,3}

Stephan Günnemann^{1,2}

¹ School of Computation, Information and Technology, Technical University of Munich

² Munich Data Science Institute, Technical University of Munich

³ Center for Computation Health, Helmholtz Munich

{si.ayadi, l.hetzel, jm.sommer, f.theis, s.guennemann}@tum.de

Abstract

Effectively designing molecular geometries is essential to advancing pharmaceutical innovations, a domain, which has experienced great attention through the success of generative models and, in particular, diffusion models. However, current molecular diffusion models are tailored towards a specific downstream task and lack adaptability. We introduce UniGuide, a framework for controlled geometric guidance of unconditional diffusion models that allows flexible conditioning during inference without the requirement of extra training or networks. We show how applications such as structure-based, fragment-based, and ligand-based drug design are formulated in the UniGuide framework and demonstrate on-par or superior performance compared to specialised models. Offering a more versatile approach, UniGuide has the potential to streamline the development of molecular generative models, allowing them to be readily used in diverse application scenarios.

1 Introduction

Diffusion models have emerged as an important class of generative models in various domains, including computer vision [1], signal processing [2], computational chemistry, and drug discovery [3–8]. By gradually adding noise to data samples and learning the reverse process of removing noise, diffusion models effectively transform noisy samples into structured data [9, 10]. In the context of drug discovery, it is essential to effectively address downstream tasks, which often pose specific geometric conditions. Examples of this include (i) Structure-based drug design (SBDD) that aims to create small ligands that fit given receptor binding sites [11], (ii) Fragment-based drug design (FBDD) that designs molecules by elaborating known scaffolds [12, 13], or (iii) Ligand-based drug design (LBDD) which generates molecules that fit a certain shape [14]. Recent works address these tasks by either incorporating specialised models or focusing on conditions that directly resemble molecular structures. In both cases, this narrow focus restricts their adaptability to new or slightly altered settings.

We address the challenge of adaptability by introducing UniGuide, a method that unifies guidance for geometry-conditioned molecular generation, see Fig. 1. The key element for achieving this unification is the *condition map*, which transforms complex geometric conditions to match the diffusion model’s

*Equal contribution

Project Page: www.cs.cit.tum.de/daml/uniguide

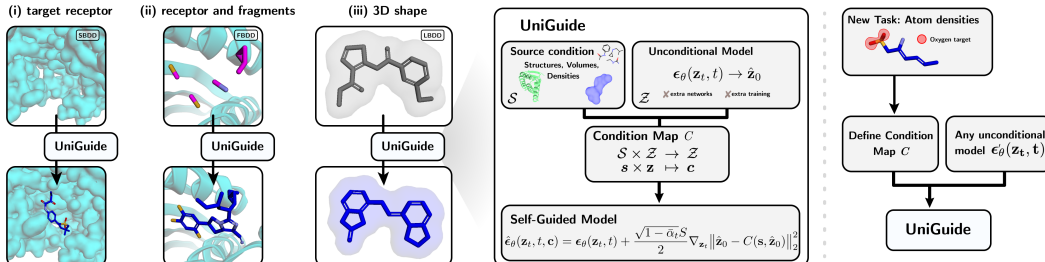


Figure 1: UniGuide handles diverse conditioning modalities for guidance, including: (i) a target receptor for SBDD, (ii) additional molecular fragments for FBDD, or (iii) a predefined 3D shape for LBDD. It combines a source condition $s \in \mathcal{S}$ and the unconditional model $\epsilon_{\theta}(\mathbf{z}_t, t)$ within its condition map to enable self-guidance. The flexible formulation of our approach can be generalised to new geometric tasks, for example, conditioning on atomic densities.

configuration space, thereby enabling self-guidance without the need for external models. Like other guidance-based approaches, UniGuide does not constrain the generality of the underlying model. Moreover, our method is the most versatile, extending beyond guiding molecular structures to leveraging complex geometric conditions such as volumes, surfaces, and densities, thereby enabling the unified tackling of diverse drug discovery tasks. For complex conditions specifically, previous works primarily rely on conditional diffusion models for effective condition encoding [12–14]. With our method, we are able to tackle the same tasks, while overcoming major drawbacks: UniGuide eliminates the need for additional training and, more importantly, avoids constraining the model to specific tasks.

We demonstrate the wide applicability of UniGuide by tackling a variety of geometry-constrained drug discovery tasks. With performance either on par with or superior to tailored models, we conclude that UniGuide offers advantages beyond its unification. Firstly, while the novelty of conditional models often stems from the condition incorporation, our method redirects focus to advancing unconditional generation, which directly benefits multiple applications. Furthermore, this separation of model training and conditioning allows us to tackle tasks with minimal data, a common scenario in the biological domain.

In summary, our contributions are as follows:

- We present UniGuide: A unified guidance method for generating geometry-conditioned molecular structures, requiring neither additional training nor external networks used to guide the generation.
- We demonstrate UniGuide’s wide applicability by tackling various conditioning scenarios in structure-based, fragment-based, and ligand-based drug design.
- We show UniGuide’s favourable performance over task-specific baselines, highlighting the practical relevance of our approach.

2 Related work

Diffusion models and controllable generation Diffusion models [9, 10] are generative models achieving state-of-the-art performance across various domains, including the generation of images [1, 9], text [15], or point clouds [16]. Conditional diffusion models [17–21] are based on the same principle but incorporate a particular condition in their training, allowing for the controlled generation. Alternatively, classifier guidance [22, 23] relies on external models for controllable generation. Prior works in this context primarily focused on global properties [22, 24], lacking the capacity to condition on the geometric conditions central to our work. For instance, Bao et al. [24] demonstrate control over molecule generation based on desired quantum properties.

De novo molecule generation Research on *de novo* molecule generation focused extensively on generating molecules using their chemical graph representations [7, 25–34]. However, these methods are limited in modelling the molecules’ conformation information and are, therefore, not ideally suited for several drug-discovery settings, such as target-aware drug design. Recently, attention has shifted towards generating molecules in 3D space, utilising variational autoencoders [35], autoregressive models [36–38], flow-based models [39, 40], and diffusion-based approaches [20, 41–47].

Conditional generation of molecules Downstream applications of molecular generation can be categorised by their condition modality. In the case of SBDD [38, 48, 49], Schneuing et al. [11] and Guan et al. [50], for example, introduce models that simultaneously operate on protein pockets and ligands. In the conditional case, the pocket context is fixed throughout the generation. Moreover, FBDD imposes (multiple) scaffolds as a constraint [11, 12, 51–53]. Igashov et al. [13] expand given scaffolds by generating the molecule around the fixed scaffolds. In a related task of FBDD, linker design with pose estimation, as discussed in [54], further generate the rotation of the given scaffolds. SBDD and FBDD rely on the availability of high-quality data of protein pockets, which is often scarce. For this reason, LBDD aims to generate molecules that match the same 3D volume of reference ligands that are known to bind to the target of interest [55, 56]. Chen et al. [14] specifically train a shape encoder to capture the molecular shape of a reference ligand and use the resulting embedding to train a conditional diffusion model.

3 Controlling the generation of diffusion models

Diffusion Models [9, 57] learn a Markov Chain that involves a forward process to perturb data from a distribution $q(\mathbf{z})$ and learn to reverse the process to generate new samples from a tractable prior, for example, a normal distribution. Given a data point sampled from the true underlying distribution, $\mathbf{z}_{\text{data}} \sim q(\mathbf{z})$, the forward process $q(\mathbf{z}_t | \mathbf{z}_{t-1})$ gradually adds Gaussian noise:

$$q(\mathbf{z}_t | \mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t | \sqrt{1 - \beta_t} \mathbf{z}_{t-1}, \beta_t \mathbf{I}), \quad (1)$$

where $\{\beta_t \in (0, 1)\}_{t=1}^T$ defines a variance schedule. Defining the forward process this way, one can readily sample from $q(\mathbf{z}_t | \mathbf{z}_{\text{data}})$:

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t} \mathbf{z}_{\text{data}} + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (2)$$

with $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$. Since the time-reverse process $q(\mathbf{z}_{t-1} | \mathbf{z}_t)$ depends on \mathbf{z}_{data} , which is not available at generation time, it is approximated by modelling $p_\theta(\mathbf{z}_{t-1} | \mathbf{z}_t)$:

$$p_\theta(\mathbf{z}_{t-1} | \mathbf{z}_t) = \mathcal{N}(\mathbf{z}_{t-1} | \boldsymbol{\mu}_\theta(\mathbf{z}_t, t), \sigma_t \mathbf{I}), \quad (3)$$

where the mean $\boldsymbol{\mu}_\theta$ is parameterised by a noise-predicting neural network $\boldsymbol{\epsilon}_\theta$ in the form of:

$$\boldsymbol{\mu}_\theta(\mathbf{z}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{z}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t) \right). \quad (4)$$

The model $\boldsymbol{\epsilon}_\theta$ is trained to optimise the variational lower bound through the simplified training objective:

$$\mathcal{L}_{\text{train}} = \frac{1}{2} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)\|_2^2. \quad (5)$$

Self-guiding diffusion models Using Bayes’ rule, the conditional probability $p_\theta(\mathbf{z}_t | \mathbf{c})$ given a condition \mathbf{c} can be expressed as

$$p_\theta(\mathbf{z}_t | \mathbf{c}) \propto p_\theta(\mathbf{z}_t) p_\theta(\mathbf{c} | \mathbf{z}_t). \quad (6)$$

This allows us to decompose the score function as follows:

$$\nabla_{\mathbf{z}_t} \log p_\theta(\mathbf{z}_t | \mathbf{c}) = \nabla_{\mathbf{z}_t} \log p_\theta(\mathbf{z}_t) + S \nabla_{\mathbf{z}_t} \log p_\theta(\mathbf{c} | \mathbf{z}_t), \quad (7)$$

where the second term is used for guiding the unconditional generation, with $S > 0$ controlling the guidance strength. Using that $\nabla_{\mathbf{z}_t} \log p_\theta(\mathbf{z}_t) = -(1 - \bar{\alpha}_t)^{-\frac{1}{2}} \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)$ [22], we can rewrite the score function from Eq. (7) and identify the modified noise predictor $\hat{\boldsymbol{\epsilon}}_\theta$:

$$\nabla_{\mathbf{z}_t} \log p_\theta(\mathbf{z}_t | \mathbf{c}) = -\frac{1}{\sqrt{1 - \bar{\alpha}_t}} \left[\underbrace{\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t) - \sqrt{1 - \bar{\alpha}_t} S \nabla_{\mathbf{z}_t} \log p_\theta(\mathbf{c} | \mathbf{z}_t)}_{=: \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_t, t, \mathbf{c})} \right] \quad (8)$$

The modified mean function $\hat{\boldsymbol{\mu}}_\theta$ then follows from the modified version of Eq. (4), enabling us to sample from $p_\theta(\mathbf{z}_{t-1} | \mathbf{z}_t, \mathbf{c}) \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_\theta(\mathbf{z}_t, t, \mathbf{c}), \sigma_t \mathbf{I})$:

$$\hat{\boldsymbol{\mu}}_\theta(\mathbf{z}_t, t, \mathbf{c}) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{z}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_t, t, \mathbf{c}) \right) = \boldsymbol{\mu}_\theta(\mathbf{z}_t, t) + \lambda(t) \nabla_{\mathbf{z}_t} \log p_\theta(\mathbf{c} | \mathbf{z}_t), \quad (9)$$

where $\lambda(t) = (\alpha_t)^{-\frac{1}{2}}\beta_t S$ balances the conditional update. Eq. (9) requires sampling from $\log p_\theta(\mathbf{c} | \mathbf{z}_t)$ to which we do not have access. Assuming the condition \mathbf{c} lies in the same space as \mathbf{z}_t , we can follow Kollovich et al. [58] and approximate $\log p_\theta(\mathbf{c} | \mathbf{z}_t)$ as a multivariate Gaussian distribution:

$$p_\theta(\mathbf{c} | \mathbf{z}_t) = \mathcal{N}(\mathbf{c} | \mathbf{f}_\theta(\mathbf{z}_t, t), \mathbf{I}) , \quad (10)$$

where $\mathbf{f}_\theta(\mathbf{z}_t, t)$ approximates the clean data point, enabling to estimate the condition in data space. Using Eq. (2), we can readily predict the clean data point given the noisy sample \mathbf{z}_t via

$$\mathbf{f}_\theta(\mathbf{z}_t, t) = \frac{\mathbf{z}_t - \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)}{\sqrt{\bar{\alpha}_t}} =: \hat{\mathbf{z}}_0 . \quad (11)$$

With this, the *guiding term* becomes a direct differentiation of the squared error with respect to the noisy sample \mathbf{z}_t :

$$\nabla_{\mathbf{z}_t} \log p_\theta(\mathbf{c} | \mathbf{z}_t) = -\frac{1}{2} \nabla_{\mathbf{z}_t} \|\mathbf{f}_\theta(\mathbf{z}_t, t) - \mathbf{c}\|_2^2 . \quad (12)$$

By directly leveraging the prediction of the unconditional model $\boldsymbol{\epsilon}_\theta$, Eq. (12) establishes our self-guiding conditioning, thereby defining the self-guided noise predictor $\hat{\boldsymbol{\epsilon}}_\theta$:

$$\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_t, t, \mathbf{c}) = \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t) + \frac{\sqrt{1 - \bar{\alpha}_t} S}{2} \nabla_{\mathbf{z}_t} \|\hat{\mathbf{z}}_0 - \mathbf{c}\|_2^2 . \quad (13)$$

4 UniGuide

To enable the application of unconditional molecular diffusion models $\boldsymbol{\epsilon}_\theta$ to geometric downstream tasks in drug discovery, we aim to develop a unified guidance framework, UniGuide, see Fig. 1. Importantly, we seek to enable guidance from arbitrary geometric conditions $\mathbf{s} \in \mathcal{S}$, where \mathcal{S} denotes a general space of source conditions. However, the source conditions \mathbf{s} cannot be directly used for the loss computation in Eq. (12) when they do not match the configuration space \mathcal{Z} .

To address this challenge, we introduce *condition maps* C , which bridge the gap between arbitrary source conditions \mathbf{s} and target conditions \mathbf{c} suitable for guidance. In Sec. 4.1, we start with its general formulation and continue to derive a condition map $C_{\mathcal{Z}}$ for the special case where $\mathcal{S} = \mathcal{Z}$. This will be useful when discussing the application of UniGuide to various drug discovery tasks in Sec. 4.2. We also demonstrate how to derive a task-specific condition map $C_{\partial V}$ for ligand-based drug design.

Notation In 3D space, the *configuration* of molecules, including proteins, can be represented by a set of tuples $\mathbf{z} = \{(\mathbf{x}_i, \mathbf{h}_i)\}_{i=1}^N \in \mathcal{Z}$, where $\mathbf{x}_i \in \mathbb{R}^3$ and $\mathbf{h}_i \in \mathbb{R}^d$ refer to coordinates and features of a node $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{h}_i)$, respectively. The space of configurations is denoted by \mathcal{Z} and includes configurations of varying size N . We distinguish between different configuration entities via superscripts, i.e. refer to molecules \mathcal{M} and proteins \mathcal{P} through $\mathbf{z}^{\mathcal{M}}$ and $\mathbf{z}^{\mathcal{P}}$, respectively. The collection of coordinates $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \in \mathbb{R}^{N \times 3} \in \mathcal{X}$ defines the *conformation* of a molecule \mathcal{M} or protein \mathcal{P} . We represent arbitrary geometric conditions with the variable $\mathbf{s} \in \mathcal{S}$, and conditions that can be used for guidance with the variables $\mathbf{c} \in \mathcal{Z}$.

4.1 Unified self-guidance from geometric conditions $\mathbf{s} \in \mathcal{S}$

The concept of a condition map C is essential to our method, enabling guidance from conditions $\mathbf{s} \in \mathcal{S}$ in a unified fashion, where \mathcal{S} represents a space of general geometric objects such as structures, densities, or surfaces. These geometric objects do not necessarily match the configuration space \mathcal{Z} , i.e. $\mathcal{S} \neq \mathcal{Z}$, preventing the computation of the guiding score function from Eq. (12). We overcome this challenge by defining C as a transformation that maps \mathbf{s} to a suitable target condition $\mathbf{c} \in \mathcal{Z}$, which is then utilised for self-guidance.

In the most general case, C takes the form of

$$\begin{aligned} C : \mathcal{S} \times \mathcal{Z} &\rightarrow \mathcal{Z} \\ \mathbf{s} \times \mathbf{z} &\mapsto \mathbf{c} , \end{aligned} \quad (14)$$

where the source condition \mathbf{s} together with a configuration \mathbf{z} are mapped to a target condition $\mathbf{c} \in \mathcal{Z}$. Including the condition map C in the guidance, we obtain our guidance signal:

$$\nabla_{\mathbf{z}_t} \log p_{\theta}(\mathbf{c} | \mathbf{z}_t) = -\frac{1}{2} \nabla_{\mathbf{z}_t} \|\hat{\mathbf{z}}_0 - C(\mathbf{s}, \hat{\mathbf{z}}_0)\|_2^2 = -\nabla_{\mathbf{z}_t} \mathcal{L}(\hat{\mathbf{z}}_0, \mathbf{s}) \quad , \quad (15)$$

where $\hat{\mathbf{z}}_0 = \mathbf{f}_{\theta}(\mathbf{z}_t, t)$ is the estimate of \mathbf{z}_0 given the unconditional model $\epsilon_{\theta}(\mathbf{z}_t, t)$ obtained according to Eq. (11) and $\mathbf{c} = C(\mathbf{s}, \hat{\mathbf{z}}_0)$ is the target condition produced by the condition map. In this formulation, \mathbf{c} can also be understood as *guidance target* of the unconditional model.

It is important to highlight that Eq. (15) should not destroy the underlying properties of the unconditional generative process. In particular, if the unconditional model ϵ_{θ} is equivariant to a set of transformations G , e.g. rotations and translations, as is common in the molecular domain, we want to retain equivariance also in the guidance signal. Hence, the self-guided model $\hat{\epsilon}_{\theta}$ should satisfy

$$\hat{\epsilon}_{\theta}(G(\mathbf{z}_t), t, \mathbf{c}) = G(\hat{\epsilon}_{\theta}(\mathbf{z}_t, t, \mathbf{c})) \quad , \quad (16)$$

for all transformations G to which ϵ_{θ} is equivariant.

Theorem 4.1. *Consider a function $C : \mathcal{S} \times \mathcal{Z} \rightarrow \mathcal{Z}$. If $C(\mathbf{s}, \mathbf{z})$ is invariant to rigid transformations G in the first argument and equivariant in the second argument, then the gradient $\nabla_{\mathbf{z}} \|\mathbf{v}\|_2^2$ of the vector $\mathbf{v} = \mathbf{z} - C(\mathbf{s}, \mathbf{z})$ is equivariant to transformations of \mathbf{z} .*

Proof. We prove Theorem 4.1 in App. B. □

Using Theorem 4.1, we can guarantee equivariant guidance signals if the condition maps $C(\mathbf{s}, \mathbf{z})$ are invariant and equivariant under rigid transformations concerning the source condition \mathbf{s} and configuration \mathbf{z} , respectively.

Guidance in the special case of $\mathcal{S} = \mathcal{Z}$ In the case where the source condition \mathbf{s} directly defines subset \mathcal{A} of $m < N$ nodes of the configuration, i.e. $\mathcal{S} = \mathcal{Z}$, we can fully specify the condition map. This is feasible because the condition map no longer needs to bridge different spaces; it only needs to ensure equivariance, as the loss computation between \mathbf{s} and the configuration is already possible. To distinguish this special case from the general setting, we denote $\mathbf{s} = \tilde{\mathbf{z}} \in \mathbb{R}^{m \times (3+d)}$ and refer to the defined subset within the configuration $\hat{\mathbf{z}}_0$ by $\hat{\mathbf{z}}_0^{\mathcal{A}}$.

In order to satisfy the requirements on $C(\tilde{\mathbf{z}}, \hat{\mathbf{z}}_0^{\mathcal{A}})$ as stated by Theorem 4.1, we align $\tilde{\mathbf{z}}$ with $\hat{\mathbf{z}}_0^{\mathcal{A}}$ by using the Kabsch algorithm [59, 60]. Denoting the resulting transformation with $T_{\hat{\mathbf{z}}_0^{\mathcal{A}}}$, we get an $\hat{\mathbf{z}}_0$ -equivariant condition map:

$$C_{\mathcal{Z}}: \mathbb{R}^{m \times (3+d)} \times \mathbb{R}^{m \times (3+d)} \rightarrow \mathbb{R}^{m \times (3+d)} \\ \tilde{\mathbf{z}} \times \hat{\mathbf{z}}_0^{\mathcal{A}} \quad \mapsto \quad T_{\hat{\mathbf{z}}_0^{\mathcal{A}}} \tilde{\mathbf{z}} \quad . \quad (17)$$

Taken together, we can compute the guidance signal based on the following loss \mathcal{L} :

$$\mathcal{L}(\hat{\mathbf{z}}_0^{\mathcal{A}}, \tilde{\mathbf{z}}) = \frac{1}{2} \|\hat{\mathbf{z}}_0^{\mathcal{A}} - T_{\hat{\mathbf{z}}_0^{\mathcal{A}}} \tilde{\mathbf{z}}\|_2^2 \quad . \quad (18)$$

We emphasise that although the loss $\mathcal{L}(\hat{\mathbf{z}}_0^{\mathcal{A}}, \tilde{\mathbf{z}})$ is computed on the subset \mathcal{A} , the gradient, as presented in Eq. (15), is still computed with respect the full configuration \mathbf{z}_t .

In summary, our method requires only an unconditionally trained model ϵ_{θ} and a suitable condition map C , eliminating the need for additional networks or training. Together, this facilitates unified self-guidance from arbitrary geometric sources. Importantly, the separation of model training and conditioning enables us to tackle tasks even with minimal data, which is crucial in practical scenarios. In the following section, we discuss the wide applicability of UniGuide by illustrating its application to multiple drug discovery tasks.

4.2 UniGuide for drug discovery

Having introduced both the guidance framework and the condition map, we will continue to discuss how to tackle a set of drug discovery tasks within the UniGuide framework. We start with its application to ligand-based drug design (LBDD), which aims to generate a ligand that satisfies a predefined molecular shape.

Ligand-based drug design LBDD aims to generate novel ligands with a similar 3D shape as a reference ligand \mathcal{M}_{ref} . In this setting, one operates on the molecule level only since the protein information is assumed to be unknown. However, to still generate active ligands that bind to a protein pocket, one leverages the 3D shape information of a reference molecule. Specifically, the goal is to modify the generative process $\hat{\epsilon}_\theta$ to generate a ligand \mathbf{z}_0 with a similar 3D shape but different molecular structure than \mathcal{M}_{ref} . With Sec. 4.1 introducing all required concepts, we can readily formulate a *surface condition map* $C_{\partial V}$ suitable to tackle the task of LBDD, see Fig. 2:

To represent \mathcal{M}_{ref} 's 3D shape, we identify our source condition \mathbf{s} with a set of K points \mathbf{y} sampled uniformly from the reference ligand's surface ∂V , $\mathbf{y} \in \mathbb{R}^{K \times 3} = \mathcal{S}$. As no features are guided, we formulate $C_{\partial V}$ with respect to the conformation space $\mathcal{X} = \mathbb{R}^{N \times 3}$:

$$C_{\partial V} : \mathbb{R}^{K \times 3} \times \mathbb{R}^{N \times 3} \rightarrow \mathbb{R}^{N \times 3} \quad (19)$$

$$\mathbf{y} \times \hat{\mathbf{x}}_0 \mapsto \mathbf{c}_\mathbf{x} ,$$

where $\hat{\mathbf{x}}_0$ denotes the conformation of the clean data point estimation $\hat{\mathbf{z}}_0$ as computed by Eq. (11). To satisfy Theorem 4.1, $C_{\partial V}$ first aligns \mathbf{y} with $\hat{\mathbf{x}}_0$ by a rotation $R_{\hat{\mathbf{x}}_0} \in \mathbb{R}^{3 \times 3}$ resulting from the ICP algorithm [61]. For every atom coordinate $\hat{\mathbf{x}}_i$, $C_{\partial V}$ subsequently computes the mean $\bar{\mathbf{y}}_i$ over $\hat{\mathbf{x}}_i$'s k closest surface points:

$$\bar{\mathbf{y}}_i = \frac{1}{k} \sum_{j \in \mathcal{N}_{\hat{\mathbf{x}}_i}} R_{\hat{\mathbf{x}}_0} \mathbf{y}_j \quad , \quad \text{with} \quad \mathcal{N}_{\hat{\mathbf{x}}_i} = \arg \min_{I \subset \{1, \dots, K\}, |I|=k} \sum_{j \in I} \|R_{\hat{\mathbf{x}}_0} \mathbf{y}_j - \hat{\mathbf{x}}_i\|_2 \quad . \quad (20)$$

Finally, the individual components $\mathbf{c}_{\mathbf{x},i}$ of the target condition compute as follows:

$$\mathbf{c}_{\mathbf{x},i} = \begin{cases} \bar{\mathbf{y}}_i + \frac{\alpha}{d}(\bar{\mathbf{y}}_i - \hat{\mathbf{x}}_i) , & \text{if } \hat{\mathbf{x}}_i \text{ outside } V \\ \bar{\mathbf{y}}_i - \frac{\alpha}{d}(\bar{\mathbf{y}}_i - \hat{\mathbf{x}}_i) , & \text{if } \hat{\mathbf{x}}_i \text{ inside } V \wedge d < \alpha \\ \hat{\mathbf{x}}_i , & \text{otherwise} , \end{cases} \quad (21)$$

where d denotes the distance to the surface, $d = \|\bar{\mathbf{y}}_i - \hat{\mathbf{x}}_i\|_2$, and α the required distance to the surface. Note that the target condition $\mathbf{c}_\mathbf{x}$ represents a valid conformation inside the surface ∂V , and that $C_{\partial V}$ effectively bridges spaces from \mathcal{S} to \mathcal{X} . Consequently, when using $C_{\partial V}$, the guidance signal is derived from Eq. (15) with the loss function $\mathcal{L}(\hat{\mathbf{z}}_0, \mathbf{y})$. The full algorithm for guidance using $C_{\partial V}$ is presented in App. D.1.

Structure-based drug design The goal of SBDD is to design a ligand that binds to a target protein pocket \mathbf{s} . In this setting, one operates on both the molecule and protein level. Technically, we are interested in generating a ligand \mathbf{z}_0^M conditioned on the protein configuration $\tilde{\mathbf{z}}^P$. With the unconditional diffusion model $\epsilon_\theta(\mathbf{z}_t, t)$, $\mathbf{z}_t = (\mathbf{z}_t^M, \mathbf{z}_t^P)$, approximating the joint distribution of ligand-protein pairs $p(\mathbf{z}_{\text{data}}^M, \mathbf{z}_{\text{data}}^P)$, one can readily see that the source condition directly corresponds to the configuration of the protein pocket. Hence, we can use $C_{\tilde{\mathbf{z}}}$ from Sec. 4.1 and identify $\tilde{\mathbf{z}}$ with $\tilde{\mathbf{z}}^P$. The guidance signal then follows from the loss $\mathcal{L}(\hat{\mathbf{z}}_0^P, \tilde{\mathbf{z}}^P)$ with $\mathbf{c}^P = C_{\tilde{\mathbf{z}}}(\tilde{\mathbf{z}}^P, \hat{\mathbf{z}}_0^P)$ as defined in Eq. (18). We describe the sampling algorithm for the SBDD task in App. E.1.

Fragment-based drug design FBDD aims to design a ligand by optimising a molecule around fragments \mathcal{F} that bind weakly to a receptor. Similarly to SBDD, one operates on both the molecule and protein level. Technically, we are interested in generating a ligand \mathbf{z}_0^M conditioned on both the protein and the fragment configuration, $\tilde{\mathbf{z}}^P$ and $\tilde{\mathbf{z}}^F$, respectively. Considering the same kind of unconditional model $\epsilon_\theta(\mathbf{z}_t, t)$ as in SBDD, we can use $C_{\tilde{\mathbf{z}}}$ from Sec. 4.1. Only now, we identify $\tilde{\mathbf{z}}$ with both $\tilde{\mathbf{z}}^P$ and $\tilde{\mathbf{z}}^F$ and write $\tilde{\mathbf{z}}^A$ with $\mathcal{A} = \mathcal{P} \cup \mathcal{F}$. Using Eq. (18), the guidance signal directly follows from $\mathcal{L}(\hat{\mathbf{z}}_0^A, \tilde{\mathbf{z}}^{P \cup F})$ with $\mathbf{c}^P = C_{\tilde{\mathbf{z}}}(\tilde{\mathbf{z}}^{P \cup F}, \hat{\mathbf{z}}_0^{P \cup F})$. The sampling algorithm is similar to the one described in App. E.1.

Several tasks exist within the FBDD setting [62–65]. Examples are scaffold hopping [64], where the core structure of \mathbf{z}_0^M has to be generated, but functional groups that interact with the receptor are

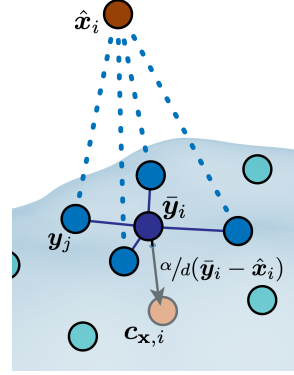


Figure 2: Surface condition map $C_{\partial V}$: For each atom coordinate \mathbf{x}_i , the closest surface points \mathbf{y}_j are computed. The target condition $\mathbf{c}_{\mathbf{x},i}$ is the projection along the mean of neighbours $\bar{\mathbf{y}}_i$ to the inside of the volume by a margin α , where $d = \|\bar{\mathbf{y}}_i - \hat{\mathbf{x}}_i\|_2$.

Table 1: Ligand-Based Drug Design. Results taken from Chen et al. [14] are indicated with (*). We highlight the best conditioning approach for the ShapeMol backbone in **bold** and underline the best approach across all methods.

	Method	only shape	Sim _S (↑)	max Sim _S (↑)	Sim _G (↓)	max Sim _G (↓)	Ratio (↑)	Diversity (↑)
Non-diffusion based	VS* [14]	✗	0.729 ± 0.04	0.807 ± 0.04	0.226 ± 0.04	0.241 ± 0.09	3.226	0.759 ± 0.02
	SQUID* [55] (λ = 0.3)	✗	0.717 ± 0.08	0.904 ± 0.07	0.349 ± 0.09	0.549 ± 0.24	2.054	0.687 ± 0.07
	SQUID* [55] (λ = 1.0)	✗	0.670 ± 0.07	0.842 ± 0.06	0.235 ± 0.05	0.271 ± 0.09	2.851	0.744 ± 0.05
Diffusion-based	ShapeMol [14]	✓	0.677 ± 0.04	0.797 ± 0.04	0.239 ± 0.05	0.240 ± 0.07	2.834	0.714 ± 0.05
	ShapeMol+g [14]	✗	0.744 ± 0.03	0.849 ± 0.03	0.242 ± 0.04	0.245 ± 0.05	3.074	0.708 ± 0.05
	UniGuide (ShapeMol [U])	✓	0.726 ± 0.04	0.827 ± 0.05	0.248 ± 0.05	0.239 ± 0.05	2.927	0.651 ± 0.05
	UniGuide (ShapeMol)	✓	0.760 ± 0.05	0.857 ± 0.06	0.240 ± 0.04	0.237 ± 0.06	3.167	0.705 ± 0.04
	UniGuide (EDM)	✓	0.749 ± 0.04	0.860 ± 0.04	<u>0.212 ± 0.04</u>	<u>0.206 ± 0.06</u>	3.536	0.736 ± 0.04

fixed, or linker design [65], where the connection between separated fragments has to be optimised through the generative process, see Fig. 5. Note that these tasks differ primarily in their application and can be treated identically from a technical perspective within UniGuide. In addition, one can also consider variations where the protein information \tilde{z}^P is discarded. This usually aligns with switching to an unconditional model ϵ_θ that solely models the distribution over molecules. We present results for this configuration in Sec. 5.3.

Furthermore, we would like to highlight that it is possible to combine guidance strategies within UniGuide. For example, one could incorporate a version of the surface condition map $C_{\partial V}$ for FBDD to provide an additional geometric guidance signal for the atoms not included in \mathcal{F} .

Limitations Drug discovery also involves tasks beyond purely geometric conditions, encompassing global graph properties [24]. These are excluded from the UniGuide framework. Additionally, UniGuide requires the unconditional model to be trained on a matching configuration space. We discuss the broader impact of our work in App. A.

5 Results

In this section, we compare UniGuide to state-of-the-art models across various drug discovery tasks. To highlight the wide range of tasks to which unconditional models can be adapted through UniGuide, we conduct experiments on ligand-based (Sec. 5.1), structure-based (Sec. 5.2) and fragment-based (Sec. 5.3) drug design. We demonstrate that UniGuide performs competitively or even surpasses specialised baseline models, underscoring its practical relevance and transferability to diverse drug discovery scenarios.

5.1 Ligand-based drug design

Dataset Following Chen et al. [14], we employ the MOSES dataset for the ligand-based drug design task [66]. We evaluate on a test set consisting of 1000 reference ligands, from which the 3D shape conditions are extracted. For every shape condition \mathcal{M}_{ref} , 50 samples are generated. We refer to App. D.1 for further details on the evaluation setup.

Baselines For the LBDD task, we compare UniGuide to ShapeMol, a conditional diffusion model that is trained by conditioning on learned latent embeddings of the molecular surfaces [14]. Chen et al. [14] also propose a correction technique that adjusts the atom positions based on their distance to the reference ligand’s nodes, which is referred to as ShapeMol+g. Additionally, we include as baselines Virtual Screening (VS) [14], a shape-based virtual screening tool, and SQUID [55], a variational autoencoder that decodes molecules by sequentially attaching fragments with fixed bond lengths and angles. For this task, we evaluate UniGuide equipped with the surface condition map $C_{\partial V}$ from Eq. (21) in conjunction with two *unconditionally trained* diffusion models, ShapeMol [U] and EDM [14, 20] as well as the *conditional* model ShapeMol [14]. The “only shape” column in Tab. 1 indicates whether a method uses solely the reference ligand’s shape or also incorporates its atom positions.

We compare UniGuide with an alternative guidance approach adapted from Guan et al. [67] in App. D.4 and refer to App. C and App. D.3 for further information on the unconditional models and the guidance parameters, respectively. In addition, inspired by the performance of UniGuide on the

Table 2: Structure-Based Drug Design. Quantitative comparison of generated ligands for target pockets from the CrossDocked and Binding MOAD test sets. Results taken from the respective works are indicated with (*). We highlight the best conditioning approach for the DiffSBDD backbone in **bold** and underline the best approach over all methods.

		Method	Vina Score (\downarrow)	Vina Min (\downarrow)	Vina Dock (\downarrow)	QED (\uparrow)	SA (\uparrow)
CrossDocked	Non-Diff.	Test Set	-6.362 ± 3.14	-6.707 ± 2.50	-7.450 ± 2.33	0.48	0.73
		3D-SBDD* [38]	-5.754 ± 3.25	-6.180 ± 2.42	-6.746 ± 4.02	0.51	0.63
		Pocket2Mol* [48]	-5.139 ± 3.17	-6.415 ± 2.93	-7.152 ± 4.90	0.56	<u>0.74</u>
	Diffusion-based	DecompDiff* (No Drift) [67]	$-4.750 \pm -$	$-6.170 \pm -$	-	-	-
		TargetDiff* [50]	-5.466 ± 8.32	<u>-6.643 ± 4.94</u>	-7.802 ± 3.62	0.48	0.58
		DiffSBDD-cond [11]	-3.684 ± 11.3	-4.670 ± 6.06	-6.941 ± 4.33	0.47	0.58
		DiffSBDD [11]	-4.097 ± 11.3	-6.306 ± 5.00	-7.889 ± 2.61	0.57	0.64
	UniGuide	-5.103 ± 8.39	-6.610 ± 4.20	-7.921 ± 2.43	0.57	0.64	
Binding MOAD	Diffusion-based	Test Set	-6.748 ± 2.77	-7.563 ± 2.53	-8.297 ± 2.03	0.60	0.64
		DiffSBDD-cond [11]	-4.466 ± 2.63	-6.309 ± 2.52	-7.482 ± 1.84	0.43	0.56
		DiffSBDD [11]	-4.744 ± 7.70	-6.586 ± 2.59	-7.767 ± 2.06	0.55	0.62
		UniGuide	-5.074 ± 6.75	-6.622 ± 2.57	-7.911 ± 1.97	0.56	0.61

LBDD task, we further motivate its applicability for the generation of molecules given atom densities, see App. G.

Evaluation The goal of LBDD is to discover novel molecules that fit within a given 3D shape. This can be quantified by a high 3D shape similarity and low graph similarity compared to the reference ligand, as illustrated in Fig. 3 as well as App. D.2. We highlight this trade-off by reporting the ratio of these similarities in Tab. 1 as $\text{Sim}_S/\text{Sim}_G$, which constitutes the most important metric for this task. We follow Chen et al. [14] and further evaluate the mean and maximum shape similarities Sim_S and max Sim_S , respectively, per reference ligand, measured via the volume overlap between the two aligned molecules. Additionally, we report the graph similarity Sim_G defined as the Tanimoto similarity between the generated and reference ligand, and the graph similarity max Sim_G of the generated molecule with the maximum shape similarity. Further metrics concerning the quality of the generated ligands are provided in App. D.2.

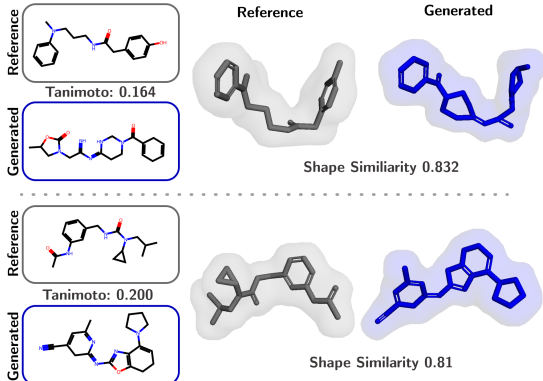


Figure 3: Examples of the two shape-conditioned ligands generated by UniGuide. The goal is to have *low* molecular graph similarity and *high* shape similarity.

Both in terms of shape similarity and graph similarity, guiding the generation of EDM with UniGuide outperforms other task-specific conditioning mechanisms and even the Virtual Screening baseline. Emphasised by the Ratio metric across all evaluated methods, UniGuide demonstrates that it is able to generate diverse molecules with very similar shapes compared to the reference ligand. Remarkably, UniGuide achieves higher shape similarity than ShapeMol+g, even though the conditional model is explicitly guided towards the position of the reference ligand through the position correction technique. UniGuide, on the other hand, does not require information about the reference’s atom positions at all to generate novel, high-quality ligands. This highlights how UniGuide and the design of condition maps enables unconditional models like EDM, that have not been tailored or trained for the LBDD task, to achieve state-of-the-art performance on new tasks.

5.2 Structure-based drug design

Datasets Following Schneuing et al. [11], we evaluate UniGuide on two protein-ligand datasets: the CrossDocked dataset [68] and the Binding MOAD dataset [69]. For the CrossDocked dataset, we follow the preprocessing as described by [38] and conduct the evaluation on 100 test protein pockets.

Table 3: Linker Design. Results taken from Igashov et al. [13] are indicated with (*). We underline the best method overall.

Method		QED (\uparrow)	SA (\downarrow)	No. Rings (\uparrow)	Valid (\uparrow)	Unique (\uparrow)	2D Filters (\uparrow)	Recovery (\uparrow)
Non-diffusion based	DeLinker + ConfVAE + MMFF [53]*	0.64 \pm 0.16	<u>3.11 \pm 0.68</u>	0.21 \pm 0.42	<u>98.3</u>	44.2	84.8	80.2
	3DLinker [52]*	<u>0.65 \pm 0.16</u>	3.14 \pm 0.68	0.24 \pm 0.43	71.5	29.2	83.7	93.5
	3DLinker (given anchors) [52]*	0.65 \pm 0.16	<u>3.11 \pm 0.67</u>	0.23 \pm 0.42	99.3	29.0	84.2	<u>94.0</u>
Diffusion-based	DiffLinker [13]*	<u>0.65 \pm 0.15</u>	3.19 \pm 0.77	0.32 \pm 0.54	90.6	51.4	<u>87.9</u>	70.7
	DiffLinker (given anchors) [13]*	<u>0.65 \pm 0.15</u>	3.24 \pm 0.81	0.36 \pm 0.59	94.8	50.9	84.7	77.5
	UniGuide (EDM)	0.64 \pm 0.16	3.63 \pm 1.08	<u>0.49 \pm 0.62</u>	89.1	<u>72.1</u>	<u>87.9</u>	58.8

The Binding MOAD dataset is preprocessed as discussed in Schneuing et al. [11], resulting in 130 test proteins. Per target pocket, 100 ligands are generated. We evaluate the generation of ligands on models that are trained on the full-atom context of the pockets in Tab. 2 and results of models trained on the C_α representation of the pockets are provided in App. E.5.

Baselines We compare UniGuide to two autoregressive models designed for the SBDD task: 3D-SBDD [38] and Pocket2Mol [48]. We further include TargetDiff [50] and DecompDiff [67], conditional diffusion models for SBDD that fix the protein pocket context during every step of the diffusion process. We exclude approaches with explicit drift terms like Guan et al. [67] and Huang et al. [70] from the comparison, as UniGuide’s SBDD condition map does not include drift terms currently, but can be readily extended to do so. Schneuing et al. [11] present two techniques for controlled structure-based generation: (i) DiffSBDD-cond, a conditional diffusion model similar to [50] and (ii) DiffSBDD, an inpainting-inspired technique that modifies the generative process of an unconditional diffusion model that jointly generates protein-ligand pairs. Across datasets, both UniGuide and DiffSBDD control the same unconditional ligand-protein diffusion model. We provide more information and further evaluation regarding this base model in App. E.2 and App. E.3 and investigate the influence of the guidance scale S as well as the resampling trick [71], a technique that modifies the generative process to better harmonise the generated ligand with the controlled pockets, in App. E.4 and App. E.5.

Evaluation As the task of SBDD is to generate ligands that bind well to a given protein pocket, we assess generated ligands based on affinity-related metrics (Vina Score, Vina Min and Vina Dock), which estimate the binding affinity between the generated ligands and a given test receptor [72]. Additionally, we measure the quality of the generated ligands using two chemical properties: the drug-likeness (QED) and the synthetic accessibility (SA) [66, 73].

Tab. 2 demonstrates that, without additional training or external networks, UniGuide performs competitively with even the highly specialised conditional models like TargetDiff and DecompDiff. Our results indicate that not fully converging to the target protein pocket due to soft guidance, compared to, for example, DiffSBDD’s inpainting-inspired technique, is not a limitation in practice. Rather, it suggests that utilising self-guidance in combination with a suitable condition map generates well-harmonised ligand-protein pairs. This is also reflected in the properties of the generated ligands, where UniGuide achieves good drug-likeness (QED) and synthetic accessibility (SA) scores. We provide additional qualitative examples for the SBDD task in Fig. 4, which showcase that UniGuide not only generates drug-like ligands but is even able to improve over the VINA Dock metric of the reference ligand.

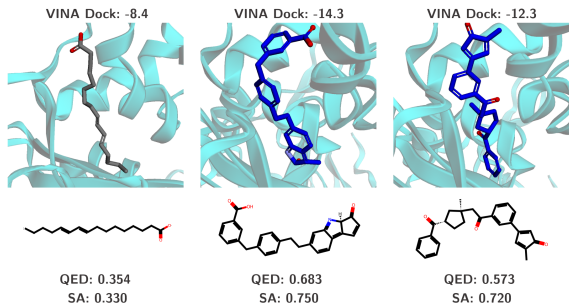


Figure 4: Qualitative example of a test protein pocket (6c0b) from the Binding MOAD dataset. We show the reference ligand (grey) and samples generated by UniGuide (blue).

5.3 Fragment-based drug design

Datasets & Baselines In the following, we investigate linker design, a subfield of fragment-based drug design. We follow Igashov et al. [13] and decompose ligands from the ZINC dataset [74] with the MMPA algorithm [75]. Note that the ZINC dataset does not contain pocket information, and the evaluated approaches operate solely at the molecular level. We compare UniGuide to DiffLinker [13], a diffusion-based conditional model that fixes fragments in space. Additionally, we evaluate

the variational autoencoder-based methods DeLinker [53] and 3DLinker [52], adapted as described in Igashov et al. [13]. We provide more information on the experimental setup as well as the unconditionally trained EDM model in App. F.1 and App. C.

Evaluation Following Igashov et al. [13], we evaluate the generated linkers and ligands with respect to their properties (SA, QED, Number of Rings and 2D Filters). We additionally measure (i) the uniqueness of the generated samples, (ii) the recovery of the reference ligands, and (iii) the validity, which combines the chemical validity and the successful linking of the fragments.

Using UniGuide to control the EDM generation enables the successful combination of the condition fragments and the generation of diverse linkers. Even compared to task-specific models, UniGuide is able to perform competitively across different metrics. Importantly, UniGuide enables the same unconditional model (EDM) to tackle both the linker design task as presented in Tab. 3 as well as the LBDD task as presented in Tab. 1 without additional training. Note that, while DiffLinker is specifically designed to generate linkers, UniGuide readily generalises to other tasks within the FBDD setting, such as fragment growing and scaffolding, see Fig. 5. Additionally, UniGuide is agnostic to the fragmentation procedure used to obtain the condition scaffolds, meaning that UniGuide will generalise to unseen fragments as long as the underlying molecule fits within the training distribution. In App. F.2, we demonstrate how the same unconditional model can be adapted for these tasks. Our quantitative evaluation highlights the benefits achieved through the unification of controlled generation provided by UniGuide.

6 Conclusion

In this work, we present UniGuide, a unified way of controlling the generation of molecular diffusion models towards geometric constraints. UniGuide generalises to a multitude of drug discovery tasks without the need for conditioning networks or specialised training protocols, enabling UniGuide to find applicability also in scenarios where little data is available. By demonstrating that specialisation is not a necessity and that a more flexible, unified method outperforms specialised approaches across tasks and datasets, we open up new avenues for streamlined and flexible generative models with wide-ranging applications.

Acknowledgements SA, LH, and JS are thankful for valuable feedback from Marcel Kollovich, Leo Schwinn, and Alessandro Palma from the DAML group and Theis Lab. SA is supported by the DAAD programme Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the Federal Ministry of Education and Research. LH is supported by the Helmholtz Association under the joint research school “Munich School for Data Science - MUDS”. FJT acknowledges support from the Helmholtz Association’s Initiative and Networking Fund through Helmholtz AI (ZT-I-PF-5-01). FJT further acknowledges support by the BMBF (01IS18053A). In addition, FJT consults for Immunai Inc., Singularity Bio B.V., CytoReason Ltd, and Omniscope Ltd and has an ownership interest in Dermagnostix GmbH and Cellarity.

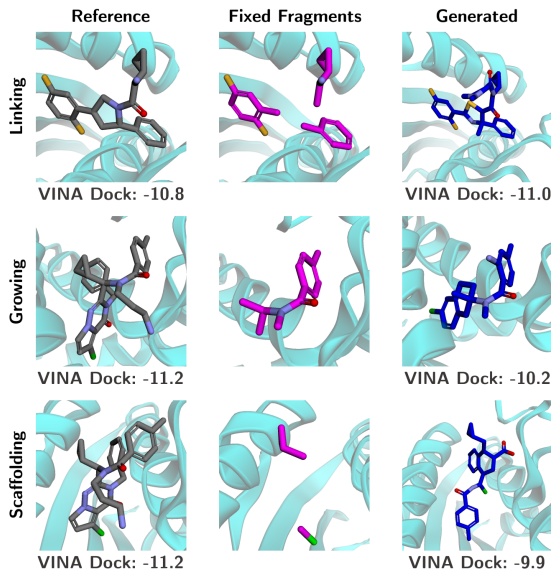


Figure 5: For various pocket-conditioned FBDD tasks, we show reference ligands (grey), desired fragments (magenta), and ligands generated by UniGuide (blue).

References

- [1] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [2] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020.
- [3] N Anand and T Achim. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv*, 2022. doi: 10.48550. *arXiv preprint arXiv:2205.15019*.
- [4] Brian L. Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem, March 2023. URL <http://arxiv.org/abs/2206.04119>. *arXiv:2206.04119* [cs, q-bio, stat].
- [5] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, and others. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023. Publisher: Nature Publishing Group UK London.
- [6] Gabriele Corso, Bowen Jing, Regina Barzilay, Tommi Jaakkola, and others. DiffDock: Diffusion steps, twists, and turns for molecular docking. In *International conference on learning representations (ICLR 2023)*, 2023.
- [7] Yuanqi Du, Tianfan Fu, Jimeng Sun, and Shengchao Liu. MolGenSurvey: A Systematic Survey in Machine Learning Models for Molecule Design. Technical Report *arXiv:2203.14500*, *arXiv*, March 2022. URL <http://arxiv.org/abs/2203.14500>. *arXiv:2203.14500* [cs, q-bio] type: article.
- [8] Leon Hetzel, Simon Böhm, Niki Kilbertus, Stephan Günemann, Mohammad Lotfollahi, and Fabian Theis. Predicting single-cell perturbation responses for unseen drugs. Technical report, ICLR2022 Machine Learning for Drug Discovery, April 2022. *arXiv:2204.13545* [cs, q-bio, stat] type: article.
- [9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [10] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- [11] Arne Schneuing, Yuanqi Du, Charles Harris, Arian Jamasb, Ilia Igashov, Weitao Du, Tom Blundell, Pietro Lió, Carla Gomes, Max Welling, Michael Bronstein, and Bruno Correia. Structure-based Drug Design with Equivariant Diffusion Models, June 2023. URL <http://arxiv.org/abs/2210.13695>. *arXiv:2210.13695* [cs, q-bio].
- [12] Jos Torge, Charles Harris, Simon V Mathis, and Pietro Lio. Diffhopp: A graph diffusion model for novel drug design via scaffold hopping. *arXiv preprint arXiv:2308.07416*, 2023.
- [13] Ilia Igashov, Hannes Stärk, Clément Vignac, Victor Garcia Satorras, Pascal Frossard, Max Welling, Michael Bronstein, and Bruno Correia. Equivariant 3D-Conditional Diffusion Models for Molecular Linker Design, October 2022. URL <http://arxiv.org/abs/2210.05274>. *arXiv:2210.05274* [cs, q-bio].
- [14] Ziqi Chen, Bo Peng, Srinivasan Parthasarathy, and Xia Ning. Shape-conditioned 3D molecule generation via equivariant diffusion models. *arXiv preprint arXiv:2308.11890*, 2023.
- [15] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022.

- [16] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. LION: Latent point diffusion models for 3D shape generation. *arXiv preprint arXiv:2210.06978*, 2022.
- [17] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [18] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.
- [19] Weilun Wang, Jianmin Bao, Wengang Zhou, Dongdong Chen, Dong Chen, Lu Yuan, and Houqiang Li. Semantic image synthesis via diffusion models. *arXiv preprint arXiv:2207.00050*, 2022.
- [20] Emiel Hooeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant Diffusion for Molecule Generation in 3D, June 2022. URL <http://arxiv.org/abs/2203.17003>. arXiv:2203.17003 [cs, q-bio, stat].
- [21] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18208–18218, 2022.
- [22] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- [23] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 843–852, 2023.
- [24] Fan Bao, Min Zhao, Zhongkai Hao, Peiyao Li, Chongxuan Li, and Jun Zhu. Equivariant Energy-Guided SDE for Inverse Molecular Design, February 2023. URL <http://arxiv.org/abs/2209.15408>. arXiv:2209.15408 [physics, q-bio].
- [25] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction Tree Variational Autoencoder for Molecular Graph Generation. Technical report, International Conference on Machine Learning, 2018. URL <https://arxiv.org/abs/1802.04364>.
- [26] Krzysztof Maziarz, Henry Jackson-Flux, Pashmina Cameron, Finton Sirockin, Nadine Schneider, Nikolaus Stiefl, Marwin Segler, and Marc Brockschmidt. Learning to Extend Molecular Scaffolds with Structural Motifs. Technical Report arXiv:2103.03864, arXiv, April 2022. URL <http://arxiv.org/abs/2103.03864>. arXiv:2103.03864 [cs, q-bio] type: article.
- [27] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Hierarchical Generation of Molecular Graphs using Structural Motifs. Technical report, International Conference on Machine Learning, 2020.
- [28] Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete Denoising diffusion for graph generation, October 2022. URL <http://arxiv.org/abs/2209.14734>. arXiv:2209.14734 [cs].
- [29] Xiangzhe Kong, Wenbing Huang, Zhixing Tan, and Yang Liu. Molecule Generation by Principal Subgraph Mining and Assembling, September 2022. URL <http://arxiv.org/abs/2106.15098>. arXiv:2106.15098 [cs, q-bio].
- [30] Nicola De Cao and Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. Technical Report arXiv:1805.11973, arXiv, May 2018. URL <http://arxiv.org/abs/1805.11973>. arXiv:1805.11973 [cs, stat] type: article.
- [31] Leon Hetzel, Johanna Sommer, Bastian Rieck, Fabian Theis, and Stephan Günnemann. MAG-Net: Motif-agnostic generation of molecules from shapes. *arXiv preprint arXiv:2305.19303*, 2023.

- [32] Zijie Geng, Shufang Xie, Yingce Xia, Lijun Wu, Tao Qin, Jie Wang, Yongdong Zhang, Feng Wu, and Tie-Yan Liu. De Novo Molecular Generation via Connection-aware Motif Mining, February 2023. URL <http://arxiv.org/abs/2302.01129>. arXiv:2302.01129 [cs].
- [33] Johanna Sommer, Leon Hetzel, David Lüdke, Fabian J. Theis, and Stephan Günnemann. The Power of Motifs as Inductive Bias for Learning Molecular Distributions. March 2023. URL https://openreview.net/forum?id=cS3_jJ0se3z.
- [34] Mohamed Amine Ketata, Nicholas Gao, Johanna Sommer, Tom Wollschläger, and Stephan Günnemann. Lift Your Molecules: Molecular Graph Generation in Latent Euclidean Space, June 2024. URL <http://arxiv.org/abs/2406.10513>. arXiv:2406.10513.
- [35] Matthew Ragoza, Tomohide Masuda, and David Ryan Koes. Learning a continuous representation of 3D molecular structures with deep generative models. *arXiv preprint arXiv:2010.08687*, 2020.
- [36] Niklas W. A. Gebauer, Michael Gastegger, and Kristof T. Schütt. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules, January 2020. URL <http://arxiv.org/abs/1906.00957>. G-SchNet.
- [37] Youzhi Luo and Shuiwang Ji. An Autoregressive Flow Model for 3D Molecular Geometry Generation from Scratch. October 2021. URL <https://openreview.net/forum?id=C03Ajc-NS5W>. G-SphereNet.
- [38] Shitong Luo, Jiaqi Guan, Jianzhu Ma, and Jian Peng. A 3D generative model for structure-based drug design. *Advances in Neural Information Processing Systems*, 34:6229–6239, 2021.
- [39] Victor Garcia Satorras, Emiel Hoogeboom, Fabian B. Fuchs, Ingmar Posner, and Max Welling. E(n) Equivariant Normalizing Flows, January 2022. URL <http://arxiv.org/abs/2105.09016>. arXiv:2105.09016 [physics, stat].
- [40] Yuxuan Song, Jingjing Gong, Minkai Xu, Ziyao Cao, Yanyan Lan, Stefano Ermon, Hao Zhou, and Wei-Ying Ma. Equivariant flow matching with hybrid probability transport for 3D molecule generation. In *Thirty-seventh conference on neural information processing systems*, 2023.
- [41] Minkai Xu, Alexander Powers, Ron Dror, Stefano Ermon, and Jure Leskovec. Geometric Latent Diffusion Models for 3D Molecule Generation, May 2023. URL <http://arxiv.org/abs/2305.01140>. GeoLDM.
- [42] Lemeng Wu, Chengyue Gong, Xingchao Liu, Mao Ye, and Qiang Liu. Diffusion-based Molecule Generation with Informative Prior Bridges, September 2022. URL <http://arxiv.org/abs/2209.00865>. Bridge.
- [43] Alex Morehead and Jianlin Cheng. Geometry-Complete Diffusion for 3D Molecule Generation and Optimization, June 2023. URL <http://arxiv.org/abs/2302.04313>. GCDM.
- [44] Bo Qiang, Yuxuan Song, Minkai Xu, Jingjing Gong, Bowen Gao, Hao Zhou, Wei-Ying Ma, and Yanyan Lan. Coarse-to-fine: a hierarchical diffusion model for molecule generation in 3d. In *International conference on machine learning*, pages 28277–28299. PMLR, 2023.
- [45] Clement Vignac, Nagham Osman, Laura Toni, and Pascal Frossard. Midi: Mixed graph and 3d denoising diffusion for molecule generation. *arXiv preprint arXiv:2302.09048*, 2023.
- [46] Han Huang, Leilei Sun, Bowen Du, and Weifeng Lv. Learning Joint 2D & 3D Diffusion Models for Complete Molecule Generation, June 2023. URL <http://arxiv.org/abs/2305.12347>. JODO.
- [47] Lei Huang, Hengtong Zhang, Tingyang Xu, and Ka-Chun Wong. MDM: Molecular Diffusion Model for 3D Molecule Generation, September 2022. URL <http://arxiv.org/abs/2209.05710>. MDM.
- [48] Xingang Peng, Shitong Luo, Jiaqi Guan, Qi Xie, Jian Peng, and Jianzhu Ma. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. In *International conference on machine learning*, pages 17644–17655. PMLR, 2022.

- [49] Meng Liu, Youzhi Luo, Kanji Uchino, Koji Maruhashi, and Shuiwang Ji. Generating 3d molecules for target protein binding. *arXiv preprint arXiv:2204.09410*, 2022.
- [50] Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. *arXiv preprint arXiv:2303.03543*, 2023.
- [51] Fergus Imrie, Thomas E Hadfield, Anthony R Bradley, and Charlotte M Deane. Deep generative design with 3D pharmacophoric constraints. *Chemical science*, 12(43):14577–14589, 2021. Publisher: Royal Society of Chemistry.
- [52] Yinan Huang, Xingang Peng, Jianzhu Ma, and Muhan Zhang. 3DLinker: an E (3) equivariant variational autoencoder for molecular linker design. *arXiv preprint arXiv:2205.07309*, 2022.
- [53] Fergus Imrie, Anthony R Bradley, Mihaela van der Schaar, and Charlotte M Deane. Deep generative models for 3D linker design. *Journal of chemical information and modeling*, 60(4):1983–1995, 2020. Publisher: ACS Publications.
- [54] Jiaqi Guan, Xingang Peng, PeiQi Jiang, Yunan Luo, Jian Peng, and Jianzhu Ma. LinkerNet: Fragment Poses and Linker Co-Design with 3D Equivariant Diffusion. *Advances in Neural Information Processing Systems*, 36:77503–77519, December 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f4821075019a058700f6e6738eea1365-Abstract-Conference.html.
- [55] Keir Adams and Connor W Coley. Equivariant shape-conditioned generation of 3d molecules for ligand-based drug design. *arXiv preprint arXiv:2210.04893*, 2022.
- [56] Siyu Long, Yi Zhou, Xinyu Dai, and Hao Zhou. Zero-shot 3d drug design by sketching and generating. *Advances in Neural Information Processing Systems*, 35:23894–23907, 2022.
- [57] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.
- [58] Marcel Kollovich, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. *arXiv preprint arXiv:2307.11494*, 2023.
- [59] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976. Publisher: International Union of Crystallography.
- [60] Wolfgang Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 34(5):827–828, 1978. Publisher: International Union of Crystallography.
- [61] Paul J Besl and Neil D McKay. Method for registration of 3-D shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.
- [62] Robert Abel and Sathesh Bhat. Chapter Seven - Free Energy Calculation Guided Virtual Screening of Synthetically Feasible Ligand R-Group and Scaffold Modifications: An Emerging Paradigm for Lead Optimization. In Robert A. Goodnow, editor, *Annual Reports in Medicinal Chemistry*, volume 50 of *Platform Technologies in Drug Discovery and Validation*, pages 237–262. Academic Press, January 2017. doi: 10.1016/bs.armc.2017.08.007. URL <https://www.sciencedirect.com/science/article/pii/S0065774317300106>.
- [63] Qingxin Li. Application of Fragment-Based Drug Discovery to Versatile Targets. *Frontiers in Molecular Biosciences*, 7:180, 2020. ISSN 2296-889X. doi: 10.3389/fmolb.2020.00180.
- [64] Hans-Joachim Böhm, Alexander Flohr, and Martin Stahl. Scaffold hopping. *Drug Discovery Today. Technologies*, 1(3):217–224, December 2004. ISSN 1740-6749. doi: 10.1016/j.ddtec.2004.10.009.

- [65] Chunquan Sheng and Wannian Zhang. Fragment informatics and computational fragment-based drug design: an overview and update. *Medicinal Research Reviews*, 33(3):554–598, May 2013. ISSN 1098-1128. doi: 10.1002/med.21255.
- [66] Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alan Aspuru-Guzik, and Alex Zhavoronkov. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models, October 2020. URL <http://arxiv.org/abs/1811.12823>. arXiv:1811.12823 [cs, stat].
- [67] Jiaqi Guan, Xiangxin Zhou, Yuwei Yang, Yu Bao, Jian Peng, Jianzhu Ma, Qiang Liu, Liang Wang, and Quanquan Gu. DecompDiff: Diffusion Models with Decomposed Priors for Structure-Based Drug Design, 2024. URL <https://arxiv.org/abs/2403.07902>. Version Number: 1.
- [68] Paul G Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B Iovanisci, Ian Snyder, and David R Koes. 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. Publisher: ACS Publications.
- [69] Liegi Hu, Mark L Benson, Richard D Smith, Michael G Lerner, and Heather A Carlson. Binding MOAD (mother of all databases). *Proteins: Structure, Function, and Bioinformatics*, 60(3): 333–340, 2005. Publisher: Wiley Online Library.
- [70] Zhilin Huang, Ling Yang, Xiangxin Zhou, Zhilong Zhang, Wentao Zhang, Xiawu Zheng, Jie Chen, Yu Wang, CUI Bin, and Wenming Yang. Protein-ligand interaction prior for binding-aware 3d molecule diffusion models. *The Twelfth International Conference on Learning Representations 2024*.
- [71] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022.
- [72] Amr Alhossary, Stephanus Daniel Handoko, Yuguang Mu, and Chee-Keong Kwoh. Fast, accurate, and reliable molecular docking with QuickVina 2. *Bioinformatics (Oxford, England)*, 31(13):2214–2216, 2015. Publisher: Oxford University Press.
- [73] Greg Landrum and others. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8:31, 2013.
- [74] John J. Irwin, Khanh G. Tang, Jennifer Young, Chinzorig Dandarchuluun, Benjamin R. Wong, Munkhzul Khurelbaatar, Yuri S. Moroz, John Mayfield, and Roger A. Sayle. ZINC20—A Free Ultralarge-Scale Chemical Database for Ligand Discovery. *Journal of Chemical Information and Modeling*, 2020. ISSN 1549-9596. doi: 10.1021/acs.jcim.0c00675.
- [75] Alexander G Dossetter, Edward J Griffen, and Andrew G Leach. Matched molecular pair analysis in drug discovery. *Drug Discovery Today*, 18(15-16):724–731, 2013. Publisher: Elsevier.
- [76] Priyank Jaini, Lars Holdijk, and Max Welling. Learning equivariant energy based models with equivariant stein variational gradient descent. *Advances in Neural Information Processing Systems*, 34:16727–16737, 2021.
- [77] Maciej Wójcikowski, Piotr Zielenkiewicz, and Pawel Siedlecki. Open Drug Discovery Toolkit (ODDT): a new open-source player in the drug discovery field. *Journal of cheminformatics*, 7(1):1–6, 2015. Publisher: BioMed Central.
- [78] Mikko J Vainio, J Santeri Puranen, and Mark S Johnson. ShaEP: molecular overlay based on shape and electrostatic potential, 2009.

- [79] Freyr Sverrisson, Jean Feydy, Bruno E. Correia, and Michael M. Bronstein. Fast end-to-end learning on protein surfaces. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15267–15276, June 2021. doi: 10.1109/CVPR46437.2021.01502. URL <https://ieeexplore.ieee.org/document/9577686>. ISSN: 2575-7075.
- [80] Maciej Wójcikowski, Piotr Zielenkiewicz, and Pawel Siedlecki. Open Drug Discovery Toolkit (ODDT): a new open-source player in the drug discovery field. *Journal of Cheminformatics*, 7(1):26, December 2015. ISSN 1758-2946. doi: 10.1186/s13321-015-0078-2. URL <https://jcheminf.biomedcentral.com/articles/10.1186/s13321-015-0078-2>.
- [81] Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, Tommi Jaakkola, and Andreas Krause. Independent se (3)-equivariant models for end-to-end rigid protein docking. *arXiv preprint arXiv:2111.07786*, 2021.
- [82] Jan Zaucha, Charlotte A. Softley, Michael Sattler, Dmitrij Frishman, and Grzegorz M. Popowicz. Deep learning model predicts water interaction sites on the surface of proteins using limited-resolution data. *Chemical Communications*, 56(98):15454–15457, December 2020. ISSN 1364-548X. doi: 10.1039/D0CC04383D. URL <https://pubs.rsc.org/en/content/articlelanding/2020/cc/d0cc04383d>. Publisher: The Royal Society of Chemistry.
- [83] Huimin Zhu, Renyi Zhou, Dongsheng Cao, Jing Tang, and Min Li. A pharmacophore-guided deep learning approach for bioactive molecular generation. *Nature Communications*, 14(1): 6234, October 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-41454-9. URL <https://www.nature.com/articles/s41467-023-41454-9>. Publisher: Nature Publishing Group.
- [84] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '87, pages 163–169, New York, NY, USA, August 1987. Association for Computing Machinery. ISBN 978-0-89791-227-3. doi: 10.1145/37401.37422. URL <https://dl.acm.org/doi/10.1145/37401.37422>.

A Impact Statement

Our research holds the promise of significant contributions to the advancement of drug discovery, possibly assisting in the discovery of novel pharmaceutical compounds. Nevertheless, because of its applications in drug discovery, this strategy is not without its hazards. The ability to produce various molecules with desired properties may not only serve the purpose of beneficial drug development but may also unintentionally result in the creation of dangerous substances or compounds with unexpected effects. These concerns underline the critical need for careful handling when working with the structures this method can generate.

B Proof of Theorem 4.1

First, recall Theorem 4.1 that we provide in Sec. 4:

Theorem 1. Consider a function $C : \mathcal{S} \times \mathcal{Z} \rightarrow \mathcal{Z}$. If $C(\mathbf{s}, \mathbf{z})$ is invariant to rigid transformations G in the first argument and equivariant in the second argument, then the gradient $\nabla_{\mathbf{z}} \|\mathbf{v}\|_2^2$ of the vector $\mathbf{v} = \mathbf{z} - C(\mathbf{s}, \mathbf{z})$ is equivariant to transformations of \mathbf{z} .

Proof. We start the proof by showing that $\|\mathbf{v}\|_2$ is invariant to transformations of both \mathbf{z} and \mathbf{s} .

1. $\|\mathbf{z} - C(\mathbf{s}, \mathbf{z})\|_2$ is invariant to transformations in \mathbf{z} :

$$\begin{aligned} \|G\mathbf{z} - C(\mathbf{s}, G\mathbf{z})\|_2 &= \|G\mathbf{z} - GC(\mathbf{s}, \mathbf{z})\|_2 && [C \text{ is equivariant in } \mathbf{z}] \\ &= \|G(\mathbf{z} - C(\mathbf{s}, \mathbf{z}))\|_2 \\ &= \|\mathbf{z} - C(\mathbf{s}, \mathbf{z})\|_2 && [G \text{ is a rigid transformation}] \end{aligned} \tag{22}$$

2. $\|\mathbf{z} - C(\mathbf{s}, \mathbf{z})\|_2$ is invariant to transformations \mathbf{s} follows immediately:

$$\|\mathbf{z} - C(G\mathbf{s}, \mathbf{z})\|_2 = \|\mathbf{z} - C(\mathbf{s}, \mathbf{z})\|_2 \quad [C \text{ is invariant in } \mathbf{s}] \tag{23}$$

In a second step, we make use of the fact that for a group of transformations G , it holds that if $\mathcal{L}(\cdot, \cdot)$ is a G -invariant function, $\nabla_x \mathcal{L}(\cdot, x)$ is G -equivariant [76]. From the invariance of $\|\mathbf{v}\|_2$, it follows immediately that $\nabla_{\mathbf{z}} \|\mathbf{z} - C(\mathbf{s}, \mathbf{z})\|_2^2$ is equivariant to transformations of \mathbf{z} . \square

C Unconditional Equivariant Diffusion Model

UniGuide guides an unconditional diffusion model given an arbitrary condition and a natural choice for a model operating only on the molecule level is the EDM model as proposed in Hoogeboom et al. [20].

We adapt this model for two tasks presented in this work, namely the LBDD task discussed in Sec. 5.1 and the Linker Design task as presented in Sec. 5.3. For these tasks, we train an unconditional EDM model both on the MOSES dataset [66] in the configuration as described in Chen et al. [14] and on the ZINC dataset [74] as described in Igashov et al. [13]. For both trainings, we employ the hyperparameter configuration for the GEOM dataset as described in Hoogeboom et al. [20]. We run multi-GPU trainings on 4 NVIDIA A100 GPUs until convergence, however, a single NVIDIA A100 GPU is sufficient for this training and will only increase the training time. For inference, we employ the Resampling trick as discussed in Lugmayr et al. [71] with $R = 10$ resampling steps and $T = 100$ timesteps. EDM is available under the MIT License.

D Ligand-based drug design

D.1 Implementation details

We train two unconditional diffusion models, ShapeMol [U] and EDM, to generate 3D molecules on the MOSES dataset [66], licensed under the MIT License, for which we generate 3D conformers with RDKit [73], available under the BSD 3-Clause License. We use 1,593,653 training samples

and randomly select 1000 samples for validation. The model architecture of ShapeMol[U] is an unconditional version of the ShapeMol model proposed in Chen et al. [14], and it is trained with 1000 diffusion steps. ShapeMol [U] is trained with a batch size of 32 on two NVIDIA A100 GPUs for 500 epochs. Unlike ShapeMol, we do not concatenate the molecular surface embedding of the ligands to the features. For the shape-conditioned generation with position correction (ShapeMol+g), we follow the scheme proposed in Chen et al. [14]. It provides further guidance to the conditional generation by sampling 20 query points from a Gaussian distribution centred around every atom in the reference ligand. The position correction adjusts the coordinates of the predicted atom positions during every generation step by pushing the coordinates close to the query points as follows:

$$\hat{\mathbf{x}} = (1 - \sigma)\hat{\mathbf{x}} + \sigma \sum_{\mathbf{z} \in n(\hat{\mathbf{x}}, \mathcal{Q})} \mathbf{z}/n, \text{ if } \sum_{\mathbf{z} \in n(\hat{\mathbf{x}}, \mathcal{Q})} d(\hat{\mathbf{x}}, \mathbf{z})/n > \gamma, \quad (24)$$

where $d(\hat{\mathbf{x}}, \mathbf{z})$ is the Euclidean distance, $n(\hat{\mathbf{x}}, \mathcal{Q})$ is the set of n nearest neighbors of $\hat{\mathbf{x}}$ in \mathcal{Q} and $\gamma > 0$ is a distance threshold. We follow the implementation of Chen et al. [2] for the position correction method by setting $\gamma = 0.2$ and only guiding during the first 700 denoising steps.

For the shape-conditioned generation with UniGuide, we extract the mesh of the condition ligand using the Open Drug Discovery Toolkit [77], which is available under the BSD 3-Clause revised License. The query points we use for guidance are 512 points sampled uniformly on the mesh surface. For the evaluation, we measure the shape similarity Sim_S as the volume overlap between the aligned generated ligand and the condition ligand. For the alignment, we utilise the ShaEP tool [78].

We provide a detailed description of the LBDD sampling algorithm in Algorithm 1.

Algorithm 1: Sampling algorithm to generate a ligand that is conditioned on a reference ligand \mathcal{M}_{ref} 's surface, using an unconditional model $\epsilon_\theta(\mathbf{z}_t, t)$ modelling the distribution over molecules. The points $\mathbf{y} \in \mathbb{R}^{K \times 3}$ are sampled uniformly from the surface of \mathcal{M}_{ref} , enclosing the volume V .

Require: \mathbf{y} , α : desired margin to surface, k : number of nearest neighbours
 $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ {Sample from normal prior}

for $t = T$ **to** 1 **do**

$\mathbf{x}_t, \mathbf{h}_t = \mathbf{z}_t$

$\hat{\mathbf{x}}_0 = \frac{\mathbf{x}_t - \sqrt{1 - \alpha_t} \epsilon_\theta^*(\mathbf{z}_t, t)}{\sqrt{\alpha_t}}$ {Compute the conformation $\hat{\mathbf{x}}_0$ of the clean approximation $\hat{\mathbf{z}}_0$ }

For every atom $\hat{\mathbf{x}}_i$ in $\hat{\mathbf{x}}_0$ **do**:

$\bar{\mathbf{y}}_i = \frac{1}{k} \sum_{\mathbf{y} \in \mathcal{N}_{\hat{\mathbf{x}}_i}} \mathbf{y}$ {Compute the mean of k nearest neighbors of $\hat{\mathbf{x}}_i$ in \mathbf{y} }

 Compute $(\mathbf{c}_x)_i$ based on Eq. (21) {Compute component-wise condition map}

$\mathcal{L} = \mathcal{L}(\hat{\mathbf{x}}_0, \mathbf{c}_x)$

$\mathbf{g} = \nabla_{\mathbf{x}_t} \mathcal{L}$ {Compute gradient of guidance loss}

$\boldsymbol{\mu}_t = \boldsymbol{\mu}_\theta(\mathbf{z}_t, t) - \lambda(t) \cdot \mathbf{g}$ {Update the mean function}

$\mathbf{z}_{t-1} \sim \mathcal{N}(\boldsymbol{\mu}_t, \sigma_t \mathbf{I})$

end for

return \mathbf{z}_0

D.2 Additional results

For completeness, we report additional quantitative evaluation of the generated ligands' properties in Tab. 4. We also provide further qualitative results of the generated ligands for the LBDD task in Fig. 6. UniGuide generates ligands with better shape similarity to the reference ligands compared to the conditional model ShapeMol with the position correction technique.

Table 4: Additional ligand property results for the methods discussed in Sec. 5.1. We report mean and standard deviation and highlight the best result in **bold**.

method	Connect. (\uparrow)	Unique (\uparrow)	QED	SA (\uparrow)	LogP (\uparrow)	Lipinski (\uparrow)
ShapeMol	98.8%	99.9%	0.753	0.640 \pm 0.104	2.001 \pm 1.360	4.979 \pm 0.156
ShapeMol+g	97.0%	99.8%	0.751	0.630 \pm 0.110	1.908 \pm 1.508	4.874 \pm 0.170
UniGuide+ ShapeMol[U]	98.0%	100%	0.736	0.625 \pm 0.103	1.828 \pm 1.463	4.974 \pm 0.186
UniGuide (ShapeMol)	99.0%	100%	0.750	0.641 \pm 0.107	2.002 \pm 1.374	4.982 \pm 0.152
UniGuide+ EDM	99.8%	99.99%	0.742	0.636 \pm 0.088	1.833 \pm 1.221	4.994 \pm 0.082

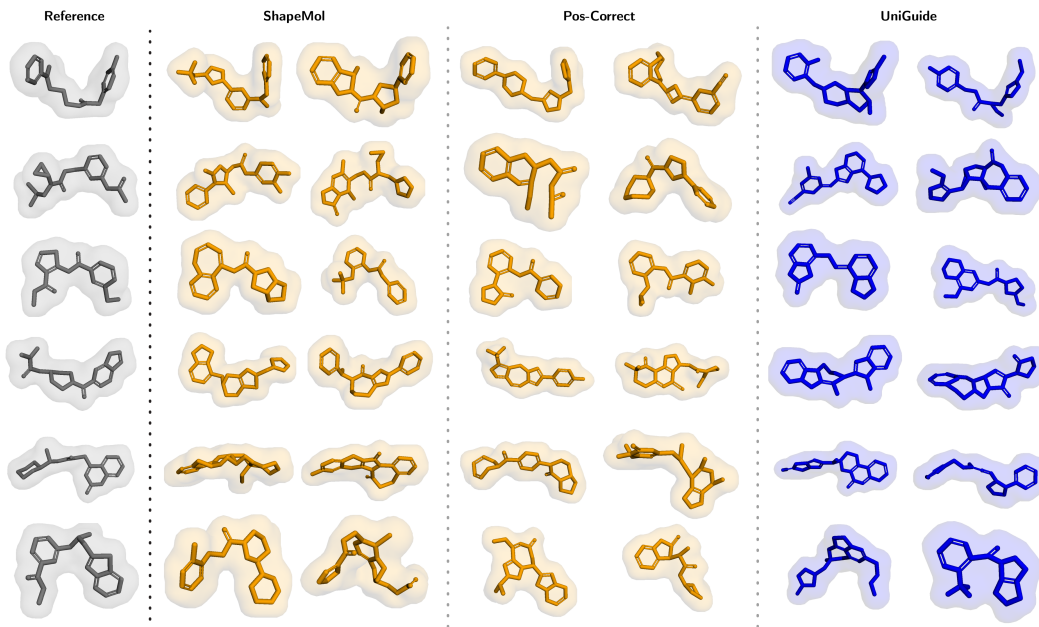


Figure 6: Examples of the ligands generated by *ShapeMol*, *Pos-Correct* and *UniGuide*. *Pos-Correct* is the position correction technique proposed by Chen et al. [14]. Both *Pos-Correct* and *UniGuide* are combined with the unconditionally trained model *ShapeMol* [U]. We plot the reference ligand as well as the generated ligands with their shapes.

Table 5: Comparison of *UniGuide* with validity guidance for shape-based generation. We highlight the ratio metric as the most critical indicator, reflecting the balance between shape similarity and graph dissimilarity.

	Sim_S (\uparrow)	maxSim_S (\uparrow)	Sim_G (\downarrow)	maxSim_G (\downarrow)	Ratio (\uparrow)	Connect. (\uparrow)	Unique. (\uparrow)	Diversity (\uparrow)	QED (\uparrow)
Validity Guidance	0.59	0.76	0.20	0.20	2.96	97%	100%	0.76	0.69
<i>UniGuide</i> (EDM)	0.74	0.86	0.21	0.20	3.53	99%	99%	0.73	0.74

D.3 Guidance parameters

For the LBDD task, the guidance strength S is weighted by an exponentially decreasing function $\frac{\beta_t}{\sqrt{\alpha_t}}$. For the guided generation using the unconditional *ShapeMol* [U] model under the *UniGuide* framework, we define a scale scheduler that increases with an exponent of 1.01 and weight it with $\frac{\beta_t}{\sqrt{\alpha_t}}$ and guide from the diffusion step 1000 to the diffusion step 200. For the guided generation using the EDM model, we use a linear scale function that increases from 5 to 15. The guidance is applied from the diffusion step 920 to the last timestep 1.

D.4 Comparison of *UniGuide* with an alternative loss formulation

We adapt the validity guidance loss from Guan et al. [50] to the LBDD setting. The proposed loss is grounded in the smooth distance function $S(x)$ from Sverrisson et al. [79], which computes as:

$$S(x) = -\sigma \log \left(\sum_i^N \exp(-\|x_t - y_i\|_2^2 / \sigma) \right) .$$

This function provides an alternative approach to shape-based generation by deriving an appropriate loss function $\sum_x S(x)$, rather than modifying the condition map as proposed by *UniGuide*. Here, $S(x)$ implicitly defines a surface through $S(x) - \gamma = 0$ and points x_t inside satisfy $S(x_t) < \gamma$.

On a technical level, the gradient for validity guidance computes as follows:

$$\begin{aligned}\nabla_{x_t} \mathcal{S}(x_t) &= \nabla_{x_t} \left[-\sigma \log \left(\sum_i^N \exp(-\|x_t - y_i\|_2^2 / \sigma) \right) \right] \\ &= \frac{1}{\sum \exp(\dots)} \sum_i^N \underbrace{\exp(\dots)}_{\omega_i} \nabla_{x_t} \|x_t - y_i\|_2^2 \\ &= \frac{1}{\sum \omega_i} \nabla_{x_t} \sum_i^N \omega_i \|x_t - y_i\|_2^2 \quad .\end{aligned}$$

This gradient formulation is quite similar (up to the weighting) to UniGuide’s special case $\mathcal{S} = \mathcal{Z}$, as it computes an L_2 loss on a given conformation ($\{y_i\}$), see Eq. (18), meaning that it does not generalise to arbitrary geometric conditions.

We emphasise that UniGuide is more broadly applicable because it separates surface computation from gradient computation, offering two key benefits. First, since the condition map does not require differentiability, there is greater flexibility in computing surface points. Second, the precise geometric intuition behind the condition map makes it easier to adapt to new scenarios, as demonstrated by our application to generating density-guided molecules.

For the empirical comparison, we selected the hyperparameters σ and γ in the surface loss computation to achieve a high DICE score between the implicitly defined surface and the meshes UniGuide utilises for LBDD ($\sigma = 1, \gamma = 2, \text{DICE} > 0.8$). Our surface calculations use the Open Drug Discovery Toolkit (ODDT), which assigns specific radii to individual atom types and employs the marching cubes algorithm to generate meshes [80].

We performed several runs around the above-specified hyperparameter configuration. The runs performed similarly, and we report the best result in Tab. 5. Although validity guidance for LBDD yields low graph similarity, the shape similarity remains suboptimal compared to UniGuide. Additionally, we frequently encounter numerical instability when computing the guidance term, an issue not present with UniGuide’s formulation of LBDD. One possible explanation for this numerical instability is that the surface is defined implicitly, unlike UniGuide where it is explicitly defined. The explicit definition in UniGuide allows for relating the gradient updates directly to the surface, as shown in Eq. (21).

E Structure-based drug design

Algorithm 2: Sampling algorithm to generate a ligand conditioned on a protein pocket $\tilde{\mathbf{z}}^P$ using the unconditional joint model $\epsilon_\theta(\mathbf{z}_t, t)$, where $\mathbf{z}_t = [\mathbf{z}_t^M, \mathbf{z}_t^P]$, that models the distribution $P(\mathbf{z}^M, \mathbf{z}^P)$. The guidance signal is controlled via the guidance strength S . Note that samples from the generative process $p_\theta(\mathbf{z}_{t-1} | \mathbf{z}_t)$ are assumed to be CoM-free.

```

Require:  $\tilde{\mathbf{z}}^P, S$ 
 $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  { Sample from normal prior }
for  $t = T$  to 1 do
   $\hat{\mathbf{z}}_0^P = \mathbf{z}_t^P - \sqrt{1 - \alpha_t} \epsilon_\theta^P(\mathbf{z}_t, t) / \sqrt{\alpha_t}$  { Compute the clean data of the pocket }
   $\mathcal{L} = \mathcal{L}(\hat{\mathbf{z}}_0^P, \tilde{\mathbf{z}}^P)$ 
   $\mathbf{g} = (\nabla_{\mathbf{x}_t} \mathcal{L} - \nabla_{\mathbf{x}_t} \overline{\mathcal{L}}, \nabla_{\mathbf{h}_t} \mathcal{L})$  { Compute gradient and subtract the CoM }
   $\boldsymbol{\mu}_t = \boldsymbol{\mu}_\theta(\mathbf{z}_t, t) - \lambda(t) \cdot \mathbf{g}$  { Update the mean of the pocket }
   $\mathbf{z}_{t-1} \sim \mathcal{N}(\boldsymbol{\mu}_t, \sigma_t \mathbf{I})$ 
end for
return  $\mathbf{z} = (\mathbf{z}_0^M, \mathbf{z}_0^P)$ 

```

E.1 SBDD sampling algorithm

We provide the algorithm for inference in the SBDD task scenario in Algorithm 2.

E.2 Ligand-protein generative joint model

SBDD aims to generate a ligand given a protein pocket: $p_{\theta}(\mathbf{z}^{\mathcal{M}} | \mathbf{z}_{\text{test}}^{\mathcal{P}}, t)$. We adopt DiffSBDD [11], an unconditional joint diffusion model that approximates the joint distribution $p(\mathbf{z}_{\text{data}}^{\mathcal{M}}, \mathbf{z}_{\text{data}}^{\mathcal{P}})$ of generating ligand-protein pairs, where the noise predictor $\epsilon_{\theta}(\mathbf{z}_t^{\mathcal{M}}, \mathbf{z}_t^{\mathcal{P}}, t)$ is parametrised by EGNN. DiffSBDD is available under the MIT License. To process ligand and pocket nodes with a single GNN, atom types and residue types are embedded jointly. Atom and residue features are then decoded separately using atom decoder and residue decoder to $\epsilon_{\theta}^{\mathcal{M}}(\mathbf{z}_t^{\mathcal{M}}, \mathbf{z}_t^{\mathcal{P}}, t)$ and $\epsilon_{\theta}^{\mathcal{P}}(\mathbf{z}_t^{\mathcal{M}}, \mathbf{z}_t^{\mathcal{P}}, t)$ [11].

For the unconditional sampling with the joint model, the number of ligand and pocket nodes is sampled from the joint node distribution $p(N^{\mathcal{M}}, N^{\mathcal{P}})$, measured across a training set of $(\mathcal{M}, \mathcal{P})$ pairs. During the modified generative process with the inpainting-inspired technique or with UniGuide the number of pocket nodes is set to be equal to the number of nodes in $\mathcal{P}_{\text{test}}$, while the size of the ligand is generated from a conditional distribution $p(N^{\mathcal{M}} | N^{\mathcal{P}})$. Since this sampling procedure leads to ligands that are much smaller compared to the reference ligands found in the test set, the mean size of sampled ligands is increased by 10 for Binding MOAD and 5 for CrossDocked during ligand generation [11]. We utilize the unconditional base models from Schneuing et al. [11], which are trained on either the C_{α} or full-atom context from the Binding MOAD or CrossDocked datasets. However, we retrain the DiffSBDD model specifically on the full-atom context of the CrossDocked data, as we were unable to reproduce the reported results in this configuration from Schneuing et al. [11]. We find that contrary to what is reported in Schneuing et al. [11], the model converges early and does not need a full 1000 epochs to fully train. We employ this checkpoint to evaluate both the DiffSBDD inpainting-inspired approach as well as UniGuide. We train the model on four NVIDIA A100 GPU with a batch size of 2. 8 training epochs take approximately 24 hours.

Table 6: Hyperparameters of ligand and proteins graphs in joint models

	CROSSDOCKED		BINDING MOAD	
	JOINT C_{α} MODEL	JOINT FULL-ATOM MODEL	JOINT C_{α} MODEL	JOINT FULL-ATOM MODEL
EDGES (LIGAND-LIGAND)	FULLY CONNECTED	FULLY CONNECTED	FULLY CONNECTED	FULLY CONNECTED
EDGES (LIGAND-POCKET)	< 5 Å	< 5 Å	< 8 Å	< 7 Å
EDGES (POCKET-POCKET)	< 5 Å	< 5 Å	< 8 Å	< 4 Å

Representing ligands and proteins as graphs Proteins consist of amino acids, where every amino acid is a set of amino (NH), carboxyl (CO), α -carbon atom and a side chain (R) that is specific to every amino acid type [81]. The C_{α} -representation of a protein pocket is a residue-level graph, in which the node features of the protein are represented as one-hot encodings of the amino acid type. The full-atom representation of the receptor is an atom-level graph and represents the full context of the protein pocket. Details on processed graphs of the joint model $p(\mathbf{z}^{\mathcal{M}}, \mathbf{z}^{\mathcal{P}})$ are provided in Tab. 6. We refer the reader to Schneuing et al. [11] for more information on the hyperparameters of the joint model.

Table 7: Quantitative evaluation of samples generated by the unconditional joint models [11] trained on Crossdocked (C.D.) and Binding MOAD (B.M). We report the mean over all generated ligands.

DATASET	R	T	QED (\uparrow)	SA (\uparrow)	LIPINSKI (\uparrow)	DIVERSITY (\uparrow)	CONNECTIVITY (\uparrow)	VALIDITY (\uparrow)
C.D. (C_{α})	1	500	0.535	0.660	4.741	0.772	0.893	0.986
C.D. (C_{α})	10	50	0.578	0.752	4.836	0.774	0.994	0.986
B.M. (C_{α})	1	500	0.471	0.608	4.783	0.824	0.839	0.985
B.M. (C_{α})	10	50	0.544	0.665	4.883	0.823	0.961	0.992

E.3 Further Comparison to DiffSBDD

In addition to Tab. 2, we follow the experimental setup as utilised in Schneuing et al. [11] to compare UniGuideto DiffSBDD, which uses the same base model, in particular. In Tab. 8, we further investigate the advantages of using self-guidance in combinations with UniGuide over both the conditional DiffSBDD model (DiffSBDD-cond) as well as the inpainting-inspired technique

Table 8: Quantitative comparison of generated ligands for target pockets from the CrossDocked and Binding MOAD test sets. Results taken from Schneuing et al. [11] are indicated with (*). We report mean and standard deviation and highlight the best diffusion-based approach in **bold**.

	Vina (\downarrow)	Vina Top 10% (\downarrow)	QED (\uparrow)	SA (\uparrow)	Lipinski (\uparrow)	Diversity (\uparrow)	RMSD (\downarrow)	
Test Set	-6.865 \pm 2.35	-	0.476 \pm 0.20	0.728 \pm 0.14	4.340 \pm 1.14	-	-	
CrossDocked	3D-SBDD* [38]	-5.888 \pm 1.91	-7.289 \pm 2.34	0.502 \pm 0.17	0.675 \pm 0.14	4.787 \pm 0.51	0.742 \pm 0.09	
	Pocket2Mol* [48]	-7.058 \pm 2.80	-8.712 \pm 3.18	0.572 \pm 0.16	0.752 \pm 0.12	4.936 \pm 0.27	0.735 \pm 0.15	
	Graph-BP* [49]	-4.719 \pm 4.03	-7.165 \pm 1.40	0.502 \pm 0.12	0.307 \pm 0.09	4.883 \pm 0.37	0.844 \pm 0.01	
	TargetDiff* [50]	-7.318 \pm 2.47	-9.669 \pm 2.55	0.483 \pm 0.20	0.584 \pm 0.13	4.594 \pm 0.83	0.718 \pm 0.09	0.000 \pm 0.00
	DiffSBDD-cond*	-6.950 \pm 2.06	-9.120 \pm 2.16	0.469 \pm 0.21	0.578 \pm 0.13	4.562 \pm 0.89	0.728 \pm 0.07	0.000 \pm 0.00
	DiffSBDD	-7.216 \pm 2.54	-9.490 \pm 2.00	0.571 \pm 0.19	0.639 \pm 0.14	4.808 \pm 0.50	0.707 \pm 0.09	0.045 \pm 0.01
	UniGuide	-7.320 \pm 2.27	-9.514 \pm 2.04	0.571 \pm 0.19	0.638 \pm 0.14	4.822 \pm 0.47	0.705 \pm 0.08	0.047 \pm 0.01
Bind. MOAD	Test Set	-8.331 \pm 2.05	-	0.602 \pm 0.15	0.636 \pm 0.08	4.838 \pm 0.37	-	
	Graph-BP* [49]	-4.843 \pm 2.24	-6.629 \pm 0.95	0.512 \pm 0.11	0.310 \pm 0.09	4.945 \pm 0.27	0.826 \pm 0.01	0.000 \pm 0.00
	DiffSBDD-cond	-7.172 \pm 1.88	-9.174 \pm 2.13	0.430 \pm 0.20	0.564 \pm 0.12	4.526 \pm 0.80	0.711 \pm 0.08	0.000 \pm 0.00
	DiffSBDD	-7.263 \pm 4.19	-9.776 \pm 2.25	0.546 \pm 0.21	0.618 \pm 0.12	4.777 \pm 0.54	0.740 \pm 0.05	53 \pm 31
	UniGuide	-7.661 \pm 2.99	-9.864 \pm 2.13	0.556 \pm 0.20	0.605 \pm 0.12	4.799 \pm 0.50	0.723 \pm 0.05	55 \pm 31

(DiffSBDD). UniGuide reliably achieves superior VINA Dock scores compared to both DiffSBDD models and performs competitively with the conditional TargetDiff model. In App. E.4 and App. E.5, we expand on this experimental comparison with further analysis of the effects of Resampling as well as the guidance strength.

E.4 Resampling

Inpainting is introduced for diffusion models to condition outputs with fixed parts [71] and can be applied for structure-based molecular tasks. Given a model that generates $(\mathbf{z}_t^M, \mathbf{z}_t^P)$ pairs at denoising step t , the protein pocket P_t is replaced with the noised representation of protein context $\tilde{\mathbf{z}}_t^P$. This noised representation can be obtained through the forward process of diffusion models as specified in Eq. (2). However, the direct application of this method leads to locally harmonised samples that struggle to incorporate the global context [71]. In order to effectively harmonise the generated information during the entire generative process, Lugmayr et al. [71] propose a technique they call ‘‘Resampling’’. This modifies the reverse Markov chain by moving back and forth in the diffusion process to enable the model to better incorporate the replaced components.

Schneuing et al. [11] propose to use the same resampling technique to harmonise the replaced protein context with the ligand, since the replaced receptor is sampled independently of the ligand. During resampling, each latent representation is repeatedly diffused back and forth before advancing to the next time step. We found that resampling further improves the general performance of the unconditional generation, and thus improves the guided generation as well. We report results for this in App. E.5, where we evaluate how the unconditional generation of the joint model is improved across different metrics with added resampling steps. We follow Schneuing et al. [11] in using the setting of $R = 10$ resampling steps and $T = 50$ timesteps. While DiffSBDD resamples the ligand and the noised target protein pocket, we resample the guided protein pocket and ligand with UniGuide. In general, the concept of resampling can be applied to harmonise the configuration \mathbf{z}_t with the condition c .

E.5 Guidance parameters

The guidance scale S controls the strength of the guiding signal, see Eq. (7) and it is weighted by $w(t) = \frac{\beta(t)}{\sqrt{\alpha_t}}$ during the generation. We use a constant scale S for structure-based drug design experiments and evaluate for several guidance scale values in Tab. 9 and Tab. 10 for models trained on the Binding MOAD dataset with C_α and full-atom representation respectively. The quantitative evaluation on the CrossDocked data is shown under Tab. 11 and Tab. 12 with additional metrics reported in Tab. 7. For the generation with the C_α -models, we generate 100 samples for every test pocket with a batch size of 50. The full generation takes approximately 5 hours for Binding MOAD and 6 hours for CrossDocked. For the DiffSBDD model trained on the Binding MOAD fullatom pocket data, we use a batch size of 15 for the generation. We use a batch size of 2 to sample with the DiffSBDD model trained on CrossDocked (fullatom).

Table 9: Results for the Binding MOAD test set with the unconditional DiffSBDD base model trained on the C_α -representation of the pockets combined with UniGuide and the inpainting-inspired technique DiffSBDD [11]. We provide results for varying the guidance scales S during our controlled generation. We also report results for the DiffSBDD-cond (C_α) model trained on the C_α pockets.

METHOD	S	R/T	VINA (\downarrow)	VINA TOP 10% (\downarrow)	QED (\uparrow)	SA (\uparrow)	LIPINSKI (\uparrow)	DIVERSITY (\uparrow)	RMSD (\downarrow)
DIFFSBDD-COND (C_α)	-	-	-6.628 \pm 1.59	-8.291 \pm 1.26	0.481 \pm 0.20	0.554 \pm 0.11	4.651 \pm 0.70	0.714 \pm 0.04	0.000 \pm 0.00
DIFFSBDD	-	1/500	-6.362 \pm 3.04	-8.179 \pm 1.24	0.452 \pm 0.20	0.541 \pm 0.11	4.604 \pm 0.76	0.734 \pm 0.03	0.008 \pm 0.01
UNI GUIDE	1.0	1/500	-6.519 \pm 2.05	-8.227 \pm 1.23	0.464 \pm 0.20	0.540 \pm 0.11	4.627 \pm 0.73	0.733 \pm 0.03	0.125 \pm 0.01
UNI GUIDE	2.0	1/500	-6.568 \pm 2.13	-8.268 \pm 1.25	0.471 \pm 0.20	0.543 \pm 0.11	4.636 \pm 0.73	0.735 \pm 0.04	0.105 \pm 0.25
UNI GUIDE	3.0	1/500	-6.667 \pm 1.92	-8.305 \pm 1.28	0.468 \pm 0.20	0.542 \pm 0.11	4.622 \pm 0.73	0.737 \pm 0.03	0.072 \pm 0.03
UNI GUIDE	4.0	1/500	-6.587 \pm 1.86	-8.293 \pm 1.29	0.470 \pm 0.20	0.544 \pm 0.11	4.636 \pm 0.72	0.735 \pm 0.03	0.058 \pm 0.01
UNI GUIDE	6.0	1/500	-6.568 \pm 1.93	-8.284 \pm 1.26	0.468 \pm 0.20	0.542 \pm 0.11	4.630 \pm 0.73	0.734 \pm 0.03	0.045 \pm 0.01
UNI GUIDE	7.0	1/500	-6.575 \pm 1.86	-8.296 \pm 1.28	0.469 \pm 0.20	0.544 \pm 0.11	4.636 \pm 0.72	0.735 \pm 0.03	0.043 \pm 0.05
DIFFSBDD	-	10/50	-6.896 \pm 3.10	-8.962 \pm 1.37	0.547 \pm 0.20	0.578 \pm 0.20	4.754 \pm 0.50	0.709 \pm 0.05	0.007 \pm 0.01
UNI GUIDE	1.0	10/50	-6.845 \pm 3.68	-8.972 \pm 1.36	0.547 \pm 0.19	0.578 \pm 0.13	4.756 \pm 0.53	0.709 \pm 0.05	0.216 \pm 0.21
UNI GUIDE	2.0	10/50	-6.889 \pm 3.83	-9.018 \pm 1.40	0.547 \pm 0.19	0.577 \pm 0.13	4.756 \pm 0.52	0.707 \pm 0.04	0.279 \pm 0.03
UNI GUIDE	3.0	10/50	-7.050 \pm 2.38	-9.051 \pm 1.39	0.551 \pm 0.18	0.575 \pm 0.14	4.763 \pm 0.50	0.706 \pm 0.04	0.220 \pm 0.01
UNI GUIDE	4.0	10/50	-7.016 \pm 2.93	-9.023 \pm 1.38	0.552 \pm 0.18	0.578 \pm 0.14	4.765 \pm 0.50	0.708 \pm 0.03	0.168 \pm 0.05
UNI GUIDE	6.0	10/50	-7.053 \pm 2.91	-9.067 \pm 1.39	0.550 \pm 0.18	0.579 \pm 0.14	4.761 \pm 0.51	0.703 \pm 0.04	0.146 \pm 0.01
UNI GUIDE	7.0	10/50	-7.076 \pm 2.27	-9.038 \pm 1.38	0.550 \pm 0.18	0.579 \pm 0.14	4.767 \pm 0.50	0.704 \pm 0.04	0.131 \pm 0.01

For all tables, we conduct the experiments both with and without resampling. The VINA Dock score is measured with QuickVina2 [72], available under the Apache License, and the chemical properties (QED, SA, Lipinski) are measured with RDKit. We note that in all ablation tables we measure the VINA Dock score on the processed molecules, following Schneuing et al. [11], while the VINA Dock score in Tab. 2 is measured following Guan et al. [67]. Both the VINA Dock score and chemical properties improve with additional resampling steps ($R = 10, T = 50$) for both datasets. Additionally, increasing the guidance scale improves the RMSD with respect to the target protein, and results in generating ligands with an improved binding affinity (lower VINA).

E.6 Additional Results for SBDD

Supplementary to Tab. 2 we provide additional metrics for the evaluation of the generated ligands in Tab. 14: the validity as measured by RDKit [73] and the connectivity, representing the percentage of valid molecules without any disconnected fragments. Additionally, we report the uniqueness and novelty of the valid connected ligands.

E.7 Runtime Comparison

In Tab. 13, we provide a comparison of the different controlled generation mechanisms regarding their runtime. While UniGuide has a higher runtime compared to other conditioning mechanisms, as it has to compute gradients through the diffusion model at inference time, it stays comparable to other mechanisms such as inpainting.

Table 10: Results for the Binding MOAD test set with the unconditional DiffSBDD base model trained on the full-atom context of the pockets combined with UniGuide and the inpainting-inspired technique DiffSBDD [11]. We provide results for varying the guidance scales S during our controlled generation. We also report results for the conditional diffusion model DiffSBDD-cond.

METHOD	S	R/T	VINA (\downarrow)	VINA TOP 10% (\downarrow)	QED (\uparrow)	SA (\uparrow)	LIPINSKI (\uparrow)	DIVERSITY (\uparrow)	RMSD (\downarrow)
DIFFSBDD-COND	-	-	-7.172 \pm 1.88	-9.174 \pm 2.13	0.430 \pm 0.20	0.564 \pm 0.12	4.526 \pm 0.80	0.711 \pm 0.08	0.0 \pm 0.0
DIFFSBDD	-	1/500	-6.540 \pm 2.00	-8.427 \pm 1.39	0.413 \pm 0.20	0.531 \pm 0.11	4.611 \pm 0.77	0.748 \pm 0.03	55 \pm 31
UNI GUIDE	6.0	1/500	-6.696 \pm 1.78	-8.561 \pm 1.58	0.407 \pm 0.19	0.527 \pm 0.11	4.587 \pm 0.78	0.740 \pm 0.04	55 \pm 31
UNI GUIDE	7.0	1/500	-6.683 \pm 1.91	-8.575 \pm 1.52	0.406 \pm 0.19	0.524 \pm 0.11	4.579 \pm 0.80	0.738 \pm 0.04	55 \pm 31
UNI GUIDE	8.0	1/500	-6.682 \pm 1.77	-8.555 \pm 1.52	0.407 \pm 0.19	0.526 \pm 0.11	4.591 \pm 0.78	0.740 \pm 0.04	55 \pm 31
UNI GUIDE	9.0	1/500	-6.689 \pm 1.74	-8.541 \pm 1.50	0.403 \pm 0.19	0.524 \pm 0.11	4.589 \pm 0.78	0.738 \pm 0.04	55 \pm 31
DIFFSBDD	-	10/50	-7.263 \pm 4.19	-9.776 \pm 2.25	0.546 \pm 0.21	0.618 \pm 0.12	4.777 \pm 0.54	0.740 \pm 0.05	53 \pm 31
UNI GUIDE	5.0	10/50	-7.470 \pm 2.97	-9.621 \pm 1.84	0.563 \pm 0.20	0.605 \pm 0.12	4.807 \pm 0.50	0.723 \pm 0.05	55 \pm 31
UNI GUIDE	6.0	10/50	-7.570 \pm 3.20	-9.731 \pm 1.90	0.566 \pm 0.20	0.606 \pm 0.12	4.815 \pm 0.48	0.722 \pm 0.05	55 \pm 31
UNI GUIDE	7.0	10/50	-7.639 \pm 2.39	-9.793 \pm 2.06	0.559 \pm 0.20	0.605 \pm 0.12	4.804 \pm 0.49	0.723 \pm 0.05	54 \pm 31
UNI GUIDE	8.0	10/50	-7.635 \pm 2.71	-9.821 \pm 2.07	0.558 \pm 0.20	0.605 \pm 0.12	4.804 \pm 0.50	0.720 \pm 0.05	54 \pm 31
UNI GUIDE	9.0	10/50	-7.661 \pm 2.99	-9.864 \pm 2.13	0.556 \pm 0.20	0.605 \pm 0.12	4.799 \pm 0.50	0.723 \pm 0.05	55 \pm 31

Table 11: Evaluation of the samples generated for the CrossDocked test set using the joint ligand-protein diffusion model trained on the C_α pocket representation for varying guidance scales S . The base model is combined either with the inpainting-inspired technique (DiffSBDD) or UniGuide. We further report the evaluation of the molecules generated by the conditional model DiffSBDD-cond that is trained on the C_α pocket representation.

METHOD	S	R/T	VINA (\downarrow)	VINA TOP 10% (\downarrow)	QED (\uparrow)	SA (\uparrow)	LIPINSKI (\uparrow)	DIVERSITY (\uparrow)	RMSE (\downarrow)
DIFFSBDD-COND (C_α)	-	-	-6.770 \pm 2.73	-8.796 \pm 1.75	0.475 \pm 0.22	0.612 \pm 0.12	4.536 \pm 0.91	0.725 \pm 0.06	0.000 \pm 0.00
DIFFSBDD	-	1/500	-6.485 \pm 2.50	-8.472 \pm 1.62	0.510 \pm 0.21	0.619 \pm 0.12	4.640 \pm 0.73	0.735 \pm 0.06	0.053 \pm 0.03
UNI GUIDE	2.0	1/500	-6.528 \pm 2.64	-8.527 \pm 1.67	0.518 \pm 0.21	0.623 \pm 0.12	4.649 \pm 0.73	0.739 \pm 0.05	0.085 \pm 0.01
UNI GUIDE	3.0	1/500	-6.604 \pm 2.57	-8.556 \pm 1.64	0.519 \pm 0.21	0.622 \pm 0.12	4.657 \pm 0.72	0.738 \pm 0.05	0.070 \pm 0.01
UNI GUIDE	4.0	1/500	-6.578 \pm 2.72	-8.563 \pm 1.68	0.518 \pm 0.21	0.623 \pm 0.12	4.659 \pm 0.71	0.741 \pm 0.05	0.059 \pm 0.02
UNI GUIDE	5.0	1/500	-6.563 \pm 2.58	-8.549 \pm 1.66	0.516 \pm 0.21	0.624 \pm 0.12	4.646 \pm 0.72	0.741 \pm 0.05	0.052 \pm 0.01
UNI GUIDE	6.0	1/500	-6.658 \pm 2.50	-8.578 \pm 1.69	0.527 \pm 0.21	0.629 \pm 0.12	4.683 \pm 0.69	0.741 \pm 0.05	0.045 \pm 0.01
DIFFSBDD	-	10/50	-7.030 \pm 3.39	-9.057 \pm 1.79	0.559 \pm 0.21	0.730 \pm 0.12	4.729 \pm 0.60	0.720 \pm 0.07	0.052 \pm 0.01
UNI GUIDE	1.0	10/50	-6.909 \pm 3.35	-9.069 \pm 1.79	0.563 \pm 0.21	0.734 \pm 0.12	4.743 \pm 0.57	0.721 \pm 0.06	0.711 \pm 0.12
UNI GUIDE	2.0	10/50	-7.015 \pm 3.20	-9.115 \pm 1.79	0.562 \pm 0.21	0.733 \pm 0.12	4.735 \pm 0.60	0.721 \pm 0.07	0.188 \pm 0.02
UNI GUIDE	3.0	10/50	-7.081 \pm 2.95	-9.140 \pm 1.83	0.560 \pm 0.20	0.732 \pm 0.11	4.742 \pm 0.57	0.723 \pm 0.07	0.127 \pm 0.01
UNI GUIDE	4.0	10/50	-7.086 \pm 3.27	-9.125 \pm 1.81	0.561 \pm 0.19	0.731 \pm 0.10	4.729 \pm 0.60	0.719 \pm 0.06	0.102 \pm 0.01
UNI GUIDE	5.0	10/50	-7.117 \pm 2.78	-9.127 \pm 1.78	0.561 \pm 0.20	0.731 \pm 0.12	4.738 \pm 0.59	0.722 \pm 0.07	0.090 \pm 0.01
UNI GUIDE	6.0	10/50	-7.113 \pm 3.00	-9.133 \pm 1.80	0.556 \pm 0.20	0.731 \pm 0.12	4.734 \pm 0.60	0.720 \pm 0.32	0.077 \pm 0.01

F Fragment-based drug design

F.1 Linker Design

For the experimental evaluation of the linker design task, we follow Igashov et al. [13], employ the ZINC dataset [74] and preprocess it following Igashov et al. [13]. That is, 3D conformers are generated from the SMILES strings present in the dataset with RDKit [73]. We fragment the dataset ligands using an MMPA-based algorithm [75, 73], generating multiple fragment conditions per molecule. We train an unconditional EDM model for this task as specified in App. C. For the evaluation metrics, we follow Igashov et al. [13]. Note that the synthetic accessibility score computation (SA) in Tab. 3 differs from the remaining experimental evaluations. While Igashov et al. [13] report the SA score s_{SA} directly, Schneuing et al. [11] report the SA score as $(10 - s_{SA})/9$.

For the task of linker design, we adjust the condition map as discussed in Sec. 4.2 slightly to include anchor information, similar in spirit to the DiffLinker model incorporating anchor information [13]. That is, additionally to guiding parts of the molecule to the desired fragment configuration, we additionally define a cuboid’s surface that is defined from the specified anchor atoms. We can then utilise this surface condition $C_{\partial V}$ to guide the linker atoms in accordance with Eq. (21). Additionally, we can expand this surface based on the linker size to ensure chemical validity of the generated linker. This condition map highlights the flexibility of UniGuide condition maps in various tasks, especially through the combination of two definitions of the condition map. For the experimental evaluation, we sample the size of the linker nodes uniformly in accordance with Igashov et al. [13] and compare to the DiffLinker model without an external network to predict the linker size. Note, however, that also the unconditional EDM model combined with UniGuide can be adapted to include such predictors.

Table 12: Results for the CrossDocked test set with the joint model trained on the full-atom pocket representation of the pocket for varying guidance scales S . The unconditional model is either controlled by the inpainting-inspired technique (DiffSBDD) or UniGuide.

METHOD	S	R/T	VINA (\downarrow)	VINA TOP 10% (\downarrow)	QED (\uparrow)	SA (\uparrow)	LIPINSKI (\uparrow)	DIVERSITY (\uparrow)	RMSE (\downarrow)
DIFFSBDD-COND	-	-	-6.950 \pm 2.06	-9.120 \pm 2.16	0.469 \pm 0.21	0.578 \pm 0.13	4.562 \pm 0.89	0.728 \pm 0.07	0.000 \pm 0.00
DIFFSBDD	-	1/500	-6.225 \pm 1.77	-8.115 \pm 1.64	0.469 \pm 0.20	0.573 \pm 0.11	4.691 \pm 0.70	0.778 \pm 0.04	0.049 \pm 0.01
UNI GUIDE	5.0	1/500	-6.346 \pm 1.74	-8.208 \pm 1.62	0.482 \pm 0.20	0.570 \pm 0.12	4.718 \pm 0.67	0.773 \pm 0.04	0.040 \pm 0.01
UNI GUIDE	6.0	1/500	-6.335 \pm 1.72	-8.225 \pm 1.61	0.484 \pm 0.20	0.571 \pm 0.12	4.715 \pm 0.66	0.775 \pm 0.04	0.039 \pm 0.01
UNI GUIDE	7.0	1/500	-6.338 \pm 1.73	-8.218 \pm 1.60	0.481 \pm 0.19	0.571 \pm 0.12	4.710 \pm 0.67	0.774 \pm 0.04	0.039 \pm 0.01
UNI GUIDE	8.0	1/500	-6.366 \pm 1.72	-8.261 \pm 1.57	0.485 \pm 0.20	0.570 \pm 0.12	4.717 \pm 0.66	0.773 \pm 0.03	0.039 \pm 0.01
DIFFSBDD	-	10/50	-7.216 \pm 2.54	-9.490 \pm 2.00	0.571 \pm 0.19	0.639 \pm 0.14	4.808 \pm 0.50	0.707 \pm 0.09	0.045 \pm 0.01
UNI GUIDE	6.0	10/50	-7.295 \pm 2.22	-9.441 \pm 1.95	0.574 \pm 0.19	0.641 \pm 0.14	4.825 \pm 0.47	0.706 \pm 0.08	0.047 \pm 0.01
UNI GUIDE	7.0	10/50	-7.320 \pm 2.27	-9.514 \pm 2.04	0.571 \pm 0.19	0.638 \pm 0.14	4.822 \pm 0.47	0.705 \pm 0.08	0.047 \pm 0.01
UNI GUIDE	8.0	10/50	-7.298 \pm 2.21	-9.460 \pm 2.01	0.568 \pm 0.19	0.641 \pm 0.14	4.818 \pm 0.47	0.703 \pm 0.09	0.048 \pm 0.01
UNI GUIDE	9.0	10/50	-7.265 \pm 2.45	-9.495 \pm 2.05	0.577 \pm 0.19	0.640 \pm 0.14	4.821 \pm 0.47	0.706 \pm 0.08	0.049 \pm 0.01

Table 13: We evaluate the runtime of UniGuide and compare it to DiffSBDD-cond and DiffSBDD from Schneuing et al. [11]. We report the average time (in seconds) to generate 100 ligands per pocket for the CrossDocked (C_α), Binding Moad (C_α) and Binding Moad (fullatom).

DATASET	MODEL	RUNTIME (S)
CROSSDOCKED (C_α)	DIFFSBDD-COND	60 ± 68
	DIFFSBDD	141 ± 55
	UNI GUIDE	193 ± 61
BINDING MOAD (C_α)	DIFFSBDD-COND	54 ± 42
	DIFFSBDD	61 ± 17
	UNI GUIDE	104 ± 36
BINDING MOAD (FULL)	DIFFSBDD-COND	345 ± 55
	DIFFSBDD	398 ± 95
	UNI GUIDE	453 ± 120

Table 14: Additional metrics for the methods discussed in Sec. 5.2.

		VALIDITY (↑)	CONNECTIVITY (↑)	UNIQUENESS (↑)	NOVELTY (↑)
CROSS-DOCKED	TEST SET	100%	100%	96.00%	96.88%
	DIFFSBDD-COND (C_α)	95.32%	80.63%	99.97%	99.81%
	DIFFSBDD-COND	97.32%	78.91%	99.99%	99.91%
	DIFFSBDD (C_α)	99.20%	98.14%	99.26%	99.16%
	DIFFSBDD	97.76%	89.84%	99.94%	99.87%
	UNI GUIDE (C_α)	99.12%	98.35%	99.50%	99.24%
	UNI GUIDE	97.40%	93.18%	99.93%	99.76%
BINDING MOAD	TEST SET	97.69%	100%	38.58%	77.55%
	DIFFSBDD-COND (C_α)	94.43%	77.17%	100%	100%
	DIFFSBDD-COND	96.20%	63.20%	100%	100%
	DIFFSBDD (C_α)	98.54%	91.45%	100%	100%
	DIFFSBDD	94.22%	75.60%	100%	100%
	UNI GUIDE (C_α)	98.44%	93.12%	100%	99.99%
	UNI GUIDE	93.85%	79.95%	100%	100%

F.2 General Fragment Conditions

To assess the performance of UniGuide for the task of FBDD, we create an experimental setup with the goal of generating ligands conditioned on desired fragments roughly following [13]. We select 10 random protein targets from the Binding MOAD dataset and decompose their corresponding reference ligands using an MMPA-based algorithm [75, 73]. This decomposition results in a set of 40 different scenarios, including separated fragments we want to link, a fragment to grow or small functional groups to perform scaffolding. For every set of fixed fragments, we aim to guide the unconditional generation of ligands towards the generation of a ligand containing the desired fragments. As the protein is not the target of the guidance, we employ the DiffSBDD-cond model, which is conditionally trained on the (C_α)-representation of the protein pocket. For every set of fixed fragments, we generate 100 ligands and use a constant guidance scale of 8.

We provide quantitative results for the task of fragment-based drug design in Tab. 15. On the one hand, the task requires the desired fragments to be present in the generated molecule. Thus, we measure the success rate of recovery (Hit Ratio) and the RMSD between the generated fragments and desired fragments. On the other hand, given that the target fragments are met in the generated ligand, the generation has to achieve favourable chemical properties, high binding affinity, as well as high diversity within the set of generated ligands and low similarity to the reference ligand. As the Inpaint mechanism enforces the fragment during generation more strictly, it is able to achieve a better Hit Ratio and RMSD. Nevertheless, UniGuide achieves competitive results but also better VINA docking scores, better properties, and lower similarity compared to the reference ligand.

The FBDD task puts a hard constraint on the generated ligands, namely that a set of desired fragments has to be present in the generated ligand. However, neither DiffSBDD nor UniGuide can guarantee that the condition fragments are present in the generated samples.

We provide further qualitative results of the generated ligands for the FBDD task in Fig. 7.

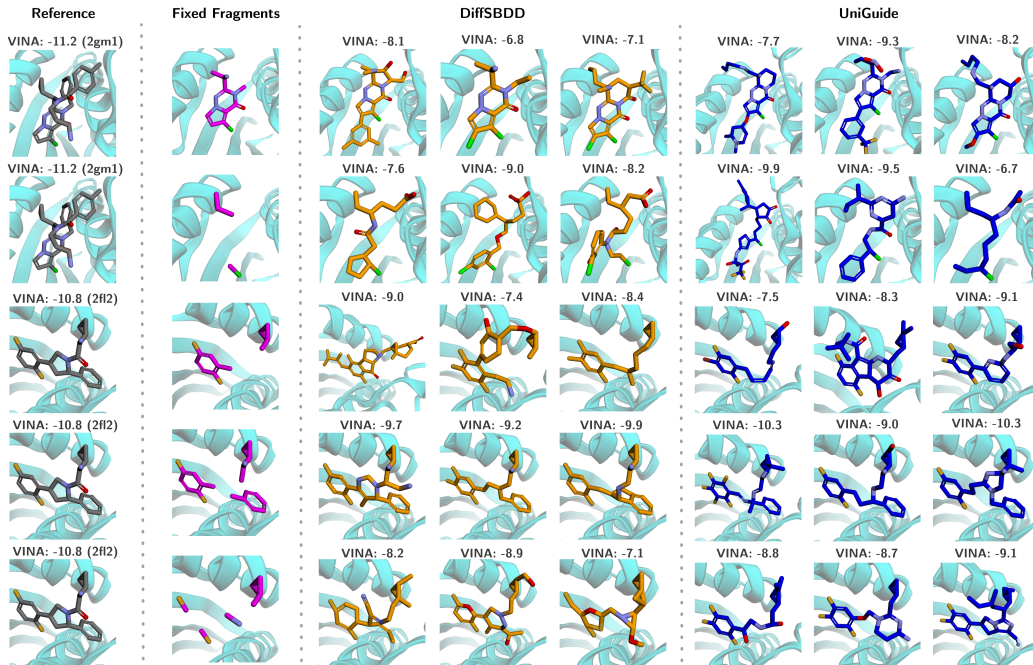


Figure 7: Examples of the generated fragment conditioned ligands.

Table 15: Quantitative comparison between DiffSBDD and UniGuide for the FBDD task on the Binding MOAD (C_{α}) dataset. As the condition in this FBDD scenario is a hard constraint that entails the condition to be exactly present in the generation, we add a post-hoc step for both methods where we replace the inpainted or guided parts with the exact condition atoms. We report mean and standard deviation and highlight the best method in **bold**.

	DIFFSBDD	UNI GUIDE
VINA (\downarrow)	-7.406 \pm 0.79	-7.924 \pm 0.89
QED (\uparrow)	0.612 \pm 0.11	0.639 \pm 0.09
SA (\uparrow)	0.703 \pm 0.11	0.691 \pm 0.10
LIPINSKI (\uparrow)	4.819 \pm 0.28	4.875 \pm 0.19
DIVERSITY (\uparrow)	0.653 \pm 0.28	0.669 \pm 0.23
SIMILARITY (\downarrow)	0.172 \pm 0.02	0.177 \pm 0.02
VALIDITY (\uparrow)	93.35 %	94.41 %
CONNECTIVITY (\uparrow)	66.87 %	68.30 %

G Atom densities in 3D space

Similar to the guidance by the volume enclosed by the molecular surface, UniGuide allows to guide towards multiple point clouds simultaneously. A natural extension of LBDD would be to harness atom densities as described in Zaucha et al. [82]. Such a setting combines aspects of LBDD and SBDD as it provides conditions also on the feature space, yet the source can only be represented by point clouds.

In particular, we anticipate UniGuide to be useful in scenarios where explicit information about advantageous features of the ligand is provided in the form of 3D densities. Examples of this include a) volumetric densities that indicate beneficial placement of certain atom types, such as oxygen atoms [82] or b) pharmacophore-like retrieval of advantageous positions for aromatic rings, as utilised in e.g. Zhu et al. [83]. On a technical level, this setting assumes that instead of a reference ligand’s structure, we only have access to (multiple) atom type densities that indicate preferred locations for optimal interaction with the protein. Additionally, instead of conditioning on a reference ligand’s shape, we could condition on a protein pocket’s surface, which primarily defines exclusion zones rather than precise atom placement.

Adapting UniGuide for such scenarios requires only minor adjustments, as the protein surface can be treated like shapes in standard LBDD, defining an exclusion zone based on proximity to the surface. The atom densities are thresholded to reflect regions of high interest and converted to surfaces using the marching cubes algorithm [84]. To also include feature information, we effectively employ a modified condition map similar to Eq. (21) that extends the transformation from the conformation to the configuration space. Moreover, the number of atoms guided by each density is adjusted based on its volume, reflecting the varying influence of each density, and guidance is only applied if atoms are sufficiently close.

We show explorative results for the guided generation of molecules towards desired atom densities using UniGuide in Fig. 8. While our current approach represents a promising first step in tackling this task, we acknowledge the potential for further refinement and are eager to explore future improvements within the UniGuide framework.

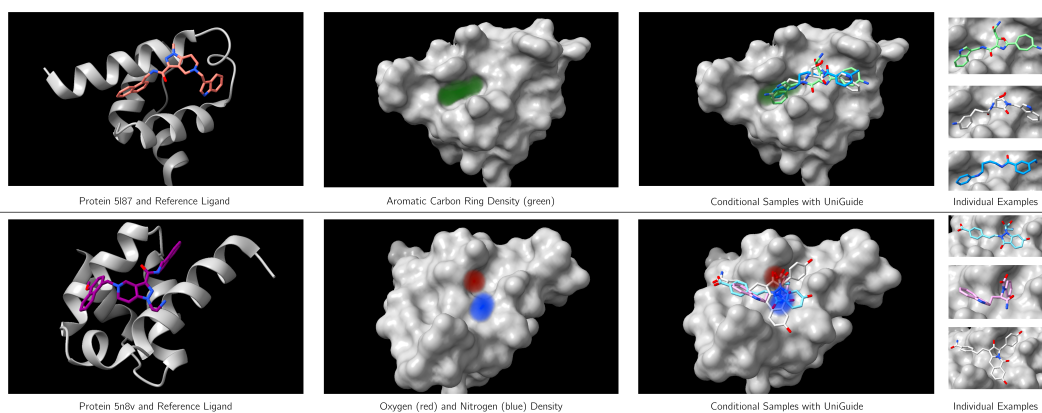


Figure 8: Given a source density of oxygens, we can extend UniGuide to generate ligands satisfying the condition.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The claims made in the abstract and introduction reflect the paper's contribution and scope: Sec. 4 details how UniGuide is readily adaptable to various tasks in drug design, attesting to the unification provided by the UniGuide framework. Sec. 5 emphasises this aspect through competitive or superior performance across various tasks, even when compared to task-specific baselines.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss the limitations of UniGuide in Sec. 4.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We discuss in Sec. 4 that the generative process retains equivariance with an appropriately chosen condition map and provide a full proof for this discussion in App. B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The setup of all experimental evaluations is described in App. E, App. F and App. D for the SBDD, FBDD and LBDD tasks respectively, including hyperparameters for UniGuide, dataset preprocessing and inference algorithms. For experimental evaluations performed according to previous work, we reference them accordingly.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

- (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We made the code available as part of the supplementary material with the submission. We have included the link to UniGuide’s project page, which will reference the public codebase.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Both the discussion of the experiments provided in Sec. 5 as well as the supplementary information provided throughout the appendix ensures that the results are sufficiently contextualised for the reader.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Throughout the experimental evaluation we provide the mean and standard deviation for all metrics that can be computed e.g. per-sample or per-pocket to ensure statistical significance of the presented results. In cases where the metric aggregates the entire set of samples, we report the mean.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide details on the hardware requirements for the training of the evaluated unconditional models in App. E.2, App. D.1 and App. C for the DiffSBDD, ShapeMol and EDM model respectively. Additionally, we provide runtime comparisons for the inference with UniGuide compared to the evaluated baselines in App. E.7.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: The research presented in this work conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss the broader impact of our work in App. A. We discuss the positive societal impacts of the proposed unification and the resulting flexibility of unconditional models to be adapted to various new drug discovery tasks in Sec. 1 and Sec. 6.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The research discussed in this paper does not require safeguards to be put in place.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Where applicable, we credit and cite owners and authors of previous works and the accompanying codebases or datasets and provide the license under which the assets were made public.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Accompanying the supplementary material, we provide documentation and instructions to navigate and utilise the UniGuide codebase.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This work did not conduct research on human subjects or crowdsourcing experiments.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This work did not conduct experiments where human subject were involved and therefore does not require IRB approvals.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.