A Detail description of Bayesian Flow Network

A.1 Key Distribution

A Bayesian Flow Network (BFN) maintains four distributions at each step: the input distribution $p_{\mathtt{I}}(\mathbf{m} \mid \boldsymbol{\theta})$, the output distribution $p_{\mathtt{O}}(\mathbf{m} \mid \boldsymbol{\theta}, t)$, the sender distribution $p_{\mathtt{S}}(\mathbf{y} \mid \mathbf{m}; \alpha)$, and the receiver distribution $p_{\mathtt{R}}(\mathbf{y} \mid \boldsymbol{\theta}, t, \alpha)$. Semantically, the input distribution represents the model's current belief (prior or posterior) over the data; the output distribution is the network's predicted distribution (allowing context) given the input parameters; the sender distribution is a noisy perturbation of the true data; and the receiver distribution is the model's predicted distribution over such noisy messages (averaging over the output distribution). Mathematically, all four are factorized over the data dimensions for tractability.

Input distribution $p_{\mathtt{I}}(\mathbf{m} \mid \boldsymbol{\theta})$. This is a factorized distribution over the data $\mathbf{m} = (\mathbf{m}^{(1)}, \ldots, \mathbf{m}^{(D)})$ with parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(D)})$. For example, each $\boldsymbol{\theta}^{(d)}$ might parametrize a univariate Gaussian or categorical for $\mathbf{m}^{(d)}$. Initially $p_{\mathtt{I}}$ is a simple prior (e.g. $\mathcal{N}(0, 1)$ or a uniform categorical). During sampling, $p_{\mathtt{I}}$ is updated via Bayes's rule when new observations arrive, so its parameters $\boldsymbol{\theta}$ become progressively more informative about $\mathbf{m}$.

Output distribution $p_{\mathtt{O}}(\mathbf{m} \mid \boldsymbol{\theta}, t)$. Given input parameters $\boldsymbol{\theta}$ and the current (discrete or continuous) process time t, a neural network computes an output vector that parameterizes $p_{\mathtt{O}}$. However, unlike $p_{\mathtt{I}}$ it integrates context across dimensions via the network: the parameters of $p_{\mathtt{O}}$ depend jointly on all of $\boldsymbol{\theta}$ and t, allowing modeling of correlations among data dimensions . In effect, $p_{\mathtt{O}}$ represents the empirical sample distribution accumulated up to step t, providing a comprehensive prediction that integrates the gathered evidence θ together with contextual information, and it will be used to predict the clean data distribution.

Sender distribution $p_{\mathtt{S}}(\mathbf{y} \mid \mathbf{m}; \alpha)$. This is the distribution over a noisy observation $\mathbf{y}$ given the true data $\mathbf{m}$, with accuracy (noise level) α_i. It is also factorized across dimensions: $p_{\mathtt{S}}(\mathbf{y}_i \mid \mathbf{m}; \alpha_i) = \prod_{d=1}^{D} p_{\mathtt{s}}\left(\mathbf{y}_i^{(d)} \mid \mathbf{m}^{(d)}; \alpha_i\right)$. The parameter α_i controls informativeness: at $i = 0$ the sender's sample is pure noise (uninformative about $\mathbf{m}$), and as $i \to \infty$ the sample concentrates on $\mathbf{y} = \mathbf{m}$. In practice, α increases through the transmission steps, so that each sender sample carries more refined information about the true data. Intuitively, $p_{\mathtt{S}}$ defines the "message" drawn from data: Alice adds controlled Gaussian or categorical noise to x to form the sender distribution and then draws a sample $y \sim p_{\mathtt{S}}(\cdot \mid x; \alpha)$ (For the Alice-and-Bob example, we recommend consulting the original BFN paper [18]).

Receiver distribution $p_{\mathtt{R}}(\mathbf{y} \mid \boldsymbol{\theta}, t, \alpha_i)$. This is the model's predictive distribution over possible noisy observations $\mathbf{y}$ given the output distribution. Formally, it is obtained by marginalizing out the unknown true data $\mathbf{m}$: $p_{\mathtt{R}}(\mathbf{y}_i \mid \boldsymbol{\theta}_{i-1}; t_i, \alpha_i) = \mathop{\mathbb{E}}_{\hat{\mathbf{m}} \sim p_{\mathtt{O}}(\hat{\mathbf{m}} \mid \theta_{i-1}; t_i)} p_{\mathtt{S}}\left(\mathbf{y} \mid \hat{\mathbf{m}}; \alpha_i\right)$. In other words, for every candidate $\mathbf{m}$, we consider the sender distribution $p_{\mathtt{S}}(\mathbf{y} \mid \mathbf{m}; \alpha)$ that would have been used if $\mathbf{m}$ were the truth, and weight these by the network's probability $p_{\mathtt{O}}(\mathbf{m} \mid \boldsymbol{\theta}, t)$. The receiver distribution thus captures both the "known unknown" due to the sender noise (entropy of p_S) and the "unknown unknown" from the output distribution's uncertainty.

Bayesian update distribution $p_{\mathtt{U}}(\boldsymbol{\theta}_i \mid \boldsymbol{\theta}_{i-1}; \alpha_i)$. This distribution specifies how the parameter vector of the input distribution evolves after assimilating a noisy observation. Concretely, let the closed-form update rule be $\boldsymbol{\theta}_i = h(\boldsymbol{\theta}_{i-1}, \mathbf{y}_i, \alpha_i)$ where $\mathbf{y}_i \sim p_{\mathtt{S}}(\mathbf{y}_i \mid \mathbf{m}; \alpha_i)$. Conditioning on the current parameters $\boldsymbol{\theta}_i$ and the accuracy level α_i, the update distribution is defined as the push-forward of the sender distribution:

$$p_{\mathtt{U}}\left(\boldsymbol{\theta}_i \mid \boldsymbol{\theta}_{i-1}; \alpha_i\right) = \mathop{\mathbb{E}}_{p_{\mathtt{O}}(\mathbf{m} \mid \theta_{i-1}; t_i)} \left[\mathop{\mathbb{E}}_{p_{\mathtt{S}}(\mathbf{y}_i \mid \mathbf{m}; \alpha_i)} \left[\delta\left(\boldsymbol{\theta}_i - h\left(\boldsymbol{\theta}_{i-1}, \mathbf{y}_i, \alpha_i\right)\right) \right] \right] \tag{16}$$

Because $h(\cdot)$ is applied component-wise under the factorisation of $p_{\mathtt{S}}$, $p_{\mathtt{U}}$ factorises across parameter dimensions, preserving tractability. The role of $p_{\mathtt{U}}$ is to enact a one-step Bayesian posterior update on $\boldsymbol{\theta}$: at early iterations, when α_i is small, $p_{\mathtt{U}}$ is broad and admits substantial parameter movement, whereas at later iterations, larger α_i makes the update sharply concentrated, refining $\boldsymbol{\theta}$ toward the values most consistent with the clean data. Thus $p_{\mathtt{U}}$ provides the formal bridge between each noisy message and the progressively more informative input distribution maintained by the network.

A.2 Modality-Agnostic Representation and Progressive Refinement

Because BFN operate on distribution parameters rather than raw data, the same framework applies uniformly to discrete, discretized, and continuous modalities. For example, discrete data are represented by categorical distribution parameters (probabilities), which lie on a probability simplex and thus provide continuous inputs to the network. In fact, the parameters of a categorical distribution are real-valued probabilities, so the inputs to the network are continuous even when the data is discrete. Likewise, continuous data use Gaussian or other continuous distributions. In all cases the model processes continuous-valued parameters, avoiding discontinuities common in discrete diffusion models. This modality-agnostic formulation means that the same Bayes-and-network machinery can generate text, images, or other data with only minimal adaptation.

A.3 Generative Sampling Process

After training, sample generation proceeds by iteratively refining the distribution parameters via Bayesian updates. Starting from an initial parameter $\boldsymbol{\theta}_0$ (e.g., a prior at initial noise level t_0), the model performs N iterations of the following procedure for $i = 1, \ldots, N$:

1. **Sample from output distribution:** An intermediate sample $\mathbf{m}'_i$ is drawn from the output distribution $\mathbf{m}'_i \sim p_{\mathsf{O}}(\cdot \mid \boldsymbol{\theta}_{i-1}, t_i)$ given the current parameter $\boldsymbol{\theta}_{i-1}$ and scheduled noise level t_i
2. **Sample from sender distribution:** A noisy observation $\mathbf{y}_i$ is then drawn from the sender distribution, conditional on $\mathbf{m}'_i$, via $\mathbf{y}'_i \sim p_{\mathsf{S}}(\cdot \mid \mathbf{m}'_i, \alpha_i)$, where α_i is the accuracy (inverse noise variance) prescribed for step i.
3. **Bayesian parameter update:** The distribution parameter is updated by incorporating the observation $\mathbf{y}_i$ through the Bayesian update function: $\boldsymbol{\theta}_i = h(\boldsymbol{\theta}_{i-1}, \mathbf{y}_i, \alpha_i)$. Here $h(\boldsymbol{\theta}_{i-1}, \mathbf{y}_i, \alpha_i)$ computes the posterior parameter after observing y_i with precision α_i, given the prior $\boldsymbol{\theta}_{i-1}$.

Repeating the above steps yields a final parameter $\boldsymbol{\theta}_N$ after N iterations. This $\boldsymbol{\theta}_N$ characterizes a highly concentrated distribution (approximately a Dirac delta distribution in the limit of large N), from which the final data sample can be obtained by drawing $\mathbf{m} \sim p_{\mathsf{O}}(\cdot \mid \theta_N, t_N)$. Importantly, the receiver distribution p_{R} is not explicitly used during generation – its effect is implicitly achieved by the two-step sampling (p_{O} followed by p_{S}) at each iteration. This ensures that sampling relies solely on the forward update pattern $\mathbf{m}'_i \to \mathbf{y}_i \to \boldsymbol{\theta}_i$ described above, in line with the canonical BFN [18] formulation. A simplified schematic illustration of this is provided in Figure 7.

B Score-Driven Sampling in DDPM vs. BFN: A Rigorous Comparison

B.1 Continuous Variable Modeling

Diffusion based sampling

$$
\begin{aligned}
\mathbf{x}_{t-1} &= \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0 + \sigma_t \mathbf{z} \\
&= \frac{\sqrt{\alpha_t}\,(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\,(1-\alpha_t)}{1-\bar{\alpha}_t}\mathbf{x}_0 + \sigma_t \mathbf{z} \\
&= \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\nabla_{\mathbf{x}_t} \log p\left(\mathbf{x}_t\right) + \sigma_t \mathbf{z} \\
\text{guidance} \to \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t &+ \frac{1-\alpha_t}{\sqrt{\alpha_t}}\nabla_{\mathbf{x}_t} \log p\left(\mathbf{x}_t \mid y\right) + \sigma_t \mathbf{z} \qquad (17) \\
&= \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\nabla_{\mathbf{x}_t} \log p\left(\mathbf{x}_t\right) + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\nabla_{\mathbf{x}_t} \log p\left(y \mid \mathbf{x}_t\right) + \sigma_t \mathbf{z} \qquad (18) \\
&\simeq \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\nabla_{\mathbf{x}_t} \log p\left(\mathbf{x}_t\right) + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\nabla_{\mathbf{x}_t} \log p\left(y \mid \mathbf{x}_0\right) + \sigma_t \mathbf{z} \qquad (19)
\end{aligned}
$$

BFN based sampling

$$\boldsymbol{\theta}_i = \frac{\rho_{i-1}}{\rho_i}\boldsymbol{\theta}_{i-1} + \frac{\alpha_i}{\rho_i}\mathbf{y} \tag{20}$$

$$= \frac{\rho_{i-1}}{\rho_i}\boldsymbol{\theta}_{i-1} + \frac{\alpha}{\rho_i}\mathbf{x}_0 + \frac{1}{\rho_i}\nabla_{\mathbf{x}_0}\log p(\mathbf{x}_0) \tag{21}$$

$$\text{guidance} \rightarrow \frac{\rho_{i-1}}{\rho_i}\boldsymbol{\theta}_{i-1} + \frac{\alpha}{\rho_i}\mathbf{x}_0 + \frac{1}{\rho_i}\nabla_{\mathbf{x}_0}\log p(\mathbf{x}_0 \mid y) \tag{22}$$

$$= \frac{\rho_{i-1}}{\rho_i}\boldsymbol{\theta}_{i-1} + \frac{\alpha}{\rho_i}\mathbf{x}_0 + \frac{1}{\rho_i}\nabla_{\mathbf{x}_0}\log p(\mathbf{x}_0) + \frac{1}{\rho_i}\nabla_{\mathbf{x}_0}\log p(y \mid \mathbf{x}_0) \tag{23}$$

$$= \frac{\rho_{i-1}}{\rho_i}\boldsymbol{\theta}_{i-1} + \frac{\alpha_i}{\rho_i}\mathbf{y} + \frac{1}{\rho_i}\nabla_{\mathbf{x}_0}\log p(y \mid \mathbf{x}_0) \tag{24}$$

B.2 Categorical Variable Modeling

Diffusion based sampling

$$\mathbf{x}_{t-1} \sim \mathcal{C}\left(\mathbf{x}_{t-1} \mid \boldsymbol{\theta}_{\text{post}}(\mathbf{x}_t, \hat{\mathbf{x}}_0)\right) \tag{25}$$

$$\text{, where } \boldsymbol{\theta}_{\text{post}}(\mathbf{x}_t, \hat{\mathbf{x}}_0) = \frac{\left[\alpha_t\mathbf{x}_t + \frac{1-\alpha_t}{K}\right] \odot \left[\bar{\alpha}_{t-1}\hat{\mathbf{x}}_0 + \frac{1-\bar{\alpha}_{t-1}}{K}\right]}{\sum_{k=1}^{K}\left(\left[\alpha_t\mathbf{x}_t + \frac{1-\alpha_t}{K}\right] \odot \left[\bar{\alpha}_{t-1}\hat{\mathbf{x}}_0 + \frac{1-\bar{\alpha}_{t-1}}{K}\right]\right)_k} \tag{26}$$

Here, $\odot$ denotes element-wise multiplication, ensuring the probabilities are normalized over all categories. And, $\mathcal{C}$ denotes categorical distribution to model discrete atom types $\mathbf{v}$.

BFN based Sampling

$$\boldsymbol{\theta}_i = \text{Softmax}(e^{\mathbf{y}} \cdot \boldsymbol{\theta}_{i-1}) \tag{27}$$

$$= \text{Softmax}(e^{\alpha(K\cdot\mathbf{e}_\mathbf{x}-\mathbf{1})+\nabla_{\mathbf{e}_\mathbf{x}}\log p(\mathbf{e}_\mathbf{x})} \cdot \boldsymbol{\theta}_{i-1}) \tag{28}$$

$$\text{guidance} \rightarrow \text{Softmax}(e^{\alpha(K\cdot\mathbf{e}_\mathbf{x}-\mathbf{1})+\nabla_{\mathbf{e}_\mathbf{x}}\log p(\mathbf{e}_\mathbf{x}|1)} \cdot \boldsymbol{\theta}_{i-1}) \tag{29}$$

$$= \text{Softmax}\left(e^{\mathbf{y}} \cdot \boldsymbol{\theta}_{i-1}^{\mathbf{v}} \cdot \ e^{\nabla_{\mathbf{e}_\mathbf{x}}\log p(1|\mathbf{e}_\mathbf{x})}\right) \tag{30}$$

B.3 Theoretical Distinctions between Diffusion Models and Bayesian Flow Networks

Bayesian Flow Networks (BFN) differ fundamentally from Diffusion Models (DMs) in their mechanism of uncertainty injection and the domain of parameter updates. Unlike diffusion-based generative processes, which explicitly manipulate the data sample x and inject Gaussian noise at each step to maintain stochasticity, BFN iteratively update distribution parameters θ through Bayesian inference, integrating noisy observations y_i drawn from a sender distribution $p_S(y \mid x; \alpha_i)$.

In contrast to diffusion methods, BFN have no explicit forward noise injection process. Instead, the generation is achieved by a deterministic parameter update conditional upon the observed

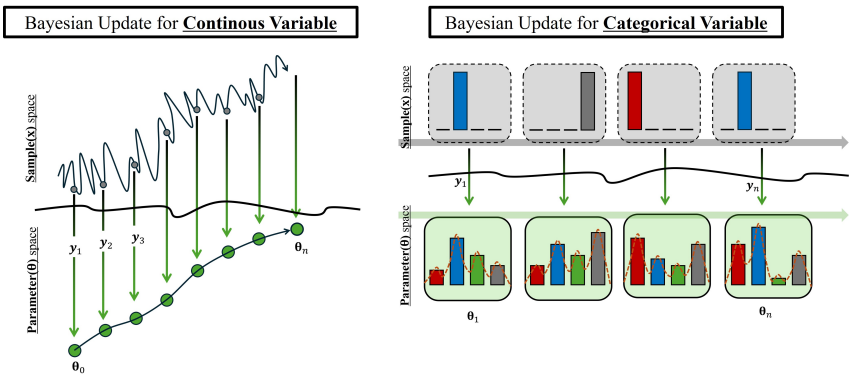

Figure 7: Schematic illustration of Bayesian updates for each variable type

sample y_i, implicitly encoding uncertainty through the random sampling of y_i. Specifically, for continuous variables, the sender distribution $p_{\mathtt{S}}(y_i \mid x; \alpha_i)$ is typically Gaussian, with precision (inverse variance) controlled by α_i. A small value of α_i implies high noise, rendering the observed value y_i nearly independent of the true data x, whereas a large α_i indicates a highly informative (low-noise) observation. Formally, the BFN update can be expressed as:

$$\theta_i = \left(1 - \frac{\alpha_i}{\rho_i}\right)\theta_{i-1} + \frac{\alpha_i}{\rho_i} y_i, \quad \text{where} \quad \rho_i = \rho_{i-1} + \alpha_i. \tag{31}$$

This update rule can be directly interpreted through the lens of Kalman filtering for scalar Gaussian models. In this analogy, each update of θ_i is drawn toward the observed sample y_i proportionally to the precision parameter α_i, thus progressively reducing uncertainty as ρ_i increases. Given the sampled y_i, the Bayesian update function remains strictly deterministic, and no additional random noise is explicitly introduced beyond the implicit stochasticity already present in y_i.

This principle of implicit uncertainty introduction also extends naturally to categorical variables. Suppose the parameter vector θ_i represents a categorical probability distribution over K classes, i.e., $\theta_i = (p_i^{(1)}, p_i^{(2)}, \ldots, p_i^{(K)})$. In this case, the sender distribution can be viewed as emitting symbols y_i according to a confusion matrix parameterized by α_i. Specifically, we have:

$$p_{\mathtt{S}}(y_i = c \mid x = k; \alpha_i) \approx \begin{cases} \frac{1}{K}, & \text{when } \alpha_i = 0 \quad (\text{uninformative observation}), \\ \mathbf{1}_{c=k}, & \text{as } \alpha_i \to \infty \quad (\text{perfectly informative}). \end{cases} \tag{32}$$

Upon observing the symbol $y_i = c$, the Bayesian update for the parameter θ_i follows:

$$\theta_{i+1}(j) \propto \theta_i(j) \cdot \Pr_{\rho_i}[Y = c \mid X = j]. \tag{33}$$

Although y_i is originally sampled as a continuous vector from a Gaussian distribution centered around a one-hot encoding of the true class, the Bayesian update interpretation treats y_i as if it were decoded into a discrete class index $c = \arg\max_k y_k$. For rigorous derivation and proof of this decoding interpretation, we refer the reader to Graves et al. [18]. The resulting posterior update normalizes probabilities over all classes j, illustrating explicitly how the sampled observation y_i reweights prior probabilities $\theta_i(j)$ based on the likelihood of observing y_i under each class hypothesis $X = j$. Thus, the sampled variable y_i serves directly as an informative mediator driving the parameter dynamics.

Unlike the arbitrary Gaussian noise z used in diffusion processes, the sampled variable y_i in BFN is inherently linked to the unknown true data x through the sender distribution $p_{\mathtt{S}}(y_i \mid x)$. As the inference procedure progresses (i.e., as i increases and noise decreases), y_i provides increasingly precise information, causing the parameter θ_i to converge toward a distribution sharply concentrated around the true data. This iterative Bayesian updating mechanism effectively transfers information from the sample space into the parameter space.

In summary, two critical distinctions between BFN and diffusion-based generative methods emerge from our theoretical analysis:

1. **Implicit versus Explicit Noise Injection:** BFN deterministically update parameters given noisy observations, implicitly capturing uncertainty. In contrast, diffusion models explicitly add random noise to samples at each step to maintain stochasticity.
2. **Parameter-space versus Sample-space Updates:** Diffusion models perform both training and inference entirely within the sample space. Conversely, BFN operate fundamentally within the parameter space, integrating information from the sample space through occasional noisy observations, thus creating a natural integration of information between the two spaces.

C Related Work

C.1 Molecule Generation in Structure-based Drug Design

With the rapid accumulation of protein structural data, generative methods for molecule design have become increasingly important in structure-based drug discovery. Initial approaches, such as [44], employed sequence-based generative models to produce SMILES representations informed by protein binding sites. However, the advent of powerful geometric and 3D modeling methods has shifted the paradigm toward directly constructing molecules within three-dimensional spaces. For

example, [38] represented molecules using voxelized atomic density grids and utilized Variational Autoencoders (VAEs) for molecular synthesis. Meanwhile, [31], [34], and [36] introduced sequential, autoregressive approaches for placing atoms or functional groups step-by-step into target binding pockets. Building upon these autoregressive techniques, fragment-based strategies, as seen in FLAG [49] and DrugGPS [48], integrated chemically meaningful fragments, thereby improving the structural realism of generated ligands.

In parallel, diffusion-based generative methods have emerged, achieving remarkable success across various generative tasks such as image and text synthesis. Adaptations of these models in molecular contexts, exemplified by recent studies [20, 39, 25, 21], iteratively refine atom identities and coordinates, leveraging symmetry-preserving architectures such as SE(3)-equivariant neural networks to ensure chemical validity and structural accuracy.

Despite substantial advancements, current generative frameworks frequently encounter difficulties in simultaneously optimizing multiple pharmacologically relevant properties, including binding affinity, synthetic accessibility, and low toxicity. In practical drug development scenarios, the simultaneous control and optimization of these attributes are typically mandatory requirements rather than optional criteria [9]. Thus, developing generative strategies capable of effectively incorporating multiple property constraints remains a critical challenge in the field.

C.2 Optimization based Molecule Generation in Structure-based Drug Design

Recent research in molecular generative modeling has moved beyond approximating training data distributions toward explicit optimization strategies aimed at producing molecules with desirable properties, such as high target protein binding affinity and synthetic accessibility (SA).

For example, RGA [16] employs genetic algorithms specifically tailored to structure-based drug design (SBDD), explicitly incorporating target protein structures into molecular optimization. DecompOpt [51] combines a pre-trained, structure-aware equivariant diffusion model to initially identify suitable molecular substructures complementary to target binding pockets, followed by a docking-based greedy iterative optimization to enhance binding affinity. TacoGFN [43] leverages generative flow networks (G-FlowNets) to identify pharmacophoric interactions with target proteins, subsequently utilizing reinforcement learning to optimize binding affinity and synthetic accessibility. Additionally, ALIDiff [19] introduces a preference-based optimization method that aligns pre-trained generative models to specified molecular properties through fine-tuning.

Despite their demonstrated effectiveness, these approaches share an intrinsic limitation: since optimization methods are tightly integrated into their training processes, adapting the models to new property requirements inevitably necessitates retraining. In contrast, our proposed approach distinguishes itself by leveraging a pre-trained generative model, enabling effective multi-property optimization directly through modifications at sampling time, thus avoiding costly retraining and enhancing flexibility in targeting diverse pharmacologically relevant properties.

C.3 Bayesian Flow Network

Recently, Bayesian Flow Network (BFN) have gained attention as effective models for protein sequence modeling [2] and molecular structure generation [46, 37], demonstrating promising capabilities particularly in the generation of realistic three-dimensional molecular structures. Despite these advancements, the development of controllable generation methods within BFN, aimed at optimizing diverse molecular properties required for viable drug candidates, remains largely unexplored. Thus, establishing a theoretical link between generative models based on maximum likelihood estimation, such as diffusion models and BFN, and subsequently formulating gradient-guidance strategies within the BFN framework, represents an essential step forward. Such advancements could significantly inform future directions in the field of structure-based drug design.

D Implement Detail

D.1 Predictor with Bayesian Neural Network for Uncertainty

To achieve property-driven generation without retraining the generative model, we introduce an external Bayesian neural network (BNN) predictor modeling the conditional distribution $p(\mathtt{l} \mid \mathbf{m}, \mathbf{p})$ as a Gaussian distribution $\mathcal{N}\big(\mathtt{l}; \boldsymbol{\mu}(\mathbf{m}, \mathbf{p}), \boldsymbol{\sigma}(\mathbf{m}, \mathbf{p})^2\big)$ with mean $\boldsymbol{\mu}_\vartheta(\mathbf{m}, \mathbf{p})$ and variance $\boldsymbol{\sigma}_\vartheta(\mathbf{m}, \mathbf{p})^2$. Unlike point estimates, this BNN provides uncertainty-aware predictions for properties (e.g., binding affinity), enabling informed guidance. During molecule generation, the gradient $\nabla_{\mathbf{m}} \log p(\mathtt{l} \mid \mathbf{m}, \mathbf{p})$,

which incorporates predictive uncertainty, guides the generative process. This uncertainty integration allows the model to appropriately moderate its guidance, preventing overly confident predictions toward regions unsupported by the predictor.

We approximate the overall predicted label distribution as Gaussian with predictive mean and variance computed from these samples. In particular, the mean is $\hat{\boldsymbol{\mu}}_\vartheta(\mathbf{x},\mathbf{p}) = M^{-1}\sum_{i=1}^{M}\boldsymbol{\mu}_{\vartheta,i}(\mathbf{x},\mathbf{p})$, and the predictive variance $\hat{\boldsymbol{\sigma}}^2_\vartheta(\mathbf{x},\mathbf{p}) = M^{-1}\sum_i\left(\boldsymbol{\sigma}^2_{\vartheta,i}(\mathbf{x}) + \boldsymbol{\mu}_i^2(\mathbf{x})\right) - \boldsymbol{\mu}^2_{\vartheta,i}(\mathbf{x})$ combines the average of the BNN's output variances with the variance of its output means. Using the law of total variance, we decompose the predictive uncertainty into aleatoric and epistemic components:

$$\begin{aligned}\boldsymbol{\sigma}^2_\vartheta(\mathbf{m},\mathbf{p}) &= M^{-1}\sum_i \boldsymbol{\sigma}^2_{\vartheta,i}(\mathbf{m},\mathbf{p}) + M^{-1}\sum_i \boldsymbol{\mu}_i^2(\mathbf{m},\mathbf{p}) - \boldsymbol{\mu}^2_{\vartheta,i}(\mathbf{m},\mathbf{p}) \\ &= \mathbb{E}\left[\boldsymbol{\sigma}^2_\vartheta(\mathbf{m},\mathbf{p})\right] + \mathbb{E}\left[\boldsymbol{\mu}^2(\mathbf{m},\mathbf{p})\right] - \mathbb{E}\left[\boldsymbol{\mu}_\vartheta(\mathbf{m},\mathbf{p})\right]^2 \\ &= \underbrace{\mathbb{E}\left[\boldsymbol{\sigma}^2_\vartheta(\mathbf{m},\mathbf{p})\right]}_{\text{Aleatoric Uncertainty}} + \underbrace{\mathrm{Var}\left[\boldsymbol{\mu}(\mathbf{m},\mathbf{p})\right]}_{\text{Epistemic Uncertainty}}\end{aligned} \tag{34}$$

where ϑ represents the BNN's parameters (random due to the weight posterior). The first term $\boldsymbol{\sigma}^2_{\text{aleatoric}}$ is the expected predictive variance (reflecting inherent noise or irreducible uncertainty in the property given $(\mathbf{x},\mathbf{p})$), while the second term $\boldsymbol{\sigma}^2_{\text{epistemic}}$ is the variance of the predicted means (reflecting uncertainty in the model parameters due to limited training data). Our CByG can thus modulate the influence of the guidance signal in proportion to the predictor's confidence, improving robust controllability.

We train the property predictor on an external dataset of protein–ligand complexes, allowing it to learn a mapping from 3D structures to property values independent of the generative model. In particular, we use the CrossDocked2020 dataset [15] to train the predictor. Each training sample provides a protein structure $\mathbf{p}$, a ligand $\mathbf{m}$, and a ground-truth label $\mathbf{l}$ (e.g., an experimental or docking-derived affinity score). We optimize the BNN by maximizing the likelihood of the true labels under its predicted Gaussian distribution. The negative log-likelihood (NLL) loss for a single sample is given by (For notational simplicity, we omit $\mathbf{p}$):

$$\mathcal{L}_{\text{NLL}}\left(\mathbf{y}_n,\mathbf{x}_n\right) = \frac{\log\boldsymbol{\sigma}^2_\vartheta\left(\mathbf{x}_n\right)}{2} + \frac{\left(\boldsymbol{\mu}_\vartheta\left(\mathbf{x}_n\right) - \mathbf{y}_n\right)^2}{2\boldsymbol{\sigma}^2_\vartheta\left(\mathbf{x}_n\right)}, \tag{35}$$

where we omit the constant $\frac{1}{2}\log(2\pi)$ for brevity. In addition to the standard NLL, we also employ a β-weighted NLL variant (denoted β-NLL) to improve training stability in the presence of heteroscedastic uncertainty [41], as follow:

$$\mathcal{L}_{\beta-\text{NLL}}\left(\mathbf{y}_n,\mathbf{x}_n\right) = \text{stop}\left(\sigma^{2\beta}\right)\mathcal{L}_{\text{NLL}}\left(\mathbf{y}_n,\mathbf{x}_n\right), \tag{36}$$

for some hyperparameter $\beta > 0$. Setting $\beta = 0$ recovers the original NLL loss, while a positive β increases the relative penalty for large predicted variances.

D.2 Implement detail

The generative backbone architecture employed in our proposed CByG framework directly inherits the architecture and pretrained weights from MolCRAFT [37]. For the property predictor backbone, we adapted the graph transformer architecture from TargetDiff [20] by removing the equivariant head, resulting in a novel SE(3)-invariant graph transformer. This design choice naturally aligns with the inherent symmetry of protein-ligand complexes, where binding affinity and synthetic accessibility scores remain invariant to rotations and translations.

Specifically, the property predictor comprises 16 attention heads, with each attention block consisting of three SE(3)-invariant layers featuring a hidden dimension of 64. Key and value embeddings are generated through a two-layer MLP. Layer normalization is uniformly applied throughout the network, and the Swish activation function is adopted. The learning rate is exponentially decayed by a factor of 0.6, with a lower bound set at 1×10^{-6}. This decay is triggered if no improvement is observed in the validation loss for ten consecutive evaluations, which occur every 1000 training steps. Finally, the property predictor's scoring function is defined as $\text{Score} = \left(\frac{\text{DS}}{-20} \times \text{SA}\right)$, where DS means docking score and SA means synthetic accessibility score.

D.3 Sampling

Algorithm 1 Sampling Procedure CBYG

Input:
Protein pocket $\mathbf{p}$,
Pre-trained `output` network:
  $p_{\mathtt{0}}(\mathbf{m} \mid \boldsymbol{\theta}_{i-1}, \mathbf{p}; t) \leftarrow \Psi_{\mathtt{output}}(\boldsymbol{\theta}_{i-1}, \mathbf{p}, t_i)$,
Property predictor network providing property predictions and uncertainty:
  $p_{\mathtt{1}}(\mathtt{1} \mid \mathbf{m}, \mathbf{p}) \leftarrow \mathcal{N}\left(\mathtt{1}; \boldsymbol{\mu}_\vartheta(\mathbf{m}, \mathbf{p}), \boldsymbol{\sigma}_\vartheta(\mathbf{m}, \mathbf{p})^2\right)$,
Pre-defined precision schedule for coordinate and type $(\alpha_{\mathbf{x},i}, \alpha_{\mathbf{v},i})$ according to [18],
Guidance scale for coordinate and type $\leftarrow \lambda_{\mathbf{x}}, \lambda_{\mathbf{v}}$
$\boldsymbol{\theta}_0^{\mathbf{x}}, \boldsymbol{\theta}_0^{\mathbf{v}} \leftarrow \mathbf{0}, \left[\frac{1}{K}\right]_{\mathbb{N}_{\mathrm{M}} \times K}$
for $i = 1$ to N **do**
  $t \leftarrow \frac{i-1}{n}$
  $\mathbf{m} : [\hat{\mathbf{x}}, \hat{\mathbf{v}}] \sim p_{\mathtt{0}}(\mathbf{m} \mid \boldsymbol{\theta}_{i-1}, \mathbf{p}; t)$
  $\mathbf{y}_{\mathbf{x},i} \sim p_{\mathtt{s}}(\mathbf{y}_i \mid \hat{\mathbf{x}}; \alpha_{\mathbf{x},i})$
  $\mathbf{y}_{\mathbf{v},i} \sim p_{\mathtt{s}}(\mathbf{y}_i \mid \hat{\mathbf{v}}; \alpha_{\mathbf{v},i})$
  $\boldsymbol{\mu}_\vartheta, \boldsymbol{\sigma}_\vartheta^2 \leftarrow p_{\mathtt{1}}(\mathtt{1} \mid \mathbf{m}, \mathbf{p})$ ▷ Mean and uncertainty from property predictor
  $\boldsymbol{\theta}_i^{\mathbf{x}} \leftarrow \frac{\alpha_i}{\rho_i}\mathbf{y}_{\mathbf{x},i} + \frac{\rho_{i-1}}{\rho_i}\boldsymbol{\theta}_{i-1}^{\mathbf{x}} + \boldsymbol{\sigma}_\vartheta^2 \cdot \lambda_{\mathbf{x}} \cdot \frac{1}{\rho_i}\nabla_{\mathbf{x}} \log p_{\mathtt{1}}(\mathtt{1} \mid [\hat{\mathbf{x}}, \hat{\mathbf{v}}], \mathbf{p})$
  $\boldsymbol{\theta}_i^{\mathbf{v}} \leftarrow \mathrm{Softmax}\left(e^{\mathbf{y}_{\mathbf{v},i}} \cdot \boldsymbol{\theta}_{i-1}^{\mathbf{v}} \cdot e^{\mathbf{h}}\right)$
    where $\mathbf{h} = \boldsymbol{\sigma}_\vartheta^2 \cdot \lambda_{\mathbf{v}} \cdot \nabla_{\mathbf{e}_{\mathbf{v}}} \log p_{\mathtt{1}}(\mathtt{1} \mid [\hat{\mathbf{x}}, \hat{\mathbf{v}}], \mathbf{p}), \mathbf{e}_{\mathbf{v}} = \mathrm{GumbelSoftmax}(\hat{\mathbf{v}})$
end for
$\mathbf{m} \sim p_{\mathtt{0}}(\mathbf{m} \mid \boldsymbol{\theta}_N, \mathbf{p}; t)$
return $[\hat{\mathbf{x}}, \hat{\mathbf{v}}]$

D.4 SE(3)-Equivariance

Since our proposed guidance injection strategy is integrated into an SE(3)-equivariant generative model, the designed guidance itself must inherently preserve the SE(3)-equivariance property. As described by [40], aligning the protein-ligand complex to center the pocket at the origin removes translation equivariance, requiring only O(3)-equivariance. The following provides a formal proof verifying this property.

Proposition D.1. *Suppose the property preditor* $\Psi_{\mathtt{prop}}([\mathbf{x}_{\mathrm{M}}, \mathbf{v}_{\mathrm{M}}], [\mathbf{x}_{\mathrm{P}}, \mathbf{v}_{\mathrm{P}}]) \leftarrow p_{\mathtt{1}}(\mathtt{1} \mid \mathbf{m}, \mathbf{p})$ *is invariant such that* $\Psi_{\mathtt{prop}}([\mathbf{x}_{\mathrm{M}}, \mathbf{v}_{\mathrm{M}}], [\mathbf{x}_{\mathrm{P}}, \mathbf{v}_{\mathrm{P}}]) = \Psi_{\mathtt{prop}}([T_g(\mathbf{x}_{\mathrm{M}}), \mathbf{v}_{\mathrm{M}}], [T_g(\mathbf{x}_{\mathrm{P}}), \mathbf{v}_{\mathrm{P}}])$. *Denoting* T_g *as the group of O(3)-transformation,* $T_g(\mathbf{x}) = \mathbf{R}\mathbf{x}$, *where* $\mathbf{R} \in \mathbb{R}^{3\times3}$ *is the rotation matrix, and* $\mathbf{b} \in \mathbb{R}^3$ *is the translation vector. Then, gradient guidance function is orthogonal equivariant such that* $\nabla_{\mathbf{x}_{\mathrm{M}}}\Psi_{\mathtt{prop}}(T_g(\mathbf{x}_{\mathrm{M}}), T_g(\mathbf{x}_{\mathrm{P}})) = T_g(\nabla_{\mathbf{x}_{\mathrm{M}}}\Psi_{\mathtt{prop}}(\mathbf{x}_{\mathrm{M}}, \mathbf{x}_{\mathrm{P}}))$. *Variables corresponding to type* $\mathbf{v}$ *are omitted from the notation, as they remain unaffected by O(3) transformations.*

Proof. Given the invariance of the property predictor $\Psi_{\mathtt{prop}}(\mathbf{x}_{\mathrm{M}}, \mathbf{x}_{\mathrm{P}})$ under O(3) transformations, it follows that $\Psi_{\mathtt{prop}}(\mathbf{R}\mathbf{x}_{\mathrm{M}}, \mathbf{R}\mathbf{x}_{\mathrm{P}}) = \Psi_{\mathtt{prop}}(\mathbf{x}_{\mathrm{M}}, \mathbf{x}_{\mathrm{P}})$. Differentiating both sides of the equation with respect to $\mathbf{x}_{\mathrm{M}}$ yields:

$$\Psi_{\mathtt{prop}}(\mathbf{x}_{\mathrm{M}}, \mathbf{x}_{\mathrm{P}}) = \Psi_{\mathtt{prop}}(\mathbf{R}\mathbf{x}_{\mathrm{M}}, \mathbf{R}\mathbf{x}_{\mathrm{P}}) \tag{37}$$

$$\begin{aligned}\nabla_{\mathbf{x}_{\mathrm{M}}}\Psi_{\mathtt{prop}}(\mathbf{x}_{\mathrm{M}}, \mathbf{x}_{\mathrm{P}}) &= \nabla_{\mathbf{x}_{\mathrm{M}}}\Psi_{\mathtt{prop}}(\mathbf{R}\mathbf{x}_{\mathrm{M}}, \mathbf{R}\mathbf{x}_{\mathrm{P}}) \\ &= \left(\frac{\partial(\mathbf{R}\mathbf{x}_{\mathrm{M}})}{\partial\mathbf{x}_{\mathrm{M}}}\right)^\top \nabla_{\mathbf{x}_{\mathrm{M}}}\Psi_{\mathtt{prop}}(\mathbf{R}\mathbf{x}_{\mathrm{M}}, \mathbf{R}\mathbf{x}_{\mathrm{P}}) \\ &= \mathbf{R}^\top\nabla_{\mathbf{x}_{\mathrm{M}}}\Psi_{\mathtt{prop}}(\mathbf{R}\mathbf{x}_{\mathrm{M}}, \mathbf{R}\mathbf{x}_{\mathrm{P}}) \end{aligned} \tag{38}$$

$$\begin{aligned}\mathbf{R}\nabla_{\mathbf{x}_{\mathrm{M}}}\Psi_{\mathtt{prop}}(\mathbf{x}_{\mathrm{M}}, \mathbf{x}_{\mathrm{P}}) &= \mathbf{R}\mathbf{R}^\top\nabla_{\mathbf{x}_{\mathrm{M}}}\Psi_{\mathtt{prop}}(\mathbf{R}\mathbf{x}_{\mathrm{M}}, \mathbf{R}\mathbf{x}_{\mathrm{P}}) \\ &= \nabla_{\mathbf{x}_{\mathrm{M}}}\Psi_{\mathtt{prop}}(\mathbf{R}\mathbf{x}_{\mathrm{M}}, \mathbf{R}\mathbf{x}_{\mathrm{P}}) \end{aligned} \tag{39}$$

Therefore, $\nabla_{\mathbf{x}_{\mathrm{M}}}\Psi_{\mathtt{prop}}(\cdot)$ exhibits equivariance under the transformation $T_g(\cdot)$, completing the proof. □

E *TargetOpt* Implement Detail

To evaluate the effectiveness of our proposed guidance injection within the Bayesian Flow Network framework, we implemented a comparable guidance method within a diffusion-based generative

model. Specifically, we adopted the TargetDiff [20] architecture, enhancing it with gradient-based guidance strategies. Two distinct gradient guidance approaches were considered: (1) gradient computation based on property prediction at arbitrary intermediate time steps $\mathbf{x}_t$, and (2) gradient computation via posterior sampling, leveraging property prediction at the fully denoised state $\hat{\mathbf{x}}_0$. For fair comparison, the model utilized for property prediction at $\hat{\mathbf{x}}_0$ was identical to that employed in our CBYG model. Below, we present the unconditional denoising procedures used by TargetDiff for each variable type; additional details can be found in the original paper [20].

$$q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)=\mathcal{N}\left(\mathbf{x}_{t-1} ; \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right), \tilde{\beta}_t \mathbf{I}\right), \quad q\left(\mathbf{v}_{t-1} \mid \mathbf{v}_t, \mathbf{v}_0\right)=\mathcal{C}\left(\mathbf{v}_{t-1} \mid \tilde{\boldsymbol{c}}_t\left(\mathbf{v}_t, \mathbf{v}_0\right)\right) \tag{40}$$

, where $\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right)=\frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1-\bar{\alpha}_t} \mathbf{x}_0+\frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t} \mathbf{x}_t, \tilde{\beta}_t=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t$, and $\tilde{\boldsymbol{c}}_t\left(\mathbf{v}_t, \mathbf{v}_0\right)=\boldsymbol{c}^* / \sum_{k=1}^K c_k^*$ and $\boldsymbol{c}^*\left(\mathbf{v}_t, \mathbf{v}_0\right)=\left[\alpha_t \mathbf{v}_t+\left(1-\alpha_t\right) / K\right] \odot\left[\bar{\alpha}_{t-1} \mathbf{v}_0+\left(1-\bar{\alpha}_{t-1}\right) / K\right]$.
Under the scenario of property prediction at arbitrary intermediate time steps $\mathbf{x}_t$, the update procedure for atomic coordinate and type can be modified as follows.

$$\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right)=\frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1-\bar{\alpha}_t} \mathbf{x}_0+\frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t} \mathbf{x}_t+\frac{1-\alpha_t}{\sqrt{\alpha_t}} \nabla_{\mathbf{x}_t} \log p\left(\mathtt{1} \mid \boldsymbol{x}_t, \mathbf{p}\right) \tag{41}$$

$$\tilde{\boldsymbol{c}}_t\left(\mathbf{v}_t, \mathbf{v}_0\right)=\left(\frac{\boldsymbol{c}^*}{\sum_{k=1}^K c_k^*}+\boldsymbol{\delta}\right) \cdot e^{\nabla_{\mathbf{v}_t} p\left(\mathtt{1} \mid \mathbf{v}_t, \mathbf{p}\right)} \tag{42}$$

Additionally, when employing posterior sampling for gradient computation, the gradient form can be reformulated analogous to DPS as follows.

$$\nabla_{\mathbf{x}_t} \log p\left(\mathtt{1} \mid \mathbf{x}_t\right) \simeq \nabla \log \mathbb{E}_{\mathbf{x}_0 \sim p\left(\mathbf{x}_0 \mid \mathbf{x}_t\right)}\left[p\left(y \mid \hat{\mathbf{x}}_0\right)\right] \tag{43}$$

$$\nabla_{\mathbf{v}_t} \log p\left(\mathtt{1} \mid \mathbf{v}_t\right) \simeq \nabla \log \mathbb{E}_{\mathbf{v}_0 \sim p\left(\mathbf{v}_0 \mid \mathbf{v}_t\right)}\left[p\left(y \mid \hat{\mathbf{v}}_0\right)\right] \tag{44}$$

F Baseline Model for Evaluation

For a fair comparison, we categorize baseline models into two groups: "Only Generation Type" and "Optimization based Generation Type". Movevoer, we selected state-of-the-art generative models exhibiting competitive performance as representative models for the "Only Generation Type" category.

Only Generation Type

- **AR** [31] leverages Markov chain Monte Carlo techniques to sequentially infer molecular structures from spatial atomic density representations.
- **Pocket2Mol** [34] incrementally synthesizes molecules through sequential prediction of atoms and associated bonds using an E(3)-equivariant architecture, selectively expanding frontier atoms to significantly improve sampling efficiency.
- **TargetDiff** [20] enhances dual-modality diffusion methodologies, distinctly processing continuous and discrete modalities through parallel diffusion pipelines, demonstrating improved outcomes over purely continuous formulations such as DiffSBDD.
- **DecompDiff** [21] adopts molecular decomposition techniques, separating the molecular structure into functional arms and connective scaffolds, thereby integrating chemically-informed priors within diffusion-based generative mechanisms.
- **MolCRAFT** [37] exploits Bayesian Flow Networks coupled with sophisticated sampling schemes, showing significant enhancements relative to contemporary diffusion-based methodologies. *We directly utilized the publicly available molecules provided through their respective GitHub repositories for our experiments.*

Optimization based Generation Type

- **RGA** [16] extends the evolutionary optimization approach of AutoGrow4 by integrating a reinforcement learning-based policy conditioned on target binding pockets, effectively constraining exploratory randomness during molecular search procedures. *Since the molecules generated by these models for CrossDocked2020 benchmark are not publicly available on GitHub, we trained these models ourselves to produce molecules for evaluation.*

- **DecompOpt** [51] employs conditional generative modeling of chemically meaningful fragments aligned with receptor sites, iteratively optimizing molecular generation via guided resampling within a structured diffusion latent space, informed by fragment-based oracle rankings. *Since the molecules generated by these models for CrossDocked2020 benchmark are not publicly available on GitHub, we trained these models ourselves to produce molecules for evaluation.*
- **TacoGFN** [42] leverages a G-FlowNet-based autoregressive approach, incrementally assembling molecular structures fragment-by-fragment while identifying key pharmacophoric interactions with target proteins. It integrates a reward-based optimization mechanism, simultaneously promoting advantageous properties such as binding affinity and synthetic accessibility (SA score) throughout the generative process. *We directly utilized the publicly available molecules provided through their respective GitHub repositories for our experiments.*
- **ALIDiff** [19] is an SE(3)-equivariant diffusion generative model that incorporates recent reinforcement learning from human feedback techniques, leveraging Energy Preference Optimization to effectively generate molecules exhibiting superior properties such as enhanced binding affinity and other desirable attributes. *We directly utilized the publicly available molecules provided through their respective GitHub repositories for our experiments.*
- **TargetOpt** is a diffusion-based generative model developed specifically in this study to enable gradient-based guidance propagation, serving as a comparative baseline for evaluating the effectiveness of guidance mechanisms between BFN and diffusion-based frameworks. Although several studies have adopted similar optimization strategies, we exclude these approaches from consideration, as they either exclusively optimize for binding affinity or neglect guidance on categorical atom types.

G Additional Experiment Metrics

G.1 Experimental Setup of Section 6.2

We employ multiple metrics to comprehensively evaluate the binding affinity and intrinsic properties of the generated molecules. Specifically, we utilize SMINA, GNINA, and AutoDock Vina to independently calculate two distinct forms of binding affinity: (1) the intrinsic docking scores of the generated molecules themselves (denoted as Score.), and (2) the affinity scores obtained via re-docking procedures (denoted as Dock.). Additionally, we introduce the High Affinity metric, defined as the percentage of generated molecules exhibiting superior binding affinity compared to a given reference ligand for each target protein. To quantify the intrinsic molecular properties, we measure the Synthetic Accessibility (SA) and Diversity metrics. Finally, to capture the overall stability and validity of generated molecules (both intrinsically and in complex with the target protein) we utilize the PB-valid score from the PoseBusters benchmark. The metric PB-valid denotes the proportion of generated molecules considered valid under the PoseBusters benchmark, assuming that violation of any of its 17 evaluation criteria renders the molecule invalid.

G.2 Experimental Setup of Section 6.3

We quantitatively evaluate the synthetic feasibility of generated molecules using six key metrics provided by AiZynthFinder (Solved, Routes, Solved Routes, and Top Score). The reported values in Table 2 are averages computed across all generated molecules per model. Specifically, Solved indicates whether AiZynthFinder successfully identified at least one valid retrosynthetic route for a given molecule. A higher number of nodes indicates a more extensive exploration of possible reaction pathways by the algorithm, which may reflect the inherent complexity of the target molecule, diversity in applicable reaction templates, or increased search depth. While a large node count can imply a thorough and comprehensive search, it might also signal inefficiencies if numerous unproductive pathways were evaluated. Thus, this metric provides valuable insights into the balance between computational effort and search comprehensiveness.

Routes counts the total number of distinct retrosynthetic pathways (complete reaction sequences from purchasable precursors to the target molecule) identified by AiZynthFinder. This metric quantifies the diversity of retrosynthetic solutions identified by the algorithm. A higher number of identified routes suggests multiple viable synthetic strategies for the target molecule, providing chemists with alternative synthetic options. Nevertheless, not all identified routes may be equally practical or feasible; thus, this metric should be interpreted alongside complementary measures such as the number of solved routes or the top-scoring pathways.

Solved Routes is the subset of these routes comprising exclusively purchasable precursors listed in commercially available databases (e.g., ZINC), thus representing practically realizable synthetic pathways. This metric enables rapid assessment of synthetic feasibility given a predefined inventory of building blocks. Specifically, if at least one retrosynthetic pathway is successfully identified, the target molecule is considered synthesizable, making this a straightforward, high-level indicator of synthetic achievability. However, as it does not capture pathway quality or diversity, it should be interpreted in conjunction with complementary metrics.

Lastly, Top Score reflects the highest-ranked synthetic route as evaluated by AiZynthFinder's scoring function, which aggregates criteria such as precursor availability, reaction step count, and reaction feasibility (e.g., average template frequency). This metric quantitatively represents the quality of the highest-ranked synthetic route, assisting chemists in prioritizing retrosynthetic pathways for consideration. A higher score reflects routes with greater feasibility, efficiency, and desirability. This measure is particularly useful for comparing alternative pathways or selecting the most promising candidates for subsequent experimental validation.

H Additional Experiment result

H.1 Experimental Analysis of Section 6.2

Our proposed model consistently outperforms baseline methods across 12 evaluation metrics covering binding affinity and intrinsic molecular properties, as shown Table 1. In particular, our model significantly outperforms baseline methods in the 'Score' metric, which measures the binding affinity of generated molecules prior to any docking procedure. This result indicates that our model inherently generates molecules possessing high binding affinity, even without additional docking optimization. Similar superior performance is observed in the Synthetic Accessibility (SA) and PoseBusters validity (PB-Valid) metrics, indicating that gradients derived from binding affinity and SA scores are effectively propagated during the guidance-based sampling process.

Furthermore, we observe that different baseline models excel depending on the affinity evaluation tool used; specifically, ALIDiff outperforms DecompDiff when evaluated by SMINA, whereas DecompDiff achieves better results when GNINA is used. Considering the variability in performance across different evaluation metrics, the consistently strong performance of our proposed model across all three docking tools (Vina, SMINA, and GNINA) highlights its robustness and reliability in generating molecules with high binding affinity, as well as its generalizable efficacy across diverse evaluation standards. Additionally, our model uniquely exhibits minimal differences in binding affinity between pre-docking and post-docking evaluations. This minimal discrepancy indicates our model's ability to intrinsically predict stable and energetically favorable binding poses, explicitly capturing meaningful protein-ligand interactions.

Table 4 demonstrates the effectiveness of jointly applying gradient-based guidance to both atomic coordinates and atom types. As clearly indicated by the results, simultaneous guidance across both modalities consistently outperforms methods employing guidance on a single modality alone. This outcome aligns with fundamental chemical principles, as molecular properties and the corresponding energy landscape inherently depend upon intricate interactions between atomic types and their spatial configurations.

In addition to the results presented in Figure 5, we conducted supplementary experiments using the PoseCheck benchmark to further assess the generated molecules' structural stability and validity. Crucially, molecules evaluated in this additional experiment were directly sampled from generative models, without employing docking. This methodological choice ensures that the evaluation reflects the inherent capability of the generative models, rather than improvements arising from docking-based optimizations or adjustments by docking tools. Their reliance on external docking for generating final 3D conformations makes it unsuitable to accurately evaluate these models from the perspective of generating intrinsically stable 3D molecular structures. Consequently, models such as RGA and TacoGFN, which initially generate molecules as SMILES strings or 2D graphs and subsequently rely on docking software to derive the final 3D conformations, were excluded from this comparative analysis.

Experimental results (Table 5) indicate that the CByG model outperforms baseline models in terms of the "Clash" and "Strain Energy" metrics, whereas DecompOpt achieves the best performance in the "Intermolecular Interaction" metric. Considering that DecompOpt explicitly optimizes molecules by fixing protein-interacting fragments, this result aligns naturally with its optimization strategy.

Table 4: Summary of binding affinity and molecular properties of reference molecules and molecules generated by CBYG and baselines. (↑)/(↓) denotes whether a larger/smaller number is preferred. Top 2 results are bolded and underlined, respectively.

Methods	SMINA (↓)		GNINA (↓)		Vina (↓)		SA (↑)		PB-Valid (↑)	
	Score.	Dock.	Score.	Dock.	Score.	Dock.	Avg.	Med.	Avg.	Med.
Reference	-6.37	-7.92	-7.06	-7.61	-6.36	-7.45	0.73	0.74	95.0%	95.0%
CBYG	**-7.74**	**-9.61**	**-7.63**	**-8.33**	**-8.60**	**-9.16**	**0.84**	**0.87**	**94.9%**	**96.0%**
CBYG `w/o pos guidance`	-7.05	-8.60	-6.88	-7.42	-7.74	-8.12	0.78	0.79	90.4%	91.1%
CBYG `w/o type guidance`	-6.92	-8.45	-6.70	-7.20	-7.60	-7.95	0.76	0.77	88.3%	89.0%
CBYG `w/o uncertainty`	-7.25	-8.81	-7.10	-7.64	-7.90	-8.31	0.75	0.76	87.1%	87.7%
CBYG ($\lambda_x = 40, \lambda_v = 40$)	**-7.74**	**-9.61**	**-7.63**	**-8.33**	**-8.60**	**-9.16**	**0.84**	**0.87**	**94.9%**	**96.0%**
CBYG ($\lambda_x = 30, \lambda_v = 30$)	-7.13	-8.64	-6.95	-7.47	-7.79	-8.17	0.79	0.81	91.0%	91.4%
CBYG ($\lambda_x = 50, \lambda_v = 50$)	-7.33	-8.94	-7.22	-7.79	-8.07	-8.49	0.74	0.76	85.6%	86.2%
CBYG ($\lambda_x = 30, \lambda_v = 40$)	-7.28	-8.89	-7.15	-7.68	-7.98	-8.42	0.77	0.78	89.2%	89.7%
CBYG ($\lambda_x = 40, \lambda_v = 30$)	-7.10	-8.59	-6.92	-7.42	-7.76	-8.10	0.80	0.82	91.8%	92.3%

Table 5: Lipinski results for all methods.

	CBYG	TargetDiff	DecompDiff	MolCraft	DecompOpt	ALIDiff
Avg. Clash (↓)	**3.71**	10.54	13.66	5.72	17.05	8.71
Avg. Strain Energy (↓)	$\mathbf{5.85\times10^{7}}$	1.41×10^{14}	1.44×10^{9}	7.57×10^{11}	1.18×10^{11}	6.22×10^{16}
Avg. Interaction (↑)	15.65	17.15	18.26	15.92	**18.77**	17.74

H.2 Experimental Analysis of Section 6.3

In this experiment, both our proposed model and the RGA model exhibited overall high performance. Here, it is important to consider molecular complexity in relation to binding affinity. Typically, molecules with greater complexity tend to possess higher binding affinity due to increased opportunities for intermolecular interactions with target proteins; conversely, simpler molecules usually exhibit lower affinity. Given this, the performance of the RGA model on the AiZynthFinder benchmark aligns logically with its relatively lower binding affinity scores reported in Table 1. Applying the same perspective to our proposed model, it is particularly noteworthy that our model not only achieves top-tier performance in binding affinity but also exhibits near state-of-the-art results on the AiZynthFinder benchmark. This indicates the ability of our approach to generate molecules that are simultaneously effective in terms of biological efficacy and practical retrosynthetic feasibility.

Interestingly, we observe that several models show no clear correlation between their performance on the AiZynthFinder benchmark and the SA score reported in Table 1. This highlights the necessity for evaluating synthetic feasibility in SBDD research using multiple diverse criteria beyond just the SA score. A notable concern is that despite high SA scores (approaching 0.8 for RGA and surpassing 0.8 for our proposed model) the fraction of molecules classified as synthetically feasible under the AiZynthFinder 'Solved' metric remains below 50%. This outcome suggests that synthetic feasibility deserves greater attention in future SBDD research, underscoring the need for broader consideration and deeper analysis of retrosynthetic practicality.

H.3 Experimental Analysis of Section 6.4

As demonstrated in Figure 6, guidance scores obtained using the BFN-based approach consistently surpass those derived from diffusion-based methods across the entire generative trajectory. Furthermore, guidance scores from diffusion models utilizing predictions of the final clean state ($\hat{\mathbf{x}}_0$) exhibit marked instability, underscoring the robustness of the BFN-guided generation procedure. Notably, diffusion-based methods (green and gray plots) yield higher absolute guidance scores when employing predictions of the final molecular states; however, these scores simultaneously exhibit increased variance as the generative process advances. This behavior highlights a fundamental trade-off within diffusion-based guidance strategies in 3D molecular generation: gradients derived from predicted clean states facilitate higher guidance scores but necessitate point estimation toward the final state in the sample space, inherently introducing instability into intermediate guidance steps. In contrast, BFN operates in a continuous parameter space rather than directly in sample space, enabling stable and continuous gradient propagation even for categorical variables. Consequently, BFN-based guidance using predicted final states provides inherently more stable gradient trajectories, making it particularly advantageous for robust and controllable 3D molecular generation.

H.4 Experimental Analysis of Section 6.5

To evaluate the selectivity control capabilities of the proposed CBYG model, we conducted experiments using the selectivity benchmark set specifically constructed for this study. We primarily assessed and compared selectivity performance before and after applying guidance in two generative model categories: diffusion-based models and Bayesian Flow Network-based models. Notably, optimization-based generation models were excluded from this comparison due to their intrinsic requirement for retraining to optimize for different molecular properties, highlighting the versatility of our proposed model in addressing diverse property optimization objectives.

Molecule generation was directed toward enhancing binding affinity to designated on-target proteins, while guidance was explicitly designed to minimize binding affinity to specified off-target proteins. In Table 3, the "Succ.Rate" represents the proportion of generated molecules demonstrating superior affinity for the on-target protein relative to the off-target, whereas the "Δ Score" quantifies the differential affinity between on-target and off-target interactions.

Experimental results revealed that even without explicit selectivity-guidance, both model categories produced molecules with superior affinity toward the on-target protein in more than half of the generated cases. This outcome can be attributed to the inherent advantage of structure-based generative models, which explicitly encode and leverage the structural context of the target proteins during molecule generation. Nevertheless, Bayesian Flow Network-based models consistently demonstrated superior performance compared to diffusion-based models, and this advantage was markedly amplified when selectivity guidance was employed. These findings collectively underscore the efficacy and versatility of the proposed CBYG framework in achieving controlled generation not only for conventional metrics such as synthetic accessibility (SA score) but also for critical properties such as selectivity.

H.5 Comparative Visualization of Generated Ligands

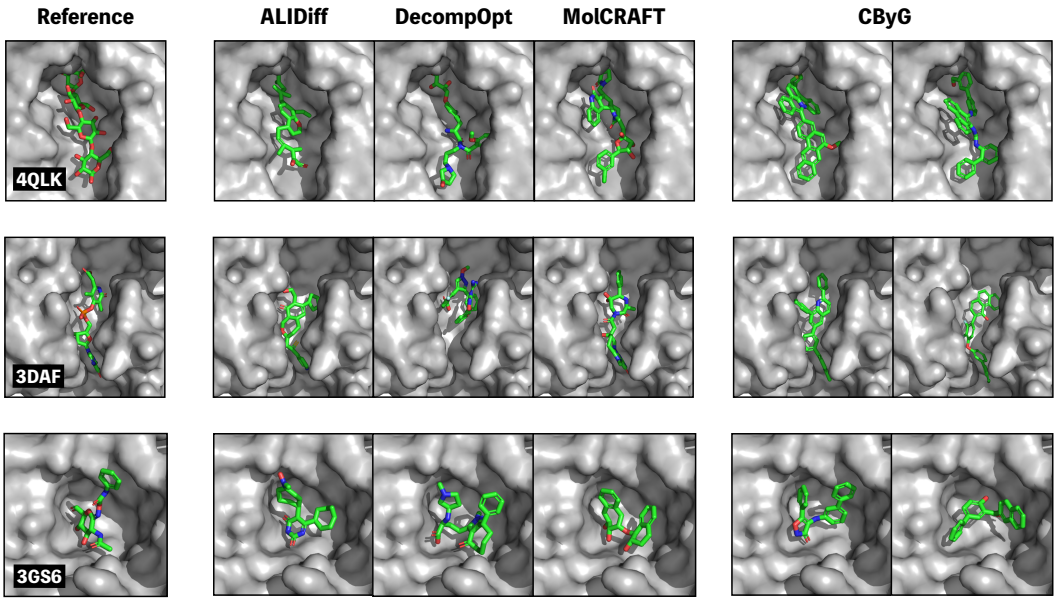

Figure 8: Visualization of generated ligands for protein pockets, with a reference molecule (left) and corresponding outputs from ALIDiff, DecompOpt, MolCRAFT, and CBYG.

I Selectivity Dataset

We constructed a selectivity-focused dataset based on kinase inhibitor selectivity data. Initially, we identified the selectivity profiles of 285 proteins across 38 kinase inhibitors, as reported in a study on the quantitative analysis of kinase inhibitor selectivity [27, 10]. Subsequently, for each inhibitor, we categorized the proteins into on-target and off-target groups and extracted their corresponding Entrez Gene Symbols. These gene symbols were then used to systematically gather protein-related data from the UniProt [8] database via REST APIs. The UniProt web crawling process was structured in

three stages. First, the Entrez Gene Symbols were URL-encoded and filtered by the human taxonomy ID (9606) to retrieve corresponding UniProt IDs in JSON format. Second, protein sequences were obtained by querying the UniProt FASTA API, with FASTA headers subsequently removed. Third, ATP binding site information—including binding site positions and sequences—was extracted from the UniProt feature sections. The collected data, comprising UniProt IDs, sequences, and binding site details, were integrated using the initial set of 285 Entrez Gene Symbols as a reference. Proteins lacking ATP binding site data (six in total) were excluded, yielding a final dataset of 279 proteins prepared for further analysis. Protein structures were predicted using AlphaFold3 [1], and model structure files were retrieved in CIF format. These files were converted to PDB format using the Bio.PDB module of Biopython. The ATP binding site information was then employed to define and extract protein pocket structures, specifically targeting the binding site residues and the surrounding region within a 5Å radius. Structural similarity among the protein pockets was evaluated using TM-score and RMSD metrics. The TM-score quantifies topological similarity between protein structures, with values ranging from (0, 1]; scores above 0.5 typically indicate identical protein folds, while scores below 0.17 suggest unrelated structures. We classified and organized the extracted protein structures into directories based on a TM-score threshold of 0.4 and an RMSD of 1Å, reflecting structurally similar protein pockets suitable for downstream selectivity analyses. This process ultimately facilitated the construction of on-target (primary) and off-target pairs.

J Rethinking

J.1 Addressing Fundamental Challenges of Diffusion-based Guidance in 3D Molecular Generation

A core objective in Structure-based Drug Design (SBDD) is generating molecules that bind specifically to target proteins while simultaneously satisfying desired properties. Diffusion-based generative models are particularly suited to this task, as they can incorporate external predictors for property-guided sampling. Specifically, these models leverage guidance derived from predictors to direct the generative process toward property-specific regions of molecular space. However, as briefly mentioned in previous sections, 3D molecular structures inherently comprise hybrid data types, consisting of continuous variables (e.g., Cartesian coordinates) typically modeled by Gaussian distributions, and categorical variables (e.g., atom types such as oxygen or nitrogen) typically represented by categorical distributions. This hybrid nature presents fundamental challenges for conventional gradient-based guidance approaches.

First, since coordinates and atom types are sampled from fundamentally distinct distributions, guidance gradients tend to propagate independently across these data types. Consequently, the guidance mechanism often fails to accurately capture the critical chemical interdependencies between atomic coordinates and categorical atom identities, thereby undermining the chemical coherence of the generated molecules.

Second, categorical variables in diffusion models rely on discrete sampling processes involving an argmax operation at each reverse sampling step. Due to the discrete nature of argmax operations, direct application of gradient-based guidance becomes infeasible, as minor guidance gradients typically do not influence the argmax outcome unless excessively amplified. Yet, increasing the guidance scale excessively can cause the distribution during reverse sampling to become dominated by guidance gradients, resulting in unstable and unrealistic molecular structures. Alternatively, attempting to circumvent this issue by artificially converting categorical distributions into continuous or discretized variables introduces unnatural assumptions and significantly increases the complexity of model design.

Lastly, injecting guidance gradients directly into the denoising process, which operates in the molecular sample space, risks destabilizing intermediate molecular configurations. This instability arises due to the numerical sensitivity of 3D coordinates, potentially causing molecules to lose chemical validity and structural coherence during intermediate generation stages. Thus, this approach substantially hinders effective controllability over molecular properties, and leads to unstable molecular outcomes.

In the context of 3D molecules (with continuous coordinates and discrete atom types), diffusion-based generative guidance methods face several fundamental limitations.

First, guidance gradients often fail to capture interdependencies between modalities (e.g. coordinate updates and atom-type assignments may be misaligned if treated separately) as evidenced by the need for separate latent spaces or noise schedules for different variable types in prior diffusion approaches.

Second, guidance in categorical diffusion is unstable and often ineffective: choosing atom types via an argmax during the denoising process introduces a discontinuous, non-differentiable operation that disrupts gradient-based optimization. Third, the denoising of spatial coordinates is structurally fragile – adding noise to atomic positions can break chemical bonds or distort interatomic distances beyond physical limits, leaving intermediate states chemically invalid and uninformative. Given these challenges, BFN offer a promising alternative for property-guided molecular generation. BFN operate in a fully differentiable parameter space and provide a unified probabilistic treatment of continuous and categorical modalities, thereby inherently modeling cross-modal dependencies and avoiding the need for modality-specific hacks. In contrast to diffusion, BFN do not require per-step argmax sampling for atom types; instead, they maintain a probability simplex representation for categorical variables, preserving gradient information throughout the generative process. This unified and differentiable approach enables stable gradient-based guidance on molecular properties, making BFN a robust paradigm for 3D molecule generation under complex hybrid objectives.

J.2 Rethinking Posterior Guidance: From Intermediate States to Predicted Final Structures

In generative frameworks such as diffusion models, conditional generation typically involves controlling the generative process by leveraging gradient guidance from a posterior conditioned on labels (attributes) $\mathtt{l}$. According to the theoretical foundations of the reverse process, the introduced posterior term can be represented as $\nabla_{\mathbf{x}_t} p(\mathtt{l} \mid \mathbf{x}_t)$, commonly known as the conditional score function, which is usually learned using a dedicated neural network. Here, it is crucial to reassess whether the intermediate state x_t, employed as input for the conditional score function, is a sensible variable for attribute prediction [11, 45, 22]. In domains like image generation, where score-based diffusion models were initially introduced, intrinsic structural characteristics of the data enable meaningful predictions of labels even from intermediate noisy states. Thus, utilizing the conditional score function in the form $p(\mathtt{l} \mid \mathbf{x}_t)$ has proven reasonable in these contexts.

However, unlike image data, intermediate states of 3D molecular structures with added noise lose their chemical validity, rendering derived molecular properties essentially meaningless. Consequently, employing a posterior conditioned directly on the intermediate noisy state $\mathbf{x}_t$, i.e., $p(\mathtt{l} \mid \mathbf{x}_t)$, is fundamentally unreasonable as guidance for molecule generation tasks. To overcome this limitation, recent studies have adopted a posterior sampling strategy, originally proposed in inverse problems within the image generation domain [7, 23]. Specifically, these methods predict the final, noise-free molecular structure x_0 and leverage this prediction to guide gradient-based generation, i.e., $p(\mathtt{l} \mid \hat{\mathbf{x}}_0)$. Further efforts have also extended such posterior sampling approaches to the conditional generation of 3D molecules. However, existing methods primarily focus on general conditional molecular generation rather than the specialized task of structure-based drug design (SBDD), which involves molecular binding to specific proteins. Therefore, substantial modifications and further methodological advancements are necessary for applying these approaches to the SBDD task. Notably, existing frameworks discretize categorical variables representing atom types into continuous representations, which is inherently unnatural given the data's discrete characteristics.

In summary, predicting the posterior for the final molecular structure x_0 and subsequently using it for calculating guidance gradients is more principled in the context of SBDD tasks. We propose that this principle is broadly applicable across generative modeling frameworks, including not only diffusion models but also BFN.

J.3 Limitations of Current Evaluation Methods

In previous research on structure-based drug design (SBDD), evaluating the binding affinity between generated molecules and their target proteins has been a common practice to assess model performance. Most studies traditionally relied heavily on AutoDock Vina to measure binding affinity. Although AutoDock Vina provides three distinct scoring metrics, these metrics inherently depend on the same underlying computational algorithm, potentially introducing bias due to reliance on a single scoring method. To enhance the generalizability and reliability of affinity assessments, incorporating multiple docking algorithms in the evaluation process is necessary. Accordingly, in Section 6.2, we present a detailed experimental setup employing several docking tools, including AutoDock Vina, to enable a more comprehensive and robust evaluation of generative model performance.

Furthermore, previous SBDD research has commonly utilized the Synthetic Accessibility (SA) score to evaluate the synthetic feasibility of generated molecules. The SA score quantitatively integrates chemical fragment contributions and structural complexity penalties into a single metric ranging between 0 and 1, with higher scores indicating greater synthetic accessibility. However, molecules

possessing very high SA scores (e.g., greater than 0.9) frequently lack viable retrosynthetic pathways, making their actual synthesis infeasible. Regardless of a molecule's theoretical efficacy, its practical value is severely limited without an achievable synthetic route. Therefore, rigorously assessing realistic synthetic accessibility is critical, although research addressing this aspect has been relatively limited. Recognizing the importance of this issue, we introduce the AiZynthFinder benchmark, an evaluation method for retrosynthetic analysis based on practically available chemical building blocks.

From the viewpoint of practical drug development, selectivity is equally important as binding affinity for identifying promising drug candidates. Selectivity refers to the ability of a candidate molecule to specifically bind to its intended target protein without significant interactions with off-target proteins. Molecules lacking sufficient selectivity may interact with unintended proteins, potentially causing side effects or adverse reactions, thereby reducing or negating the desired pharmacological effects. Recently, selectivity has received increased attention in the field of 3D molecular generation, and several diffusion-based guidance strategies have been proposed to address this requirement.

However, existing selectivity-focused strategies typically require prior training of classifiers that distinguish between positive (binding) and negative (non-binding) protein-ligand pairs. Furthermore, the CrossDocked2020 dataset, commonly used in docking studies, was not originally constructed for selectivity evaluations. Thus, leveraging this dataset for selectivity assessments necessitates extensive additional docking computations. Moreover, the absence of clear criteria for identifying true binding molecules and the significantly greater number of false binding molecules relative to true binding molecules pose substantial challenges for obtaining generalizable guidance signals. Consequently, deriving selectivity metrics based solely on the CrossDocked2020 dataset inherently risks bias due to these intrinsic dataset limitations. Most critically, the CrossDocked dataset may not adequately reflect biologically meaningful selectivity, limiting its practical utility for reliable selectivity assessment. Therefore, establishing rigorous, standardized benchmark datasets capable of objectively evaluating selectivity is essential. Additionally, there is an urgent need to develop novel, efficient controllable generation strategies capable of effectively ensuring molecular selectivity.