FUSEDBFN: PROBABILISTIC COMPOSITION FOR DUAL-TARGET MOLECULAR GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Dual-target drug design aims to generate 3D molecules that can be compatible with two distinct protein binding pockets, yet the scarcity of dual-target structural data makes direct model training difficult. We study this problem in a zero-shot setting by employing pretrained single-target generative models and investigate where target-specific information should be composed within the generative process. We propose FusedBFN, which combines two target-aware sender likelihoods using a weighted Product of Experts before performing a shared Bayesian update. To examine composition at different probabilistic levels, we further introduce CompBFN and FusedDiff, which, together with CompDiff, instantiate distribution fusion and transition-kernel composition under both Bayesian flow and diffusion backbones. Our analysis reveals that seemingly different composition mechanisms can become equivalent under matched settings, while their differences otherwise depend on the induced uncertainty and generative dynamics. We also introduce a chemically weighted prior-based alignment method and a prior-free pocket alignment strategy to construct aligned dual-target contexts. Extensive experiments show that FusedBFN achieves strong dual-target generation performance, and further reveal that reducing sampling uncertainty around the fused signal can substantially improve average dual-target binding. These results clarify how the probabilistic formulation of composition interacts with the chosen generative framework in dual-target molecular design. The source code is provided at `https://anonymous.4open.science/r/FusedBFN/`.

1 INTRODUCTION

Structure-based drug design (SBDD) aims to generate small molecules that geometrically and chemically complement the 3D structure of a target protein, where generative models have achieved promising results in this task (Guan et al., 2023; Qu et al., 2024; Gu et al., 2024; Zhou et al., 2026). However, most existing methods are developed for the single-target scenario, limiting their applicability when therapeutic intervention requires the coordinated modulation of multiple targets (He et al., 2016; Ramsay et al., 2018). Dual-target design instead seeks a single molecule compatible with two distinct binding environments, while the scarcity of dual-target structural data makes it difficult to train dedicated 3D generators (Southiratn et al., 2026). A practical alternative is therefore to reuse a pretrained single-target model in a zero-shot manner, applying it separately to the two protein pockets and fusing the resulting target-specific information during generation (Zhou et al., 2024a). This raises a fundamental and underexplored modeling question: how can two target-aware signals be composed at different probabilistic levels, and how do the resulting fusion strategies behave within different generative frameworks? Two natural choices are to fuse target-specific distributions before a generative update or to compose transition kernels after constructing separate target-specific updates. These operations may not induce the same generative process, particularly when molecular states comprise both continuous coordinates and discrete atom types.

Existing dual-target generation methods provide only a partial view of this question. Most methods operate on sequence-based molecular representations without jointly conditioning generation on the 3D geometries of both binding pockets. Isigkeit et al. (2024) and Srinivasan & Bajorath (2024) generate SMILES strings by fine-tuning chemical language models to explore the chemical space of dual-target compounds, whereas Munson et al. (2024) and Southiratn et al. (2026) employ reinforcement learning and search-based approaches to optimize molecular candidates against multiple

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

affinity objectives. In the structure-based setting, Zhou et al. (2024a) reprogram the pretrained target-aware diffusion model (DM) TargetDiff (Guan et al., 2023) through two fusion strategies. CompDiff composes target-conditioned generative transition kernels at the output level. DualDiff further combines target-specific hidden-state and coordinate messages within every layer of the SE(3)-equivariant network. Although these methods demonstrate the potential of zero-shot structure-based composition, existing approaches remain confined to diffusion models and do not investigate alternative probabilistic fusion strategies. Consequently, it is not yet well understood how the choice of composition level affects dual-target generation.

To address this question, we propose FusedBFN, a **Fused B**ayesian **F**low **N**etwork method for dual-target drug design. Motivated by the Bayesian Flow Network (BFN) (Graves et al., 2023) mechanism of updating the parameters of an input distribution via Bayesian inference, FusedBFN fuses two target-aware sender likelihoods using a weighted Product of Experts (wPoE) (Hinton, 2002; Zhang et al., 2020; 2025) formulation before performing a shared Bayesian update. Since structural data for dual-target ligand–protein complexes are scarce, we build FusedBFN upon a pretrained target-aware BFN model MolCRAFT (Qu et al., 2024), extending the generative knowledge learned from single-target complexes to dual-target generation in a zero-shot manner. To investigate fusion at different probabilistic levels, we further introduce CompBFN, which transfers the transition kernel fusion applied in CompDiff to the BFN generative process, and FusedDiff, which fuses target-specific distributions using wPoE before each denoising update. Together with FusedBFN and CompDiff, these variants enable us to examine two fusion strategies within both the BFN and diffusion frameworks. For pocket alignment, we extend the ligand-prior alignment proposed by Zhou et al. (2024a) to consider the chemical semantics of different atoms in the protein–ligand interaction prior, and introduce a prior-free strategy to simplify the alignment process. Through theoretical analysis and extensive experiments, we characterize the relationships between different probabilistic formulations of composition and study how their induced sampling behaviors affect dual-target molecular generation.

Our main contributions can be summarized as follows:

- We propose FusedBFN, a fused Bayesian flow network for dual-target generation that utilizes wPoE to fuse target-specific sender likelihoods before a shared Bayesian update, thereby extending a pretrained single-target BFN to dual-target generation.
- We introduce CompBFN, which transfers the transition-kernel fusion used in CompDiff to BFN, and FusedDiff, which fuses target-aware distributions before each diffusion denoising update. Together, these methods instantiate two different fusion strategies within both the BFN and diffusion frameworks.
- We theoretically characterize when the two composition mechanisms coincide or differ, and empirically analyze the resulting sampling behaviors within each generative framework.
- We extend ligand-prior alignment by accounting for the chemical semantics of different atoms in the protein–ligand interaction prior, and introduce a prior-free strategy that operates without ligand priors.

2 Related Works

Single- and Dual-Target Drug Design Current SBDD methods are typically formulated for single-target scenarios. Autoregressive methods (Luo et al., 2021; Peng et al., 2022; Zhang et al., 2023; Zhang & Liu, 2023) sequentially place atoms or molecular fragments within a target binding site. Recently, non-autoregressive generative frameworks, including diffusion models (Guan et al., 2023; 2024; Qian et al., 2024; Gu et al., 2024; Zhou et al., 2025), flow matching (Zhang et al., 2024; Zhou et al., 2026), and Bayesian flow networks (Qu et al., 2024; Qiu et al., 2025), have also been extensively applied to SBDD. These methods iteratively refine all atoms at each step and have demonstrated promising performance. In contrast, most existing work on dual-target molecular generation adopts sequence-based approaches. Isigkeit et al. (2024); Srinivasan & Bajorath (2024) fine-tune chemical language models to generate SMILES sequences. Munson et al. (2024) employ reinforcement learning to optimize generated molecules toward multiple targets. Southiratn et al. (2026) introduce a Pareto Monte Carlo tree search framework over a synthesizable fragment space. In the structure-based setting, Zhou et al. (2024a) extend dual-target generation to 3D space: CompDiff composes target-aware reverse transitions at the model output, whereas DualDiff com-

bines hidden-state and coordinate messages within each layer of the denoising network. However, both methods are confined to the diffusion framework and do not explore the composition of target-specific information at different probabilistic levels.

Product of Experts The Product of Experts (PoE) (Hinton, 2002) combines multiple probabilistic experts by multiplying and renormalizing their densities, thereby favoring regions jointly supported by all experts. In ordinary PoE, every expert contributes with unit strength, so their relative influence cannot be adjusted explicitly. The generalized Product of Experts (gPoE) (Cao & Fleet, 2014) introduces a separate power for each expert to reflect its reliability. Weighted Product of Experts (wPoE) (Zhang et al., 2020; 2025) follows the same powered-product principle but emphasizes controlling expert contributions through weights that may be prescribed, learned, or optimized. In our setting, wPoE serves as a tool for fusing target-specific distributions.

3 PRELIMINARIES

Notation and Problem Formulation A ligand molecule with N_M atoms is represented as $\mathbf{m} = [\mathbf{x}_M, \mathbf{v}_M]$, where $\mathbf{x}_M \in \mathbb{R}^{N_M \times 3}$ is the matrix of 3D coordinates and $\mathbf{v}_M \in \mathbb{R}^{N_M \times K}$ contains the one-hot vectors encoding the atom types, with K being the atom type dimension. For each target $j \in \{1, 2\}$, we denote $\mathbf{p}^{(j)} = [\mathbf{x}_P^{(j)}, \mathbf{v}_P^{(j)}]$ as its protein binding pocket, where $\mathbf{x}_P^{(j)} \in \mathbb{R}^{N_P^{(j)} \times 3}$ contains the coordinates of its $N_P^{(j)}$ protein atoms and $\mathbf{v}_P^{(j)} \in \mathbb{R}^{N_P^{(j)} \times N_f}$ encodes their N_f-dimensional atom features.

Since protein binding pockets exhibit diverse geometric shapes and chemical characteristics, spatial alignment between the two pockets is required when modeling the conditional distribution given both pockets (Zhou et al., 2024a). We align $\mathbf{p}^{(1)}$ to $\mathbf{p}^{(2)}$ through a rigid transformation $\mathcal{T}$ parameterized by a rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ and a translation vector $\mathbf{b} \in \mathbb{R}^3$, yielding $\mathcal{T}\mathbf{p}^{(1)} = [\mathbf{x}_P^{(1)}\mathbf{R}^\top + \mathbf{1}_{N_P^{(1)}}\mathbf{b}^\top, \mathbf{v}_P^{(1)}]$, where $\mathbf{1}_{N_P^{(1)}} \in \mathbb{R}^{N_P^{(1)}}$ is the all-ones vector. For convenience, we write $\widetilde{\mathbf{p}}^{(1)} = \mathcal{T}\mathbf{p}^{(1)}$ and $\widetilde{\mathbf{p}}^{(2)} = \mathbf{p}^{(2)}$. Dual-target molecular generation is then formulated as learning the conditional distribution $p(\mathbf{m} \mid \widetilde{\mathbf{p}}^{(1)}, \widetilde{\mathbf{p}}^{(2)})$.

Molecular Design via BFNs BFNs formulate generation as a sequence of message exchanges between a sender and a receiver. Let $i \in \{1, \ldots, N_{\mathrm{BFN}}\}$ denote the discrete BFN update-step index, where N_{BFN} is the total number of updates, and $\tau_i \in [0, 1]$ is defined as the continuous schedule time associated with step i. At step i, the molecular input distribution $p_I(\mathbf{m} \mid \boldsymbol{\theta}_{i-1})$ is parameterized by $\boldsymbol{\theta}_{i-1} = [\boldsymbol{\theta}_{i-1}^x, \boldsymbol{\theta}_{i-1}^v]$. The sender transmits a noisy message $\mathbf{y}_i = [\mathbf{y}_i^x, \mathbf{y}_i^v]$ according to the sender distribution $p_S(\mathbf{y}_i \mid \mathbf{m}; \boldsymbol{\alpha}_i)$. Here, $\boldsymbol{\alpha}_i = (\alpha_i^x, \alpha_i^v) \in \mathbb{R}^2_{>0}$ contains the accuracy increments at step i. For target j, the pretrained SE(3)-equivariant GNN $\boldsymbol{\Psi}$ predicts the clean molecular estimate $\hat{\mathbf{m}}^{(j)} = [\hat{\mathbf{x}}_M^{(j)}, \hat{\mathbf{v}}_M^{(j)}] = \boldsymbol{\Psi}(\boldsymbol{\theta}_{i-1}, \widetilde{\mathbf{p}}^{(j)}, \tau_{i-1})$, which parameterizes the output distribution $p_O^{(j)}$. The corresponding target-conditioned receiver distribution is

$$p_R^{(j)}(\mathbf{y}_i \mid \boldsymbol{\theta}_{i-1}, \widetilde{\mathbf{p}}^{(j)}; \tau_i) = \mathbb{E}_{\hat{\mathbf{m}}^{(j)} \sim p_O^{(j)}(\cdot \mid \boldsymbol{\theta}_{i-1}, \widetilde{\mathbf{p}}^{(j)}; \tau_{i-1})}[p_S(\mathbf{y}_i \mid \hat{\mathbf{m}}^{(j)}; \boldsymbol{\alpha}_i)]. \tag{1}$$

After receiving the message $\mathbf{y}_i$, the BFN updates the parameters $\boldsymbol{\theta}$ via the Bayesian update function $\mathbf{h}$ derived from Bayesian inference rules: $\boldsymbol{\theta}_i = \mathbf{h}(\boldsymbol{\theta}_{i-1}, \mathbf{y}_i, \boldsymbol{\alpha}_i)$. The update function induces the Bayesian update distribution

$$p_U(\boldsymbol{\theta}_i \mid \boldsymbol{\theta}_{i-1}, \mathbf{m}, \widetilde{\mathbf{p}}^{(j)}; \boldsymbol{\alpha}_i) = \mathbb{E}_{\mathbf{y}_i \sim p_S}[\delta(\boldsymbol{\theta}_i - \mathbf{h}(\boldsymbol{\theta}_{i-1}, \mathbf{y}_i, \boldsymbol{\alpha}_i))], \tag{2}$$

where $\delta(\cdot)$ is the Dirac delta distribution. Due to the additivity of accuracy (Graves et al., 2023), the Bayesian flow distribution can be derived by marginalizing all intermediate parameters:

$$p_F(\boldsymbol{\theta}_i | \mathbf{m}, \widetilde{\mathbf{p}}^{(j)}; \tau_i) = \mathop{\mathbb{E}}_{\boldsymbol{\theta}_{1\ldots i-1} \sim p_U} p_U(\boldsymbol{\theta}_i | \boldsymbol{\theta}_{i-1}, \mathbf{m}, \widetilde{\mathbf{p}}^{(j)}; \alpha_i) = p_U(\boldsymbol{\theta}_i | \boldsymbol{\theta}_0, \mathbf{m}, \widetilde{\mathbf{p}}^{(j)}; \boldsymbol{\beta}(\tau_i)), \tag{3}$$

where $\boldsymbol{\beta}(\tau_i) = (\beta_i^x, \beta_i^v) = (\sum_{r=1}^i \alpha_r^x, \sum_{r=1}^i \alpha_r^v)$. After each update, the receiver feeds the updated parameters of p_I into $\boldsymbol{\Psi}$, which parameterizes the next p_O. This procedure is repeated for N_{BFN} steps until the receiver can predict the molecule with sufficient accuracy.

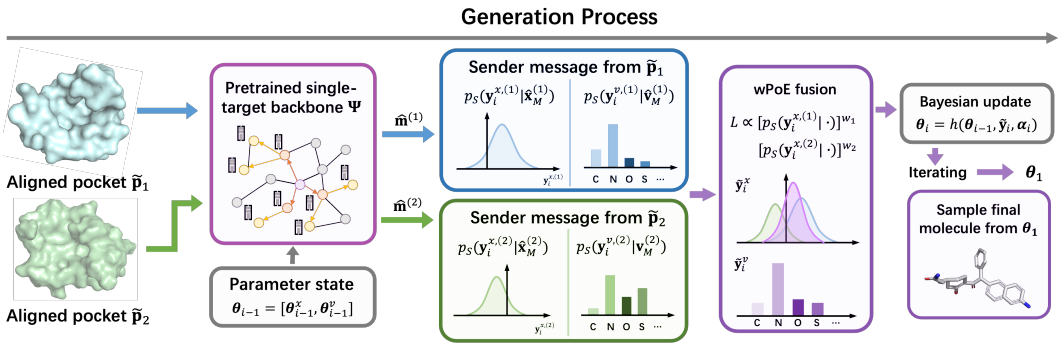

Figure 1: Overview of FusedBFN for dual-target molecular generation. Given two aligned pockets, a shared pretrained network Ψ predicts two target-specific molecular estimates. FusedBFN combines information from the two targets using wPoE before performing a shared Bayesian update. Iterating this process yields a final molecule jointly conditioned on both target pockets.

Molecular Design via DMs Diffusion models define a forward corruption process and a learned reverse generative process. The diffusion step index is denoted by $t \in \{1, \dots, T\}$, where T is the total number of diffusion steps. We reserve t exclusively for diffusion and distinguish it from the BFN update-step index i and the BFN schedule time τ_i. The initial state $\mathbf{m}_0 = \mathbf{m}$ is the clean ligand molecule, whereas $\mathbf{m}_t$ is its noisy state at diffusion step t. The forward and reverse processes are defined as the Markov chains

$$q(\mathbf{m}_{1:T} \mid \mathbf{m}_0) = \prod_{t=1}^{T} q(\mathbf{m}_t \mid \mathbf{m}_{t-1}), \quad p_{\boldsymbol{\omega}}(\mathbf{m}_{0:T-1} \mid \mathbf{m}_T, \tilde{\mathbf{p}}^{(j)}) = \prod_{t=1}^{T} p_{\boldsymbol{\omega}}(\mathbf{m}_{t-1} \mid \mathbf{m}_t, \tilde{\mathbf{p}}^{(j)}). \quad (4)$$

The SE(3)-equivariant network $\boldsymbol{\Phi}_{\boldsymbol{\omega}}$ predicts $[\hat{\mathbf{x}}^{(j)}_{M,0}, \hat{\mathbf{v}}^{(j)}_{M,0}] = \boldsymbol{\Phi}_{\boldsymbol{\omega}}(\mathbf{m}_t, \tilde{\mathbf{p}}^{(j)}, t)$, which are substituted into the reverse posterior to instantiate the reverse transition kernel: $K_t^{(j)}(\mathbf{m}_{t-1} \mid \mathbf{m}_t) := p_{\boldsymbol{\omega}}(\mathbf{m}_{t-1} \mid \mathbf{m}_t, \tilde{\mathbf{p}}^{(j)})$. Detailed single-target formulations for the BFN and diffusion backbones are presented in Appendices A.2 and A.1, respectively.

4 Methods

CompDiff composes transition kernels after the corresponding target-aware updates have been constructed. In contrast, we employ wPoE to fuse the target-specific distributions before performing a shared generative update. We refer to these two mechanisms as distribution fusion and transition-kernel composition, respectively.

Table 1: Summary of the four methods.

Mechanism	BFN	Diffusion
Distribution fusion	FusedBFN	FusedDiff
Kernel composition	CompBFN	CompDiff

For distribution fusion, let z denote the random variable shared by two distributions $p^{(1)}(z)$ and $p^{(2)}(z)$. Given nonnegative fusion weights $w_1, w_2 \geq 0$, we fuse the two distributions using wPoE. Transition-kernel composition operates on the target-specific kernels $K^{(j)}(z' \mid z)$, which map the current generative state z to the next state z', with $\eta > 0$ controlling the composition strength. The two mechanisms are defined as

$$p(z) \propto [p^{(1)}(z)]^{w_1} [p^{(2)}(z)]^{w_2}, \qquad K(z' \mid z) \propto [K^{(1)}(z' \mid z) K^{(2)}(z' \mid z)]^{\eta}. \quad (5)$$

We first introduce FusedBFN to realize wPoE distribution fusion within the BFN framework. To investigate how distribution fusion and transition-kernel composition behave under different generative backbones, we further develop FusedDiff by transferring distribution fusion to diffusion and CompBFN by transferring CompDiff-style transition-kernel composition to BFN. Together with CompDiff, these methods instantiate both mechanisms across BFN and diffusion, as summarized in Table 1, allowing us to systematically examine their behavior in the two generative frameworks.

4.1 FUSEDBFN: WPOE DISTRIBUTION FUSION

At step i, the target-conditioned receiver distributions defined in equation 1 produce the clean molecular estimates $\hat{\mathbf{m}}^{(j)}$. FusedBFN fuses the corresponding target-specific distributions through wPoE and performs one shared Bayesian update. Let $S := w_1 + w_2, Q := w_1^2 + w_2^2$, where S controls the information strength of the fused distribution and Q determines the sampling variance induced by combining two target-specific messages. By default, we use equal fusion weights $w_1 = w_2 = 1/2$.

Continuous coordinates The continuous atom coordinates are modeled as a Gaussian distribution $\mathcal{N}(\mathbf{x}_M|\boldsymbol{\mu}_i, \rho_i^{-1}\mathbf{I})$, with parameters $\boldsymbol{\theta}_i^x = \{\boldsymbol{\mu}_i, \rho_i\}$, where $\boldsymbol{\mu}_i$ is learned and ρ_i is specified by the noise factor α. The prior $\boldsymbol{\theta}_0^x$ is set as a standard Gaussian. Conditional on the target-specific coordinate estimates, the two branches independently generate sender messages

$$\mathbf{y}_i^{x,(j)} \sim \mathcal{N}(\hat{\mathbf{x}}_M^{(j)}, (\alpha_i^x)^{-1}\mathbf{I}), \tag{6}$$

where $\mathbf{y}_i^{x,(j)}$ is the message generated by expert j. Here, expert j denotes the target-specific branch obtained by conditioning the shared pretrained network on the pocket $\tilde{p}^{(j)}$. Applying wPoE to the two sender factors, the fused likelihood can be defined as

$$L(\mathbf{x}_M; \mathbf{y}_i^{x,(1)}, \mathbf{y}_i^{x,(2)}, \alpha_i^x) := [p_S(\mathbf{y}_i^{x,(1)} \mid \mathbf{x}_M; \alpha_i^x)]^{w_1}[p_S(\mathbf{y}_i^{x,(2)} \mid \mathbf{x}_M; \alpha_i^x)]^{w_2}. \tag{7}$$

Substituting equation 6 into equation 7 gives $L(\mathbf{x}_M; \mathbf{y}_i^{x,(1)}, \mathbf{y}_i^{x,(2)}, \alpha_i^x) \propto \mathcal{N}(\mathbf{x}_M; \tilde{\mathbf{y}}_i^x, (S\alpha_i^x)^{-1}\mathbf{I})$, where the fused message is distributed as

$$\tilde{\mathbf{y}}_i^x := \frac{w_1\mathbf{y}_i^{x,(1)} + w_2\mathbf{y}_i^{x,(2)}}{S} \sim \mathcal{N}\left(\frac{w_1\hat{\mathbf{x}}_M^{(1)} + w_2\hat{\mathbf{x}}_M^{(2)}}{S}, \frac{Q}{S^2\alpha_i^x}\mathbf{I}\right). \tag{8}$$

Conditioned on the two target-specific coordinate estimates, marginalizing the sender messages and intermediate parameters yields the fused coordinate flow

$$p_F(\boldsymbol{\mu}_i \mid \hat{\mathbf{x}}_M^{(1)}, \hat{\mathbf{x}}_M^{(2)}, \widetilde{\mathbf{p}}^{(1)}, \widetilde{\mathbf{p}}^{(2)}; \tau_i) = \mathcal{N}\left(\boldsymbol{\mu}_i; \frac{\beta^x(\tau_i)(w_1\hat{\mathbf{x}}_M^{(1)} + w_2\hat{\mathbf{x}}_M^{(2)})}{1 + S\beta^x(\tau_i)}, \frac{\beta^x(\tau_i)Q}{[1 + S\beta^x(\tau_i)]^2}\mathbf{I}\right), \tag{9}$$

where where $\beta^x(\tau_i) := \sum_{r=1}^{i} \alpha_r^x$ and $\rho_i = 1 + S\beta^x(\tau_i)$. The complete derivation is provided in Appendix B.1.

Discrete atom types For ligand atom $n \in \{1, \ldots, N_M\}$, let $v_M[n] \in \{1, \ldots, K\}$ denote its observed atom-type category. Its one-hot representation is $\mathbf{v}_M[n] := \mathbf{e}_{v_M[n]} \in \mathbb{R}^K$, where $\mathbf{e}_k$ is the k-th standard basis vector. The categorical input parameter is $\boldsymbol{\theta}_i^v[n] \in \Delta^{K-1}$, where Δ^{K-1} is the $(K-1)$-dimensional probability simplex. The prior is set to $\boldsymbol{\theta}_0^v[n] = K^{-1}\mathbf{1}_K$.

During sampling, expert j predicts $\hat{\mathbf{v}}_M^{(j)}[n] \in \Delta^{K-1}$ for ligand atom n and the two branches independently generate sender messages

$$\mathbf{y}_i^{v,(j)}[n] \sim \mathcal{N}(\alpha_i^v(K\hat{\mathbf{v}}_M^{(j)}[n] - \mathbf{1}_K), \alpha_i^v K\mathbf{I}_K), \tag{10}$$

where $\mathbf{1}_K \in \mathbb{R}^K$ is the all-ones vector, and $\mathbf{I}_K$ is the $K \times K$ identity matrix. For a candidate atom type $v_M[n] = k$, equivalently $\mathbf{v}_M[n] = \mathbf{e}_k$, we apply wPoE to the two senders and define the fused likelihood as

$$\begin{aligned} &L(v_M[n] = k; \mathbf{y}_i^{v,(1)}[n], \mathbf{y}_i^{v,(2)}[n], \alpha_i^v) \\ &\quad := [p_S(\mathbf{y}_i^{v,(1)}[n] \mid \mathbf{v}_M[n] = \mathbf{e}_k; \alpha_i^v)]^{w_1}[p_S(\mathbf{y}_i^{v,(2)}[n] \mid \mathbf{v}_M[n] = \mathbf{e}_k; \alpha_i^v)]^{w_2}. \end{aligned} \tag{11}$$

Substituting equation 10 into equation 11 and retaining the terms that depend on k gives $L(v_M[n] = k; \mathbf{y}_i^{v,(1)}[n], \mathbf{y}_i^{v,(2)}[n], \alpha_i^v) \propto \exp((\tilde{\mathbf{y}}_i^v[n])_k)$, where the fused message satisfies

$$\tilde{\mathbf{y}}_i^v[n] := w_1\mathbf{y}_i^{v,(1)}[n] + w_2\mathbf{y}_i^{v,(2)}[n] \sim \mathcal{N}(\alpha_i^v[K(w_1\hat{\mathbf{v}}_M^{(1)}[n] + w_2\hat{\mathbf{v}}_M^{(2)}[n]) - S\mathbf{1}_K], \alpha_i^v KQ\mathbf{I}_K). \tag{12}$$

Denote $\widetilde{\mathbf{Y}}_i^v[n] \in \mathbb{R}^K$ as the accumulated fused message. Marginalizing the sender messages and intermediate parameters gives

$$p_F(\boldsymbol{\theta}_i^v[n] \mid \hat{\mathbf{v}}_M^{(1)}[n], \hat{\mathbf{v}}_M^{(2)}[n], \widetilde{\mathbf{p}}^{(1)}, \widetilde{\mathbf{p}}^{(2)}; \tau_i) = \mathbb{E}_{\widetilde{\mathbf{Y}}_i^v[n]}[\delta(\boldsymbol{\theta}_i^v[n] - \mathrm{softmax}(\widetilde{\mathbf{Y}}_i^v[n]))], \tag{13}$$

where

$$\widetilde{\mathbf{Y}}_i^v[n] \sim \mathcal{N}(\beta^v(\tau_i)[K(w_1\hat{\mathbf{v}}_M^{(1)}[n] + w_2\hat{\mathbf{v}}_M^{(2)}[n]) - S\mathbf{1}_K], \beta^v(\tau_i)KQ\mathbf{I}_K). \tag{14}$$

The complete derivation is provided in Appendix B.2.

Sampling Concentration To explicitly control the stochastic uncertainty, we introduce a concentration parameter c while keeping the fused means and all other settings unchanged. Specifically, we replace the covariance terms by $\mathrm{Cov}(\boldsymbol{\mu}_i) = \frac{c\,\beta^x(\tau_i)}{[1+\beta^x(\tau_i)]^2}\mathbf{I}, \mathrm{Cov}(\widetilde{\mathbf{Y}}_i^v[n]) = c\,\beta^v(\tau_i)K\mathbf{I}$. Under the default setting, FusedBFN uses $c = Q = 1/2$. $c < Q$ produce a more concentrated sampling process around the same fused center, whereas $c > Q$ retain greater stochastic uncertainty. The limiting case $c = 0$ yields a deterministic variant. Importantly, when $c \neq Q$, this modification should be viewed as a concentration control around the fused signal rather than as the covariance directly implied by the wPoE construction.

4.2 Cross-Framework Variants: FusedDiff and CompBFN

FusedDiff: wPoE distribution fusion in diffusion We transfer wPoE distribution fusion to the diffusion model without modifying or retraining the pretrained single-target backbone. For continuous coordinates, we fuse the two target-aware forward marginals given in equation 26. Let $w_1, w_2 \geq 0$ and $S = w_1 + w_2$. Because the two Gaussian distributions have the same covariance, their wPoE fused distribution has the closed form

$$q(\mathbf{x}_{M,t}|\hat{\mathbf{x}}_{M,0}^{(1)}, \hat{\mathbf{x}}_{M,0}^{(2)}) \propto \prod_{j=1}^{2} q(\mathbf{x}_{M,t}|\hat{\mathbf{x}}_{M,0}^{(j)})^{w_j} = \mathcal{N}(\mathbf{x}_{M,t}; \sqrt{\bar{a}_t}\frac{w_1\hat{\mathbf{x}}_{M,0}^{(1)} + w_2\hat{\mathbf{x}}_{M,0}^{(2)}}{S}, \frac{1-\bar{a}_t}{S}\mathbf{I}).$$

To preserve the coordinate variance specified by the original diffusion schedule, the weights must satisfy $S = 1$. Deriving the reverse posterior induced by the fused distribution shows that FusedDiff performs coordinate updates using $\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_{M,t}\, w_1\hat{\mathbf{x}}_{M,0}^{(1)} + w_2\hat{\mathbf{x}}_{M,0}^{(2)})$, where $\tilde{\boldsymbol{\mu}}_t$ is defined in equation 29.

For discrete atom types, we apply wPoE fusion to the two target-specific categorical marginals following equation 27:

$$q(\boldsymbol{v}_{M,t}[n]|\hat{\mathbf{v}}_{M,0}^{(1)}[n], \hat{\mathbf{v}}_{M,0}^{(2)}[n]) \propto \prod_{j=1}^{2} q(\mathbf{v}_{M,t}[n]|\hat{\mathbf{v}}_{M,0}^{(j)}[n])^{w_j}. \tag{15}$$

Combining the fused distribution with the one-step categorical transition gives the reverse posterior $p\left(\mathbf{v}_{M,t-1}[n] \mid \mathbf{m}_t, \widetilde{\mathbf{p}}_1, \widetilde{\mathbf{p}}_2\right) = \mathrm{Cat}\left(\mathbf{v}_{M,t-1}[n]; \tilde{\mathbf{c}}_{t,n}^{\mathrm{Fuse}}\right)$, where

$$\left(\tilde{\mathbf{c}}_{t,n}^{\mathrm{Fuse}}\right)_k := \frac{q\left(\mathbf{v}_{M,t}[n] \mid \mathbf{v}_{M,t-1}[n] = \mathbf{e}_k\right) q(\mathbf{e}_k|\hat{\mathbf{v}}_{M,0}^{(1)}[n], \hat{\mathbf{v}}_{M,0}^{(2)}[n])}{\sum_{\ell=1}^{K} q\left(\mathbf{v}_{M,t}[n] \mid \mathbf{v}_{M,t-1}[n] = \mathbf{e}_\ell\right) q(\mathbf{e}_\ell|\hat{\mathbf{v}}_{M,0}^{(1)}[n], \hat{\mathbf{v}}_{M,0}^{(2)}[n])}. \tag{16}$$

FusedDiff therefore fuses the target-specific distributions first and performs only one reverse-posterior update for each modality. Complete derivations are provided in Appendix C and its categorical relation to CompDiff is analyzed in Appendix C.4.

CompBFN: transition-kernel composition in BFN CompBFN realizes transition-kernel composition within the BFN generation. As in CompDiff, the target-aware transition kernels are first constructed separately and then composed through a powered product with the default strength $\eta = 1/2$.

For the coordinate parameters $\boldsymbol{\theta}_i^x = \{\boldsymbol{\mu}_i, \rho_i\}$, the precision is updated deterministically as $\rho_i = \rho_{i-1} + \alpha_i^x$. Therefore, the target-specific coordinate kernel is:

$$K_i^{x,(j)}(\boldsymbol{\mu}_i \mid \boldsymbol{\theta}_{i-1}^x, \hat{\mathbf{x}}_M^{(j)}) = \mathcal{N}(\boldsymbol{\mu}_i; \bar{\boldsymbol{\mu}}_i^{(j)}, \boldsymbol{\Sigma}_i^x), \tag{17}$$

where $\tilde{\boldsymbol{\mu}}_i^{(j)} := \frac{\rho_{i-1}\boldsymbol{\mu}_{i-1} + \alpha_i^x\hat{\mathbf{x}}_M^{(j)}}{\rho_i}, \boldsymbol{\Sigma}_i^x := \frac{\alpha_i^x}{\rho_i^2}\mathbf{I}$. CompBFN composes the two kernels as

$$K_i(\boldsymbol{\mu}_i \mid \boldsymbol{\theta}_{i-1}^x) \propto [K_i^{(1)}(\boldsymbol{\mu}_i \mid \boldsymbol{\theta}_{i-1}^x, \hat{\mathbf{x}}_M^{(1)})K_i^{(2)}(\boldsymbol{\mu}_i \mid \boldsymbol{\theta}_{i-1}^x, \hat{\mathbf{x}}_M^{(2)})]^\eta = \mathcal{N}(\boldsymbol{\mu}_i; \frac{\tilde{\boldsymbol{\mu}}_i^{(1)} + \tilde{\boldsymbol{\mu}}_i^{(2)}}{2}, \frac{\boldsymbol{\Sigma}_i^x}{2\eta}). \tag{18}$$

Under a standard Gaussian prior, the corresponding coordinate flow is

$$p_F(\boldsymbol{\mu}_i \mid \hat{\mathbf{x}}_M^{(1)}, \hat{\mathbf{x}}_M^{(2)}; \tau_i) = \mathcal{N}(\boldsymbol{\mu}_i; \frac{\beta^x(\tau_i)}{1+\beta^x(\tau_i)}\frac{\hat{\mathbf{x}}_M^{(1)} + \hat{\mathbf{x}}_M^{(2)}}{2}, \frac{\beta^x(\tau_i)}{2\eta\,[1+\beta^x(\tau_i)]^2}\mathbf{I}). \tag{19}$$

Since $\boldsymbol{\theta}_i^v[n]$ lies on the probability simplex, we represent it using the log-ratio vector $\mathbf{r}_i[n]$ for convenience, whose k-th component is $(\mathbf{r}_i[n])_k := \log \frac{(\boldsymbol{\theta}_i^v[n])_k}{(\boldsymbol{\theta}_i^v[n])_K}$, $k \in \{1, \dots, K-1\}$. For expert j, the target-specific transition kernel induced by the categorical Bayesian update is

$$K_i^{v,(j)}(\mathbf{r}_i[n] \mid \mathbf{r}_{i-1}[n], \hat{\mathbf{v}}_M^{(j)}[n]) = \mathcal{N}(\mathbf{r}_i[n]; \mathbf{r}_{i-1}[n] + \alpha_i^v K \mathbf{D} \hat{\mathbf{v}}_M^{(j)}[n], \alpha_i^v K \mathbf{D}\mathbf{D}^\top), \tag{20}$$

where the contrast matrix $\mathbf{D} := [\mathbf{I}_{K-1} \quad -\mathbf{1}_{K-1}] \in \mathbb{R}^{(K-1)\times K}$. The powered product yields the composed categorical kernel

$$K_i\left(\mathbf{r}_i[n] \mid \mathbf{r}_{i-1}[n]\right) \propto \mathcal{N}(\mathbf{r}_i[n]; \mathbf{r}_{i-1}[n] + \alpha_i^v K \mathbf{D} \frac{\hat{\mathbf{v}}_M^{(1)}[n] + \hat{\mathbf{v}}_M^{(2)}[n]}{2}, \frac{\alpha_i^v K}{2\eta} \mathbf{D}\mathbf{D}^\top). \tag{21}$$

The accumulated flow induced by this composed kernel can be expressed using an equivalent K-dimensional Gaussian message

$$\mathbf{Y}_i^v[n] \sim \mathcal{N}(\beta^v(\tau_i)[K \frac{\hat{\mathbf{v}}_M^{(1)}[n] + \hat{\mathbf{v}}_M^{(2)}[n]}{2} - \mathbf{1}_K], \frac{\beta^v(\tau_i) K}{2\eta} \mathbf{I}_K). \tag{22}$$

Under the uniform categorical prior, the resulting atom-type flow is

$$p_F(\boldsymbol{\theta}_i^v[n] \mid \hat{\mathbf{v}}_M^{(1)}[n], \hat{\mathbf{v}}_M^{(2)}[n]; \tau_i, \eta) = \mathbb{E}_{\mathbf{Y}_i^v[n]}[\delta(\boldsymbol{\theta}_i^v[n] - \mathrm{softmax}(\mathbf{Y}_i^v[n]))]. \tag{23}$$

The equivalent message above represents the flow induced by composing two transition kernels. It does not imply that CompBFN fuses the target-specific sender distributions before the Bayesian update. The complete derivation is provided in Appendix D. Notably, under weights $w_1 = w_2 = 1/2$, CompBFN with $\eta = 1$ exactly recovers FusedBFN in both coordinate and atom-type flows. We discuss this intersection of the two composition families in Appendix D.3.

4.3 Dual-Pocket Alignment

Spatial alignment establishes a shared coordinate frame for the two target pockets and serves as a preprocessing component before generation. Following the ligand-prior alignment, we dock the same probe ligand into the two pockets and align its resulting poses. Let $\ell_a^{(j)} \in \mathbb{R}^3$ denote the coordinate of probe-ligand atom $a \in \{1, \dots, N_{\mathrm{pr}}\}$ in the pose associated with pocket j, where N_{pr} is the number of probe atoms. We estimate a rotation matrix $R^\star \in \mathrm{SO}(3)$ and a translation vector $b^\star \in \mathbb{R}^3$ through the chemically weighted registration $(R^\star, b^\star) = \underset{R\in\mathrm{SO}(3),\, b\in\mathbb{R}^3}{\arg\min} \sum_{a=1}^{N_{\mathrm{pr}}} \lambda_a \|\ell_a^{(2)} - (R\ell_a^{(1)} + b)\|_2^2$, where

$$\lambda_a = \begin{cases} \lambda_{\mathrm{heavy}}, & \text{if atom } a \text{ is a heavy atom,} \\ \lambda_{\mathrm{light}}, & \text{otherwise,} \end{cases} \qquad \lambda_{\mathrm{heavy}} > \lambda_{\mathrm{light}} > 0. \tag{24}$$

The motivation is that heavy atoms usually form the main topological scaffold of the ligand and provide more distinctive geometric constraints, whereas light atoms contribute less to transformation estimation and are more susceptible to local perturbations.

When a reliable probe ligand is unavailable, we additionally provide a prior-free alternative. It first extracts pocket-surface atoms using P2Rank (Polák et al., 2025), estimates a coarse rigid registration with RANSAC (Fischler & Bolles, 1981), and refines the transformation using ICP (Besl & McKay, 1992). All alignment hyperparameters are detailed in Appendix E.6.

5 Experiments

5.1 Experimental Setup

Dataset We evaluate all methods on the dual-target benchmark introduced by Zhou et al. (2024a). The benchmark contains 12,917 pharmacologically relevant target pairs, with a reference ligand provided for each target (details are provided in Appendix E.1). Following the established protocol, all target pairs are used for evaluation and each method generates 10 molecules for every target pair.

Table 2: Summary of the binding affinity and molecular properties of reference ligands and molecules generated in the dual-target setting. (↑)/(↓) indicates that higher/lower values are preferred. Top 2 results are highlighted with **bold text** and underlined text, respectively.

Methods	P-1 Vina Dock (↓)		P-2 Vina Dock (↓)		Max Vina Dock (↓)		Dual High Aff. (↑)		QED (↑)		SA (↑)		Diversity (↑)	
	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.
Reference	-7.67	-7.83	-4.90	-7.33	-4.37	-7.10	-	-	0.53	0.55	0.74	0.77	-	-
TargetDiff	-8.62	-8.61	-6.89	-7.68	-6.56	-7.39	44.6%	42.9%	0.50	0.51	0.58	0.58	0.70	0.71
MolCRAFT	**-9.01**	**-8.88**	-7.60	-7.92	-7.30	-7.63	52.2%	50.0%	0.52	0.53	0.66	0.65	0.70	0.71
DualDiff	-8.38	-8.48	-8.43	-8.52	-7.60	-7.84	51.2%	50.0%	0.55	0.57	0.59	0.58	0.66	0.67
CompDiff	-8.35	-8.48	-8.42	-8.53	-7.50	-7.80	51.2%	50.0%	0.55	0.56	0.59	0.59	**0.72**	**0.72**
FusedDiff	-8.31	-8.41	-8.37	-8.44	-7.57	-7.79	49.7%	50.0%	0.53	0.55	0.59	0.59	0.69	0.69
CompBFN	-8.46	-8.44	-8.46	-8.44	-7.77	-7.83	52.1%	50.0%	0.55	0.57	0.66	0.65	0.68	0.68
FusedBFN*	-8.52	-8.50	-8.51	-8.47	-7.81	-7.86	53.0%	55.6%	0.55	0.56	0.67	0.67	0.70	0.70
FusedBFN	-8.74	-8.70	-8.73	-8.69	-8.02	-8.05	57.8%	60.0%	0.56	0.58	0.69	0.69	0.69	0.69
FusedBFN^{Det}	-8.93	-8.90	**-8.91**	**-8.88**	**-8.17**	**-8.23**	**62.9%**	**71.4%**	**0.60**	**0.63**	**0.72**	**0.72**	0.71	0.71

Baselines We include **TargetDiff** and **MolCRAFT** as pretrained single-target references, **CompDiff** and **DualDiff** as diffusion-based dual-target baselines, and **FusedBFN** together with the variants **FusedDiff**, **CompBFN**, **FusedBFN***, and **FusedBFN**^{Det}. FusedDiff and CompDiff use the pretrained TargetDiff backbone, whereas all BFN-based variants use MolCRAFT. Following Zhou et al. (2024a), CompDiff, CompBFN, FusedDiff, and FusedBFN* use the ligand-prior alignment, whereas FusedBFN and FusedBFN-Det use our chemically weighted alignment. FusedBFN-Det further removes sampling noise by setting $c = 0$. We use $w_1 = w_2 = \frac{1}{2}$ and set $\eta = \frac{1}{2}$ following Zhou et al. (2024a). Full configurations are provided in Appendix E.5.

Evaluation metrics We evaluate binding affinity using AutoDock Vina (Trott & Olson, 2010). **P-1** and **P-2 Vina Dock** are the docking scores against the first and second targets. **Max Vina Dock** is the larger of the two scores and thus reflects binding to the less favored target. **Dual High Affinity** measures the proportion of generated molecules that outperform the corresponding reference ligands on both targets simultaneously. We additionally report **QED** (Bickerton et al., 2012) for drug-likeness, **SA** (Ertl & Schuffenhauer, 2009) for synthetic accessibility, and molecular **Diversity**.

5.2 Main Results

We evaluate all methods under the dual-target setting, with results reported in Table 2. **Single-target references:** TargetDiff and MolCRAFT perform well for single-target design, yet their docking results drop markedly on P-2, showing that single-target generation does not directly translate to effective dual-target design. **Diffusion-based comparison:** CompDiff performs slightly better overall than FusedDiff. Since their coordinate updates coincide under $w_1 = w_2 = \frac{1}{2}$ and $\eta = \frac{1}{2}$, we investigate the difference of the categorical trajectories in Section 5.3. **BFN-based comparison:** FusedBFN* outperforms CompBFN in overall dual-target binding, with significant paired improvements (Table 19). Under the current weight setting, the two methods share the same fused mean but differ in covariance, whose effect is isolated in Table 3. **Effect of chemically weighted alignment:** FusedBFN further improves all binding metrics over FusedBFN* (Table 19). Since FusedBFN and FusedBFN* only differ in alignment, this improvement demonstrates the contribution of our alignment strategy. **Effect of sampling concentration:** FusedBFN^{Det}, which retains the same fused signal and alignment as FusedBFN but removes sampling noise, achieves the strongest overall affinity performance. This full-benchmark result is consistent with Table 3, showing that reduced stochastic uncertainty can further benefit generation around the fused signal.

5.3 Analysis and Ablation Studies

Atom-Type Confidence during Sampling As detailed in Appendix C.4, FusedDiff and CompDiff use the same coordinate update under the matched settings, while their categorical rules differ in both composition order and effective expert exponent. To characterize the resulting atom-type distributions, we define the top-1/top-2 log-probability margin for ligand atom n at sampling step t as $m_{t,n}^{\log} = \log \frac{p_{t,n,(1)}}{p_{t,n,(2)}}$, where $p_{t,n,(1)}$ and $p_{t,n,(2)}$ are the largest and second-largest probabilities in the predicted categorical distribution, respectively. Averaged over 100 trajectories generated for 10 randomly selected target pairs, Figure 2 shows that FusedDiff exhibits consistently smaller mar-

gins than CompDiff, particularly during the later sampling steps. This indicates that FusedDiff has a softer categorical distribution with lower confidence in the top predicted atom type, as theoretically analyzed in Appendix C.4. Additional experiments in Appendix F.4 show that differences in categorical concentration alone are insufficient to explain the performance gap between the two methods.

Table 3: Effect of the covariance scale c on FusedBFN*.

c	P-1 Vina Dock (↓)		P-2 Vina Dock (↓)		Max Vina Dock (↓)		Dual High Aff. (↑)		QED (↑)		SA (↑)		Diversity (↑)	
	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.
1.5	-8.33	-8.31	-8.35	-8.33	-7.73	-7.75	47.1%	44.4%	0.53	0.54	0.63	0.62	0.67	0.67
1.0	-8.53	-8.52	-8.57	-8.53	-7.90	-7.94	50.6%	50.0%	0.54	0.55	0.65	0.64	0.68	0.67
0.5	-8.60	-8.57	-8.63	-8.57	-7.95	-7.99	56.5%	60.0%	0.53	0.55	0.66	0.66	0.69	0.69
0.25	-8.94	-8.93	-8.99	-8.95	-8.24	-8.30	60.0%	70.0%	0.56	0.58	0.69	0.69	0.68	0.69
0	-8.98	-8.93	-9.02	-8.97	-8.29	-8.33	61.8%	70.0%	0.59	0.62	0.71	0.70	0.70	0.70

Study of Sampling Concentration Table 3 examines the effect of sampling concentration under the BFN backbone using 1,000 randomly selected target pairs, with 10 molecules generated for each pair. The fused means are kept fixed while varying the covariance scale c. Under the matched setting, $c = 0.5$ corresponds to FusedBFN*, whereas $c = 1$ matches the covariance scale of CompBFN with $\eta = 1/2$. We therefore vary c to disentangle the effect of sampling concentration from the probabilistic location of composition.

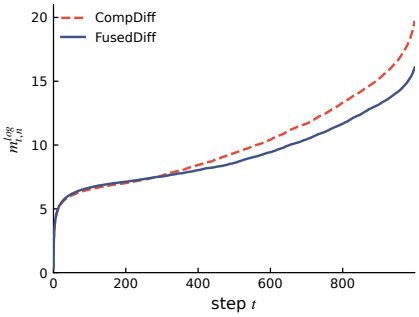

Figure 2: Atom-type confidence during sampling measured by the mean $m_{t,n}^{\log}$.

The results show that all affinity-related metrics improve as c decreases. When the experts agree and the fused center is reliable, reducing c suppresses stochastic deviations from their shared direction and prevents such perturbations from propagating through subsequent sampling steps. Notably, diversity remains high at $c = 0$, indicating that this covariance is not the sole source of molecular diversity. Nevertheless, for target pairs with strong expert disagreement, the deterministic fused center may be an unreliable compromise, and moderate covariance can instead be beneficial (Appendix F.5).

Effect of Dual-Target Alignment Table 5 compares different alignment strategies on 10,000 molecules sampled across 1,000 randomly selected target pairs, with implementation details provided in Appendix E.6. Lig-ChemPrior achieves the best overall performance, which shows that incorporating the chemical semantics of different atoms leads to a more stable alignment pattern. It also explains the additional improvement from FusedBFN* to FusedBFN observed in Table 2. Surface-RI remains competitive even without interaction priors, offering a practical solution when reliable probe ligands are unavailable and providing a simplified alignment pipeline.

6 Conclusion

We present FusedBFN, a zero-shot framework for dual-target molecular generation that fuses target-specific sender likelihoods through wPoE before a shared Bayesian update. By further introducing CompBFN and FusedDiff, together with CompDiff, we systematically study different probabilistic formulations of dual-target composition under both BFN and diffusion backbones. Our analysis reveals that distinct composition constructions do not necessarily induce distinct sampling processes: under BFN, FusedBFN and CompBFN can become exactly equivalent under matched settings, whereas their default configurations share the same fused center and differ primarily in covariance; under diffusion, their coordinate transitions can coincide while categorical updates remain distinct. Experiments show strong dual-target binding performance of FusedBFN, while further analyses identify sampling concentration and chemically weighted pocket alignment as important contributors to the observed improvements. Reducing sampling uncertainty around a reliable fused signal improves average performance, although moderate covariance can remain beneficial when the target-specific experts disagree. These results suggest that the behavior of dual-target composition depends jointly on composition level, induced uncertainty, and the generative framework. Future

work will explore adaptive fusion weights and covariance scales tailored to each target pair, and validate generated candidates experimentally.

AI USE STATEMENT

We used generative AI tools to assist with checking mathematical derivations, designing experiments and ablation studies, implementing and modifying research code. We also used these tools for figure refinement, and drafting and editing the manuscript. All AI-assisted experimental designs, code, figures, and text were reviewed and verified by the authors, and literature claims and citations were checked against the original sources. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work is entirely computational and does not involve human subjects, personal data, or animal experiments. The proposed framework is intended to support computational drug discovery by generating molecular candidates conditioned on protein targets. Generated molecules should be regarded only as computational hypotheses, as predicted binding affinity does not establish biochemical activity, safety, or clinical efficacy. Any downstream use therefore requires appropriate expert review and experimental validation.

REPRODUCIBILITY STATEMENT

We provide detailed theoretical derivations, sampling algorithms, and implementation settings to facilitate reproduction of our results. Appendices A–D contain the backbone formulations and complete derivations and sampling procedures for FusedBFN, FusedDiff, and CompBFN. Appendix E describes the dataset, featurization, network and sampling configurations, and dual-pocket alignment procedures, while Appendix F reports additional experiments, ablations, and statistical tests. All experiments use publicly available pretrained MolCRAFT and TargetDiff checkpoints without retraining or fine-tuning. Anonymous source code is available via the link provided in the abstract.

REFERENCES

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, pp. 586–606. Spie, 1992.

G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.

Yanshuai Cao and David J Fleet. Generalized product of experts for automatic and principled fusion of gaussian process predictions. *arXiv preprint arXiv:1410.7827*, 2014.

Seungyeon Choi, Hwanhee Kim, Chihyun Park, Dahyeon Lee, Seungyong Lee, Yoonju Kim, Hyoungjoon Park, Sein Kwon, Youngwan Jo, and Sanghyun Park. Controllable 3d molecular generation for structure-based drug design through bayesian flow networks and gradient integration. *Advances in Neural Information Processing Systems*, 38:125612–125647, 2026.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1:1–11, 2009.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

Alex Graves, Rupesh Kumar Srivastava, Timothy Atkinson, and Faustino Gomez. Bayesian flow networks. *arXiv preprint arXiv:2308.07037*, 2023.

Siyi Gu, Minkai Xu, Alexander Powers, Weili Nie, Tomas Geffner, Karsten Kreis, Jure Leskovec, Arash Vahdat, and Stefano Ermon. Aligning target-aware molecule diffusion models with exact energy optimization. *Advances in Neural Information Processing Systems*, 37:44040–44063, 2024.

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.

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. *arXiv preprint arXiv:2403.07902*, 2024.

Charles Harris, Kieran Didi, Arian Rokkum Jamasb, Chaitanya K Joshi, Simon V Mathis, Pietro Lio, and Tom Leon Blundell. Posecheck: Generative models for 3d structure-based drug design produce unrealistic poses. 2023.

Bing He, Cheng Lu, Guang Zheng, Xiaojuan He, Maolin Wang, Gao Chen, Ge Zhang, and Aiping Lu. Combination therapeutics in complex diseases. *Journal of cellular and molecular medicine*, 20(12):2231–2240, 2016.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Laura Isigkeit, Tim Hörmann, Espen Schallmayer, Katharina Scholz, Felix F Lillich, Johanna HM Ehrler, Benedikt Hufnagel, Jasmin Büchner, Julian A Marschner, Jörg Pabel, et al. Automated design of multi-target ligands by generative deep learning. *Nature Communications*, 15(1):7946, 2024.

Craig Knox, Mike Wilson, Christen M Klinger, Mark Franklin, Eponine Oler, Alex Wilson, Allison Pon, Jordan Cox, Na Eun Chin, Seth A Strawbridge, et al. Drugbank 6.0: the drugbank knowledgebase for 2024. *Nucleic acids research*, 52(D1):D1265–D1275, 2024.

Haitao Lin, Guojiang Zhao, Odin Zhang, Yufei Huang, Lirong Wu, Cheng Tan, Zicheng Liu, Zhifeng Gao, and Stan Z Li. Cbgbench: fill in the blank of protein-molecule complex binding graph. In *International Conference on Learning Representations*, volume 2025, pp. 19861–19887, 2025.

Hui Liu, Wenhao Zhang, Bo Zou, Jinxian Wang, Yuanyuan Deng, and Lei Deng. Drugcombdb: a comprehensive database of drug combinations toward the discovery of combinatorial therapy. *Nucleic acids research*, 48(D1):D871–D881, 2020.

Zhihai Liu, Minyi Su, Li Han, Jie Liu, Qifan Yang, Yan Li, and Renxiao Wang. Forging the basis for developing protein–ligand interaction scoring functions. *Accounts of chemical research*, 50 (2):302–309, 2017.

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.

Andrew T McNutt, Yanjing Li, Rocco Meli, Rishal Aggarwal, and David Ryan Koes. Gnina 1.3: the next increment in molecular docking with deep learning. *Journal of Cheminformatics*, 17(1): 28, 2025.

Brenton P Munson, Michael Chen, Audrey Bogosian, Jason F Kreisberg, Katherine Licon, Ruben Abagyan, Brent M Kuenzi, and Trey Ideker. De novo generation of multi-target compounds using deep generative chemistry. *Nature Communications*, 15(1):3636, 2024.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

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*, pp. 17644–17655. PMLR, 2022.

Lukáš Polák, Petr Škoda, Kamila Riedlová, Radoslav Krivák, Marian Novotnỳ, and David Hoksza. Prankweb 4: a modular web server for protein–ligand binding site prediction and downstream analysis. *Nucleic Acids Research*, 53(W1):W466–W471, 2025.

Hao Qian, Wenjing Huang, Shikui Tu, and Lei Xu. Kgdiff: towards explainable target-aware molecule generation with knowledge guidance. *Briefings in Bioinformatics*, 25(1):bbad435, 2024.

Keyue Qiu, Yuxuan Song, Jie Yu, Hongbo Ma, Ziyao Cao, Zhilong Zhang, Yushuai Wu, Mingyue Zheng, Hao Zhou, and Wei-Ying Ma. Empower structure-based molecule optimization with gradient guided bayesian flow networks. In *International Conference on Machine Learning*, pp. 50645–50671. PMLR, 2025.

Yanru Qu, Keyue Qiu, Yuxuan Song, Jingjing Gong, Jiawei Han, Mingyue Zheng, Hao Zhou, and Wei-Ying Ma. Molcraft: Structure-based drug design in continuous parameter space. *arXiv preprint arXiv:2404.12141*, 2024.

Rona R Ramsay, Marija R Popovic-Nikolic, Katarina Nikolic, Elisa Uliassi, and Maria Laura Bolognesi. A perspective on multi-target drug discovery and design for complex diseases. *Clinical and translational medicine*, 7(1):3, 2018.

Thibaud Southiratn, Bonil Koo, Yijingxiu Lu, and Sun Kim. Combimots: Combinatorial multi-objective tree search for dual-target molecule generation. *arXiv preprint arXiv:2604.23307*, 2026.

Sanjana Srinivasan and Jürgen Bajorath. Generation of dual-target compounds using a transformer chemical language model. *Cell Reports Physical Science*, 5(11), 2024.

Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022.

Qihang Zhang, Muchen Li, Ziao Wang, Renjie Liao, and Lele Wang. Test-time steering for lossless text compression via weighted product of experts. In *EMNLP (Findings)*, pp. 2076–2088, 2025.

Ruixiang Zhang, Masanori Koyama, and Katsuhiko Ishiguro. Learning structured latent factors from dependent data: A generative model framework from information-theoretic perspective. In *International Conference on Machine Learning*, pp. 11141–11152. PMLR, 2020.

Zaixi Zhang and Qi Liu. Learning subpocket prototypes for generalizable structure-based drug design. In *International Conference on Machine Learning*, pp. 41382–41398. PMLR, 2023.

Zaixi Zhang, Yaosen Min, Shuxin Zheng, and Qi Liu. Molecule generation for target protein binding with structural motifs. In *The eleventh international conference on learning representations*, 2023.

Zaixi Zhang, Mengdi Wang, and Qi Liu. Flexsbdd: Structure-based drug design with flexible protein modeling. *Advances in Neural Information Processing Systems*, 37:53918–53944, 2024.

Jingyuan Zhou, Dengwei Zhao, Hao Qian, Shikui Tu, and Lei Xu. Multi-objective structure-based drug design using causal discovery. *IEEE Transactions on Computational Biology and Bioinformatics*, 2025.

Jingyuan Zhou, Hao Qian, Shikui Tu, and Lei Xu. Prior-guided flow matching for target-aware molecule design with learnable atom number. *Advances in Neural Information Processing Systems*, 38:121513–121543, 2026.

Xiangxin Zhou, Jiaqi Guan, Yijia Zhang, Xingang Peng, Liang Wang, and Jianzhu Ma. Reprogramming pretrained target-specific diffusion models for dual-target drug design. *Advances in Neural Information Processing Systems*, 37:87255–87281, 2024a.

Ying Zhou, Yintao Zhang, Donghai Zhao, Xinyuan Yu, Xinyi Shen, Yuan Zhou, Shanshan Wang, Yunqing Qiu, Yuzong Chen, and Feng Zhu. Ttd: Therapeutic target database describing target druggability information. *Nucleic acids research*, 52(D1):D1465–D1477, 2024b.

A Diffusion Models and Bayesian Flow Networks for SBDD

Diffusion models (DMs) (Ho et al., 2020) define a forward corruption process that progressively perturbs data and learn reverse transitions that recover the data distribution. In 3D molecular generation, continuous coordinates are commonly corrupted with Gaussian noise, whereas discrete atom types follow a categorical diffusion process. TargetDiff (Guan et al., 2023) parameterizes the corresponding reverse transitions with an SE(3)-equivariant network that jointly predicts clean coordinates and atom type probabilities conditioned on a protein pocket. By contrast, Bayesian Flow Networks (BFNs) (Graves et al., 2023) formulate generation through communication between a sender and a receiver: sender distributions transmit noisy observations, Bayesian inference updates the parameters of an input distribution, and a neural network predicts clean data from the evolving parameter state. MolCRAFT (Qu et al., 2024) instantiates this framework for SBDD by jointly modeling continuous coordinates and discrete atom types in a continuous parameter space. These different probabilistic structures provide complementary settings for examining where target-specific information should be composed.

A.1 Single-Target Diffusion Models

Following Guan et al. (2023), this section summarizes the single-target diffusion formulation. Recall that $t \in \{1, \dots, T\}$ denotes the diffusion time-step index and is reserved exclusively for diffusion, whereas i and τ_i denote, respectively, the discrete update-step index and continuous schedule time of a BFN. For a clean ligand molecule $\mathbf{m}_0 = \mathbf{m} = [\mathbf{x}_{M,0}, \mathbf{v}_{M,0}]$, let $\mathbf{m}_t = [\mathbf{x}_{M,t}, \mathbf{v}_{M,t}]$ denote its noisy state at diffusion step t.

Forward diffusion At diffusion step t, the noise rate is $b_t \in (0, 1)$, and define

$$a_t := 1 - b_t, \qquad \bar{a}_t := \prod_{r=1}^{t} a_r, \qquad \bar{a}_0 := 1.$$

Although the schedules can be different in practice, we use the same notation for conciseness.

The forward transition factorizes over the coordinate and atom-type modalities:

$$q(\mathbf{m}_t \mid \mathbf{m}_{t-1}) = q(\mathbf{x}_{M,t} \mid \mathbf{x}_{M,t-1}) q(\mathbf{v}_{M,t} \mid \mathbf{v}_{M,t-1}).$$

For coordinates, the one-step transition is

$$q(\mathbf{x}_{M,t} \mid \mathbf{x}_{M,t-1}) = \mathcal{N}\left(\mathbf{x}_{M,t}; \sqrt{a_t}\, \mathbf{x}_{M,t-1}, b_t \mathbf{I}\right). \tag{25}$$

The corresponding marginal distribution at any diffusion step t is

$$q(\mathbf{x}_{M,t} \mid \mathbf{x}_{M,0}) = \mathcal{N}\left(\mathbf{x}_{M,t}; \sqrt{\bar{a}_t}\, \mathbf{x}_{M,0}, (1 - \bar{a}_t)\mathbf{I}\right). \tag{26}$$

Let $\mathbf{u}_K := \frac{1}{K}\mathbf{1}_K \in \Delta^{K-1}$ denote the uniform probability vector over the K categories. For ligand atom $n \in \{1, \dots, N_M\}$, the one-step categorical transition is

$$q\left(\mathbf{v}_{M,t}[n] \mid \mathbf{v}_{M,t-1}[n]\right) = \mathrm{Cat}\left(\mathbf{v}_{M,t}[n]; a_t \mathbf{v}_{M,t-1}[n] + b_t \mathbf{u}_K\right).$$

The transition factorizes over ligand atoms and its closed-form marginal is

$$q\left(\mathbf{v}_{M,t}[n] \mid \mathbf{v}_{M,0}[n]\right) = \mathrm{Cat}\left(\mathbf{v}_{M,t}[n]; \bar{a}_t \mathbf{v}_{M,0}[n] + (1 - \bar{a}_t)\mathbf{u}_K\right). \tag{27}$$

Reverse posteriors Given the clean coordinates $\mathbf{x}_{M,0}$ and the noisy coordinates $\mathbf{x}_{M,t}$, the exact coordinate posterior is

$$q\left(\mathbf{x}_{M,t-1} \mid \mathbf{x}_{M,t}, \mathbf{x}_{M,0}\right) = \mathcal{N}\left(\mathbf{x}_{M,t-1}; \widetilde{\boldsymbol{\mu}}_t(\mathbf{x}_{M,t}, \mathbf{x}_{M,0}), \widetilde{b}_t \mathbf{I}\right), \tag{28}$$

where

$$\widetilde{\boldsymbol{\mu}}_t(\mathbf{x}_{M,t}, \mathbf{x}_{M,0}) = \frac{\sqrt{\bar{a}_{t-1}}\, b_t}{1 - \bar{a}_t}\mathbf{x}_{M,0} + \frac{\sqrt{a_t}(1 - \bar{a}_{t-1})}{1 - \bar{a}_t}\mathbf{x}_{M,t}, \qquad \widetilde{b}_t = \frac{1 - \bar{a}_{t-1}}{1 - \bar{a}_t} b_t. \tag{29}$$

For the discrete modality, define the unnormalized vector

$$\mathbf{c}^{\star}_{t,n}\left(\mathbf{v}_{M,t}[n], \mathbf{v}_{M,0}[n]\right) := \left[a_t \mathbf{v}_{M,t}[n] + b_t \mathbf{u}_K\right] \odot \left[\bar{a}_{t-1} \mathbf{v}_{M,0}[n] + (1 - \bar{a}_{t-1})\mathbf{u}_K\right],$$

where $\odot$ denotes elementwise multiplication. Its normalized categorical parameter is

$$\widetilde{\mathbf{c}}_{t,n}\left(\mathbf{v}_{M,t}[n], \mathbf{v}_{M,0}[n]\right) = \frac{\mathbf{c}^{\star}_{t,n}\left(\mathbf{v}_{M,t}[n], \mathbf{v}_{M,0}[n]\right)}{\mathbf{1}_K^{\top}\mathbf{c}^{\star}_{t,n}\left(\mathbf{v}_{M,t}[n], \mathbf{v}_{M,0}[n]\right)}.$$

The categorical posterior is therefore

$$q\left(\mathbf{v}_{M,t-1}[n] \mid \mathbf{v}_{M,t}[n], \mathbf{v}_{M,0}[n]\right) = \mathrm{Cat}\left(\mathbf{v}_{M,t-1}[n]; \widetilde{\mathbf{c}}_{t,n}\left(\mathbf{v}_{M,t}[n], \mathbf{v}_{M,0}[n]\right)\right).$$

Target-aware generation Given a single target pocket $\mathbf{p}$, the pretrained SE(3)-equivariant network Φ_ω predicts

$$[\hat{\mathbf{x}}_{M,0}, \hat{\mathbf{v}}_{M,0}] = \Phi_\omega(\mathbf{m}_t, \mathbf{p}, t).$$

The predicted clean coordinates are substituted into the Gaussian posterior mean:

$$p_\omega(\mathbf{x}_{M,t-1} \mid \mathbf{m}_t, \mathbf{p}) = \mathcal{N}\left(\mathbf{x}_{M,t-1}; \boldsymbol{\mu}_\omega\left(\mathbf{x}_{M,t}, \hat{\mathbf{x}}_{M,0}\right), \sigma_t^2\mathbf{I}\right).$$

Likewise, substituting the predicted probabilities $\hat{\mathbf{v}}_{M,0}$ into the categorical posterior map gives

$$p_\omega\left(\mathbf{v}_{M,t-1}[n] \mid \mathbf{m}_t, \mathbf{p}\right) = \mathrm{Cat}\left(\mathbf{v}_{M,t-1}[n]; \mathbf{c}_{\omega,n}\left(\mathbf{v}_{M,t}[n], \hat{\mathbf{v}}_{M,0}[n]\right)\right).$$

The resulting single-target generative process is

$$p_\omega(\mathbf{m}_{t-1} \mid \mathbf{m}_t, \mathbf{p}) = p_\omega(\mathbf{x}_{M,t-1} \mid \mathbf{m}_t, \mathbf{p}) \prod_{n=1}^{N_M} p_\omega\left(\mathbf{v}_{M,t-1}[n] \mid \mathbf{m}_t, \mathbf{p}\right),$$

which starts from the prior $\mathbf{x}_{M,T} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{v}_{M,T}[n] \sim \mathrm{Cat}(\mathbf{u}_K)$.

A.2 Single-Target Bayesian Flow Networks

In this section, we provide a detailed description of the BFN implementation for single-target drug design following Qu et al. (2024). The input distribution factorizes over molecular coordinates and atom types:

$$p_I(\mathbf{m} \mid \boldsymbol{\theta}_i) = p_I(\mathbf{x}_M \mid \boldsymbol{\theta}_i^x)\, p_I(\mathbf{v}_M \mid \boldsymbol{\theta}_i^v).$$

Although the conditional pocket $\mathbf{p}$ is retained in the update and flow distributions in the main text to emphasize target-aware generation, the sender distribution and the Bayesian update are conditionally independent of the pocket once $\mathbf{m}$ is given. We therefore omit $\mathbf{p}$ below.

Continuous coordinates The coordinate input distribution is an isotropic Gaussian,

$$p_I(\mathbf{x}_M \mid \boldsymbol{\theta}_i^x) = \mathcal{N}\left(\mathbf{x}_M; \boldsymbol{\mu}_i, \rho_i^{-1}\mathbf{I}\right), \qquad \boldsymbol{\theta}_i^x := \{\boldsymbol{\mu}_i, \rho_i\},$$

where $\boldsymbol{\mu}_i$ is learned and ρ_i is specified by the noise factor α_i^x. The initial parameters are $\boldsymbol{\mu}_0 = \mathbf{0}$ and $\rho_0 = 1$, corresponding to a standard Gaussian prior.

Given the clean coordinates $\mathbf{x}_M$, the sender distribution is

$$p_S(\mathbf{y}_i^x \mid \mathbf{x}_M; \alpha_i^x) = \mathcal{N}\left(\mathbf{y}_i^x; \mathbf{x}_M, (\alpha_i^x)^{-1}\mathbf{I}\right). \tag{30}$$

The corresponding Bayesian update is

$$\rho_i = \rho_{i-1} + \alpha_i^x, \qquad \boldsymbol{\mu}_i = \frac{\rho_{i-1}\boldsymbol{\mu}_{i-1} + \alpha_i^x \mathbf{y}_i^x}{\rho_i}. \tag{31}$$

Since $\mathbf{y}_i^x$ is Gaussian, the update function in equation 31 induces

$$\begin{aligned} &p_U\left(\boldsymbol{\theta}_i^x \mid \boldsymbol{\theta}_{i-1}^x, \mathbf{x}_M; \alpha_i^x\right) \\ &\quad = \delta\left(\rho_i - \rho_{i-1} - \alpha_i^x\right) \mathcal{N}\left(\boldsymbol{\mu}_i; \frac{\rho_{i-1}\boldsymbol{\mu}_{i-1} + \alpha_i^x \mathbf{x}_M}{\rho_i}, \frac{\alpha_i^x}{\rho_i^2}\mathbf{I}\right). \end{aligned}$$

Define the accumulated coordinate accuracy as $\beta^x(\tau_i) := \sum_{r=1}^{i} \alpha_r^x$. Marginalizing the intermediate parameters gives

$$\rho_i = 1 + \beta^x(\tau_i), \quad p_F(\boldsymbol{\mu}_i \mid \mathbf{x}_M; \tau_i) = \mathcal{N}\left(\boldsymbol{\mu}_i; \frac{\beta^x(\tau_i)}{1 + \beta^x(\tau_i)}\mathbf{x}_M, \frac{\beta^x(\tau_i)}{\left[1 + \beta^x(\tau_i)\right]^2}\mathbf{I}\right).$$

Discrete atom types For ligand atom $n \in \{1, \dots, N_M\}$, the categorical input parameter is $\boldsymbol{\theta}_i^v[n] \in \Delta^{K-1}$, where Δ^{K-1} is the $(K-1)$-dimensional probability simplex. The input distribution factorizes over ligand atoms:

$$p_I(\mathbf{v}_M \mid \boldsymbol{\theta}_i^v) = \prod_{n=1}^{N_M} \mathrm{Cat}\left(\mathbf{v}_M[n]; \boldsymbol{\theta}_i^v[n]\right), \qquad \boldsymbol{\theta}_0^v[n] = \frac{1}{K}\mathbf{1}_K,$$

where $\mathbf{1}_K \in \mathbb{R}^K$ is the all-ones vector. The sender message associated with the atom type of atom n is $\mathbf{y}_i^v[n] \in \mathbb{R}^K$, with sender distribution

$$p_S\left(\mathbf{y}_i^v[n] \mid \mathbf{v}_M[n]; \alpha_i^v\right) = \mathcal{N}\left(\mathbf{y}_i^v[n]; \alpha_i^v\left(K\mathbf{v}_M[n] - \mathbf{1}_K\right), \alpha_i^v K \mathbf{I}_K\right), \tag{32}$$

where $\mathbf{I}_K$ is the $K \times K$ identity matrix. Given $\mathbf{y}_i^v[n]$, the Bayesian update is

$$\left(\boldsymbol{\theta}_i^v[n]\right)_k = \frac{\exp\left(\left(\mathbf{y}_i^v[n]\right)_k\right)\left(\boldsymbol{\theta}_{i-1}^v[n]\right)_k}{\sum_{\ell=1}^K \exp\left(\left(\mathbf{y}_i^v[n]\right)_\ell\right)\left(\boldsymbol{\theta}_{i-1}^v[n]\right)_\ell}, \qquad k \in \{1, \dots, K\}.$$

Equivalently,

$$\boldsymbol{\theta}_i^v[n] = \mathrm{softmax}\left(\log \boldsymbol{\theta}_{i-1}^v[n] + \mathbf{y}_i^v[n]\right). \tag{33}$$

Define the accumulated sender message as $\mathbf{Y}_i^v[n] := \sum_{r=1}^i \mathbf{y}_r^v[n]$, which, by the additivity of independent Gaussian messages, satisfies

$$\mathbf{Y}_i^v[n] \sim \mathcal{N}\left(\beta^v(\tau_i)\left(K\mathbf{v}_M[n] - \mathbf{1}_K\right), \beta^v(\tau_i) K \mathbf{I}_K\right),$$

where $\beta^v(\tau_i) := \sum_{r=1}^i \alpha_r^v$.

Since the initial parameter is uniform, repeated application of equation 33 yields $\boldsymbol{\theta}_i^v[n] = \mathrm{softmax}\left(\mathbf{Y}_i^v[n]\right)$. Consequently, the atom-type flow distribution is

$$\begin{aligned} &p_F\left(\boldsymbol{\theta}_i^v[n] \mid \mathbf{v}_M[n]; \tau_i\right) \\ &\quad = \mathbb{E}_{\mathbf{Y}_i^v[n] \sim \mathcal{N}\left(\beta^v(\tau_i)\left(K\mathbf{v}_M[n] - \mathbf{1}_K\right), \beta^v(\tau_i) K \mathbf{I}_K\right)}\left[\delta\left(\boldsymbol{\theta}_i^v[n] - \mathrm{softmax}\left(\mathbf{Y}_i^v[n]\right)\right)\right]. \end{aligned}$$

Single-target receiver and sampling At BFN update step i, the pretrained SE(3)-equivariant GNN $\boldsymbol{\Psi}$ outputs

$$\hat{\mathbf{m}} = [\hat{\mathbf{x}}_M, \hat{\mathbf{v}}_M] = \boldsymbol{\Psi}\left(\boldsymbol{\theta}_{i-1}, \mathbf{p}, \tau_{i-1}\right).$$

These predictions parameterize the output distribution p_O and, through the sender distributions in equation 30 and equation 32, the receiver distribution p_R.

To avoid introducing excessive noise, Qu et al. (2024) adopt the parameter-space sampling strategy, in which information flows as $\boldsymbol{\theta}_{i-1} \xrightarrow{\boldsymbol{\Psi}} \hat{\mathbf{m}} \xrightarrow{p_F} \boldsymbol{\theta}_i$, thereby bypassing the noisy data sampling required by Bayesian updating.

B Full Derivation of FusedBFN

This appendix provides the intermediate derivations omitted from Section 4.1, together with the sampling procedure and the SE(3)-equivariance proof. The closed-form flow distributions are derived conditionally on fixed target-specific network predictions. During sampling, these predictions are recomputed from the current parameter state at every step and substituted into the corresponding flow distribution.

B.1 Derivation of Continuous Coordinates

Recall that the normalized fused distribution $L(\mathbf{x}_M; \mathbf{y}_i^{x,(1)}, \mathbf{y}_i^{x,(2)}, \alpha_i^x)$ has precision $S\alpha_i^x$. Combining it with the input distribution $p_I(\mathbf{x}_M \mid \boldsymbol{\theta}_{i-1}^x) = \mathcal{N}\left(\mathbf{x}_M; \boldsymbol{\mu}_{i-1}, \rho_{i-1}^{-1}\mathbf{I}\right)$ gives the deterministic precision update $\rho_i = \rho_{i-1} + S\alpha_i^x$ and the corresponding mean update

$$\boldsymbol{\mu}_i = \frac{\rho_{i-1}\boldsymbol{\mu}_{i-1} + S\alpha_i^x \widetilde{\mathbf{y}}_i^x}{\rho_i}.$$

To obtain the update distribution, we first rescale the Gaussian message in equation 8. Since an affine transformation of a Gaussian variable remains Gaussian,

$$S\alpha_i^x \widetilde{\mathbf{y}}_i^x \sim \mathcal{N}\Big(\alpha_i^x \left(w_1 \hat{\mathbf{x}}_M^{(1)} + w_2 \hat{\mathbf{x}}_M^{(2)}\right), \alpha_i^x Q \mathbf{I}\Big).$$

Substituting this result into the mean update and marginalizing the sender messages yields

$$p_U\Big(\boldsymbol{\mu}_i \mid \boldsymbol{\theta}_{i-1}^x, \hat{\mathbf{x}}_M^{(1)}, \hat{\mathbf{x}}_M^{(2)}; \alpha_i^x\Big) = \mathcal{N}\left(\boldsymbol{\mu}_i; \frac{\rho_{i-1}\boldsymbol{\mu}_{i-1} + \alpha_i^x \left(w_1 \hat{\mathbf{x}}_M^{(1)} + w_2 \hat{\mathbf{x}}_M^{(2)}\right)}{\rho_i}, \frac{\alpha_i^x Q}{\rho_i^2}\mathbf{I}\right).$$

The precision component can be included explicitly as $\delta(\rho_i - \rho_{i-1} - S\alpha_i^x)$. Define the natural parameter $\mathbf{z}_i := \rho_i \boldsymbol{\mu}_i$. The Bayesian update above can then be written as

$$\mathbf{z}_i = \mathbf{z}_{i-1} + S\alpha_i^x \widetilde{\mathbf{y}}_i^x.$$

Conditioned on the two target-specific coordinate predictions, the natural-parameter increment satisfies

$$\mathbf{z}_i - \mathbf{z}_{i-1} \sim \mathcal{N}\Big(\alpha_i^x \left(w_1 \hat{\mathbf{x}}_M^{(1)} + w_2 \hat{\mathbf{x}}_M^{(2)}\right), \alpha_i^x Q \mathbf{I}\Big).$$

Because the sender noises at different update steps are independent, the natural-parameter increments are additive. Using the standard coordinate prior $\boldsymbol{\mu}_0 = \mathbf{0}$ and $\rho_0 = 1$, we therefore obtain

$$\mathbf{z}_i = \sum_{r=1}^{i} S\alpha_r^x \widetilde{\mathbf{y}}_r^x \sim \mathcal{N}\Big(\beta^x(\tau_i) \left(w_1 \hat{\mathbf{x}}_M^{(1)} + w_2 \hat{\mathbf{x}}_M^{(2)}\right), \beta^x(\tau_i) Q \mathbf{I}\Big).$$

Meanwhile, repeated application of the precision update gives

$$\rho_i = \rho_0 + S \sum_{r=1}^{i} \alpha_r^x = 1 + S\beta^x(\tau_i).$$

Finally, since $\boldsymbol{\mu}_i = \mathbf{z}_i / \rho_i$, applying the corresponding linear transformation to the Gaussian distribution of $\boldsymbol{z}_i$ gives

$$p_F\Big(\boldsymbol{\mu}_i \mid \hat{\mathbf{x}}_M^{(1)}, \hat{\mathbf{x}}_M^{(2)}, \widetilde{\mathbf{p}}^{(1)}, \widetilde{\mathbf{p}}^{(2)}; \tau_i\Big) = \mathcal{N}\left(\boldsymbol{\mu}_i; \frac{\beta^x(\tau_i) \left(w_1 \hat{\mathbf{x}}_M^{(1)} + w_2 \hat{\mathbf{x}}_M^{(2)}\right)}{1 + S\beta^x(\tau_i)}, \frac{\beta^x(\tau_i) Q}{[1 + S\beta^x(\tau_i)]^2}\mathbf{I}\right),$$

which is equation 9 in the main text. For $w_1 = 1$ and $w_2 = 0$, this result reduces to the standard single-target BFN coordinate flow. Under the default setting $w_1 = w_2 = \frac{1}{2}$, $S = 1$ preserves the pretrained BFN accuracy scale, while $Q = \frac{1}{2}$ determines the variance induced by aggregating the two independent target-specific messages.

B.2 Derivation of Discrete Atom Types

We now derive the atom-type flow in equation 13 from the fused message in equation 12. Conditioned on the two target-specific atom-type predictions, equation 12 gives

$$\widetilde{\mathbf{y}}_i^v[n] \sim \mathcal{N}\Big(\alpha_i^v \left[K \left(w_1 \hat{\mathbf{v}}_M^{(1)}[n] + w_2 \hat{\mathbf{v}}_M^{(2)}[n]\right) - S\mathbf{1}_K\right], \alpha_i^v K Q \mathbf{I}_K\Big).$$

Given the fused message $\widetilde{\mathbf{y}}_i^v[n]$, the categorical Bayesian update is

$$\boldsymbol{\theta}_i^v[n] = \mathrm{softmax}\big(\log \boldsymbol{\theta}_{i-1}^v[n] + \widetilde{\mathbf{y}}_i^v[n]\big). \tag{34}$$

To marginalize the intermediate parameters, we first express the result after repeated updates. At the preceding step, we have

$$\boldsymbol{\theta}_{i-1}^v[n] = \mathrm{softmax}\big(\log \boldsymbol{\theta}_{i-2}^v[n] + \widetilde{\mathbf{y}}_{i-1}^v[n]\big). \tag{35}$$

Substituting equation 35 into equation 34 gives

$$\boldsymbol{\theta}_i^v[n] = \mathrm{softmax}\big(\log \boldsymbol{\theta}_{i-2}^v[n] + \widetilde{\mathbf{y}}_{i-1}^v[n] + \widetilde{\mathbf{y}}_i^v[n]\big).$$

Applying this argument recursively gives

$$\boldsymbol{\theta}_i^v[n] = \mathrm{softmax}\left(\log \boldsymbol{\theta}_0^v[n] + \sum_{r=1}^{i} \widetilde{\mathbf{y}}_r^v[n]\right).$$

Define the accumulated fused message $\widetilde{\mathbf{Y}}_i^v[n] := \sum_{r=1}^{i} \widetilde{\mathbf{y}}_r^v[n]$. Conditioned on fixed target-specific predictions, the message noises at different update steps are independent. The sum of the Gaussian messages is therefore Gaussian. Its mean is

$$\begin{aligned}\mathbb{E}[\widetilde{\mathbf{Y}}_i^v[n]] &= \sum_{r=1}^{i} \alpha_r^v \left[K\left(w_1 \hat{\mathbf{v}}_M^{(1)}[n] + w_2 \hat{\mathbf{v}}_M^{(2)}[n]\right) - S\mathbf{1}_K\right] \\ &= \beta^v(\tau_i)\left[K\left(w_1 \hat{\mathbf{v}}_M^{(1)}[n] + w_2 \hat{\mathbf{v}}_M^{(2)}[n]\right) - S\mathbf{1}_K\right],\end{aligned}$$

whereas its covariance is

$$\mathrm{Cov}(\widetilde{\mathbf{Y}}_i^v[n]) = \sum_{r=1}^{i} \alpha_r^v KQ\mathbf{I}_K = \beta^v(\tau_i) KQ\mathbf{I}_K.$$

Hence,

$$\widetilde{\mathbf{Y}}_i^v[n] \sim \mathcal{N}\Big(\beta^v(\tau_i)\left[K\left(w_1 \hat{\mathbf{v}}_M^{(1)}[n] + w_2 \hat{\mathbf{v}}_M^{(2)}[n]\right) - S\mathbf{1}_K\right], \beta^v(\tau_i) KQ\mathbf{I}_K\Big).$$

Under the uniform categorical prior, we have $\log \boldsymbol{\theta}_0^v[n] = -\log K\, \mathbf{1}_K$. Since softmax is invariant to adding the same scalar to every component,

$$\boldsymbol{\theta}_i^v[n] = \mathrm{softmax}\Big(\log \boldsymbol{\theta}_0^v[n] + \widetilde{\mathbf{Y}}_i^v[n]\Big) = \mathrm{softmax}\Big(\widetilde{\mathbf{Y}}_i^v[n]\Big).$$

Marginalizing the accumulated fused message therefore gives

$$\begin{aligned}p_F\Big(\boldsymbol{\theta}_i^v[n] \mid \hat{\mathbf{v}}_M^{(1)}[n], \hat{\mathbf{v}}_M^{(2)}[n], \widetilde{\mathbf{p}}^{(1)}, \widetilde{\mathbf{p}}^{(2)}; \tau_i\Big) &= \int \delta(\boldsymbol{\theta}_i^v[n] - \mathrm{softmax}(\mathbf{Y}))\, p_{\widetilde{\mathbf{Y}}_i^v[n]}(\mathbf{Y})\, \mathrm{d}\mathbf{Y} \\ &= \mathbb{E}_{\widetilde{\mathbf{Y}}_i^v[n]}\left[\delta\Big(\boldsymbol{\theta}_i^v[n] - \mathrm{softmax}(\widetilde{\mathbf{Y}}_i^v[n])\Big)\right],\end{aligned}$$

which is equation 13 in the main text. Under the conditional atomwise factorization of the BFN input distribution, the corresponding atom-type flow is the product of these per-atom flow distributions over $n = 1, \ldots, N_M$.

B.3 Sampling

Algorithm 1 summarizes the sampling procedure of FusedBFN.

Algorithm 1 Sampling Procedure of FusedBFN

function UPDATE($\hat{\mathbf{m}}^{(1)}$, $\hat{\mathbf{m}}^{(2)}$, $\beta^x(\tau)$, $\beta^v(\tau)$, w_1, w_2)
  $[\hat{\mathbf{x}}_M^{(j)}, \hat{\mathbf{v}}_M^{(j)}] \leftarrow \hat{\mathbf{m}}^{(j)}, j \in \{1, 2\}$
  $S \leftarrow w_1 + w_2$, $Q \leftarrow w_1^2 + w_2^2$, $\rho \leftarrow 1 + S\beta^x(\tau)$
  $\boldsymbol{\mu} \sim \mathcal{N}(\frac{\beta^x(\tau)(w_1\hat{\mathbf{x}}_M^{(1)} + w_2\hat{\mathbf{x}}_M^{(2)})}{1 + S\beta^x(\tau)}, \frac{\beta^x(\tau)Q}{[1 + S\beta^x(\tau)]^2}\mathbf{I})$
  for $n = 1, \ldots, N_M$ **do**
    $\widetilde{\mathbf{Y}}^v[n] \sim \mathcal{N}(\beta^v(\tau)[K(w_1\hat{\mathbf{v}}_M^{(1)}[n] + w_2\hat{\mathbf{v}}_M^{(2)}[n]) - S\mathbf{1}_K], \beta^v(\tau)KQ\mathbf{I}_K)$
    $\boldsymbol{\theta}^v[n] \leftarrow \mathrm{softmax}(\widetilde{\mathbf{Y}}^v[n])$
  end for
  $\boldsymbol{\theta} \leftarrow [\{\boldsymbol{\mu}, \rho\}, \boldsymbol{\theta}^v]$
  return $\boldsymbol{\theta}$
end function
Require: Pretrained network $\boldsymbol{\Psi}$; aligned pockets $\widetilde{\mathbf{p}}^{(1)}$ and $\widetilde{\mathbf{p}}^{(2)}$; schedule times $\{\tau_i\}_{i=0}^{N_{\mathrm{BFN}}}$; accumulated accuracy schedules β^x and β^v; fusion weights w_1, w_2; N_M ligand atoms; K atom types
$S \leftarrow w_1 + w_2$, $Q \leftarrow w_1^2 + w_2^2$
$\boldsymbol{\mu}_0 \leftarrow \mathbf{0}$, $\rho_0 \leftarrow 1$, $\boldsymbol{\theta}_0^v \leftarrow [K^{-1}\mathbf{1}_K]_{n=1}^{N_M}$, $\boldsymbol{\theta}_0 \leftarrow [\{\boldsymbol{\mu}_0, \rho_0\}, \boldsymbol{\theta}_0^v]$
for $i = 1, \ldots, N_{\mathrm{BFN}}$ **do**
  $\hat{\mathbf{m}}^{(1)} \leftarrow \boldsymbol{\Psi}(\boldsymbol{\theta}_{i-1}, \widetilde{\mathbf{p}}^{(1)}, \tau_{i-1})$, $\hat{\mathbf{m}}^{(2)} \leftarrow \boldsymbol{\Psi}(\boldsymbol{\theta}_{i-1}, \widetilde{\mathbf{p}}^{(2)}, \tau_{i-1})$
  $\boldsymbol{\theta}_i \leftarrow$ UPDATE$(\hat{\mathbf{m}}^{(1)}, \hat{\mathbf{m}}^{(2)}, \beta^x(\tau_i), \beta^v(\tau_i), w_1, w_2)$
end for
$[\hat{\mathbf{x}}_M^{(1)}, \hat{\mathbf{v}}_M^{(1)}] \leftarrow \boldsymbol{\Psi}(\boldsymbol{\theta}_{N_{\mathrm{BFN}}}, \widetilde{\mathbf{p}}^{(1)}, 1)$, $[\hat{\mathbf{x}}_M^{(2)}, \hat{\mathbf{v}}_M^{(2)}] \leftarrow \boldsymbol{\Psi}(\boldsymbol{\theta}_{N_{\mathrm{BFN}}}, \widetilde{\mathbf{p}}^{(2)}, 1)$
$\mathbf{x}_M \leftarrow \frac{w_1\hat{\mathbf{x}}_M^{(1)} + w_2\hat{\mathbf{x}}_M^{(2)}}{S}$
for $n = 1, \ldots, N_M$ **do**
  $\boldsymbol{\pi}_M^v[n] \leftarrow \frac{w_1\hat{\mathbf{v}}_M^{(1)}[n] + w_2\hat{\mathbf{v}}_M^{(2)}[n]}{S}$
  $\mathbf{v}_M[n] \sim \mathrm{Categorical}(\boldsymbol{\pi}_M^v[n])$
end for
$\mathbf{m} \leftarrow [\mathbf{x}_M, \mathbf{v}_M]$
return $\mathbf{m}$

C Full Derivation of FusedDiff

C.1 Derivation of Continuous Coordinates

Section 4.2 establishes that, under $S = 1$, the fused coordinate marginal preserves the variance schedule of the pretrained diffusion model:

$$q\left(\mathbf{x}_{M,t} \mid \hat{\mathbf{x}}_{M,0}^{(1)}, \hat{\mathbf{x}}_{M,0}^{(2)}\right) = \mathcal{N}\left(\mathbf{x}_{M,t}; \sqrt{\bar{a}_t}\,\hat{\mathbf{x}}_{M,0}^{\mathrm{Fuse}}, (1 - \bar{a}_t)\mathbf{I}\right), \tag{36}$$

where we denote $\hat{\mathbf{x}}_{M,0}^{\mathrm{Fuse}} := w_1\hat{\mathbf{x}}_{M,0}^{(1)} + w_2\hat{\mathbf{x}}_{M,0}^{(2)}$. Throughout the following derivation, the two target-specific predictions are treated as fixed at the current reverse step. They are recomputed from the current noisy molecular state at every step during sampling. Applying the same fused marginal at time $t - 1$ gives

$$q\left(\mathbf{x}_{M,t-1} \mid \hat{\mathbf{x}}_{M,0}^{(1)}, \hat{\mathbf{x}}_{M,0}^{(2)}\right) = \mathcal{N}\left(\mathbf{x}_{M,t-1}; \sqrt{\bar{a}_{t-1}}\,\hat{\mathbf{x}}_{M,0}^{\mathrm{Fuse}}, (1 - \bar{a}_{t-1})\mathbf{I}\right).$$

Combining this marginal with the one-step forward transition through Bayes' rule yields

$$q\left(\mathbf{x}_{M,t-1} \mid \mathbf{x}_{M,t}, \hat{\mathbf{x}}_{M,0}^{(1)}, \hat{\mathbf{x}}_{M,0}^{(2)}\right) \propto q\left(\mathbf{x}_{M,t} \mid \mathbf{x}_{M,t-1}\right) q\left(\mathbf{x}_{M,t-1} \mid \hat{\mathbf{x}}_{M,0}^{(1)}, \hat{\mathbf{x}}_{M,0}^{(2)}\right). \tag{37}$$

Substituting equation 25 and equation 36 into equation 37 gives

$$q\left(\mathbf{x}_{M,t-1} \mid \mathbf{x}_{M,t}, \hat{\mathbf{x}}_{M,0}^{(1)}, \hat{\mathbf{x}}_{M,0}^{(2)}\right) = \mathcal{N}\left(\mathbf{x}_{M,t-1}; \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_{M,t}, \hat{\mathbf{x}}_{M,0}^{\mathrm{Fuse}}\right), \tilde{b}_t\mathbf{I}\right),$$

where the posterior mean follows the definition in equation 28:

$$\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_{M,t}, \hat{\mathbf{x}}_{M,0}^{\text{Fuse}}\right) = \frac{\sqrt{\bar{a}_{t-1}}\, b_t}{1-\bar{a}_t}\hat{\mathbf{x}}_{M,0}^{\text{Fuse}} + \frac{\sqrt{a_t}(1-\bar{a}_{t-1})}{1-\bar{a}_t}\mathbf{x}_{M,t}.$$

The posterior mean is affine in its clean-coordinate argument. Since $S = w_1 + w_2 = 1$, it follows that

$$\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_{M,t}, \hat{\mathbf{x}}_{M,0}^{\text{Fuse}}\right) = w_1\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_{M,t}, \hat{\mathbf{x}}_{M,0}^{(1)}\right) + w_2\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_{M,t}, \hat{\mathbf{x}}_{M,0}^{(2)}\right). \tag{38}$$

Indeed, the coefficient of $\mathbf{x}_{M,t}$ on the right-hand side is multiplied by $w_1 + w_2 = 1$, while the clean-coordinate terms reduce to the wPoE-fused prediction. This identity provides the exact justification for performing a single reverse update using $\hat{\mathbf{x}}_{M,0}^{\text{Fuse}}$. When $w_1 = 1$ and $w_2 = 0$, this transition reduces to the standard single-target TargetDiff reverse update conditioned on the first pocket. Under the default setting $w_1 = w_2 = 1/2$, its mean is the arithmetic average of the two target-specific posterior means. For $S \neq 1$, the wPoE-fused forward covariance becomes $(1-\bar{a}_t)\mathbf{I}/S$, which does not conform to the standard DDPM (Ho et al., 2020). We therefore restrict FusedDiff to $S = 1$ and do not construct a reverse sampler for the unnormalized-weight case.

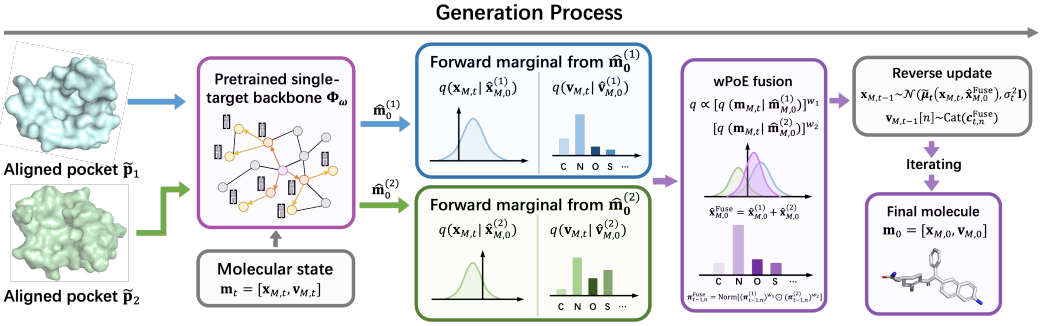

Figure 3: Overview of FusedDiff. Given two aligned pockets and the current noisy molecular state, a shared pretrained diffusion network Φ_ω predicts two target-specific clean molecular estimates. FusedDiff constructs the corresponding forward marginals for coordinates and atom types and combines them using wPoE before performing a single reverse update. Iterating this process yields final molecules that interact with both targets.

C.2 Derivation of Discrete Atom Types

We start from the wPoE-fused categorical marginal in equation 15. For ligand atom n, applying this marginal at time $t-1$, we define the target-specific categorical probability vector

$$\boldsymbol{\pi}_{t-1,n}^{(j)} := \bar{a}_{t-1}\hat{\mathbf{v}}_{M,0}^{(j)}[n] + (1-\bar{a}_{t-1})\mathbf{u}_K.$$

The corresponding fused probability vector is

$$\boldsymbol{\pi}_{t-1,n}^{\text{Fuse}} := \frac{\left(\boldsymbol{\pi}_{t-1,n}^{(1)}\right)^{w_1} \odot \left(\boldsymbol{\pi}_{t-1,n}^{(2)}\right)^{w_2}}{\mathbf{1}_K^\top\left[\left(\boldsymbol{\pi}_{t-1,n}^{(1)}\right)^{w_1} \odot \left(\boldsymbol{\pi}_{t-1,n}^{(2)}\right)^{w_2}\right]}.$$

Here, vector exponentiation and multiplication are performed elementwise. The fused categorical marginal at time $t-1$ is therefore

$$q\left(\mathbf{v}_{M,t-1}[n] \mid \hat{\mathbf{v}}_{M,0}^{(1)}[n], \hat{\mathbf{v}}_{M,0}^{(2)}[n]\right) = \text{Cat}\left(\mathbf{v}_{M,t-1}[n]; \boldsymbol{\pi}_{t-1,n}^{\text{Fuse}}\right).$$

To derive the reverse posterior, define the one-step categorical likelihood factor

$$\mathbf{g}_{t,n} := a_t\mathbf{v}_{M,t}[n] + b_t\mathbf{u}_K. \tag{39}$$

Its k-th component is the probability of the observed noisy atom type $\mathbf{v}_{M,t}[n]$ conditioned on the candidate previous type $\mathbf{v}_{M,t-1}[n] = \mathbf{e}_k$:

$$g_{t,n,k} = q\left(\mathbf{v}_{M,t}[n] \mid \mathbf{v}_{M,t-1}[n] = \mathbf{e}_k\right).$$

Applying Bayes' rule gives

$$q\left(\mathbf{v}_{M,t-1}[n] \mid \mathbf{v}_{M,t}[n], \hat{\mathbf{v}}^{(1)}_{M,0}[n], \hat{\mathbf{v}}^{(2)}_{M,0}[n]\right) \propto q\left(\mathbf{v}_{M,t}[n] \mid \mathbf{v}_{M,t-1}[n]\right) q\left(\mathbf{v}_{M,t-1}[n] \mid \hat{\mathbf{v}}^{(1)}_{M,0}[n], \hat{\mathbf{v}}^{(2)}_{M,0}[n]\right).$$

Thus, the posterior parameter is

$$\tilde{\mathbf{c}}^{\text{Fuse}}_{t,n} = \frac{\left(\mathbf{c}^{\text{Fuse}}_{t,n}\right)^*}{\mathbf{1}_K^\top \left(\mathbf{c}^{\text{Fuse}}_{t,n}\right)^*},$$

where

$$\left(\mathbf{c}^{\text{Fuse}}_{t,n}\right)^* = \mathbf{g}_{t,n} \odot \boldsymbol{\pi}^{\text{Fuse}}_{t-1,n}.$$

Consequently, the fused reverse posterior is

$$q\left(\mathbf{v}_{M,t-1}[n] \mid \mathbf{v}_{M,t}[n], \hat{\mathbf{v}}^{(1)}_{M,0}[n], \hat{\mathbf{v}}^{(2)}_{M,0}[n]\right) = \text{Cat}\left(\mathbf{v}_{M,t-1}[n]; \tilde{\mathbf{c}}^{\text{Fuse}}_{t,n}\right).$$

Expanding the k-th component gives

$$\left(\tilde{\mathbf{c}}^{\text{Fuse}}_{t,n}\right)_k = \frac{g_{t,n,k}\left(\pi^{(1)}_{t-1,n,k}\right)^{w_1}\left(\pi^{(2)}_{t-1,n,k}\right)^{w_2}}{\sum_{\ell=1}^{K} g_{t,n,\ell}\left(\pi^{(1)}_{t-1,n,\ell}\right)^{w_1}\left(\pi^{(2)}_{t-1,n,\ell}\right)^{w_2}}.$$

This expression is equivalent to equation 16. When $w_1 = 1$ and $w_2 = 0$,

$$\boldsymbol{\pi}^{\text{Fuse}}_{t-1,n} = \boldsymbol{\pi}^{(1)}_{t-1,n},$$

so the result reduces to the standard single-target categorical reverse transition in Appendix A.1.

C.3 Sampling

Algorithm 2 summarizes the complete sampling procedure of FusedDiff.

Algorithm 2 Sampling Procedure of FusedDiff

Require: Pretrained diffusion network Φ_ω; aligned target pockets $\tilde{\mathbf{p}}^{(1)}$ and $\tilde{\mathbf{p}}^{(2)}$; diffusion schedule $\{a_t, b_t, \bar{a}_t, \sigma_t^2\}_{t=1}^T$; weights w_1, w_2 satisfying $w_1 + w_2 = 1$; number of ligand atoms N_M; number of atom types K

$\mathbf{x}_{M,T} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $n = 1$ to N_M **do**
  $\mathbf{v}_{M,T}[n] \sim \text{Cat}(\mathbf{u}_K)$
end for
$\mathbf{m}_T \leftarrow [\mathbf{x}_{M,T}, \mathbf{v}_{M,T}]$
for $t = T$ down to 1 **do**
  $\hat{\mathbf{x}}^{(1)}_{M,0}, \hat{\mathbf{v}}^{(1)}_{M,0} \leftarrow \Phi_\omega(\mathbf{m}_t, \tilde{\mathbf{p}}^{(1)}, t)$, $\hat{\mathbf{x}}^{(2)}_{M,0}, \hat{\mathbf{v}}^{(2)}_{M,0} \leftarrow \Phi_\omega(\mathbf{m}_t, \tilde{\mathbf{p}}^{(2)}, t)$
  $\hat{\mathbf{x}}^{\text{Fuse}}_{M,0} \leftarrow w_1 \hat{\mathbf{x}}^{(1)}_{M,0} + w_2 \hat{\mathbf{x}}^{(2)}_{M,0}$
  $\mathbf{x}_{M,t-1} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_{M,t}, \hat{\mathbf{x}}^{\text{Fuse}}_{M,0}), \sigma_t^2 \mathbf{I})$
  for $n = 1$ to N_M **do**
    for $j = 1$ to 2 **do**
      $\boldsymbol{\pi}^{(j)}_{t-1,n} \leftarrow \bar{a}_{t-1} \hat{\mathbf{v}}^{(j)}_{M,0}[n] + (1 - \bar{a}_{t-1}) \mathbf{u}_K$
    end for
    $\boldsymbol{\pi}^{\text{Fuse}}_{t-1,n} \leftarrow \text{Normalize}[(\boldsymbol{\pi}^{(1)}_{t-1,n})^{w_1} \odot (\boldsymbol{\pi}^{(2)}_{t-1,n})^{w_2}]$
    $\tilde{\mathbf{c}}^{\text{Fuse}}_{t,n} \leftarrow \text{Normalize}[(a_t \mathbf{v}_{M,t}[n] + b_t \mathbf{u}_K) \odot \boldsymbol{\pi}^{\text{Fuse}}_{t-1,n}]$
    $\mathbf{v}_{M,t-1}[n] \sim \text{Cat}(\tilde{\mathbf{c}}^{\text{Fuse}}_{t,n})$
  end for
  $\mathbf{m}_{t-1} \leftarrow [\mathbf{x}_{M,t-1}, \mathbf{v}_{M,t-1}]$
end for
return $\mathbf{m}_0$

C.4 COMPARISON WITH COMPDIFF

We compare FusedDiff and CompDiff under $w_1 = w_2 = 1/2$ and $\eta = 1/2$.

Continuous coordinates According to equation 28, the target-specific coordinate reverse kernel for j is

$$K_t^{x,(j)} = \mathcal{N}\left(\mathbf{x}_{M,t-1}; \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_{M,t}, \hat{\mathbf{x}}_{M,0}^{(j)}\right), \sigma_t^2 \mathbf{I}\right).$$

The transition-kernel composition used by CompDiff gives

$$K_t^{\mathrm{Comp},x} \propto \left[K_t^{x,(1)} K_t^{x,(2)}\right]^{\eta} = \mathcal{N}\left(\mathbf{x}_{M,t-1}; \frac{\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_{M,t}, \hat{\mathbf{x}}_{M,0}^{(1)}\right) + \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_{M,t}, \hat{\mathbf{x}}_{M,0}^{(2)}\right)}{2}, \frac{\sigma_t^2}{2\eta}\mathbf{I}\right).$$

At $\eta = 1/2$, the composed covariance is $\sigma_t^2 \mathbf{I}$. Moreover, by the affine property established in equation 38,

$$\frac{\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_{M,t}, \hat{\mathbf{x}}_{M,0}^{(1)}\right) + \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_{M,t}, \hat{\mathbf{x}}_{M,0}^{(2)}\right)}{2} = \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_{M,t}, \frac{\hat{\mathbf{x}}_{M,0}^{(1)} + \hat{\mathbf{x}}_{M,0}^{(2)}}{2}\right).$$

This is exactly the FusedDiff reverse mean for $w_1 = w_2 = 1/2$. Therefore,

$$K_t^{\mathrm{Fuse},x} = K_t^{\mathrm{Comp},x},$$

under the matched setting considered here. With the same Gaussian noise realization, their coordinate updates are also identical samplewise. Thus, although FusedDiff reaches this transition by fusing forward distributions and CompDiff starts from target-specific reverse kernels, the resulting coordinate transitions coincide under the matched Gaussian setting.

Discrete atom types The categorical branches follow different computational orders. To express this difference compactly, define the forward-corruption operator

$$\mathcal{C}_{t-1}(\mathbf{p}) := \bar{a}_{t-1}\mathbf{p} + (1 - \bar{a}_{t-1})\mathbf{u}_K, \qquad \mathbf{p} \in \Delta^{K-1},$$

and the reverse-posterior operator for ligand atom n,

$$\mathcal{R}_{t,n}(\boldsymbol{\pi}) := \mathrm{Normalize}\left[\mathbf{g}_{t,n} \odot \boldsymbol{\pi}\right], \qquad \boldsymbol{\pi} \in \Delta^{K-1},$$

where $\mathbf{g}_{t,n}$ is defined in equation 39. FusedDiff first corrupts the two target-specific clean predictions separately and then applies fractional wPoE fusion:

$$\tilde{\mathbf{c}}_{t,n}^{\mathrm{Fuse}} = \mathcal{R}_{t,n}\left(\mathrm{Normalize}\left[\mathcal{C}_{t-1}\left(\hat{\mathbf{v}}_{M,0}^{(1)}[n]\right)^{\frac{1}{2}} \odot \mathcal{C}_{t-1}\left(\hat{\mathbf{v}}_{M,0}^{(2)}[n]\right)^{\frac{1}{2}}\right]\right).$$

By contrast, the CompDiff implementation first multiplies the two clean predictions and normalizes the result:

$$\hat{\mathbf{v}}_{M,0}^{\mathrm{Comp}}[n] = \mathrm{Normalize}\left[\hat{\mathbf{v}}_{M,0}^{(1)}[n] \odot \hat{\mathbf{v}}_{M,0}^{(2)}[n]\right],$$

and subsequently corrupts this composed prediction:

$$\tilde{\mathbf{c}}_{t,n}^{\mathrm{Comp}} = \mathcal{R}_{t,n}\left(\mathcal{C}_{t-1}\left(\hat{\mathbf{v}}_{M,0}^{\mathrm{Comp}}[n]\right)\right).$$

Consequently, for general probability vectors $\mathbf{p}^{(1)}$ and $\mathbf{p}^{(2)}$,

$$\mathrm{Normalize}\left[\mathcal{C}_{t-1}\left(\mathbf{p}^{(1)}\right)^{\frac{1}{2}} \odot \mathcal{C}_{t-1}\left(\mathbf{p}^{(2)}\right)^{\frac{1}{2}}\right] \neq \mathcal{C}_{t-1}\left(\mathrm{Normalize}\left[\mathbf{p}^{(1)} \odot \mathbf{p}^{(2)}\right]\right).$$

These expressions differ in two respects: the order of composition relative to forward corruption and the expert exponent. FusedDiff applies wPoE after corrupting each expert distribution, whereas CompDiff applies a product to the clean predictions before corruption. Uniform mixing and geometric pooling therefore do not generally commute.

This distinction remains even when the two experts agree. If both predict the same non-degenerate probability vector $\mathbf{p}$, the two marginals reduce to

$$\boldsymbol{\pi}_{t-1}^{\mathrm{Fuse}} = \mathcal{C}_{t-1}(\mathbf{p}), \qquad \boldsymbol{\pi}_{t-1}^{\mathrm{Comp}} = \mathcal{C}_{t-1}\left(\mathrm{Normalize}\left[\mathbf{p} \odot \mathbf{p}\right]\right),$$

which are generally unequal unless $\mathbf{p}$ is uniform, one-hot, or satisfies another degenerate condition. The product in CompDiff typically sharpens large probability components, whereas wPoE fusion after uniform corruption tends to produce a smoother categorical distribution. Consistently, FusedDiff exhibits smaller top-1/top-2 log-probability margins in our experiments (Figure 2).

Matched-exponent categorical control The comparison above differs in both the order of composition relative to forward corruption and the expert exponents. To separate these two factors, we additionally consider a categorical-only variant of FusedDiff in which the weights are changed from $w_1 = w_2 = 1/2$ to $w_1 = w_2 = 1$, while the coordinate branch is kept unchanged. We denote this variant as FusedDiff $(1, 1)$.

Using the corruption operator C_{t-1} defined above, its fused categorical marginal is

$$\boldsymbol{\pi}_{t-1,n}^{\mathrm{Fuse}(1,1)} = \mathrm{Normalize}\left[\mathcal{C}_{t-1}\left(\hat{\mathbf{v}}_{M,0}^{(1)}[n]\right) \odot \mathcal{C}_{t-1}\left(\hat{\mathbf{v}}_{M,0}^{(2)}[n]\right)\right].$$

The corresponding reverse-posterior parameter is

$$\tilde{\mathbf{c}}_{t,n}^{\mathrm{Fuse}(1,1)} = \mathcal{R}_{t,n}\left(\boldsymbol{\pi}_{t-1,n}^{\mathrm{Fuse}(1,1)}\right).$$

Hence, FusedDiff $(1, 1)$ and the CompDiff categorical rule use the same unit exponent but retain different computational orders:

$$\mathrm{Normalize}\left[\mathcal{C}_{t-1}(\mathbf{p}^{(1)}) \odot \mathcal{C}_{t-1}(\mathbf{p}^{(2)})\right] \neq \mathcal{C}_{t-1}\left(\mathrm{Normalize}\left[\mathbf{p}^{(1)} \odot \mathbf{p}^{(2)}\right]\right)$$

in general. Therefore, FusedDiff $(1, 1)$ removes the exponent mismatch while preserving the distinction in composition order. The empirical behavior is examined in Appendix F.4.

D Full Derivation of CompBFN

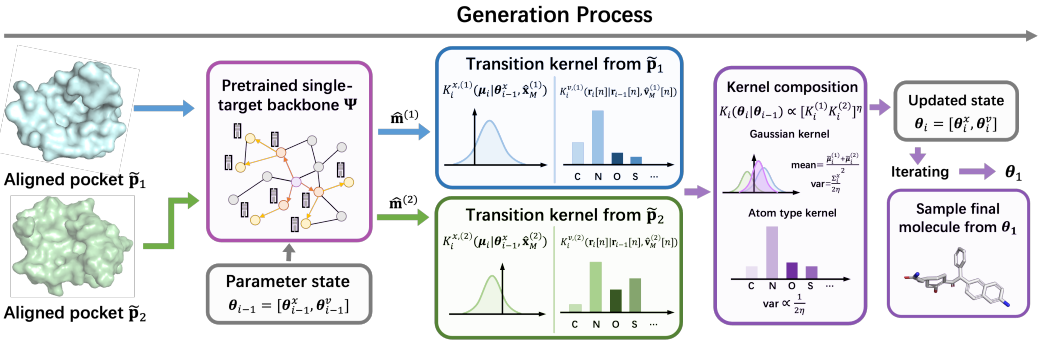

Figure 4: Overview of CompBFN, which realizes transition-kernel composition under a BFN backbone. The shared pretrained network Ψ outputs target-specific molecular estimates, from which coordinate and atom-type transition kernels are constructed. The two target-specific kernels are subsequently composed to obtain a single transition from θ_{i-1} to θ_i. Repeated composed transitions drive sampling toward a final molecule conditioned on both target pockets.

D.1 Derivation of continuous coordinates

Equation 18 gives the coordinate transition kernel obtained by composing the two target-specific Gaussian kernels. The mean and the covariance of the composed transition satisfy

$$\frac{\tilde{\boldsymbol{\mu}}_i^{(1)} + \tilde{\boldsymbol{\mu}}_i^{(2)}}{2} = \frac{\rho_{i-1}\boldsymbol{\mu}_{i-1} + \alpha_i^x\left(\hat{\mathbf{x}}_M^{(1)} + \hat{\mathbf{x}}_M^{(2)}\right)/2}{\rho_i}, \quad \frac{\boldsymbol{\Sigma}_i^x}{2\eta} = \frac{\alpha_i^x}{2\eta\rho_i^2}\mathbf{I}.$$

Consequently, the composed coordinate transition can be expressed as

$$\boldsymbol{\mu}_i \sim \mathcal{N}\left(\frac{\rho_{i-1}\boldsymbol{\mu}_{i-1} + \alpha_i^x\left(\hat{\mathbf{x}}_M^{(1)} + \hat{\mathbf{x}}_M^{(2)}\right)/2}{\rho_i}, \frac{\alpha_i^x}{2\eta\rho_i^2}\mathbf{I}\right).$$

Equivalently, this transition admits the stochastic representation

$$\boldsymbol{\mu}_i = \frac{\rho_{i-1}\boldsymbol{\mu}_{i-1} + \alpha_i^x \left(\hat{\mathbf{x}}_M^{(1)} + \hat{\mathbf{x}}_M^{(2)}\right)/2}{\rho_i} + \frac{\sqrt{\alpha_i^x}}{\sqrt{2\eta}\rho_i}\boldsymbol{\epsilon}_i^x,$$

where $\boldsymbol{\epsilon}_i^x \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The noises associated with different update steps are conditionally independent. To derive the accumulated coordinate flow, define the natural parameter $\mathbf{z}_i := \rho_i\boldsymbol{\mu}_i$. Multiplying the stochastic update by ρ_i yields

$$\mathbf{z}_i = \rho_{i-1}\boldsymbol{\mu}_{i-1} + \alpha_i^x \frac{\hat{\mathbf{x}}_M^{(1)} + \hat{\mathbf{x}}_M^{(2)}}{2} + \sqrt{\frac{\alpha_i^x}{2\eta}}\boldsymbol{\epsilon}_i^x.$$

Since $\mathbf{z}_{i-1} = \rho_{i-1}\boldsymbol{\mu}_{i-1}$, the natural-parameter recurrence becomes

$$\mathbf{z}_i = \mathbf{z}_{i-1} + \alpha_i^x \frac{\hat{\mathbf{x}}_M^{(1)} + \hat{\mathbf{x}}_M^{(2)}}{2} + \sqrt{\frac{\alpha_i^x}{2\eta}}\boldsymbol{\epsilon}_i^x.$$

Conditioned on fixed target-specific coordinate predictions, repeated application of this recurrence gives

$$\mathbf{z}_i = \mathbf{z}_0 + \sum_{r=1}^{i} \alpha_r^x \frac{\hat{\mathbf{x}}_M^{(1)} + \hat{\mathbf{x}}_M^{(2)}}{2} + \sum_{r=1}^{i} \sqrt{\frac{\alpha_r^x}{2\eta}}\boldsymbol{\epsilon}_r^x.$$

Under the standard coordinate prior, we have $\mathbf{z}_0 = \mathbf{0}$. It follows that

$$\mathbf{z}_i \sim \mathcal{N}\left(\beta^x(\tau_i)\frac{\hat{\mathbf{x}}_M^{(1)} + \hat{\mathbf{x}}_M^{(2)}}{2}, \frac{\beta^x(\tau_i)}{2\eta}\mathbf{I}\right).$$

Meanwhile, accumulating the deterministic precision increments gives $\rho_i = \rho_0 + \sum_{r=1}^{i}\alpha_r^x = 1 + \beta^x(\tau_i)$. Finally, since $\boldsymbol{\mu}_i = \mathbf{z}_i/\rho_i$, applying the corresponding linear transformation gives

$$p_F\left(\boldsymbol{\mu}_i \mid \hat{\mathbf{x}}_M^{(1)}, \hat{\mathbf{x}}_M^{(2)}; \tau_i\right) = \mathcal{N}\left(\boldsymbol{\mu}_i; \frac{\beta^x(\tau_i)}{1+\beta^x(\tau_i)}\frac{\hat{\mathbf{x}}_M^{(1)} + \hat{\mathbf{x}}_M^{(2)}}{2}, \frac{\beta^x(\tau_i)}{2\eta[1+\beta^x(\tau_i)]^2}\mathbf{I}\right),$$

which recovers the CompBFN coordinate flow in equation 19.

D.2 Derivation of Discrete Atom Types

Equation 21 already gives the normalized composed transition kernel in the $(K-1)$-dimensional log-ratio space. Since $\mathbf{D}$ has full row rank, $\mathbf{D}\mathbf{D}^\top$ is positive definite, and the composed transition can be written as

$$\mathbf{r}_i[n] = \mathbf{r}_{i-1}[n] + \alpha_i^v K\mathbf{D}\frac{\hat{\mathbf{v}}_M^{(1)}[n] + \hat{\mathbf{v}}_M^{(2)}[n]}{2} + \boldsymbol{\epsilon}_i^v[n],$$

where the transition noise satisfies

$$\boldsymbol{\epsilon}_i^v[n] \sim \mathcal{N}\left(\mathbf{0}, \frac{\alpha_i^v K}{2\eta}\mathbf{D}\mathbf{D}^\top\right).$$

Conditioned on fixed target-specific atom-type predictions, the transition noises at different update steps are independent. Repeated application of the composed transition therefore gives

$$\mathbf{r}_i[n] = \mathbf{r}_0[n] + \sum_{s=1}^{i}\alpha_s^v K\mathbf{D}\frac{\hat{\mathbf{v}}_M^{(1)}[n] + \hat{\mathbf{v}}_M^{(2)}[n]}{2} + \sum_{s=1}^{i}\boldsymbol{\epsilon}_s^v[n].$$

Under the uniform categorical prior, we have $\mathbf{r}_0[n] = \mathbf{0}$. Using $\sum_{s=1}^{i}\alpha_s^v = \beta^v(\tau_i)$, we obtain

$$\mathbf{r}_i[n] \sim \mathcal{N}\left(\beta^v(\tau_i)K\mathbf{D}\frac{\hat{\mathbf{v}}_M^{(1)}[n] + \hat{\mathbf{v}}_M^{(2)}[n]}{2}, \frac{\beta^v(\tau_i)K}{2\eta}\mathbf{D}\mathbf{D}^\top\right).$$

Here, we construct the equivalent message

$$\mathbf{Y}_i^v[n] \sim \mathcal{N}\left(\beta^v(\tau_i)\left[K\frac{\hat{\mathbf{v}}_M^{(1)}[n] + \hat{\mathbf{v}}_M^{(2)}[n]}{2} - \mathbf{1}_K\right], \frac{\beta^v(\tau_i)K}{2\eta}\mathbf{I}_K\right).$$

By $\mathbf{D}\mathbf{1}_K = \mathbf{0}$ and the affine-transformation rule for Gaussian variables, $\mathbf{D}\mathbf{Y}_i^v[n]$ and $\mathbf{r}_i[n]$ have identical means and covariances. Since both variables are Gaussian, it follows that

$$\mathbf{D}\mathbf{Y}_i^v[n] \overset{d}{=} \mathbf{r}_i[n].$$

To recover the categorical parameter from its log-ratio representation, observe that the log-ratio vector uniquely determines a probability vector in the interior of the categorical simplex. Specifically, if

$$r_k = \log \frac{\theta_k}{\theta_K}, \qquad k \in \{1, \dots, K-1\},$$

then normalization gives

$$\theta_k = \frac{\exp(r_k)}{1 + \sum_{\ell=1}^{K-1} \exp(r_\ell)}, \qquad \theta_K = \frac{1}{1 + \sum_{\ell=1}^{K-1} \exp(r_\ell)}.$$

For $\mathbf{r} = \mathbf{D}\mathbf{Y}$, we have $r_k = Y_k - Y_K$. Substituting these log-ratios into the inverse mapping gives

$$\theta_k = \frac{\exp(Y_k)}{\sum_{\ell=1}^{K} \exp(Y_\ell)}, \qquad k \in \{1, \dots, K\}.$$

Hence, the categorical parameter associated with $\mathbf{D}\mathbf{Y}$ is $\boldsymbol{\theta} = \mathrm{softmax}(\mathbf{Y})$. Since $\mathbf{D}\mathbf{Y}_i^v[n] \overset{d}{=} \mathbf{r}_i[n]$, the probability vector $\mathrm{softmax}(\mathbf{Y}_i^v[n])$ has the same distribution as the categorical parameter induced by the accumulated composed transitions. Marginalizing the equivalent Gaussian message therefore gives

$$\begin{aligned} p_F\left(\boldsymbol{\theta}_i^v[n] \mid \hat{\mathbf{v}}_M^{(1)}[n], \hat{\mathbf{v}}_M^{(2)}[n]; \tau_i, \eta\right) &= \int \delta\left(\boldsymbol{\theta}_i^v[n] - \mathrm{softmax}(\mathbf{Y})\right) p_{\mathbf{Y}_i^v[n]}(\mathbf{Y})\, \mathrm{d}\mathbf{Y} \\ &= \mathbb{E}_{\mathbf{Y}_i^v[n]}\left[\delta\left(\boldsymbol{\theta}_i^v[n] - \mathrm{softmax}(\mathbf{Y}_i^v[n])\right)\right], \end{aligned}$$

which is equivalent to equation 23. The K-dimensional Gaussian representation is not unique. For any scalar c,

$$\mathrm{softmax}\left(\mathbf{Y}_i^v[n] + c\mathbf{1}_K\right) = \mathrm{softmax}\left(\mathbf{Y}_i^v[n]\right),$$

and

$$\mathbf{D}\left(\mathbf{Y}_i^v[n] + c\mathbf{1}_K\right) = \mathbf{D}\mathbf{Y}_i^v[n].$$

Equation 22 selects an isotropic representative whose projection reproduces the mean and covariance of the accumulated log-ratio transition. Importantly, $\mathbf{Y}_i^v[n]$ is an equivalent representation of the flow induced by transition-kernel composition; it is not a message obtained by fusing the two target-specific sender distributions before the Bayesian update. Under the conditional atomwise factorization of the BFN input distribution, the complete atom-type flow is the product of these per-atom flow distributions over $n = 1, \dots, N_M$.

D.3 Relationship Between CompBFN and FusedBFN

Although FusedBFN and CompBFN perform composition at different probabilistic levels, they admit an exact equivalence under a particular parameter setting. We first characterize this equivalence and then clarify the distinction between the two constructions.

For FusedBFN with equal weights $w_1 = w_2 = 1/2$, we have $S = 1$ and $Q = 1/2$. Comparing its coordinate flow in equation 9 with the CompBFN coordinate flow in equation 19 shows that the two distributions are identical when $\eta = 1$. The same correspondence holds for atom types: the FusedBFN equivalent message associated with equation 14 and the CompBFN construction in equation 22 coincide under $\eta = 1$. Consequently, $\eta = 1$ also makes the atom-type flows identical. Conditioned on the same parameter state, both methods query the same pretrained network using the same aligned pockets and therefore output identical predictions. Starting from the same initial distribution, equality of the conditional flow implies that FusedBFN and CompBFN with $\eta = 1$ induce the same sampling process in distribution.

This equivalence does not make the two composition formulations identical in general. FusedBFN is defined by combining target-specific sender likelihoods before a shared Bayesian update, $[p_S^{(1)}]^{w_1}[p_S^{(2)}]^{w_2}$, where w_1 and w_2 directly control the relative contributions of the two target-specific experts. In particular, unequal weights shift the fused mean toward the more heavily

weighted expert, while S and Q jointly determine the effective information strength and sampling variance. By contrast, CompBFN first constructs two target-specific transition kernels and subsequently composes them through $[K_i^{(1)} K_i^{(2)}]^\eta$. Its two experts contribute symmetrically to the transition mean, whereas the composition parameter η controls the covariance of the composed kernel. Thus, the equality at $w_1 = w_2 = 1/2, \eta = 1$ should be viewed as a special intersection of two differently constructed families rather than an identity between the underlying composition mechanisms. In our experiments, CompBFN uses $\eta = 1/2$, following the original CompDiff configuration. Under this setting, CompBFN and FusedBFN retain the same fused center but have different covariance scales. The resulting comparison therefore characterizes the sampling behavior induced when the two composition mechanisms are used with their respective default formulations.

D.4 Sampling

Algorithm 3 summarizes the sampling procedure of CompBFN.

Algorithm 3 Sampling Procedure of CompBFN

function UPDATE($\hat{\mathbf{m}}^{(1)}$, $\hat{\mathbf{m}}^{(2)}$, $\beta^x(\tau)$, $\beta^v(\tau)$, η)
  $[\hat{\mathbf{x}}_M^{(j)}, \hat{\mathbf{v}}_M^{(j)}] \leftarrow \hat{\mathbf{m}}^{(j)}, j \in \{1, 2\}$
  $\rho \leftarrow 1 + \beta^x(\tau)$
  $\boldsymbol{\mu} \sim \mathcal{N}(\frac{\beta^x(\tau)}{1+\beta^x(\tau)} \frac{\hat{\mathbf{x}}_M^{(1)} + \hat{\mathbf{x}}_M^{(2)}}{2}, \frac{\beta^x(\tau)}{2\eta[1+\beta^x(\tau)]^2}\mathbf{I})$
  for $n = 1, \ldots, N_M$ **do**
    $\mathbf{Y}^v[n] \sim \mathcal{N}(\beta^v(\tau)[K \frac{\hat{\mathbf{v}}_M^{(1)}[n] + \hat{\mathbf{v}}_M^{(2)}[n]}{2} - \mathbf{1}_K], \frac{\beta^v(\tau) K}{2\eta}\mathbf{I}_K)$
    $\boldsymbol{\theta}^v[n] \leftarrow \mathrm{softmax}\,(\mathbf{Y}^v[n])$
  end for
  $\boldsymbol{\theta} \leftarrow [\{\boldsymbol{\mu}, \rho\}, \boldsymbol{\theta}^v]$
  return $\boldsymbol{\theta}$
end function
Require: Pretrained network $\boldsymbol{\Psi}$; aligned target pockets $\tilde{\mathbf{p}}^{(1)}$ and $\tilde{\mathbf{p}}^{(2)}$; schedule times $\{\tau_i\}_{i=0}^{N_{\mathrm{BFN}}}$; accumulated accuracy schedules β^x and β^v; composition strength $\eta > 0$; number of ligand atoms N_M; number of atom types K
$\boldsymbol{\mu}_0 \leftarrow \mathbf{0}, \rho_0 \leftarrow 1, \boldsymbol{\theta}_0^v \leftarrow [K^{-1}\mathbf{1}_K]_{n=1}^{N_M}, \boldsymbol{\theta}_0 \leftarrow [\{\boldsymbol{\mu}_0, \rho_0\}, \boldsymbol{\theta}_0^v]$
for $i = 1, \ldots, N_{\mathrm{BFN}}$ **do**
  $\hat{\mathbf{m}}^{(1)} \leftarrow \boldsymbol{\Psi}(\boldsymbol{\theta}_{i-1}, \tilde{\mathbf{p}}^{(1)}, \tau_{i-1}), \hat{\mathbf{m}}^{(2)} \leftarrow \boldsymbol{\Psi}(\boldsymbol{\theta}_{i-1}, \tilde{\mathbf{p}}^{(2)}, \tau_{i-1})$
  $\boldsymbol{\theta}_i \leftarrow$ UPDATE$(\hat{\mathbf{m}}^{(1)}, \hat{\mathbf{m}}^{(2)}, \beta^x(\tau_i), \beta^v(\tau_i), \eta)$
end for
$[\hat{\mathbf{x}}_M^{(1)}, \hat{\mathbf{v}}_M^{(1)}] \leftarrow \boldsymbol{\Psi}(\boldsymbol{\theta}_{N_{\mathrm{BFN}}}, \tilde{\mathbf{p}}^{(1)}, 1), [\hat{\mathbf{x}}_M^{(2)}, \hat{\mathbf{v}}_M^{(2)}] \leftarrow \boldsymbol{\Psi}(\boldsymbol{\theta}_{N_{\mathrm{BFN}}}, \tilde{\mathbf{p}}^{(2)}, 1)$
$\mathbf{x}_M \leftarrow \frac{\hat{\mathbf{x}}_M^{(1)} + \hat{\mathbf{x}}_M^{(2)}}{2}$
for $n = 1, \ldots, N_M$ **do**
  $\boldsymbol{\pi}_M^v[n] \leftarrow \frac{\hat{\mathbf{v}}_M^{(1)}[n] + \hat{\mathbf{v}}_M^{(2)}[n]}{2}$
  $v_M[n] \sim \mathrm{Categorical}\,(\boldsymbol{\pi}_M^v[n])$
end for
$\mathbf{m} \leftarrow [\mathbf{x}_M, \mathbf{v}_M]$
return $\mathbf{m}$

E Implementation Details

E.1 Dataset

We use the dual-target benchmark introduced by Zhou et al. (2024a), which was curated from synergistic drug combinations in DrugCombDB (Liu et al., 2020). A combination was retained if its zero interaction potency, Bliss, Loewe, and the highest single agent scores were all positive in at least one

human cancer cell line. Drug structures and target annotations were obtained primarily from DrugBank (Knox et al., 2024) and the Therapeutic Target Database (Zhou et al., 2024b). Protein-ligand complexes were collected from PDBBind (Liu et al., 2017) or constructed from PDB or AlphaFold structures (Varadi et al., 2022) using P2Rank and AutoDock Vina, with AlphaFold predictions of $\mathrm{pLDDT} < 70$ excluded. The resulting benchmark contains 12,917 target pairs involving 438 unique targets, with a reference ligand for each target.

E.2 Featurization

Each protein atom is described by a feature vector containing a one-hot encoding of the element type (H, C, N, O, S, Se), a 20-dimensional one-hot encoding of the amino acid type and a one-dimensional indicator of whether it belongs to the protein backbone. Ligand atoms are represented by one-hot vectors over the element set $\{\mathrm{C}, \mathrm{N}, \mathrm{O}, \mathrm{F}, \mathrm{P}, \mathrm{S}, \mathrm{Cl}\}$, together with aromaticity information.

We adopt MolCRAFT as the single-target BFN backbone and TargetDiff as the single-target diffusion backbone. Following MolCRAFT and TargetDiff, two graphs are dynamically constructed for message passing in each protein-ligand complex: a k-nearest-neighbor (kNN) graph over protein and ligand atoms, and a fully connected graph over ligand atoms. We set $k = 32$. In the kNN graph, edge features are defined as the outer product of a distance embedding and an edge-type embedding, where the distance embedding is computed by expanding pairwise distances with radial basis functions, and the edge type is represented as a 4-dimensional one-hot vector. In the ligand graph, bond information is encoded by a five-dimensional one-hot vector corresponding to non-bond, single, double, triple, and aromatic relations. All target-specific experts within a backbone share the same pretrained parameters; the two experts differ only in the aligned pocket used as the conditioning input.

E.3 SE(3)-Equivariant Network

Both pretrained backbones model protein–ligand interactions using an SE(3)-equivariant graph neural network (Guan et al., 2023). At layer l, the hidden features $\mathbf{h}^l$ and coordinates $\mathbf{x}^l$ are updated as

$$\mathbf{h}_i^{l+1} = \mathbf{h}_i^l + \sum_{j \in \mathcal{V}, i \neq j} f_h(d_{ij}^l, \mathbf{h}_i^l, \mathbf{h}_j^l, \mathbf{e}_{ij}; \phi_h) \tag{40}$$

$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + \sum_{j \in \mathcal{V}, i \neq j} (\mathbf{x}_i^l - \mathbf{x}_j^l) f_x(d_{ij}^l, \mathbf{h}_i^{l+1}, \mathbf{h}_j^{l+1}, \mathbf{e}_{ij}; \phi_x) \cdot \mathbf{l}_{mask}, \tag{41}$$

where l is the layer index, $\mathcal{V}$ denotes the set of all atoms in the protein–ligand complex, $d_{ij} = ||\mathbf{x}_i - \mathbf{x}_j||$ is the Euclidean distance and $\mathbf{e}_{ij}$ represents the relative distance information together with optional edge features between atoms i and j. The mask $\mathbf{l}_{mask}$ is applied to ligand atoms to keep the protein atom coordinates fixed during message passing. f_h and f_x with learned parameters ϕ_h and ϕ_x, respectively, are implemented as attention blocks.

E.4 Backbone and Sampling Configurations

Both MolCRAFT and TargetDiff use $L = 9$ equivariant layers, a hidden dimension of $d_h = 128$, and 16 attention heads. The key/value embeddings and attention scores are parameterized by two-layer multilayer perceptrons with ReLU activations and layer normalization. We use the released pretrained checkpoints without retraining or fine-tuning.

BFN-based methods. Following the MolCRAFT schedule, $\beta^x(\tau) = \sigma_1^{-2\tau} - 1$, with $\sigma_1 = 0.03$, and $\beta^v(\tau) = \beta_1 \tau^2$, with $\beta_1 = 1.5$. We use $N_{\mathrm{BFN}} = 100$ sampling steps. FusedBFN uses weights $w_1 = w_2 = 1/2$ and $S := w_1 + w_2 = 1$. CompBFN uses transition-kernel composition strength $\eta = 1/2$. Apart from the composition operator, all other settings are identical between FusedBFN and CompBFN.

Diffusion-based methods. We retain the original TargetDiff forward schedules. For coordinates, the diffusion noise rates follow a sigmoid schedule from $b_1^x = 10^{-7}$ to $b_T^x = 2 \times 10^{-3}$. For atom types, the noise rates follow a cosine schedule (Nichol & Dhariwal, 2021) with $s = 0.01$. The

number of diffusion steps is set to $T = 1000$. FusedDiff uses $w_1 = w_2 = 1/2$, whereas CompDiff uses $\eta = 1/2$. FusedDiff and CompDiff differ only in the composition operator.

E.5 Configurations of Method Variants

For clarity, we summarize the configurations of the dual-target variants used in our experiments. **FusedBFN*** uses the same wPoE-based fusion mechanism as FusedBFN, with $w_1 = w_2 = 1/2$ and the canonical covariance scale $c = Q = 1/2$, but adopts the ligand-prior alignment of Zhou et al. (2024a). **FusedBFN** uses the same probabilistic composition with our chemically weighted ligand-prior alignment. **FusedBFN^{Det}** shares the pretrained backbone, fusion weights, fused means, and chemically weighted alignment of FusedBFN, but sets the sampling concentration parameter to $c = 0$, thereby removing the stochasticity while retaining the same fused signal.

For the cross-framework comparisons, **CompBFN**, **FusedDiff**, and **CompDiff** all use the ligand-prior alignment of Zhou et al. (2024a). Following Zhou et al. (2024a), the probe ligands used for all methods are generated by TargetDiff. No backbone model is retrained or fine-tuned.

Table 4: Configurations of the dual-target methods.

Method	Backbone	Alignment	Setting	Equivalent cov. scale
CompDiff	TargetDiff	LigPrior	$\eta = 1/2$	–
FusedDiff	TargetDiff	LigPrior	$w_1 = w_2 = 1/2$	–
CompBFN	MolCRAFT	LigPrior	$\eta = 1/2$	$c = 1$
FusedBFN*	MolCRAFT	LigPrior	$w_1 = w_2 = 1/2$	$c = 0.5$
FusedBFN	MolCRAFT	Lig-ChemPrior	$w_1 = w_2 = 1/2$	$c = 0.5$
FusedBFN^{Det}	MolCRAFT	Lig-ChemPrior	$w_1 = w_2 = 1/2$	$c = 0$

E.6 Dual-Target Alignment

In Table 5, **Pocket-RI** directly aligns the full protein pockets using RANSAC followed by ICP refinement; **Surface-R** first extracts pocket-surface atoms and aligns them using RANSAC; **Surface-RI** further refines the RANSAC-based surface alignment with ICP; **LigPrior**, proposed by Zhou et al. (2024a), first aligns the probe ligands and then transfers the resulting transformation to the corresponding protein pockets; **Lig-ChemPrior**, introduced in this work, further incorporates chemically aware weighting during ligand alignment.

For Lig-ChemPrior, we assign alignment weights of $\lambda_{\mathrm{heavy}} = 1$ and $\lambda_{\mathrm{light}} = 0.1$ to heavy and light probe-ligand atoms, respectively, with hydrogen atoms treated as light atoms and all other atoms as heavy atoms. For the prior-free alignment, pocket-surface atoms are extracted using P2Rank, RANSAC is run for $1{,}000$ iterations to estimate a coarse rigid transformation, and ICP refines the transformation for at most 50 iterations.

Table 5: Ablation study of different alignment strategies in FusedBFN.

Methods	P-1 Vina Dock (↓)		P-2 Vina Dock (↓)		Max Vina Dock (↓)		Dual High Aff. (↑)		QED (↑)		SA (↑)		Diversity (↑)	
	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.
Pocket-RI	-8.47	-8.47	-8.46	-8.49	-7.73	-7.86	49.4%	50.0%	0.51	0.52	0.64	0.63	0.67	0.67
Surface-R	-8.47	-8.49	-8.50	-8.50	-7.82	-7.90	50.1%	50.0%	0.54	0.55	**0.68**	**0.68**	**0.70**	**0.71**
Surface-RI	-8.50	-8.49	-8.56	-8.54	-7.86	-7.93	50.7%	50.0%	0.53	0.55	0.67	0.66	0.69	0.69
LigPrior	-8.71	-8.72	-8.74	-8.74	-8.03	-8.12	56.2%	60.0%	**0.55**	**0.57**	**0.68**	**0.68**	0.69	0.69
Lig-ChemPrior	**-8.81**	**-8.78**	**-8.84**	**-8.78**	**-8.15**	**-8.16**	**56.3%**	**60.0%**	0.54	0.56	**0.68**	**0.68**	0.68	0.68

Beyond the observations discussed in the main text, Table 5 also shows Surface-RI outperforms Pocket-RI, suggesting that directly aligning complete pockets is less effective because the large number of atoms expands the alignment space and makes reliable registration more difficult. By focusing on the smaller set of pocket-surface atoms, Surface-RI yields more stable transformations. In addition, the improvement of Surface-RI over Surface-R further verifies the benefit of the ICP refinement stage.

F MORE EXPERIMENTAL RESULTS

F.1 BASELINES ON SINGLE-TARGET SETTING

To compare the performance of TargetDiff and MolCRAFT on single-target drug design, we generate 10 molecules for each of the 438 unique targets and evaluate the resulting molecules in terms of binding affinity (Vina Score, Vina Min, Vina Dock, and High Affinity) and molecular properties (drug-likeness QED, synthesizability SA, and diversity). We use the widely adopted AutoDock Vina (Trott & Olson, 2010) to estimate the mean, trimmed mean (i.e., averaging that removes the top 10% and bottom 10% of values before calculating) and median (denoted as "Avg.", "T-Avg." and "Med." respectively) of affinity-related metrics. Vina Score evaluates binding affinity based on the generated poses directly; Vina Min first applies local minimization to the pose and then estimates the affinity; Vina Dock uses a re-docking procedure to approximate the optimal binding affinity; and High Affinity measures the percentage of generated molecules whose binding affinity is better than that of the reference ligand for each test protein.

As shown in Table 6, MolCRAFT generates molecules with higher binding affinity and more favorable molecular properties. This advantage stems from the fact that the Bayesian Flow Network models continuous atomic coordinates and discrete atom types jointly in a unified continuous parameter space. By contrast, TargetDiff is less effective at handling such mixed-modality data, leading to weaker generation quality.

Table 6: Summary of the binding affinity and molecular properties of reference ligands and molecules generated by baselines in the single-target setting. $(\uparrow)/(\downarrow)$ indicates that higher/lower values are preferred. Since the mean is highly sensitive to outliers, some abnormal values of Vina Score and Vina Min are excluded.

Methods	Vina Score (↓)			Vina Min (↓)			Vina Dock (↓)			High Affinity (↑)		QED (↑)		SA (↑)		Diversity (↑)	
	Avg.	T-Avg.	Med.	Avg.	T-Avg.	Med.	Avg.	T-Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.
Reference	-	-8.05	-7.99	-	-8.10	-8.10	-8.06	-8.38	-8.29	-	-	0.54	0.55	0.74	0.78	-	-
TargetDiff	-	-7.38	-7.38	-	-7.89	-7.87	-8.76	-8.78	-8.76	56.5%	60.0%	0.51	0.53	0.58	0.58	0.69	0.70
MolCRAFT	-	-8.33	-8.28	-	-8.53	-8.47	-9.20	-9.11	-9.03	65.0%	70.0%	0.52	0.53	0.66	0.65	0.69	0.68

F.2 COMPARISON OF PROBER GENERATORS

We compare different methods for generating the probe ligand used in prior-based pocket alignment. Specifically, we use TargetDiff and MolCRAFT, respectively, to generate probe ligands, and then align their docked poses on $\mathcal{P}_1$ and $\mathcal{P}_2$ to estimate the pocket transformation. We randomly sample 1,000 dual-target pairs from the dataset and generate 10 molecules for each pair. The results are reported in Table 7. As can be seen, the quality of the generated probe ligand affects the alignment accuracy and consequently the dual-target generation performance. Using MolCRAFT to generate the prober leads to better overall results than using TargetDiff. This is because the single-target molecules generated by MolCRAFT have higher quality, which can more effectively indicate the specific spatial alignment between the two protein pockets.

Table 7: Comparison of probe ligands generated by different methods for prior-based pocket alignment.

Methods	P-1 Vina Dock (↓)		P-2 Vina Dock (↓)		Max Vina Dock (↓)		Dual High Aff. (↑)		QED (↑)		SA (↑)		Diversity (↑)	
	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.
TargetDiff-prior	-8.71	-8.72	-8.74	-8.74	-8.03	-8.12	56.2%	60.0%	0.55	0.57	0.68	0.68	0.69	0.69
MolCRAFT-prior	-8.85	-8.79	-8.85	-8.80	-8.18	-8.19	58.1%	60.0%	0.55	0.57	0.68	0.68	0.69	0.69

F.3 EFFECT OF FUSION WEIGHTS

We study the effects of both the relative and total weights in FusedBFN. As shown in Table 8, $w_1 = w_2 = 0.5$ achieves the best aggregate balance between the two targets, particularly in terms of Max Vina Dock and Dual High Affinity. Asymmetric settings shift generation toward the more

heavily weighted target: increasing w_2 favors P_2, whereas increasing w_1 favors P_1. Comparing the equal-weight settings further shows that either weakening the total weight to $w_1 + w_2 = 0.5$ or increasing it to $w_1 + w_2 = 2$ degrades balanced affinity performance. These results support normalized wPoE with $w_1 + w_2 = 1$ and equal target contributions as a robust default.

Nevertheless, equal weighting is not necessarily optimal for every target pair. For the target pair with UniProt ID P45568 and P35968, Table 9 shows that $(w_1, w_2) = (0.25, 0.75)$ improves the binding affinity for both targets and the molecular properties over equal weighting. Conversely, for P35354 and P31645, Table 10 shows that $(w_1, w_2) = (0.75, 0.25)$ substantially improves the affinity for both targets, although with a modest trade-off in QED and SA. The opposite weighting preferences in these two cases illustrate that the optimal relative weights can depend on the particular target pair. Adaptive selection or learning of w_1 and w_2 for each target pair therefore represents a promising direction for future work.

Table 8: Performance of FusedBFN under different weights.

w_1	w_2	P-1 Vina Dock (↓)		P-2 Vina Dock (↓)		Max Vina Dock (↓)		Dual High Aff. (↑)		QED (↑)		SA (↑)		Diversity (↑)	
		Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.
0.25	0.25	-8.76	-8.70	-8.73	-8.72	-7.94	-8.02	54.2%	60.0%	0.52	0.54	0.62	0.61	0.62	0.62
0.5	0.5	**-8.81**	**-8.78**	**-8.84**	**-8.78**	**-8.15**	**-8.16**	**56.3%**	**60.0%**	0.54	0.56	0.68	0.68	0.68	0.68
1	1	-8.49	-8.45	-8.41	-8.37	-7.82	-7.84	48.3%	50.0%	0.50	0.50	**0.70**	**0.71**	**0.74**	**0.74**
0.25	0.75	-8.27	-8.31	-9.10	-8.98	-7.90	-7.96	50.0%	50.0%	**0.55**	**0.57**	0.67	0.66	0.68	0.69
0.75	0.25	-9.07	-8.90	-8.25	-8.32	-7.89	-7.95	54.7%	60.0%	0.54	0.56	0.67	0.66	0.68	0.67

Table 9: Effect of weights on the target pair with UniProt ID P-1: P45568 and P-2: P35968.

w_1	w_2	P-1 Vina Dock (↓)		P-2 Vina Dock (↓)		Max Vina Dock (↓)		QED (↑)		SA (↑)		Diversity (↑)
		Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	
0.5	0.5	-8.30	-8.43	-7.48	-7.20	-7.43	-7.20	0.29	0.23	0.59	0.59	0.66
0.25	0.75	-10.09	-9.65	-10.17	-10.26	-9.81	-9.47	0.53	0.52	0.74	0.75	0.71

Table 10: Effect of weights on the target pair with UniProt ID P-1: P35354 and P-2: P31645.

w_1	w_2	P-1 Vina Dock (↓)		P-2 Vina Dock (↓)		Max Vina Dock (↓)		QED (↑)		SA (↑)		Diversity (↑)
		Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	
0.5	0.5	-10.17	-10.06	-8.21	-8.18	-9.61	-9.65	0.72	0.73	0.75	0.74	0.70
0.75	0.25	-11.55	-11.64	-10.96	-11.03	-10.86	-10.68	0.64	0.65	0.72	0.72	0.72

F.4 Effect of Categorical Exponent in FusedDiff

We evaluate FusedDiff, FusedDiff $(1, 1)$, and CompDiff on 1,000 randomly selected target pairs, generating 10 molecules for each pair, with the results reported in Table 11. Increasing the categorical weights to $(1, 1)$ does not improve FusedDiff. Although its median docking scores remain relatively close to those of the default setting, its mean affinity deteriorates substantially, indicating a heavier tail of unfavorable samples. The same pattern is observed for Max Vina Dock. This suggests that stronger categorical sharpening does not uniformly degrade the generated molecules, but instead increases the severity of poor sampling outcomes.

Figure 5 further shows that increasing the FusedDiff expert weights substantially increases categorical con-

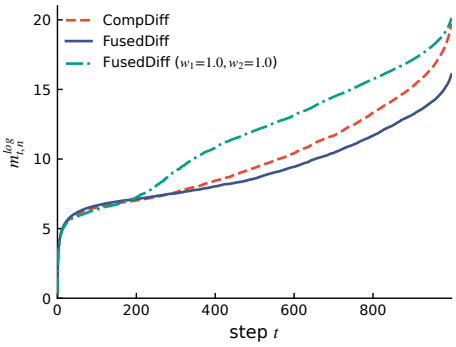

Figure 5: Atom-type confidence during sampling measured by the mean $m_{i,n}^{\log}$ over 100 generated trajectories.

centration. During most of the middle and later sampling process, FusedDiff $(1, 1)$ exhibits even larger log-probability margins than CompDiff. Therefore, the slightly better performance of CompDiff over the default FusedDiff cannot be explained simply by its sharper categorical predictions. Instead, these results indicate that the effect of categorical composition depends on how confidence is introduced within the sampling process, including the order of composition relative to forward corruption.

Table 11: Effect of categorical expert exponents in diffusion-based dual-target generation. FusedDiff $(1, 1)$ changes only the categorical weights, while keeping the coordinate branch unchanged.

Methods	P-1 Vina Dock (↓)		P-2 Vina Dock (↓)		Max Vina Dock (↓)		Dual High Aff. (↑)		QED (↑)		SA (↑)		Diversity (↑)	
	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.
FusedDiff	-8.40	-8.50	-8.45	-8.51	-7.70	-7.89	48.2%	50.0%	0.53	0.54	0.59	0.58	0.68	0.69
FusedDiff $(1, 1)$	-6.87	-8.37	-8.07	-8.39	-5.84	-7.66	44.2%	40.0%	0.54	0.56	0.55	0.56	0.72	0.72
CompDiff	-8.42	-8.56	-8.46	-8.56	-7.58	-7.89	49.6%	50.0%	0.54	0.55	0.59	0.59	0.71	0.72

F.5 Effect of Sampling Concentration

The results in Table 3 indicate that more concentrated sampling distributions can improve dual-target generation. This observation does not contradict the conventional motivation for smoothness in machine learning. The latter generally concerns regularization during training, where a smoother learned function may reduce overfitting and improve generalization. In our setting, however, the pretrained single-target models are not retrained. The concentration studied here instead controls the stochastic uncertainty retained during sampling after two target-aware experts have yielded a shared signal. When the experts agree, additional uncertainty may perturb the trajectory away from a region consistent with both targets rather than regularize the learned model. This suggests that the benefit of sampling concentration depends on the reliability of the shared dual-target signal.

To examine this dependence and identify cases in which the trend may break, we quantify the coordinate and categorical disagreement between the two experts using predictions obtained from the same molecular state. At BFN step i, let $\hat{x}^{(j)}_{M,i}$ be the clean coordinate prediction estimated by expert j. We define

$$D^x_i = \sqrt{\frac{1}{N_M}\sum_{n=1}^{N_M}\left\|\hat{x}^{(1)}_{M,i}[n] - \hat{x}^{(2)}_{M,i}[n]\right\|_2^2}, \qquad D^x_{\text{pair}} = \mathbb{E}_{\text{trajectory},i}[D^x_i].$$

For atom types, let $p^{(j)}_{i,n}$ be the atom-type distribution predicted by expert j for atom n. Define

$$k^{(j)}_{i,n} = \arg\max_{k\in\{1,\dots,K\}} p^{(j)}_{i,n,k}, \qquad g^{(j)}_{i,n} = p^{(j)}_{i,n,(1)} - p^{(j)}_{i,n,(2)},$$

where $p^{(j)}_{i,n,(1)}$ and $p^{(j)}_{i,n,(2)}$ are the largest and second-largest probabilities, respectively. The categorical expert conflict score for atom n is

$$C^v_{i,n} = \mathbf{1}\left[k^{(1)}_{i,n} \neq k^{(2)}_{i,n}\right] \min\left(g^{(1)}_{i,n}, g^{(2)}_{i,n}\right).$$

Averaging $C^v_{i,n}$ over ligand atoms, sampling steps, and trajectories gives the target-pair score C^v_{pair}. Both D^x_{pair} and C^v_{pair} are computed using only the final 50% of the sampling steps. This score is large only when the experts confidently prefer different atom types.

Table 12 summarizes the disagreement scores across $1{,}000$ randomly sampled target pairs, with two trajectories recorded per pair. Both score distributions are strongly skewed to the right, indicating that most target pairs exhibit limited expert disagreement, while a small number of pairs contain substantially conflicting predictions. Based on these distributions, we select two extreme target pairs for further analysis. The

Table 12: Summary statistics of expert disagreement.

Metrics	Avg.	Std.	Med.
D^x_{pair}	2.005	154.872	0.043
C^v_{pair}	0.004	0.019	0

P9WGR1-P23921 pair in Table 13 has exceptionally large coordinate disagreement ($D^x_{\text{pair}} = 1652.857$), whereas the Q9BZX2-P32297 pair in Table 14 exhibits particularly high categorical conflict ($C^v_{\text{pair}} = 0.164$). For both pairs, $c = 0.5$ achieves better binding affinity than $c = 0$. These cases suggest that when the target-specific experts strongly disagree, moderate covariance can be more beneficial than deterministic sampling around an unreliable fused center.

Table 13: Effect of the covariance scale c on the target pair P-1: P9WGR1 and P-2: P23921, for which $D^x_{\text{pair}} = 1652.857$ and $C^v_{\text{pair}} = 0.022$.

c	P-1 Vina Dock (↓)		P-2 Vina Dock (↓)		Max Vina Dock (↓)		QED (↑)		SA (↑)		Diversity (↑)
	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	
0.5	-5.91	-5.62	-6.43	-6.40	-5.91	-5.62	0.46	0.59	0.77	0.67	0.92
0	-5.73	-5.73	-5.83	-5.78	-5.62	-5.57	0.53	0.57	0.84	0.80	0.84

Table 14: Effect of the covariance scale c on the target pair P-1: Q9BZX2 and P-2: P32297, for which $D^x_{\text{pair}} = 0.132$ and $C^v_{\text{pair}} = 0.164$.

c	P-1 Vina Dock (↓)		P-2 Vina Dock (↓)		Max Vina Dock (↓)		QED (↑)		SA (↑)		Diversity (↑)
	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	
0.5	-7.58	-7.56	-7.15	-7.25	-7.02	-7.25	0.59	0.64	0.78	0.78	0.84
0	-7.35	-7.29	-6.46	-6.49	-6.44	-6.42	0.42	0.41	0.75	0.78	0.82

Table 15: Results of LBE and molecular size.

Methods	P-1 LBE (↑)		P-2 LBE (↑)		MW		Atom Num		
	Avg.	Med.	Avg.	Med.	Avg.	Med.	25%	50%	75%
Ref	0.359	0.365	0.296	0.332	368.392	321.40	17	23	32
TargetDiff	0.368	0.356	0.316	0.319	365.32	360.50	20	26	30
MolCRAFT	0.377	0.370	0.329	0.332	359.87	349.28	20	25	31
DualDiff	0.333	0.333	0.335	0.334	369.66	364.50	22	26	31
CompDiff	0.334	0.334	0.336	0.336	367.09	362.47	22	26	31
FusedDiff	0.332	0.331	0.334	0.332	367.09	362.47	22	26	31
CompBFN	0.351	0.348	0.351	0.348	352.89	349.47	21	25	29
FusedBFN*	0.353	0.350	0.353	0.349	350.68	347.41	21	25	29
FusedBFN	0.362	0.359	0.361	0.358	348.31	345.29	21	25	29

F.6 Effect of Chemical Alignment

To verify the general applicability of our chemically weighted alignment, we incorporate it into CompDiff and denote the resulting variant as CompDiff-CA. CompDiff uses the ligand prior alignment introduced by Zhou et al. (2024a), whereas CompDiff-CA uses our chemically weighted alignment. All other settings remain unchanged. We evaluate both methods on 1,000 randomly sampled target pairs, generating 10 molecules for each pair. As shown in Table 16, CompDiff-CA consistently achieves more favorable docking scores, while preserving molecular properties. These results show that the benefit of chemically weighted alignment is not limited to FusedBFN and suggest that it can serve as a transferable alignment strategy across different frameworks.

Table 16: Comparison of different alignment strategies for CompDiff.

Methods	P-1 Vina Dock (↓)		P-2 Vina Dock (↓)		Max Vina Dock (↓)		Dual High Aff. (↑)		QED (↑)		SA (↑)		Diversity (↑)	
	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.
CompDiff	-8.42	-8.56	-8.46	-8.56	-7.58	-7.89	49.6%	50.0%	0.54	0.55	0.59	0.59	0.71	0.72
CompDiff-CA	-8.70	-8.63	-8.74	-8.64	-8.08	-8.00	51.5%	55.6%	0.55	0.57	0.59	0.59	0.71	0.72

F.7 Ligand-Binding Efficiency and Molecular Size

To examine whether the improved docking scores arise simply from generating larger ligands, Table 15 reports ligand-binding efficacy (**LBE**), molecular weight (**MW**), and ligand atom counts (**Atom Num**). For target $j \in \{1, 2\}$, LBE is defined as $\mathrm{LBE}^{(j)} = -E^{(j)}_{\mathrm{vina}}/N_M$, where $E^{(j)}_{\mathrm{vina}}$ is the Vina docking score against target j and N_M is the number of ligand atoms (Lin et al., 2025). A higher LBE indicates a more favorable docking score per atom.

The single-target baselines exhibit a clear imbalance between the two targets, whereas the dual-target methods achieve more balanced LBE values. Among all dual-target methods, FusedBFN attains the highest LBE on both targets while generating molecules with relatively low molecular weights and atom counts. These results indicate that the docking advantage of FusedBFN reflects greater binding efficiency per atom rather than an increase in molecular size.

F.8 Conformation Evaluation

We evaluate the generated molecular conformations using strain energy (SE), steric clashes, and root mean square deviation (RMSD) between the docked poses. For a ligand docked into target $\mathbf{p}^{(j)}$, **P-j SE** measures the intramolecular energy penalty associated with the conformational adjustments required for binding, with a lower value indicating a more energetically favorable conformation. **P-j Clash** counts protein-ligand atom pairs whose distance is smaller than the sum of their van der Waals radii, using a clash tolerance of 0.5 Å. Lower clash counts indicate fewer steric conflicts with the corresponding pocket. SE and Clash are computed using PoseCheck (Harris et al., 2023). **RMSD** (Zhou et al., 2024a) is computed between corresponding ligand atoms in the two poses obtained by docking the same generated molecule into the aligned $\tilde{\mathbf{p}}^{(1)}$ and $\tilde{\mathbf{p}}^{(2)}$ pockets, thereby assessing the binding mode consistency.

Table 17 shows that FusedBFN achieves the lowest SE among all generated-molecule methods on both targets. It also obtains the lowest P-2 Clash and the lowest P-1 Clash among the dual-target methods. Within the MolCRAFT-based methods, the conformation metrics generally improve from CompBFN to FusedBFN* and then to FusedBFN. This trend indicates that distribution fusion and the chemically weighted alignment are associated with conformations of lower energy and fewer steric conflicts under the BFN backbone. By comparison, the single-target baselines exhibit relatively few clashes on $\mathbf{p}^{(1)}$ but substantially more on $\mathbf{p}^{(2)}$, showing that conformations generated for one target do not necessarily accommodate the other pocket.

Figure 6 further compares RMSD between the docked poses of each molecule on dual targets. The distributions shift toward larger values as the number of atoms increases for all methods. Nevertheless, FusedBFN generally achieves the lowest median RMSD, while FusedBFN* is typically comparable to or better than DualDiff. These results suggest that the molecules generated by FusedBFN can adopt more consistent poses across the two aligned pockets, thereby providing more reliable conformations for dual-target binding.

F.9 Evaluation with GNINA

To reduce potential evaluation bias from relying on a single docking algorithm (Choi et al., 2026), we further evaluate the generated molecules using GNINA (McNutt et al., 2025). We randomly select 1,000 target pairs and dock the generated molecules against both targets. In addition to the GNINA docking score (**GNINA Dock**), we report **CNN Score** and **CNN Affinity**. CNN Score reflects the confidence assigned to the docked pose, whereas CNN Affinity provides a learned estimate of

Table 17: Evaluation results on molecular conformation stability under the dual-target setting. For SE, we report the 25th percentile, median, and 75th percentile.

Methods	P-1 SE (↓)			P-2 SE (↓)			P-1 Clash (↓)		P-2 Clash (↓)	
	25%	50%	75%	25%	50%	75%	Avg.	Med.	Avg.	Med.
Reference	41	73	168	39	66	170	6.48	4.00	79.98	46.00
TargetDiff	386	1266	11322	395	1214	11161	9.93	7.00	68.75	47.00
MolCRAFT	123	262	833	122	262	833	**6.08**	**4.00**	66.68	48.00
DualDiff	416	1112	9287	416	1112	9287	23.65	16.00	18.27	12.00
CompDiff	521	1671	17551	521	1671	17551	28.18	21.00	23.69	17.00
FusedDiff	436	1115	8411	436	1115	8411	24.54	18.00	20.99	15.00
CompBFN	131	262	820	131	262	820	18.58	11.00	13.31	8.00
FusedBFN*	123	245	739	123	245	739	17.79	11.00	12.99	8.00
FusedBFN	**102**	**194**	**474**	**101**	**194**	**474**	16.37	10.00	**12.00**	**7.00**

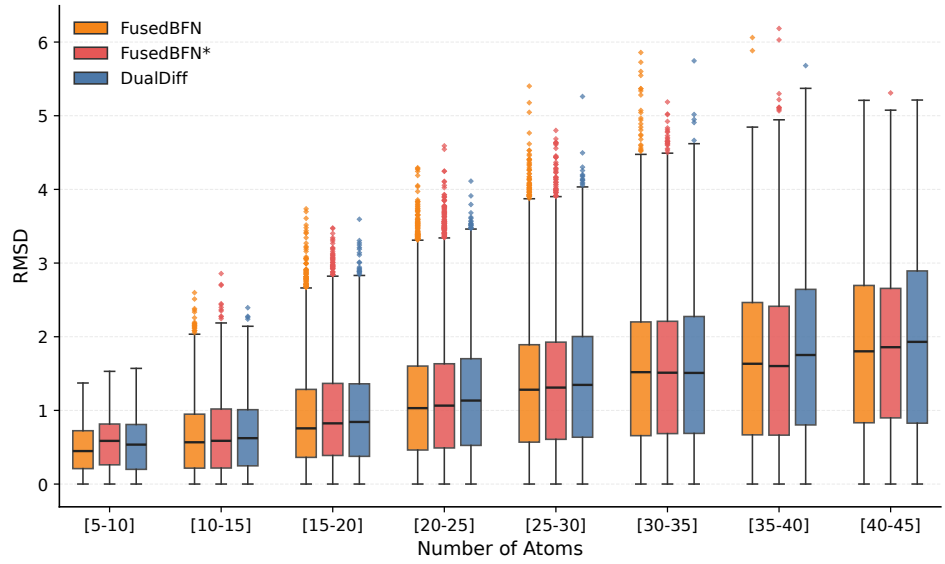

Figure 6: RMSD distributions of docked poses on dual targets grouped by the number of ligand atoms. Lower values indicate better cross-target pose consistency.

binding affinity. We also report **CNN**$_{\mathrm{VS}}$, computed for each docked pose as the product of CNN Score and CNN Affinity, thereby jointly accounting for pose confidence and predicted affinity.

As shown in Table 18, FusedBFN consistently outperforms CompBFN. The deterministic variant, FusedBFN-Det, further improves the CNN-based metrics and achieves the highest CNN$_{\mathrm{VS}}$ among the evaluated methods, providing additional evidence that reducing sampling uncertainty can be beneficial in this setting. The diffusion-based methods obtain higher CNN Affinity, whereas the BFN-based methods generally show more favorable docking and pose-confidence scores. These results suggest that different scoring functions favor different characteristics of the generated molecules.

Table 18: GNINA evaluation on 1,000 randomly selected target pairs.

Methods	P-1 GNINA Dock (↓)		P-2 GNINA Dock (↓)		P-1 CNN Score (↑)		P-2 CNN Score (↑)		P-1 CNN Affinity (↑)		P-2 CNN Affinity (↑)		CNN$_{\mathrm{VS}}$	
	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.
CompDiff	-4.95	-7.65	-4.68	-7.54	0.60	0.62	0.60	0.62	6.09	6.17	6.09	6.17	3.72	3.69
DualDiff	-5.11	-7.59	-4.96	-7.54	0.59	0.61	0.59	0.61	6.01	6.09	6.00	6.05	3.62	3.56
CompBFN	-6.62	-7.89	-6.36	-7.77	0.62	0.64	0.61	0.63	5.70	5.76	5.68	5.72	3.62	3.56
FusedBFN	-6.69	-8.15	-6.75	-8.12	0.63	0.66	0.63	0.65	5.72	5.78	5.71	5.77	3.69	3.65
FusedBFN^{Det}	-6.64	-8.22	-6.39	-8.08	0.65	0.68	0.64	0.67	5.77	5.86	5.76	5.84	3.83	3.81

F.10 Significance Tests on Affinity Metrics

To determine whether the affinity improvements reported in Table 2 are consistent across target pairs, we conduct paired t-tests. For each metric, the results obtained by the two compared methods on the same target pair are treated as paired observations. Note that lower values are preferred for P-1 Vina Dock, P-2 Vina Dock and Max Vina Dock, whereas higher values are preferred for Dual High Affinity. The paired difference is defined as the result of the first method minus that of the second method. Therefore, a negative t-statistic favors the first method for the docking metrics, while a positive t-statistic favors it for Dual High Affinity. As shown in Table 19, all tested metrics yield p-values far below 0.05, indicating statistically significant improvements in dual-target binding affinity from CompBFN to FusedBFN* and further from FusedBFN* to FusedBFN. To account for the repeated use of the same proteins across target pairs, we further conduct cluster-aware bootstrap analyses.

Since the 12,917 benchmark pairs involve only 438 unique targets, different target pairs can share one or both proteins and are therefore not fully independent. We complement the paired tests with two cluster-aware analyses. First, we use a two-way cluster bootstrap that resamples the 438 unique proteins and accounts for repeated targets at both ends of each pair. Second, we perform a target-equal-weighted bootstrap by first averaging the paired differences within each docking target and then assigning equal weight to each target, preventing frequently occurring targets from dominating the estimate. We report the pair-weighted mean difference as the raw effect size together with 95% confidence intervals from 10,000 bootstrap replicates.

As shown in Table 20, the P-1 and P-2 improvements from CompBFN to FusedBFN* remain consistent after accounting for repeated targets. The target-equal estimates are also negative, suggesting that the improvements are not driven by frequently occurring targets. The gains from FusedBFN* to FusedBFN are larger, with both the two-way cluster and target-equal confidence intervals remaining entirely below zero.

F.11 Sampling Efficiency

We further compare the sampling efficiency of different methods in Table 21, which reports the average time required to generate 10 molecules for one dual-target pair on a TITAN Xp GPU with a batch size of 10. The diffusion-based dual-target methods exhibit similar inference times, all close to that of DualDiff, whereas the BFN-based dual-target methods have runtimes comparable to FusedBFN. As a representative BFN-based method, FusedBFN requires 113.68 seconds and is nearly $10\times$ faster than DualDiff. Its runtime is approximately twice that of the single-target MolCRAFT, which is consistent with our design since it constructs two protein-ligand graphs during dual-target generation. When sufficient GPU memory is available, message passing on these two graphs can be executed in parallel, providing further opportunities to accelerate inference.

Table 19: Paired significance tests on affinity-related metrics for FusedBFN* versus CompBFN and FusedBFN versus FusedBFN*.

FusedBFN* vs CompBFN			FusedBFN vs FusedBFN*		
Metrics	t	p	Metrics	t	p
P-1 Vina Dock	-5.30	5.86×10^{-8}	P-1 Vina Dock	-39.27	$< 10^{-336}$
P-2 Vina Dock	-7.50	3.49×10^{-14}	P-2 Vina Dock	-40.27	$< 10^{-354}$
Max Vina Dock	-3.38	3.68×10^{-4}	Max Vina Dock	-30.58	$< 10^{-197}$
Dual High Aff.	8.05	4.68×10^{-16}	Dual High Aff.	37.64	$< 10^{-293}$

Table 20: Cluster-aware analysis of P-1 and P-2 Vina Dock. All confidence intervals are based on 10,000 bootstrap replicates.

Comparison	Metric	Mean diff.	Two-way cluster 95% CI	Target-equal mean [95% CI]
FusedBFN* vs. CompBFN	P-1 Vina Dock	-0.048	[-0.076, -0.021]	-0.062 [-0.090, -0.036]
	P-2 Vina Dock	-0.044	[-0.083, -0.014]	-0.077 [-0.182, -0.016]
FusedBFN vs. FusedBFN*	P-1 Vina Dock	-0.229	[-0.254, -0.207]	-0.233 [-0.265, -0.196]
	P-2 Vina Dock	-0.224	[-0.249, -0.200]	-0.225 [-0.251, -0.199]

G Limitations, Future Work and Broader Impact

Several limitations remain. First, FusedBFN and all its variants depend on the quality of pocket alignment, and inappropriate alignment may negatively affect the dual-target information fusion. Second, our evaluation is retrospective and primarily computational. Docking scores and molecular properties do not establish biochemical activity, target selectivity, or in vivo efficacy. Finally, the present study focuses on two targets and uses equal weights as the default, which may be suboptimal when targets or experts differ in importance and reliability. Future work will incorporate wet-lab validation, construct larger structural datasets of dual-target complexes for direct model training, and select fusion weights tailored to specific target pairs.

This work holds promise for facilitating the discovery of polypharmacological compounds for complex diseases. Nevertheless, generated molecules remain computational hypotheses, and incorrect or overconfident predictions could lead to compounds with poor selectivity, toxicity, or unintended off-target effects. Accordingly, outputs from the proposed framework should not be treated as clinical candidates or used for autonomous decision-making. Responsible use therefore requires expert oversight, rigorous experimental validation, and appropriate safeguards for high-risk applications.

H Visualization Examples

Table 21: Inference time required by different methods to generate 10 molecules. Lower values indicate higher generation efficiency.

Method	MolCRAFT	DualDiff	FusedBFN
Time (s)	57.27	1128.40	113.68

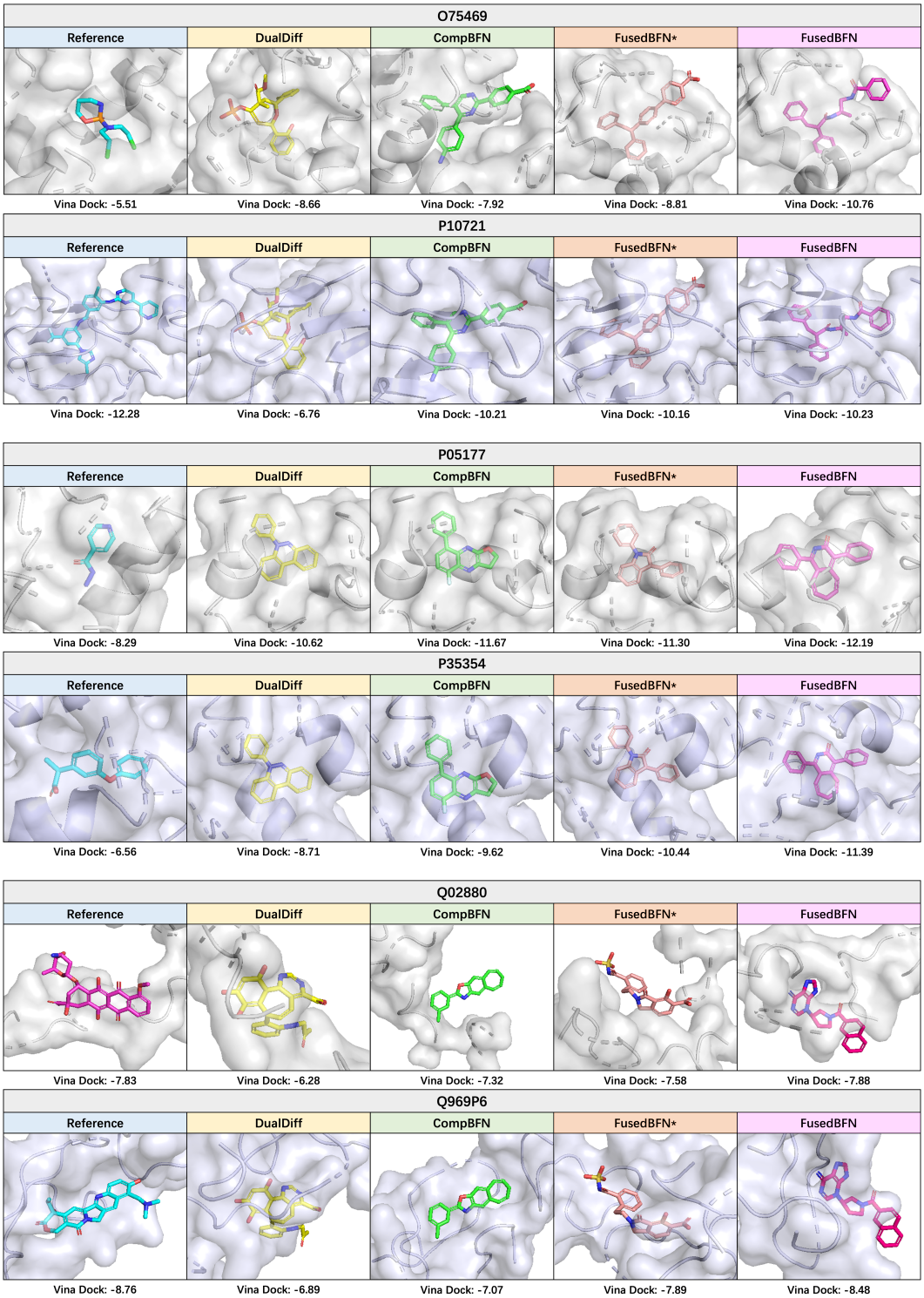

Figure 7: Visualizations of reference ligands and molecules generated by different methods