DBMol: Design of High-Affinity, Target-Specific Small Molecules through Structure Prediction Model

Anonymous Authors[1]

Abstract

Designing small molecule ligands that bind with high affinity to specific protein pockets is a fundamental goal in drug discovery. Recent breakthroughs in structure prediction, such as AlphaFold-3 and Boltz-2, enable accurate biomolecular interaction prediction and show promise as foundation models for downstream tasks, including binding-affinity prediction. We propose to leverage these models and introduce DBMol, a new structure-predictor-guided framework for de novo small molecule design. By leveraging broad protein knowledge from foundation structure models, DBMol targets a challenging setting where the protein pocket is given but no known ligand is available. DBMol formulates small molecule generation as an alternating optimization and projection process. In the optimization stage, DBMol starts from an initial molecule and uses gradient-based optimization to improve pocket-specific interactions and predicted binding affinity using a structure prediction model. In the projection stage, a flow-matching model maps the optimized molecular graph to discrete and chemically valid molecules. Optionally, a post-processing step produces synthetically accessible molecules. Experiments show that DBMol effectively optimizes the Boltz-2 proxy and generates molecules with strong predicted affinity and specificity under Boltz-2 evaluation. To reduce self-confirmation bias, we further evaluate generated molecules using held-out metrics, including AF3-based evaluation. DBMol substantially improves over unconditional generation and achieves competitive performance against stronger baselines under these held-out metrics. These results suggest that structure prediction models can provide useful optimization signals for de novo molecular design, especially in low-supervision settings where reference ligands are unavailable.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Submitted to the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026). Do not distribute.

1. Introduction

Small-molecule modulators that target specific protein pockets with high affinity are central to modern therapeutics. G protein-coupled receptors (GPCRs), for instance, represent one of the most important drug target families. More than 40% of FDA-approved drugs act on GPCRs (Hauser et al., 2018), with the objective of either activating or inhibiting receptor signaling by binding to specific conformational pockets. Achieving both high binding affinity and strong target specificity is therefore a core objective in molecule design, as it directly determines therapeutic efficacy and off-target risk.

Existing protein-ligand design methods can be broadly divided into two categories. Structure-based, pocket-aware generators, such as DiffSBDD (Schneuing et al., 2024), FLOWR (Cremer et al., 2025), and Pocket2Mol (Peng et al., 2022) explicitly condition ligand generation on protein pocket geometry. While effective, these approaches typically rely on large-scale protein-ligand complex datasets to learn interaction patterns. In contrast, ligand-centric models (Adams et al., 2024; Gao et al., 2024; Shen et al., 2025; Rekesh et al., 2025) guide generation using docking objectives, reference ligands, or pharmacophore profiles. Although these methods often achieve strong docking scores, they either rely on known binding ligands or require task-specific reinforcement learning. Consequently, methods that enables structure-conditioned ligand design without curated protein-ligand datasets, reference ligands, or task-specific retraining remains limited.

Structure-based drug design has traditionally been constrained by the limited availability and accuracy of protein-ligand complex structures. Recent advances in protein structure prediction have substantially changed this landscape. AlphaFold-3 (Abramson et al., 2024) demonstrates accurate modeling of multi-molecule complexes, including protein-small molecule interactions, enabling structure-

based in-silico ligand design workflows that were previously impractical at scale. Building on this progress, Boltz-2 (Passaro et al., 2025) adapts structure prediction models into binding-affinity predictors that generalize across diverse protein-ligand systems. These developments suggest that modern structure prediction models can serve as foundational components for molecule interaction modeling, rather than acting merely as structure predictor.

Based on this, we propose DBMol, a small-molecule design framework that uses structure prediction models to generate ligands specific to a target protein pocket. Unlike conventional generative models, DBMol does not rely on curated protein–ligand complex datasets, manually designed reward functions, or supervision from known ligands for the target. Instead, it directly leverages differentiable signals from a structure prediction model, such as Boltz-2 (Passaro et al., 2025), to guide molecule design through an alternating optimization and projection procedure, as illustrated in Figure 1. This allows DBMol to use broad protein-ligand interaction priors from structure prediction models, rather than being restricted by the coverage and bias of curated docking datasets. As a result, DBMol enables flexible de novo small-molecule generation conditioned only on a target protein sequence and target pocket positions. More specifically, DBMol first performs an optimization step (stage I in Figure 1), where a differentiable interaction objective derived from the structure prediction model is optimized to improve predicted binding affinity, pocket specificity, and predicted complex confidence. This objective provides gradient signals that guide molecule representations toward structures aligned with the target protein pocket. However, directly applying differentiable guidance to small molecules is challenging because atom types, formal charges, and bonding patterns are tightly coupled and must satisfy strict chemical constraints. Naïve gradient-based updates can therefore distort the molecular distribution and frequently produce invalid or chemically implausible structures. This necessitates an explicit projection mechanism onto the discrete, chemically valid molecule manifold. DBMol therefore uses a diffusion-based molecule completion model, such as DeFoG (Qin et al., 2025), as the projection stage (stage II in Figure 1). This stage maps optimized but potentially invalid molecule representations back to chemically valid 2D molecule graphs through iterative denoising, while preserving the optimized structural signal. Synthesizability is then encouraged by mapping each generated molecule to a structurally similar, synthetically accessible compound (Gao et al., 2024).

We evaluate DBMol on LIT-PCBA (Tran-Nguyen et al., 2020), a benchmark for target-specific molecular generation. DBMol effectively optimizes the Boltz-2 proxy and substantially improves over unconditional generation under structure-predictor-aligned metrics. These results show that the proposed optimization-and-projection pipeline can capture useful structure-predictor signals for pocket-specific molecule design. Since DBMol is optimized using Boltz-based signals, we further evaluate generated molecules using held-out AF3-based metrics as a complementary structural assessment, rather than treating the optimization proxy as ground truth. Under this held-out evaluation, DBMol remains competitive with reinforcement-learning-based and conditional generative baselines, despite operating under weaker supervision without reference ligands, curated pocket-ligand training pairs, or task-specific retraining. Beyond the benchmark setting, we further show that DBMol can be applied to protein pockets without known ligands, highlighting its intended role as a flexible de novo design framework that can benefit from continued advances in biomolecular structure prediction models.

2. Related Work

Generative models for molecule design. Molecular generative models can be broadly categorized into three classes: SMILES-based sequence models, graph-based 2D models, and 3D structure-aware models. SMILES-based models treat molecules as sequences using autoregressive or transformer-based architectures (Gómez-Bombarelli et al., 2016; Bagal et al., 2022). Graph-based 2D models represent molecules as atom–bond graphs and have been extensively studied, using variational autoencoders (Kusner et al., 2017; Liu et al., 2018; Jin et al., 2019), normalizing flows (Zang & Wang, 2020), and diffusion-based methods (Vignac et al., 2022). Discrete flow-matching frameworks such as DeFoG (Qin et al., 2025) further improve generation efficiency while preserving strong performance. 3D structure-aware models aim to jointly generate molecule topology and three-dimensional atomic coordinates, typically using E(3)-equivariant architectures with diffusion or flow matching (Hoogeboom et al., 2022; Vignac et al., 2023; Irwin et al., 2025), or keeping a non-equivariant architecture (Joshi et al., 2025; Vonessen et al., 2025). Recently, there has been increased attention to generating molecules that, beyond accurate 2D topology and geometric fidelity, are constrained to synthesizable space (Cretu et al., 2025; Koziarski et al., 2024; Shen et al., 2025; Gao et al., 2024; Rekesh et al., 2025). In this work, we focus on 2D molecule graph generation, which provides a discrete representation of molecule structure while aligning naturally with the Boltz-2 input format, and we incorporate a synthesis-aware pipeline for practical molecule design.

Structure prediction models. Structure prediction models have moved beyond single-protein folding and are in-

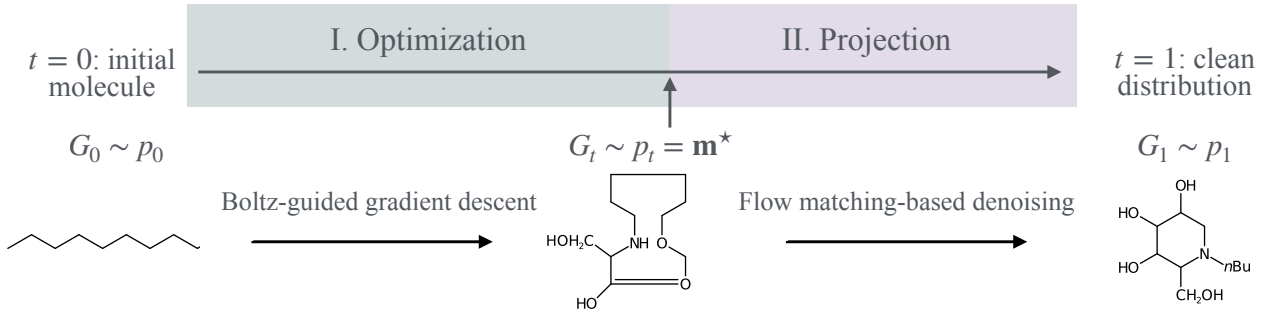

Figure 1. **Framework of DBMol**. (I) **Optimization**, where Boltz-2-conditioned gradient updates start from G_0, an uninformative initial molecule graph visualized here as a simple linear carbon chain, and refine relaxed molecule representation to improve predicted binding affinity and pocket specificity for a given protein pocket; and (II). **Projection**, where a flow-matching-based molecule denoising model maps a noisy molecule G_t (in its optimized representation $\mathbf{m}^\star$) onto discrete, chemically valid molecule graphs G_1.

creasingly used as foundation models for biomolecular systems. AlphaFold-3 (Abramson et al., 2024) showed that diffusion-based all-atom models can predict multi-component complexes, including protein-ligand interactions and nucleic-acid assemblies. Recent open-source efforts further extend structure prediction toward functional readouts such as binding affinity and conformational change. For example, Boltz-2 (Passaro et al., 2025) reports protein-ligand affinity predictions competitive with physics-based free-energy methods such as FEP, while being substantially faster, making these models attractive for virtual screening and reducing reliance on expensive simulations. Structure prediction models have also been explored for inverse design, primarily in the context of protein sequence and structure optimization. Representative approaches include all-atom sequence design with explicit ligands and ions (Yi et al., 2025), diffusion-time hallucination for de novo protein discovery (Cho et al., 2025b), and gradient-based optimization through differentiable structure predictors for protein–protein or protein–nucleic acid interfaces (Pacesa et al., 2025; Team et al., 2025; Cho et al., 2025a). However, despite these advances, a comparable differentiable, structure-predictor-guided framework for *small-molecule* design remains largely unexplored.

3. DBMol

In this section, we present DBMol, an optimization–projection framework for structure-guided molecule design. We first define the problem setting (Sec. 3.1), then describe how gradients from the differentiable structure predictor Boltz-2 are used to optimize a relaxed molecular representation (Sec. 3.2). Next, we introduce a discrete flow–matching projection step to recover chemically valid molecular graphs (Sec. 3.3). Finally, we present a refinement stage that enforces synthesizability (Sec. 3.4).

3.1. Problem Setting and Molecular Representation

We study *structure-predictor-aligned molecule design*, which can be viewed as a conditional generation problem guided by a differentiable surrogate objective. Given a protein sequence and a predefined binding pocket, we treat a structure prediction model (e.g., Boltz-2 (Passaro et al., 2025)) as a differentiable scoring function and optimize molecules to improve predicted protein-ligand interactions. Importantly, we do not assume access to ground-truth binding affinities, and we do not treat the structure predictor as a physical oracle; instead, it serves as a proxy objective whose inductive biases guide molecule design.

Molecular representation. We represent a molecule as an undirected graph with N nodes,

$$G = \big(x_{1:N}, c_{1:N}, e_{1:i<j:N}\big), \tag{1}$$

where $x_{1:N} = (x^{(n)})_{1\le n\le N}$ denote atom types, $c_{1:N} = (c^{(n)})_{1\le n\le N}$ denote formal charges, and $e_{1:i<j:N} = (e^{(ij)})_{1\le i<j\le N}$ denote bond types. Each atom type satisfies $x^{(n)} \in \mathcal{X}$, each charge $c^{(n)} \in \mathcal{C}$, and each bond type $e^{(ij)} \in \mathcal{E}$, where one category in $\mathcal{E}$ explicitly represents the absence of a bond. Following prior work (Vignac et al., 2022) on molecule graph generation, we assume a fully connected graph, allowing bonded and non-bonded atom pairs to be modeled within a unified categorical space. We adopt this discrete notation for Sec. 3.3.

To enable gradient-based optimization on Boltz-2, we adopt a *continuous relaxation* of the discrete molecule graph. Specifically, atom types $x_{1:N}$ and bond types $e_{1:i<j:N}$ are represented as categorical probability vectors on the probability simplex, while formal charges $c_{1:N}$ are represented as continuous values to align with Boltz-2. We denote this relaxed representation by $\mathbf{m}$, and distinguish it from the discrete molecule graph G. This relaxation does not carry a probabilistic interpretation over molecules, nor does it define a generative distribution; rather, it serves as a parameterization that enables constrained gradient-based design using differentiable structure prediction models. We adopt this relaxed notation for Sec. 3.2.

We next describe the structure-predictor-guided optimiza-

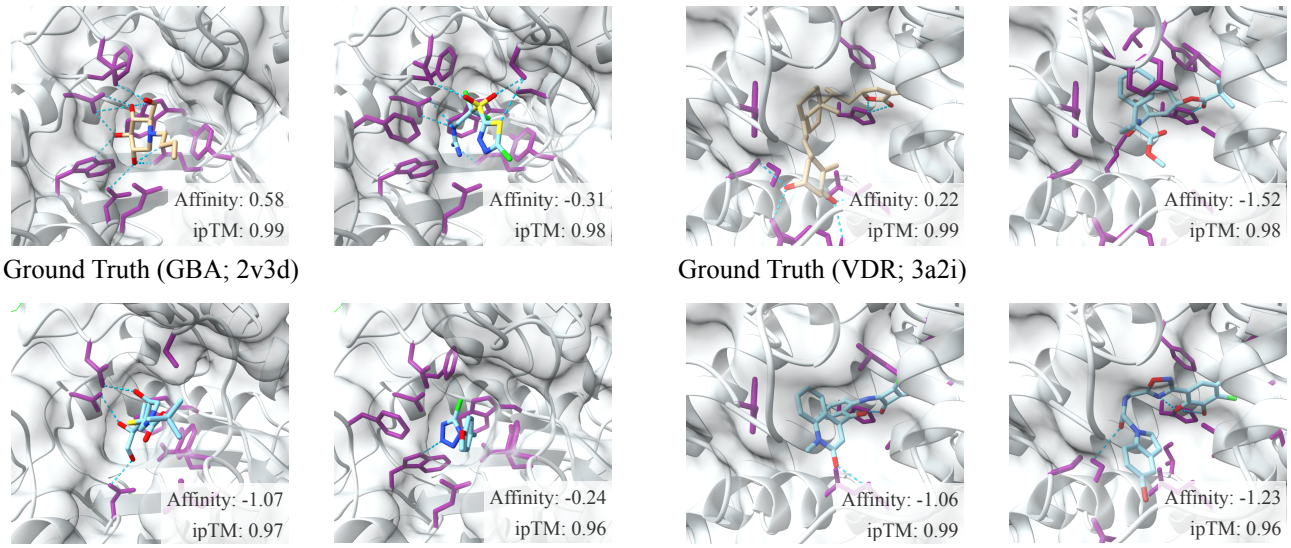

Figure 2. **Protein–ligand interactions predicted by Boltz-2.** For each target, the reference ligand and its interaction pattern are shown in beige in the upper left, while three molecules found via DBMol after SynFormer post-processing are shown in blue. The protein backbone and molecular surface are shown in gray, interacting protein side chains are highlighted in purple, and hydrogen bonds are shown as dashed light-blue lines. Across both proteins, the generated molecules bind within the target pocket and exhibit coherent predicted binding poses. The predicted binding affinities and ipTM scores are reported in the bottom right of each subfigure.

tion of DBMol under the notations above.

3.2. Optimization through Structure Prediction Models

We leverage a structure prediction model to guide molecule design through gradient-based optimization. In particular, we adopt Boltz-2 as a structure predictor that provides differentiable signals characterizing protein-ligand interactions, including binding affinity estimates and structural confidence measures. These signals define a fixed objective for optimization; DBMol does not perform model retraining, reinforcement learning, or inference over learned molecule distributions.

Objective. Given a relaxed molecule representation $\mathbf{m}$, we define a composite objective

$$\mathcal{L}(\mathbf{m}) = \mathcal{L}_{\text{affinity}}(\mathbf{m}) + w_1 \mathcal{L}_{\text{contact}}(\mathbf{m}) + w_2 \mathcal{L}_{\text{conf}}(\mathbf{m}), \tag{2}$$

where $\mathcal{L}$affinity minimizes the predicted protein-ligand binding affinity from the structure prediction model, biasing generation toward stronger binders. $\mathcal{L}_{\text{contact}}$ encourages interactions with residues in the predefined binding pocket, while $\mathcal{L}_{\text{conf}}$ aggregates Boltz-2 confidence signals, including ipLDDT, ipDE, pDE, confidence score, and pTM, to regularize the predicted complex geometry. All terms are written as losses, so lower values are preferred: higher-is-better confidence scores are negated, and error-like quantities are used as penalties. The weights (w_1, w_2) balance these objectives, with details provided in Sec. C. Overall, the objective encourages strong, target-specific binders with high-confidence predicted interfaces.

Contact loss. To encourage pocket-specific binding, we add a contact loss based on the distogram predicted by the structure prediction model. For each pocket residue i and ligand token j, the model outputs logits over D distance bins. After softmax, we obtain $p_{ij}(d)$ and define

$$\mathcal{L}_{\text{pair}}(i,j) = -\log\left(\sum_{d<d_{\text{cutoff}}} p_{ij}(d) + \epsilon\right), \tag{3}$$

where ϵ is a small constant and $d_{\text{cutoff}} = 8.0$ Å. We compute this loss only over pocket residues $\mathcal{P}$. For each residue i, we select the ligand atom with the strongest predicted contact, $j_i^\star = \arg\min_j \mathcal{L}_{\text{pair}}(i,j)$, and define

$$\mathcal{L}_{\text{pocket}}(\mathbf{m}) = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} \mathcal{L}_{\text{pair}}(i, j_i^\star). \tag{4}$$

Simultaneously enforcing many pocket–ligand contacts can make optimization unstable and lead to noisy gradients. We therefore add an anchoring term based on the single strongest pocket residue–ligand atom pair:

$$(i^\star, j^\star) = \arg\min_{i\in\mathcal{P}, j} \mathcal{L}_{\text{pair}}(i,j), \mathcal{L}_{\text{anchor}}(\mathbf{m}) = \mathcal{L}_{\text{pair}}(i^\star, j^\star). \tag{5}$$

The final contact loss averages the pocket-level and anchor-level terms:

$$\mathcal{L}_{\text{contact}}(\mathbf{m}) = \frac{1}{2}\mathcal{L}_{\text{pocket}}(\mathbf{m}) + \frac{1}{2}\mathcal{L}_{\text{anchor}}(\mathbf{m}). \tag{6}$$

The anchor term provides a stable global pull toward the target pocket, while the pocket-level term encourages broader local contacts, improving optimization stability and reducing diffuse interaction patterns.

Gradient updates in the relaxed space. Direct optimization over discrete molecule graphs is infeasible due to combinatorial constraints. We therefore perform gradient-based optimization in the relaxed molecule space defined in the previous subsection. At iteration k, we compute gradients of the objective with respect to the relaxed representation of the molecule $\mathbf{m}_k$,

$$\mathbf{g}_k = \nabla_{\mathbf{m}_k} \mathcal{L}(\mathbf{m}_k), \tag{7}$$

and apply constrained gradient descent to preserve valid categorical simplices.

For atom and bond type variables, the update takes the form

$$\tilde{\mathbf{m}}_{k+1} = \mathrm{clip}(\mathbf{m}_k - \eta \mathbf{g}_k, 0, 1) \to \mathbf{m}_{k+1} = \frac{\tilde{\mathbf{m}}_{k+1}}{\sum_l \tilde{\mathbf{m}}_{k+1}^{(l)}}.$$

Here, clipping prevents invalid negative or overly large category weights, while normalization ensures that each atom-type or bond-type vector sums to one. For continuous charge variables, we apply additive gradient-descent updates followed by clipping to a predefined range. A detailed description of how gradients affect atom types, bond types, and atomic charges is provided in Sec. C.2.

While these operations implement constrained gradient descent in the relaxed space, they do not by themselves ensure chemical feasibility. We therefore introduce a separate projection (denoising) step in Sec. 3.3. Boltz-2 is kept frozen throughout optimization: gradients are backpropagated through the structure predictor only to update the relaxed molecule variables, and are not used to update Boltz-2 or the denoising model.

3.3. Projection to Chemically Valid Molecules

While continuous relaxation enables efficient gradient-based optimization, directly discretizing the optimized representation $\mathbf{m}$ almost always produces invalid or chemically implausible molecular graphs. This occurs because atom types and bond configurations are tightly coupled, and these constraints are not enforced in the relaxed space. To address this issue, we introduce a *projection step* that maps noisy or invalid proposals to discrete molecular graphs that satisfy basic chemical validity constraints.

Learned projection via discrete denoising. We instantiate this projection using a denoising model based on discrete flow matching, specifically DeFoG (Qin et al., 2025). Discrete flow matching formulates denoising as a continuous-time process over discrete states, enabling gradual and controlled corrections of atom , bond, and charge types. This property makes it particularly well suited for projection, as local chemical inconsistencies can be resolved without overwriting the global structural signal introduced during optimization. Full technical details of the discrete flow-matching formulation and training procedure are provided in Appendix Sec. B.1.

DeFoG is trained only on chemically valid molecular graphs and learns to recover chemically consistent structures from corrupted graph inputs. We do not interpret this denoising process as probabilistic inference; instead, we use it as a learned projection operator that enforces chemical validity while preserving the structural signal introduced by optimization stage in Sec. 3.2.

Given an optimized relaxed representation $\mathbf{m}^\star$, we first sample a noisy discrete molecule graph $\tilde{G}$ according to $\mathbf{m}^\star$. Since charges are encoded as scalars in $\mathbf{m}^\star$ in Boltz-2, we discretize them in $\tilde{G}$ by rounding to the nearest integer. However, this sampled graph may violate chemical constraints. Therefore, we then apply the denoising model to project $\tilde{G}$ back onto a valid molecule graph G.

Late-time denoising. A key design choice is the initialization point of the denoising process in the flow-matching trajectory. Starting from an early denoising time introduces too much randomness and can wash out the optimization signal from the structure predictor. Starting from an overly late time, however, produces many invalid structures that the denoiser cannot reliably repair. We therefore initialize denoising at a *late diffusion time* t_0, which allows the model to ensure sufficient chemical validity while largely preserving the optimized molecule structure. In practice, we set $t_0 = 0.8$ in all experiments, which provides a stable trade-off between validity and preservation of the optimization signal.

3.4. Synthesizability-Aware Design

Chemical validity does not necessarily imply practical synthesizability: a generated molecule may satisfy valence constraints while still being unstable, implausible, or difficult to synthesize. We therefore include an optional synthesis-aware post-processing step based on SynFormer (Gao & Coley, 2020). Given a molecule generated by DBMol, SynFormer maps it to a structurally similar molecule that is reachable through chemically valid reaction pathways. This step is used only to assess whether the target-specific signal discovered by DBMol can be preserved after moving molecules toward synthesizable chemical space; it is not part of the Boltz-guided optimization objective. In our synthesis-aware variant, this mapping is applied independently to each DBMol molecule, without Boltz-2-, AF3-, Vina-, or docking-based filtering.

4. Experiments

We evaluate whether DBMol can use structure prediction models as optimization signals for de novo molecule design. Our experiments have three goals: (i) to test

whether DBMol optimizes the Boltz-2 proxy while transferring to held-out metrics as AF3-based evaluation on the seven-target LIT-PCBA protocol; (ii) to assess whether this framework extends to de novo targets beyond the original benchmark; and (iii) to validate the main design choices through ablations.

4.1. Experimental Setup

Baselines. We compare DBMol with representative baselines across conditioning and supervision regimes. DeFoG is an unconditional graph-generation reference. SynFormer (Gao et al., 2024) uses reference-ligand guidance; ShEPhERD (Adams et al., 2024) and SynCoGen (Rekesh et al., 2025) use pharmacophore guidance; CGFlow (Shen et al., 2025) uses amortized pocket rewards; and DiffSBDD (Schneuing et al., 2024) is a pocket-conditioned structure-based generator for the strict de novo setting. DiffSBDD conditions on 3D pocket information, typically from a broader 8 Å pocket region, whereas DBMol only uses pocket residue positions within 3.5 Å. Thus, DBMol operates under weaker pocket conditioning.

Together, these baselines cover the main regimes relevant to target-specific molecule generation, including no target information, reference ligands, pharmacophores, learned pocket rewards, and pocket-conditioned protein–ligand training data. Our goal is not to claim universal dominance, but to test whether structure-predictor-guided optimization can achieve competitive target-aware performance under weaker supervision.

Evaluation protocol. For the main benchmark, we report 100 molecules per target for each method. For DBMol, we generate 2048 candidates, discard chemically invalid projected graphs, and randomly sample 100 valid molecules, which takes around 10 seconds in total. This step filters only chemical validity and never uses Boltz-2, AF3, Vina, diversity, or any target-specific score.

Metrics. We evaluate all methods using held-out structural metrics, proxy-aligned reference metrics, and chemical-quality metrics, with details in Sec. C.8. Our primary evaluation uses *AlphaFold3-based held-out metrics*, since AlphaFold3 is not used during DBMol optimization. These include BindCov, measuring AF3-predicted pocket coverage; AF3 Succ., a combined success rate based on sufficient AF3 ipTM and successful pocket binding; iPAE, the AF3-predicted interface error; and Dist, the mean minimum ligand-to-pocket distance. We also report AutoDock Vina scores and molecular diversity, measured by average pairwise ECFP4 Tanimoto distance, both used only for evaluation. We additionally report *Boltz2 Success*, defined by favorable Boltz-2 affinity and sufficient pocket specificity. Since Boltz-2 provides the optimization signal, Boltz2 Success is treated only as a diagnostic proxy-aligned reference metric rather than the primary basis for comparison.

4.2. Pocket-Specific Molecule Design on LIT-PCBA

We evaluate DBMol on the seven-target LIT-PCBA protocol used in prior work. For DBMol, we use one fixed optimization configuration across all LIT-PCBA targets, including loss weights, optimization hyperparameters, and 50 optimization iterations. This avoids target-specific iteration selection or hyperparameter tuning. The full setup is reported in Appendix C. We use the ground-truth ligand size only to fix the number of atoms; no reference-ligand geometry, or interaction pattern is used.

We first ask whether the optimization-projection framework improves the proxy. The comparison in Tab. 1 shows that DBMol effectively exploits structure-predictor guidance. Under the proxy-aligned Boltz2 Success metric, DBMol improves over unconditional generation from 0.13 to 0.47, indicating that projection preserves useful optimization signals after mapping relaxed representations back to discrete molecule graphs. We then test whether this signal transfers beyond the Boltz-2 proxy using held-out AF3-based metrics.

We next ask whether Boltz-guided optimization transfers to held-out evaluation. DBMol achieves strong held-out performance despite using weaker supervision than several baselines. Using only the target pocket and the structure predictor, DBMol obtains a BCov of 0.51, an AF3 Success of 0.92, and high diversity of 0.89. Its average rank over held-out metrics is 3.83, which is competitive with pocket-conditioned and pharmacophore-guided methods. SynFormer achieves the best average rank and the strongest BCov and Dist, but it relies on the strongest conditioning setting, where a ground-truth ligand is available; CGFlow-ZS achieves a very low Vina score, supported by its training reward encouraging a proxy of this metric. Nevertheless, DBMol remains competitive under weaker supervision and achieves the highest diversity among target-aware methods. These results suggest that the Boltz-guided optimization signal transfers to independent AF3-based evaluation metrics, supporting DBMol as a flexible structure-predictor-guided design framework.

Proxy tradeoff analysis. We further analyze the proxy tradeoff by comparing DBMol with a less confidence-focused variant that optimizes only the ipTM confidence term for L_{conf}. This variant increases Boltz2 Success from 0.47 to 0.69, surpassing the strongest baseline score of 0.62. Both this variant and DBMol reach a high $ipTM$ of 0.85, indicating similar confidence under this metric. This confirms that DBMol can strongly optimize the Boltz-

Table 1. **Mean molecular generation results averaged across the seven LIT-PCBA targets.** Avg. Rank is computed over all metrics except Boltz2 Success using standard competition ranking. DBMol results are reported from 100 valid molecules without Boltz-2-, AF3-, Vina-, or docking-based post-selection. Gray rows correspond to our method, and the best Avg. Rank and Boltz2 Success are highlighted in bold.

Model	Condition	BCov ↑	AF3 Succ. ↑	iPAE ↓	Dist ↓	Div. ↑	Vina ↓	Avg. Rank ↓	Boltz Succ. ↑
DeFoG	Unconditional	0.39	0.89	11.7	4.6	0.89	-8.3	5.50	0.13
SynFormer	Ligand	0.55	0.93	10.8	3.8	0.87	-8.9	**1.83**	0.47
ShEPhERD	Pharmacophore	0.42	0.91	10.8	4.2	0.87	-8.5	4.00	0.20
SynCoGen	Pharmacophore	0.51	0.92	11.6	4.5	0.50	-10.3	3.67	**0.62**
CGFlow-ZS	Pocket-reward	0.46	0.92	10.9	4.5	0.78	-10.4	3.33	0.32
DiffSBDD	3D pocket	0.50	0.94	10.9	4.6	0.78	-8.8	3.67	0.44
DBMol	Pocket	0.51	0.92	11.8	4.6	0.89	-8.7	3.83	0.47

Table 2. **Effect of SynFormer post-processing.** We compare DBMol candidates before and after SynFormer-based synthesizability post-processing. *Retained* reports the average number of molecules kept per target after post-processing or size filtering, referred as *s.f.*.

Setting	Retained	BCov ↑	AF3 Succ. ↑	iPAE ↓	Dist ↓	Vina ↓	Div ↑	Avg. Rank ↓
DBMol	100	0.51	0.92	11.8	4.6	-8.7	0.89	**1.83**
+ SynFormer	∼97	0.44	0.88	11.7	4.9	-8.0	0.88	3.50
+ s.f. ($\geq 15\& \pm 5$)	∼59	0.47	0.91	11.3	4.5	-8.3	0.88	2.00
+ s.f. ($\geq 15\& \pm 3$)	∼41	0.48	0.91	11.5	4.4	-8.2	0.88	2.00

2 proxy, especially affinity and pocket specificity. However, this proxy gain comes with worse held-out metrics: BCov drops from 0.51 to 0.46, iPAE increases from 11.8 to 12.3, Dist increases from 4.6 to 4.7, and Vina weakens from -8.7 to -8.3. AF3 Success also slightly decreases from 0.92 to 0.91. These results show that stronger proxy optimization alone is insufficient, and that a more balanced confidence objective is important for improving held-out transfer.

Synthesizability. We evaluate DBMol-Syn to test whether the DBMol signal is preserved after moving molecules into a more synthesizable region of chemical space. For each DBMol molecule, SynFormer maps it to a structurally similar and synthetically accessible neighbor, without Boltz-2-, AF3-, Vina-, or docking-based filtering. Thus, DBMol-Syn is a synthesis-aware robustness check rather than an additional score-based selection procedure.

As shown in Table 2, SynFormer preserves much of the AF3-based signal. AF3 Success decreases from 0.92 to 0.88, and BCov decreases from 0.51 to 0.44, while iPAE remains stable. A simple size-consistency filter recovers much of this gap, reaching 0.91 AF3 Success and 0.48 BCov under the ± 3 filter, with improved Dist and iPAE. This suggests that the optimized signal is better preserved when the SynFormer neighbor remains close in molecular size, consistent with the fact that DBMol optimizes each molecule under its original size scale. Overall, synthesis-aware refinement introduces a tradeoff, but much of the DBMol signal transfers to more synthesizable molecules when the mapped neighbor is size-consistent and structurally similar.

Table 3. **De novo molecule design results on additional targets.** We compare DBMol with DiffSBDD, a pocket-conditioned structure-based baseline applicable to the strict de novo setting without reference ligands.

Method	BCov ↑	AF3 Succ. ↑	iPAE ↓	Dist ↓	Vina ↓	Div ↑	Avg. Rank ↓
LGR4							
DiffSBDD	0.13	0.34	13.5	6.6	-9.4	0.92	1.67
DBMol	0.22	0.60	12.4	5.1	-7.6	0.88	**1.33**
CD47							
DiffSBDD	0.01	0.03	8.4	15.2	-4.1	0.90	**1.50**
DBMol	0.02	0.06	8.8	18.1	-6.3	0.89	**1.50**

4.3. De Novo Molecule Design on Additional Targets

We next evaluate DBMol in a stricter de novo setting, which is its intended use case. Unlike methods requiring reference ligands, pharmacophore annotations, or pocket–ligand training pairs, DBMol only uses the target pocket positions inside the protein sequence, and a structure prediction model at test time. This makes it applicable to new or weakly annotated targets. DiffSBDD requires a well-defined 3D pocket geometry. Since these proteins do not have a complete predefined 3D pocket, we define it using the pocket region induced by a high-scoring DBMol-generated molecule.

We consider two additional targets: LGR4 (Wang et al., 2025) and CD47 (Zhao et al., 2022). These provide complementary test cases, including a GPCR target and a protein–protein interaction immune-checkpoint target. We compare DBMol against DiffSBDD using the same AF3-based metrics, Vina, and diversity as in the main evaluation. The results are summarized in Tab. 3.

On LGR4, DBMol improves all AF3-based metrics over DiffSBDD, including BCov, AF3 Success, iPAE, and Dist, while DiffSBDD obtains a better Vina score. On CD47, DBMol improves BCov, AF3 Success, and Vina, whereas DiffSBDD remains better in iPAE and Dist, leading to the

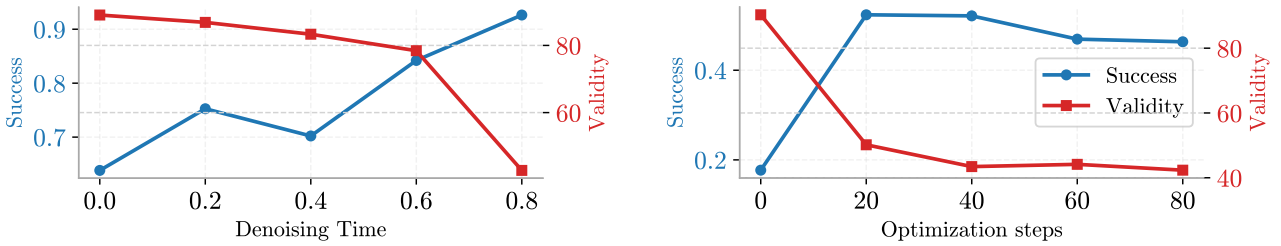

Figure 3. Success rate based on Boltz2 and molecular validity as functions of denoising time (left) and optimization steps (right).

same average rank. Overall, DBMol provides competitive results on de novo targets without reference-ligands.

4.4. Additional results

Optimization–projection tradeoff. We analyze the tradeoff between target-specific success and chemical validity. Stronger structure-predictor guidance can better preserve pocket-specific signals, but it can also move relaxed molecular representations farther from the discrete manifold of valid molecule graphs. We vary two control parameters: the denoising initialization time and the number of Boltz-guided optimization steps.

As shown in the left of Figure 3, initializing denoising at later diffusion times makes the final generated molecules closer to the optimized relaxed representation. This improves the preservation of Boltz-guided structural signals and leads to higher success rates. However, because fewer denoising steps remain, the model has less opportunity to correct chemically invalid structures, resulting in a clear decrease in chemical validity. Earlier initialization, while reducing success, provides stronger projection back to valid molecules.

The right panel of Figure 3 isolates the effect of Boltz-guided gradient refinement by evaluating the same initial relaxed candidates at different optimization steps, while keeping the projection and evaluation protocol fixed. Target-engagement metrics improve rapidly in the early stages and then saturate, indicating that the gradient updates directly contribute to the observed improvement rather than merely increasing the candidate pool. At the same time, validity decreases with excessive optimization. We therefore use a fixed 50-step protocol for all LIT-PCBA targets in the main comparison.

Additional analyses. We provide additional analyses in the appendix. In Sec. D.4, we show that DBMol supports ligand-conditioned iterative refinement by further optimizing a promising molecule found by DBMol. In Sec. D.3, we show it finds molecules close to known actives, without directly copying known ligands. In Sec. D.6, we discuss the choice of the denoiser. In Sec. D.1, we study the necessity of using both contact- and affinity-related losses.

5. Conclusion

We introduced DBMol, an optimization-centric framework that leverages differentiable signals from modern protein-ligand structure predictors for small-molecule design without docking supervision, reference ligands, or curated protein-ligand datasets. Results on LIT-PCBA show that structure-predictor-guided optimization improves pocket specificity and predicted binding affinity relative to unconditional generation, supporting the practical utility of structure-based guidance.

Several directions remain open and motivate future work. First, DBMol relies on surrogate objectives, which are inevitably imperfect and may not always align with experimental binding. As structure prediction models improve, DBMol may benefit from more accurate and better-aligned optimization signals. A key current limitation is the scarcity of high-quality protein–ligand binding datasets (Škrinjar et al., 2025); larger and more open datasets, such as OpenBind (OpenBind, 2024), could provide more robust evaluation signals closer to real-world design objectives. In addition, extending DBMol from 2D molecular graphs to full 3D protein–ligand representations would enable richer geometric reasoning and tighter integration with structure prediction models. Finally, improving computational efficiency, particularly through parallelization and reduced overall cost, remains an important direction.

Overall, DBMol substantially improves over unconditional generation under Boltz2-based metrics, while remaining competitive with reinforcement-learning-based and ligand- or interaction-conditioned baselines under held-out structural evaluation. These results suggest that structure prediction models can serve as useful differentiable surrogate objectives for target-aware molecular generation, while experimental binding validation remains an important direction for future work.

Impact Statement

The primary objective of this paper is to advance graph generation under a more flexible framework, with applications spanning general graph generation, molecular design, and digital pathology. The ability to generate graphs with discrete labels can have broad-reaching implications for fields such as drug discovery and diagnostic technologies. While this development has the potential to bring about both positive and negative societal or ethical impacts, particularly in areas like biomedical and chemical research, we currently do not foresee any immediate societal concerns associated with the proposed methodology.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pp. 1–3, 2024.

Adams, K., Abeywardane, K., Fromer, J., and Coley, C. W. ShEPhERD: Diffusing shape, electrostatics, and pharmacophores for bioisosteric drug design, 2024.

Bagal, V., Aggarwal, R., Vinod, P. K., and Priyakumar, U. D. Molgpt: Molecular generation using a transformer-decoder model. *J. Chem. Inf. Model.*, 62(9): 2064–2076, 2022. doi: 10.1021/ACS.JCIM.1C00600. URL https://doi.org/10.1021/acs.jcim.1c00600.

Campbell, A., Yim, J., Barzilay, R., Rainforth, T., and Jaakkola, T. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *International Conference on Machine Learning (ICML)*, 2024.

Cho, Y., Pacesa, M., Zhang, Z., Correia, B. E., and Ovchinnikov, S. Boltzdesign1: Inverting all-atom structure prediction model for generalized biomolecular binder design. *bioRxiv*, pp. 2025–04, 2025a.

Cho, Y., Rangel, G., Bhardwaj, G., and Ovchinnikov, S. Protein hunter: exploiting structure hallucination within diffusion for protein design. *bioRxiv*, pp. 2025–10, 2025b.

Cremer, J., Irwin, R., Tibo, A., Janet, J. P., Olsson, S., and Clevert, D.-A. Flowr: Flow matching for structure-aware de novo, interaction- and fragment-based ligand generation. *ArXiv*, abs/2504.10564, 2025. URL https://api.semanticscholar.org/CorpusID:277786923.

Cretu, M., Harris, C., Igashov, I., Schneuing, A., Segler, M., Correia, B., Roy, J., Bengio, E., and Lio, P. Synflownet: Design of diverse and novel molecules with synthesis constraints. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=uvHmnahyp1.

Gao, W. and Coley, C. W. The synthesizability of molecules proposed by generative models. *Journal of Chemical Information and Modeling*, 60(12):5714–5723, 2020.

Gao, W., Luo, S., and Coley, C. W. Generative artificial intelligence for navigating synthesizable chemical space. *arXiv preprint arXiv:2410.03494*, 2024.

Gómez-Bombarelli, R., Duvenaud, D. K., Hernández-Lobato, J. M., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4:268 – 276, 2016. URL https://api.semanticscholar.org/CorpusID:3347345.

Hauser, A. S., Chavali, S., Masuho, I., Jahn, L. J., Martemyanov, K. A., Gloriam, D. E., and Babu, M. M. Pharmacogenomics of gpcr drug targets. *Cell*, 172(1):41–54, 2018.

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

Irwin, R., Tibo, A., Janet, J. P., and Olsson, S. Semlaflow–efficient 3d molecular generation with latent attention and equivariant flow matching. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

Jin, W., Barzilay, R., and Jaakkola, T. Junction tree variational autoencoder for molecular graph generation, 2019. URL https://arxiv.org/abs/1802.04364.

Jocys, Z., Willems, H. M., and Farrahi, K. SynthFormer: Equivariant pharmacophore-based generation of molecules for ligand-based drug design. *arXiv preprint arXiv:2410.02718*, 2024.

Joshi, C. K., Fu, X., Liao, Y.-L., Gharakhanyan, V., Miller, B. K., Sriram, A., and Ulissi, Z. W. All-atom diffusion transformers: Unified generative modelling of molecules and materials. In *International Conference on Machine Learning*, 2025.

Koziarski, M., Rekesh, A., Shevchuk, D., van der Sloot, A., Gaiński, P., Bengio, Y., Liu, C., Tyers, M., and Batey, R. RGFN: Synthesizable molecular generation using GFlowNets. *Advances in Neural Information Processing Systems*, 37:46908–46955, 2024.

Kusner, M. J., Paige, B., and Hernández-Lobato, J. M. Grammar variational autoencoder. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1945–1954. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/kusner17a.html.

Liu, Q., Allamanis, M., Brockschmidt, M., and Gaunt, A. Constrained graph variational autoencoders for molecule design. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Ma, L., Lin, C., Lim, D., Romero-Soriano, A., Dokania, P. K., Coates, M., Torr, P., and Lim, S.-N. Graph inductive biases in transformers without message passing. In *International Conference on Machine Learning (ICML)*, 2023.

OpenBind. Openbind: Open protein–ligand binding data. https://openbind.uk/, 2024. Accessed: January 28, 2026.

Pacesa, M., Nickel, L., Schellhaas, C., Schmidt, J., Pyatova, E., Kissling, L., Barendse, P., Choudhury, J., Kapoor, S., Alcaraz-Serna, A., et al. One-shot design of functional protein binders with bindcraft. *Nature*, 646 (8084):483–492, 2025.

Passaro, S., Corso, G., Wohlwend, J., Reveiz, M., Thaler, S., Somnath, V. R., Getz, N., Portnoi, T., Roy, J., Stark, H., et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, pp. 2025–06, 2025.

Peng, X., Luo, S., Guan, J., Xie, Q., Peng, J., and Ma, J. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. *ArXiv*, abs/2205.07249, 2022. URL https://api.semanticscholar.org/CorpusID:248811125.

Qin, Y., Vignac, C., and Frossard, P. Sparse training of discrete diffusion models for graph generation. In *ArXiv*, 2023.

Qin, Y., Madeira, M., Thanou, D., and Frossard, P. Defog: Discrete flow matching for graph generation. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Rekesh, A., Cretu, M., Shevchuk, D., Somnath, V. R., Liò, P., Batey, R. A., Tyers, M., Koziarski, M., and Liu, C.-H. Syncogen: Synthesizable 3d molecule generation via joint reaction and coordinate modeling. *arXiv preprint arXiv:2507.11818*, 2025.

Schneuing, A., Harris, C., Du, Y., Didi, K., Jamasb, A., Igashov, I., Du, W., Gomes, C., Blundell, T. L., Lio, P., et al. Structure-based drug design with equivariant diffusion models. *Nature Computational Science*, 4(12): 899–909, 2024.

Shen, T., Seo, S., Lee, G., Pandey, M., Smith, J. R., Cherkasov, A., Kim, W. Y., and Ester, M. TacoGFN: Target-conditioned GFlownet for structure-based drug design. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=N8cPv95zOU.

Shen, T., Seo, S., Irwin, R., Didi, K., Olsson, S., Kim, W. Y., and Ester, M. Compositional flows for 3D molecule and synthesis pathway co-design. *arXiv preprint arXiv:2504.08051*, 2025.

Siraudin, A., Malliaros, F. D., and Morris, C. Cometh: A continuous-time discrete-state graph diffusion model. In *ArXiv*, 2024.

Škrinjar, P., Eberhardt, J., Durairaj, J., and Schwede, T. Have protein-ligand co-folding methods moved beyond memorisation? *BioRxiv*, pp. 2025–02, 2025.

Sterling, T. and Irwin, J. J. Zinc 15–ligand discovery for everyone. In *Journal of Chemical Information and Modeling*, 2015.

Team, P., Ren, M., Sun, J., Guan, J., Liu, C., Gong, C., Wang, Y., Wang, L., Cai, Q., Ma, W., et al. Pxdesign: Fast, modular, and accurate de novo design of protein binders. *bioRxiv*, pp. 2025–08, 2025.

Tran-Nguyen, V.-K., Jacquemard, C., and Rognan, D. Lit-pcba: An unbiased data set for machine learning and virtual screening. *Journal of Chemical Information and Modeling*, 60(9):4263–4273, 2020. doi: 10.1021/acs.jcim.0c00155. URL https://doi.org/10.1021/acs.jcim.0c00155. PMID: 32282202.

Vignac, C., Krawczuk, I., Siraudin, A., Wang, B., Cevher, V., and Frossard, P. Digress: Discrete denoising diffusion for graph generation. In *International Conference on Machine Learning (ICML)*, 2022.

Vignac, C., Osman, N., Toni, L., and Frossard, P. Midi: Mixed graph and 3d denoising diffusion for molecule generation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, 2023.

Vonessen, C., Harris, C., Cretu, M., and Liò, P. Tabasco: A fast, simplified model for molecular generation with improved physical quality, 2025. URL https://arxiv.org/abs/2507.00899.

Wang, L., Hu, F., Cui, Q., et al. Structural insights into the lgr4–rspo2–znrf3 complexes regulating wnt/β-catenin signaling. *Nature Communications*, 16:362, 2025. doi: 10.1038/s41467-024-55431-3.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

Yi, K., Jamali, K., and Scheres, S. H. All-atom inverse protein folding through discrete flow matching. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=8tQdwSCJmA.

Zang, C. and Wang, F. Moflow: an invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 617–626, 2020.

Zhao, H., Song, S., Ma, J., Yan, Z., Xie, H., Feng, Y., and Che, S. Cd47 as a promising therapeutic target in oncology. *Frontiers in Immunology*, 2022. doi: 10.3389/fimmu.2022.757480. Published 2022 Aug 22.

Zhu, W., Wen, T., Song, G., Wang, L., and Zheng, B. On structural expressive power of graph transformers. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2023.

A. Discussions

A.1. Alternatives

A possible alternative formulation of our problem is conditional generation. Since Boltz-2 provides differentiable gradients in the logits space, these gradients could, in principle, be used to modulate the rate matrix of the flow matching denoising process. This would inject structural guidance from the structure prediction model directly during sampling.

In practice, however, this approach introduces substantial computational and optimization challenges. First, it requires calling Boltz-2 to compute gradients at each denoising step. A single backward pass of Boltz-2 takes approximately 30 seconds. Even if the conditional signal is applied only at a small number of time steps, the total cost remains prohibitively high. Second, when the time variable t is close to 0, the conditional gradients are dominated by noise and become ineffective. In contrast, near the end of sampling, the logits are highly peaked and approach one-hot vectors. Meaningful modification at this stage requires very large step sizes, which leads to unstable optimization and makes the process difficult to tune.

Moreover, even after applying the mapping procedure described in Sec. C.2 to align Boltz-2 gradients, additional structural difficulties arise. The molecular graphs used by Boltz-2 and DeFoG differ in node ordering and construction. As a result, gradients must be carefully aligned through typed graph isomorphism before they can be applied. This alignment requires simultaneously constraining node types and edge types to prevent ambiguous matches caused by purely topological isomorphisms. Although we implemented this alignment pipeline, the isomorphism step itself is computationally expensive and difficult to stabilize, further increasing system complexity.

For these reasons, we adopt this two stage approach proposed in this work, which decouples structural guidance from molecular generation at the modeling level. This design avoids the high cost of step wise guidance, the instability and tuning difficulty of conditional denoising, and the implementation complexity introduced by cross representation graph isomorphism. As a result, the overall pipeline is significantly more efficient and robust.

B. Background

B.1. Discrete Flow Matching

In generative modeling, the primary goal is to generate new data samples from the underlying distribution that produced the original data, $\boldsymbol{p}_{\text{data}}$. An effective approach is to learn a mapping between a simpler distribution $\boldsymbol{p}_\epsilon$ that can be easily sampled, and $\boldsymbol{p}_{\text{data}}$.

We consider a discrete variable $z_t \in \mathcal{Z} = \{1, \dots, Z\}$ evolving over time $t \in [0, 1]$. Its marginal distribution is denoted by $\boldsymbol{p}_t \in \Delta^Z$. The initial distribution is set to a predefined noise distribution, $\boldsymbol{p}_0 = \boldsymbol{p}_\epsilon$, while $\boldsymbol{p}_1 = \boldsymbol{p}_{\text{data}}$ represents the target data distribution. We refer to the mapping $t : 1 \to 0$ as the noising process and $t : 0 \to 1$ as the denoising process.

Following (Campbell et al., 2024), the noising trajectory is defined through a simple linear interpolation conditioned on a datapoint z_1:

$$p_{t|1}(z_t|z_1) = t\,\delta(z_t, z_1) + (1 - t)\,p_0(z_t). \tag{8}$$

A common choice for p_0 is the uniform distribution over $\mathcal{Z}$.

Denoising as a CTMC. The denoising process is formulated as a continuous-time Markov chain (CTMC). A CTMC is characterized by a rate matrix $\boldsymbol{R}_t \in \mathbb{R}^{Z \times Z}$, which governs the instantaneous transition rates between states. For an infinitesimal time step $\mathrm{d}t$, the transition probabilities satisfy:

$$p_{t+\mathrm{d}t|t}(z_{t+\mathrm{d}t}|z_t) = \delta(z_t, z_{t+\mathrm{d}t}) + R_t(z_t, z_{t+\mathrm{d}t})\,\mathrm{d}t. \tag{9}$$

By definition, $R_t(z_t, z_{t+\mathrm{d}t}) \geq 0$ for $z_t \neq z_{t+\mathrm{d}t}$, and the diagonal entries are set to ensure normalization. The evolution of the marginal distribution follows the Kolmogorov forward equation, $\partial_t \boldsymbol{p}_t = \boldsymbol{R}_t^\top \boldsymbol{p}_t$.

Conditional rate construction. Similarly to the noising process, denoising is performed under conditioning on z_1. We consider a z_1-conditional rate matrix $\boldsymbol{R}_t(\cdot, \cdot|z_1)$ that satisfies the corresponding Kolmogorov equation. Under mild assumptions, (Campbell et al., 2024) derive a closed-form expression for a valid conditional rate matrix. For $z_t \neq z_{t+\mathrm{d}t}$, it is given by:

$$R_t^*(z_t, z_{t+\mathrm{d}t}|z_1) = \frac{\mathrm{ReLU}[\partial_t p_{t|1}(z_{t+\mathrm{d}t}|z_1) - \partial_t p_{t|1}(z_t|z_1)]}{Z_t^{>0}\,p_{t|1}(z_t|z_1)}. \tag{10}$$

Intuitively, this rate redistributes probability mass toward states that require increased marginal probability. The unconditional rate matrix used for sampling is obtained by marginalizing over z_1:

$$R_t(z_t, z_{t+\mathrm{d}t}) = \mathbb{E}_{p_{1|t}(z_1|z_t)}[R_t(z_{t+\mathrm{d}t}, z_t|z_1)].$$

B.2. Discrete Flow Matching for Graphs

Graph notation. We denote by $G = (x^{(1:n:N)}, c^{(1:n:N)}, e^{(1:i\neq j:N)})$ a directed graph with N nodes. Nodes, charges, and edges are categorical variables with $x^{(n)} \in \{1, \dots, X\}$, $c^{(n)} \in \{1, \dots, C\}$, and $e^{(i,j)} \in \{1, \dots, E\}$.

Training objective. The denoising model is parameterized by a neural network with parameters θ, trained to predict the clean node, charge, and edge distributions given a noisy graph G_t. The predicted distributions are denoted by $\boldsymbol{p}^\theta_{1|t}(\cdot|G_t)$.

Training is performed using a cross-entropy loss applied independently to nodes, charges, and edges:

$$\mathcal{L} = \mathbb{E}_{t, G_1, G_t}\left[-\sum_n \log p^{\theta,(n)}_{1|t}(x_1^{(n)}|G_t) - \lambda_c \sum_n \log p^{\theta,(c,n)}_{1|t}(c_1^{(n)}|G_t) - \lambda_e \sum_{i\neq j} \log p^{\theta,(i,j)}_{1|t}(e_1^{(i,j)}|G_t)\right]. \tag{11}$$

Denoising and sampling. At sampling time, denoising is formulated as a CTMC evolving from an initial graph $G_0 \sim \boldsymbol{p}_0$ toward $t = 1$. The transition kernel is defined by a rate matrix $\boldsymbol{R}_t$, approximated using the network predictions $\boldsymbol{p}^\theta_{1|t}$. In practice, the continuous-time dynamics are discretized using a finite step size Δt, resulting in an Euler-style update.

Algorithm 1 DeFoG Training

Input: Graph dataset $\mathcal{D} = \{G^1, \ldots, G^M\}$
while f_θ not converged **do**
  Sample $G \sim \mathcal{D}$
  Sample $t \sim \mathcal{T}$
  Sample $G_t \sim p_{t|1}(G_t|G)$ {Noising}
  $h \leftarrow \mathrm{RRWP}(G_t)$ {Extra features}
  $\boldsymbol{p}^\theta_{1|t}(\cdot|G_t) \leftarrow f_\theta(G_t, h, t)$ {Denoising prediction}
  loss $\leftarrow \mathrm{CE}_\lambda(G, \boldsymbol{p}^\theta_{1|t}(\cdot|G_t))$
  optimizer.step(loss)
=0

Algorithm 2 DeFoG Sampling

Input: # graphs to sample S
for $i = 1$ **to** S **do**
  Sample N from train set {# Nodes}
  Sample $G_0 \sim p_0(G_0)$
  for $t = 0$ **to** $1 - \Delta t$ **with step** Δt **do**
    $h \leftarrow \mathrm{RRWP}(G_t)$ {Extra features}
    $\boldsymbol{p}^\theta_{1|t}(\cdot|G_t) \leftarrow f_\theta(G_t, h, t)$ {Denoising prediction}
    $G_{t+\Delta t} \sim \tilde{p}_{t+\Delta t|t}(G_{t+\Delta t}|G_t)$ {Eq. (9)}
  Store G_1
=0

Algorithmic summary. For clarity, the overall training and sampling procedures of DeFoG are summarized in Alg. 1 and Alg. 2, respectively.

Discussion. Discrete flow matching offers an efficient and flexible framework for generative modeling in discrete domains. A key advantage of this formulation is the decoupling between training and sampling: the denoising model is trained to predict clean data distributions, while the sampling dynamics are specified separately through a continuous-time stochastic process. This separation is particularly advantageous for graph generation, where rigid sampling schemes can be computationally expensive due to the combinatorial structure of nodes and edges. As demonstrated in DeFoG (Qin et al., 2025), this flexibility can be leveraged through tailored sampling strategies, including modified rate constructions and adaptive denoising schedules, to substantially reduce the number of generated steps required at inference time without degrading generation quality. These properties make discrete flow matching especially suited for scalable graph generation and motivate its use as the foundation of our approach.

Algorithm 3 Boltz-guided optimization (Sec. 3.2)

Input: pocket $\mathcal{P}$, Boltz2, DeFoG denoiser f_θ
Hyperparams: weights (w_1, w_2), optimization steps K, late start t_0, step size η,
Init: simple discrete molecule G_0
$\mathbf{m}_0 \leftarrow \mathrm{Relax}(G_0)$ {node/edge probs + charges}
for $k = 0$ **to** K **do**
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{aff}}(\mathbf{m}_k) + w_1\mathcal{L}_{\text{contact}}(\mathbf{m}_k) + w_2\mathcal{L}_{\text{conf}}(\mathbf{m}_k)$
  $\mathbf{g}_k \leftarrow \nabla_{\mathbf{m}_k}\mathcal{L}(\mathbf{m}_k)$ {Boltz-2 gradients}
  $\mathbf{m}_{k+1} \leftarrow \mathrm{Update}(\mathbf{m}_k, \mathbf{g}_k, \eta)$ {distribution update}
$\tilde{G} \sim p(\cdot \mid \mathbf{m}_K)$ {sample noisy molecules}
$G_1 \leftarrow \mathrm{Denoise}(\tilde{G}, f_\theta, t_0)$ {Alg. 4}
$G_0 \leftarrow G_1$
return G_1 =0

Algorithm 4 Projection: DeFoG-based denoising (Sec. 3.3)

Input: initial graph $\tilde{G}$, denoiser f_θ, late start t_0, step Δt
Initialize $G_{t_0} \leftarrow \tilde{G}$
for $t = t_0$ **to** $1 - \Delta t$ **do**
  $\boldsymbol{p}^{\theta}_{1|t}(\cdot \mid G_t) \leftarrow f_\theta(G_t, t)$ {denoising prediction}
  $G_{t+\Delta t} \sim \tilde{p}_{t+\Delta t|t}(\cdot \mid G_t)$ {Euler step}
return G_1 =0

C. Experimental Details

This section provides further details on the experimental settings used in the paper.

C.1. Algorithms

We provide the algorithm for DBMol framework here in Alg. 3 and in Alg. 4.

C.2. Boltz2 Gradient Computation Setup.

This section describes how gradients are extracted from the Boltz2 model (Passaro et al., 2025) and incorporated into the DeFoG sampling procedure. The design goal is to use Boltz2 as an external differentiable evaluator that provides guidance signals. We therefore focus on minimal and targeted modifications to the Boltz2 inference pipeline and on a clean interface for mapping gradients into discrete sampling.

Enabling gradient support in Boltz2. To enable gradient-based guidance, Boltz2 must expose gradients with respect to molecule structure variables while preserving stable structure prediction and affinity estimation. The modifications described below are restricted to inference-time behavior and do not change the trained model parameters.

Differentiable relaxation of discrete inputs. Boltz-2 takes SMILES strings as input and preprocesses them into a structured object, denoted as `batch`, which contains all molecular features required for inference. Among these features, we focus on three components that are relevant for gradient-based optimization: atom types, bond types, and atomic charges.

First, Boltz-2 represents atom types using a large atomic vocabulary, where each atom is encoded by an index corresponding to its chemical element, for example carbon is indexed as 6, stored as `batch["ref_element"]` In contrast, DeFoG operates on a much smaller, fixed set of atom-type categories, where carbon corresponds to a single category in the DeFoG vocabulary. This discrepancy prevents atom-type representations from Boltz-2 from being updated directly in the DeFoG space. To resolve this mismatch, we first map Boltz-2 atom types to the subset of atom types supported by DeFoG, and restrict gradient updates to these matched categories. We then treat atom types as continuous variables by converting their discrete one-hot representations into continuous categorical vectors, which enables gradient-based optimization while keeping unsupported atom types fixed.

Bond types in Boltz-2 are encoded as integers and is passed through an embedding layer. To make this embedding step differentiable, we explicitly construct one-hot bond-type representations instead of directly encoding the type index integer, stored as `batch["type_bonds_onehot"]`. We then modify the implementation inside the Boltz-2 model so that the bond embedding is applied via matrix multiplication, which allows gradients to propagate through the bond-type representation during optimization.

Last, atomic charges are modeled as continuous scalar values throughout the pipeline, stored as

batch["ref_charge"]. Since this term is concatenated into the atom feature tensor, it does not naturally align with categorical representations; we therefore keep it in its original scalar form. As a result, gradients with respect to atomic charges can be computed directly, without requiring any additional relaxation or approximation.

Losses. Boltz-2 confidence outputs include several complementary estimates of complex quality, and their values may depend on recycling convergence and low-confidence or disordered regions. We therefore define $\mathcal{L}_{\text{conf}}$ as a weighted combination of selected confidence terms from the Boltz-2 confidence model output. Specifically, we use terms such as complex_iplddt, which encourages high local confidence around interface tokens; complex_ipde and complex_pde, which penalize large predicted distance errors at the interface and over the complex; ptm, which encourages reliable global complex geometry; and confidence_score, an aggregate confidence measure. For scores where higher values indicate higher confidence, we use their negative values as losses, while for error-based quantities such as PDE/ipDE, we minimize the predicted error directly. The exact weighted composition of $\mathcal{L}_{\text{conf}}$ is provided in Sec. C.

Boltz2 also predicts protein-ligand binding affinity, which reflects how strongly a small molecule binds to a target protein and provides a direct optimization signal for ligand generation. The affinity model outputs scalar predictions through affinity_pred_value. We use this value directly as the loss function and minimize it to bias generation toward higher-affinity binding candidates. At the same time, we also maximize affinity_probability_binary that suggests the predicted probability that the ligand binds to its target. Both terms are equally weighted in our loss function.

We define a contact loss based on Boltz2-predicted distance distributions (p-distogram) to encourage protein-ligand contacts. Details are provided in Sec. 3.2.

Memory-efficient gradient computation. Direct gradient computation in Boltz2 is memory-intensive. We apply three optimizations. First, the recycling mechanism is limited to a single refinement step, which substantially reduces memory usage while preserving gradient quality in practice. Second, gradient checkpointing is applied to pairwise representation update modules for lower memory consumption. Third, as mentioned in the main text, the gradient does not pass through the structure module of Boltz-2.

Requirement computational resources and running time. GPU memory usage for Boltz-2–based optimization primarily depends on protein length. In our experiments, this step consistently requires less than 65 GB of GPU memory. For molecule generation and evaluation, all setups require less than 20 GB of GPU memory. All experiments are performed on NVIDIA A100 or H100 GPUs. For efficiency, MSA searches are performed once per protein and cached. During optimization, we reuse these preprocessed MSAs, resulting in an average runtime of approximately 30 seconds per optimization step.

Hyperparameters. In our framework, the molecular size is fixed at the beginning of generation. For targets with a co-crystallized ligand, we set the number of atoms to match the ground-truth ligand size. This is the only use of reference-ligand information in our pipeline; no reference-ligand geometry, chemical identity, or interaction pattern is used during optimization.

For the main LIT-PCBA benchmark, we use a single default optimization configuration across all proteins. In particular, we fix the optimization hyperparameters and report results after 50 optimization iterations for every target, rather than selecting target-specific iteration counts. In practice, we use SGD with learning rate $\eta = 2.0$ for relaxed molecule optimization. The loss weights are fixed across all LIT-PCBA experiments, with $\omega_1 = 2.5$, and $\omega_2 = 2.5$. For the contact loss, we use $d_{\text{cutoff}} = 8.0$ Å, and assign equal weights to the pocket-averaged term and the anchoring term. The confidence loss $\mathcal{L}_{\text{conf}}$ combines the nonzero-weight Boltz-2 confidence terms, including ipTM, ipLDDT, ipDE, PDE, confidence score, and pTM, with weights 1.0, 0.3, 0.3, 0.1, 0.3, and 0.1, respectively. All other confidence terms are assigned zero weight. For the projection stage, we initialize the discrete denoising process at $t_0 = 0.8$. The same default configuration is used for the component ablations unless otherwise stated.

All generated molecules are evaluated with the same AF3-based evaluation pipeline. AF3 metrics are not used during molecular optimization; they are used after generation for evaluation and target-level reporting. The de novo hyperparameters were primarily chosen according to the Boltz-based optimization protocol, rather than through extensive AF3-guided tuning.

Due to the computational cost of AF3-based evaluation, we perform only a limited hyperparameter search over the loss terms rather than exhaustive tuning. The reported results should therefore be interpreted as performance under a practical, modest tuning budget. More systematic hyperparameter optimization may further improve performance, especially for additional de novo targets and target-specific design settings; we leave this direction to future work.

C.3. Mapping Boltz2 Gradients to DeFoG Sampling

Handling aromatic bond representations. Boltz2 and DeFoG may adopt different conventions for aromatic bonds, such as explicit aromatic categories or kekulized representations. To avoid inconsistencies during alignment and gradient transfer, we use DeFoG checkpoints trained with explicit aromatic bond types. This choice simplifies matching and removes the need to redistribute gradient mass across bond categories.

Projection of gradients into the DeFoG state space. Gradients are projected into the discrete state spaces supported by DeFoG. Atom-type gradients are mapped from Boltz's larger atomic vocabulary to the element set used by DeFoG, namely $\{$`C`, `N`, `O`, `F`, `P`, `S`, `Cl`, `Br`, `I`$\}$. Bond-type gradients are mapped to DeFoG edge categories $\{$`NO-EDGE`, `SINGLE`, `DOUBLE`, `TRIPLE`, `AROMATIC`$\}$. Charge gradients are compressed into a discrete set of charge states $\{0, +1, -1\}$.

C.4. DeFoG Setup

Denoising neural architecture. DeFoG's denoising neural network takes a noisy graph G_t as input and predicts the clean marginal probability for each node $x^{(n)}$ via $p_{1|t}^{\theta,(n)}(\cdot|G_t)$ and for each edge $e^{(ij)}$ via $p_{1|t}^{\theta,(ij)}(\cdot|G_t)$. This formulation boils down the graph generative task to a graph-to-graph mapping. While both message-passing layers and graph transformers can be used for this task, graph transformers have empirically outperformed message-passing layers in graph generation (Qin et al., 2023). DeFoG thus adopts the transformer architecture of (Vignac et al., 2022), using multi-head attention layers which encode node, edge, and graph-level features while preserving node permutation equivariance.

Enhancing model expressivity. Graph neural networks, including graph transformers, have inherent limitations in their expressive power (Xu et al., 2019; Zhu et al., 2023). An usual approach to overcome the limited representation power of graph neural networks consists of explicitly augmenting the inputs with features that the networks would otherwise struggle to learn. We adopt Relative Random Walk Probabilities (RRWP) encodings that are proved to be expressive for both discriminative (Ma et al., 2023) and generative settings (Siraudin et al., 2024). RRWP encodes the likelihood of traversing from one node to another in a graph through random walks of varying lengths, offering insights into graph dynamics across different hop distances. In particular, given a graph with an adjacency matrix A, we generate $K-1$ powers of its degree-normalized adjacency matrix, $M = D^{-1}A$, i.e., $\left[I, M, M^2, \ldots, M^{K-1}\right]$. We concatenate the diagonal entries of each power to their corresponding node embedding, while combining and appending the non-diagonal to their corresponding edge embeddings.

Training setup. We encode aromatic bonds as an additional bond type and explicitly model formal charges during training. The model is trained on a single NVIDIA A100 GPU for approximately 50 hours for 300 epochs. Model checkpoints are selected based on molecule validity evaluated on a validation set, with the final checkpoint chosen after 38 hours of training.

Sampling setup. We use the `polydec` time distortion function in DeFoG for both training and sampling to improve efficiency. During sampling, the denoising process is run for 200 steps with an initial time $t_0 = 0.8$. We find that $t_0 = 0.8$ provides a good trade-off between generating valid molecules and preserving the strength of Boltz-derived optimization signals. Reducing the number of denoising steps leads to a noticeable drop in validity. We therefore use 200 steps as a balanced choice that maintains both sample quality and computational efficiency. For each experiment, we generate 2048 molecules and randomly select 100 valid molecules for evaluation. We adopt this oversampling strategy because the guided sampling distribution can deviate substantially from the clean data distribution on which DeFoG is trained, leading to a non-negligible fraction of invalid samples. Despite this, the overall generation process remains highly efficient: sampling involves only a small number of denoising steps over relatively small molecule graphs, resulting in a total runtime of under 10 seconds (8 seconds for generation, 2 seconds for filtering) for all generated samples.

C.5. Dataset Details

LIT-PCBA. LIT-PCBA (Tran-Nguyen et al., 2020) is a carefully curated virtual screening benchmark constructed from 149 dose-response PubChem BioAssays. The dataset is filtered to remove false positives, assay artifacts, and major physicochemical imbalances, and retains only targets with experimentally resolved ligand-bound structures. After target selection and asymmetric validation embedding (AVE), LIT-PCBA consists of 15 targets with 7,844 confirmed actives and 407,381 confirmed inactives. We fix the size of the molecules during generation based on the reference ligand, which is considered as the only information leakage in our generation pipeline. We also report results for molecules with a fixed size of 25 atoms per protein, corresponding to the median reference molecule size across the 7 proteins, and use this setting for ablation studies as shown in Sec. D.1. In our experiments, we evaluate on 7 targets: Estrogen Receptor (ESR; PDB IDs: 2iok and 2p15), Glucocerebrosidase (GBA; 2v3d), Mitogen-Activated Protein Kinase 1 (MAPK1; 4zzn), Aldehyde Dehydrogenase 1 (ALDH1; 5l2m), Tumor Protein p53 (TP53; 3zme), and Vitamin D Receptor (VDR; 3a2i).

ZINC-250K. ZINC-250K is a curated subset of the ZINC database (Sterling & Irwin, 2015) containing approximately 250,000 structurally diverse, drug-like molecules. It is widely used in molecule machine learning for tasks such as property prediction and de novo molecule generation, providing a tractable yet representative sample of practical chemical space.

C.6. Synformer Setup

For the synthesis-aware variant in Tab. 2, we apply SynFormer to all 100 valid DBMol molecules per target without any pre-filtering based on Boltz-2, AF3, Vina, or docking scores. For SynFormer expansion, we use fixed hyperparameters across all experiments: exhaustiveness=128, time-limit=300, search-width=32, and num-smiles-per-target=100.

We additionally report results under simple molecule-size consistency filters. The "+size filter (+-N)" rows in Tab. 2 retain only SynFormer outputs whose heavy-atom count differs from the input DBMol molecule by at most N atoms. This filter is based solely on molecular size, not on any evaluation metric.

C.7. Baseline Setup

We mainly follow (Jocys et al., 2024) setup for our baselines. Precisely,

- SynCoGen. SynCoGen (Rekesh et al., 2025) is an amortized molecular generator that conditions directly on pharmacophore profiles, specified as interaction types and their 3D positions. During training, pharmacophore features are provided as input to the model; to encourage robustness and generalization, at most seven pharmacophore features are randomly subsampled per example. At inference time, SynCoGen generates molecules conditioned solely on the provided pharmacophore profile, without requiring reference ligands or iterative optimization.
- ShEPhERD. ShEPhERD (Adams et al., 2024) is a 3D molecular generator conditioned on pharmacophore interaction profiles. While ShEPhERD explicitly models spatial interactions, it does not impose synthesizability constraints, and chemical feasibility is not guaranteed by design.
- SynFormer. SynFormer (Jocys et al., 2024) is a synthesis-aware 2D molecular generator that produces compounds by modeling chemically valid reaction pathways. It generates molecules by conditioning on a reference compound and optimizing structural similarity under 2D fingerprint-based metrics.
- CGFlow-ZS. CGFlow (Shen et al., 2025) is a pathway-based generative model that produces molecules together with 3D poses by sampling synthesis pathways. In its original formulation, CGFlow relies on reinforcement learning with pocket-conditioned rewards. To align with amortized sampling, we use a zero-shot variant (CGFlow-ZS) that samples using the pocket-conditioned reward proposed by (Shen et al., 2024), without task-specific fine-tuning.

C.8. Evaluation Details

C.8.1. AF3-BASED STRUCTURAL EVALUATION

We evaluate each generated protein–ligand complex with AlphaFold3 (AF3). For each target protein, we provide the corresponding multiple sequence alignment (MSA) and template structures, and cofold the protein with each generated ligand. For each ligand, we run AF3 once with a fixed random seed and generate one diffusion samples.

Confidence and interface quality. AF3 reports the inter-chain predicted TM score (ipTM), which measures confidence in the predicted protein–ligand interface. We use ipTM ≥ 0.6 as the high-confidence criterion. We also report iPAE, the predicted interface aligned error, where lower values indicate higher predicted interface quality.

Pocket engagement. To evaluate whether the ligand is placed in the intended binding pocket, we use the target pocket residues defined in Sec. C.5. For each pocket residue, we compute the minimum heavy-atom distance to the ligand. We report BCov, defined as the fraction of pocket residues within 3.5 Å of any ligand atom, and Dist, defined as the mean of these per-residue minimum distances. Higher BCov indicates broader pocket coverage, while lower Dist indicates more localized pocket binding.

AF3 Success. We define AF3 Success as a heuristic thresholded structural metric built from AF3 outputs. A predicted complex is counted as successful if it satisfies both ipTM ≥ 0.6 and at least one target pocket residue lies within 3.5 Å of the ligand. The threshold was selected without considering DBMol or its variants: we evaluated several AF3 metrics and thresholds on unconditional generation and other baselines, and chose this criterion because it provided a basic separation between unconditional generation and stronger methods. Since thresholded metrics can affect rankings, our conclusions do not rely on AF3 Success alone, but on the full set of AF3-based metrics, including BCov, iPAE, and Dist.

C.8.2. Other evaluation metrics

Boltz2-aligned metrics. We report Boltz2 Success as a proxy-aligned reference metric. A molecule is counted as Boltz2-successful if it achieves predicted affinity below 0 and pocket specificity above 0.5 under Boltz-2. Because DBMol is optimized using Boltz-based signals, Boltz2 Success is reported as a reference metric rather than used as the primary basis for comparison.

Molecular diversity. We measure molecular diversity using ECFP4 fingerprints. For each generated set, we compute the average pairwise Tanimoto distance between molecules, with higher values indicating greater chemical diversity.

Docking-based reference. We report AutoDock Vina scores as an additional held-out docking-based reference metric. For each generated molecule, the receptor and ligand are converted to PDBQT format, and docking is performed in a local search box centered on the Boltz-predicted ligand region with fixed padding. We use `exhaustiveness`=32 and `num_modes`=10, and report the best-scoring pose among the returned modes. Lower Vina scores indicate stronger predicted docking affinity. Vina is used only for evaluation and does not participate in DBMol optimization, candidate selection, or hyperparameter tuning.

Known-active similarity and interaction recovery. For LIT-PCBA targets with known active compounds, we additionally evaluate whether generated molecules recover active-like chemical and interaction patterns. We report similarity to the crystallized reference ligand, active enrichment EF@1%, and interaction fingerprint similarity (IFP Sim) to the reference ligand. These metrics help assess whether generated molecules copy known binders or instead discover alternative active-like candidates with related pocket-level interactions.

D. Additional Results

D.1. Ablations

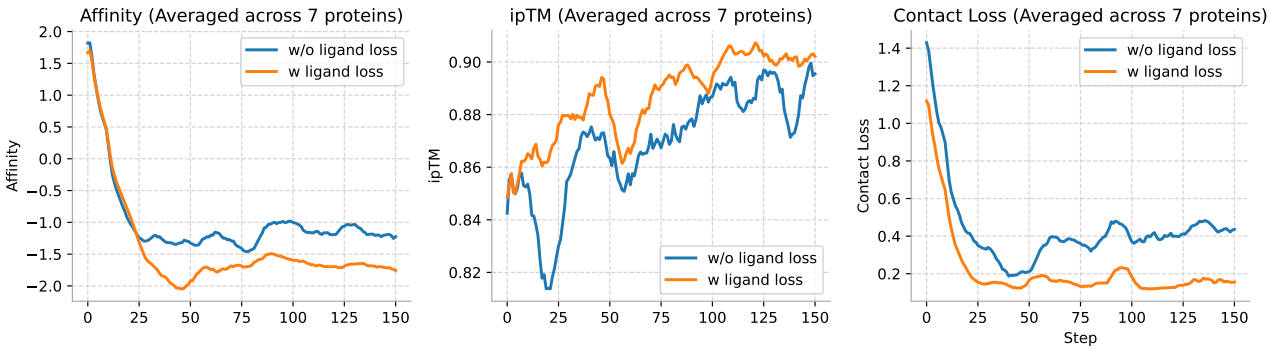

Figure 4. Ligand-centric anchoring loss helps to stabilize the optimization.

Anchoring loss. We show in Figure 4 the effectiveness of the anchoring loss. Intuitively, the anchoring loss simplifies the optimization by avoiding the need to minimize contact loss independently at all pocket positions, and instead encourages a global attraction between the ligand and the protein. This stabilizes the optimization process and leads to consistent improvements across all three objectives.

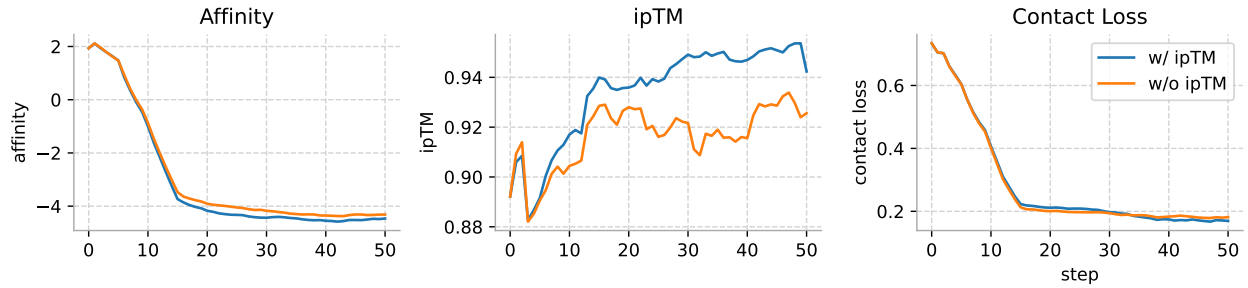

Figure 5. Effect of ipTM on Estrogen Receptor (ESR; 2p15)

Effect of ipTM loss. As shown in Figure 5, removing the ipTM loss does not noticeably affect either affinity or contact loss, while ipTM scores drop significantly across most optimization iterations. This shows the effectiveness of the optimization and also shows that improving the confidence does not sacrifice the other two objectives.

Table 4. **Ablation of affinity and contact objectives.** We compare the full DBMol objective with variants that remove either the contact term or the affinity term. Boltz2 Success is reported as a proxy-aligned reference metric, while AF3-based metrics provide held-out structural evaluation.

Method	BCov ↑	AF3 Succ. ↑	iPAE ↓	Dist ↓	Div. ↑	Vina ↓	Boltz2 Succ. ↑
DBMol	0.54	0.97	12.0	4.1	0.89	-7.6	0.47
DBMol w/o contact	0.54	0.94	12.2	3.9	0.89	-7.2	0.53
DBMol w/o affinity	0.47	0.90	12.8	4.7	0.90	-7.9	0.80

Effect of affinity and contact objectives. We ablate the two main Boltz-guided objective terms in Tab. 4 on 2v3d, 2p15, and 3zme. The full DBMol objective achieves the best overall AF3-based performance, with the highest AF3 Success and the lowest iPAE among the three variants, while maintaining competitive BCov and Dist. Removing the contact term slightly improves Dist and Boltz2 Success, and removing the affinity term further increases Boltz2 Success, but both ablations degrade the held-out AF3-based metrics, especially AF3 Success and iPAE. This indicates that the affinity and

contact terms are complementary: optimizing a single proxy term can improve some aligned scores, but the full objective provides a more balanced structural signal under held-out AF3 evaluation.

D.2. VINA correlation

We analyze the correlation between structure prediction model objectives and docking-based scores. We observe weak positive correlations between contact loss and Vina score (0.270), affinity and Vina score (0.102), and ipTM and Vina score (0.145). Notably, both contact loss and affinity are optimized to be minimized, yet they exhibit positive correlation with Vina, indicating that improvements under structure-predictor objectives do not translate directly to better Vina score.

Among these metrics, only ipTM shares a similar optimization direction with Vina, but the correlation remains weak. This suggests that structure prediction model–based signals and docking-based scores capture largely complementary aspects of protein–ligand interactions. This limited alignment helps explain why DBMol, while achieving strong structure-aligned performance, does not consistently yield optimal Vina scores.

D.3. Similarity to known actives

Table 5. **Similarity to known actives and interaction recovery on LIT-PCBA.** Sim measures similarity to the crystallized reference ligand, where lower values indicate less direct copying. EF@1% measures active enrichment in the nearest neighborhood of generated molecules. IFP Sim measures residue-level interaction fingerprint similarity to the reference ligand.

Method	Sim ↓	EF@1% ↑	IFP Sim ↑
SynFormer	0.447	1.06	0.656
ShEPhERD	0.137	0.80	0.401
SynCoGen	0.107	0.86	0.597
DiffSBDD	0.106	0.86	0.622
CGFlow	0.093	1.46	0.600
DeFoG	0.101	0.81	0.343
DBMol	0.094	1.33	0.392

We further analyze whether generated molecules simply reproduce known binders on LIT-PCBA. As shown in Tab. 5, DBMol has low similarity to the crystallized reference ligand, with Sim $= 0.094$. This is substantially lower than ligand-guided SynFormer and also lower than the unconditional DeFoG baseline, indicating that DBMol does not directly copy the reference binder. At the same time, DBMol substantially improves active enrichment over DeFoG, increasing EF@1% from 0.81 to 1.33, and achieves the second-best EF@1% among the compared methods. DBMol also modestly improves residue-level interaction recovery over DeFoG, increasing IFP Sim from 0.343 to 0.392. Overall, these results suggest that DBMol tends to generate alternative active-like molecules rather than reproducing known binders, while improving neighborhood-level active enrichment and modestly improving pocket-level interaction similarity.

Table 6. **Iterative refinement on LGR4.** DBMol-Iter initializes a second optimization round from the best first-round DBMol candidate and adds node/edge distribution regularization to keep the refined molecule close to the initialization. Avg. Rank is computed over BCov, AF3 Success, iPAE, Dist, diversity, and Vina, excluding Boltz2 Success; lower is better.

Method	BCov ↑	AF3 Succ. ↑	iPAE ↓	Dist ↓	Div. ↑	Vina ↓	Avg. Rank ↓	Boltz2 Succ. ↑
DiffSBDD	0.13	0.34	13.5	6.6	0.89	-9.4	2.50	0.27
DBMol	0.22	0.60	12.4	5.1	**0.91**	-7.6	2.00	0.21
DBMol-Iter	**0.28**	**0.73**	**11.4**	**3.6**	0.76	-8.9	**1.50**	**0.44**

D.4. Iterative refinement from a first-round candidate

We further include an exploratory ablation on LGR4 to test whether DBMol can be reused for iterative refinement. Starting from an initial DBMol batch, we select the best first-round candidate according to the Boltz-2-based selection criterion used by DBMol, without using AF3 or Vina evaluation metrics for selection. We then use this candidate to initialize a second round of optimization and projection. During this second round, we add a lightweight regularization term that penalizes deviations from the relaxed node and edge distributions of the selected first-round candidate, preventing the optimized molecule from drifting too far from the initialization. In practice, we regularize the edge distribution more strongly than

the node distribution, which empirically better preserves the molecular scaffold while still allowing local modifications.

As shown in Tab. 6, the second-round refinement improves the main structural metrics on LGR4. Compared with first-round DBMol, DBMol-Iter increases BCov from 0.22 to 0.28 and AF3 Success from 0.60 to 0.73, while reducing iPAE from 12.4 to 11.4 and Dist from 5.1 to 3.6. It also improves Boltz2 Success from 0.21 to 0.44, suggesting that reusing a promising DBMol candidate can further strengthen the structure-predictor-aligned signal. The main tradeoff is reduced molecular diversity, from 0.91 to 0.76, which is expected because the added regularization keeps the second-round search close to the selected first-round molecule. Overall, this result suggests that DBMol candidates can be further refined through iterative reuse, but we interpret this as an exploratory refinement strategy rather than a strict de novo baseline.

D.5. Runtime and inference-time compute.

To make the inference-time search effort explicit, we report the runtime of DBMol under the LIT-PCBA protocol in Tab. 7. All LIT-PCBA targets use the same fixed configuration with 50 Boltz-guided optimization steps. We report the cost of the DBMol generation pipeline before AF3 evaluation, since AF3 is used only for downstream evaluation and is not part of the molecular optimization objective. MSA search is performed once per protein and cached, and is therefore not included in the per-run runtime.

Table 7. **Runtime breakdown of DBMol under the LIT-PCBA protocol.** We report the estimated wall-clock time for the fixed 50-step optimization setting used in Tab. 1.

Component	Setting	Runtime
Boltz-guided optimization	50 steps	∼30 min
DeFoG projection / sampling	per optimized molecule set	∼10 s
DBMol generation total	50-step protocol	∼30 min
SynFormer projection	optional, 100 molecules	∼25 min

Under this protocol, DBMol uses a fixed number of Boltz calls for every target: one Boltz-guided update per optimization step, for 50 steps in total. The dominant cost is therefore the iterative Boltz-guided optimization, which takes approximately 30 minutes under our implementation. The subsequent DeFoG projection is comparatively negligible, taking about 10 seconds. When the optional DBMol-Syn variant is used, SynFormer projection adds additional runtime, reported separately in Tab. 7. Thus, the additional test-time computation used by DBMol is fixed, target-independent within LIT-PCBA, and explicitly reported.

D.6. Choice of discrete denoiser

DBMol requires a discrete molecular denoiser that can reliably project optimized relaxed representations back to chemically valid molecules, while remaining computationally efficient. DeFoG satisfies both requirements. As shown in Table 8, DeFoG achieves near-perfect validity and strong distributional alignment on the ZINC-250K benchmark, as measured by chemical validity, uniqueness, FCD, NSPDK, and scaffold diversity. We further evaluate DeFoG both with and without explicit modeling of formal atomic charges to isolate the effect of charge-aware denoising. Moreover, DeFoG adopts a flow-matching formulation, enabling high-quality molecular generation with a small number of denoising steps, which is particularly well suited for optimization-in-the-loop pipelines such as DBMol.

Table 8. Molecular generation results on the ZINC-250K dataset, comparing DeFoG with other diffusion and flow matching based baselines. Best results are highlighted in bold.

Model	Val. ↑	Uniqueness ↑	FCD ↓	NSPDK ↓	Scaffold ↑
GruM	98.65	–	2.257	0.0015	0.5299
GBD	97.87	–	2.248	0.0018	0.5042
CatFlow	99.21	**100.00**	13.211	–	–
DeFoG (w/o charges)	99.22	99.99	**1.425**	**0.0008**	0.5903
DeFoG (w/ charges)	**99.28**	**100.00**	1.850	0.0013	**0.9281**