REWARD DESIGN FOR POST-TRAINING MOLECULAR GENERATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion and flow-matching models can generate 3D molecules conditioned on protein pockets, but their training objectives do not directly optimize molecular properties relevant to practitioners. Post-training can shift the generative distribution toward molecules with better properties, yet its success depends on how rewards are constructed: molecular properties may be undefined for chemically invalid outputs, and optimizing one objective can degrade geometric plausibility or other desirable properties. We investigate these challenges by adapting DiffusionNFT, a recently proposed method for negative-aware finetuning, to jointly align the continuous coordinates and categorical outputs of a pocket-conditioned molecular generator. We first show that excluding chemically invalid generated samples from the alignment objective can cause validity collapse, whereas incorporating a validity reward largely prevents this failure. Building on this validity-aware formulation, we present the FlowR.NFT model family to systematically investigate combinations of chemical validity, Vina, QED, SA, and PoseBusters geometry rewards and their respective trade-offs. We identify reward compositions that improve Vina affinity, synthetic accessibility, and drug-likeness over the baseline model, with FlowR.NFT$_{\mathrm{multi}}$ achieving a redocked Vina score of $-10.14\,\mathrm{kcal/mol}$ on SPINDR while largely preserving molecular validity. These property gains transfer well to testing on CrossDocked2020, with a Vina docking score of $-9.10\,\mathrm{kcal/mol}$, better than those of all prior baselines in our benchmark. However, the plausibility of generated ligand poses is less well preserved across datasets. Together, our results highlight the importance of careful invalid-sample handling and balancing molecular properties during fine-tuning of ligand generators.

1 INTRODUCTION

The de novo design of small molecules that bind a protein target while satisfying desirable molecular properties is a central challenge in drug discovery. Structure-Based Drug Design (SBDD) addresses this task by leveraging the three-dimensional structure of a protein binding pocket to guide the generation of high-affinity ligands. The vast size and complexity of chemical space and the small fraction of possible molecules that satisfy the desired properties make this challenging. Recently, diffusion and flow-matching models have shown strong performance in generating molecular structures conditioned on three dimensional protein binding sites (Guan et al., 2023; Schneuing et al., 2024; 2025; Cremer et al., 2025a;b).

These models are typically trained with denoising or flow-matching objectives that encourage them to learn the training data distribution. However, they do not directly optimize downstream objectives relevant to practitioners, such as binding affinity, drug-likeness, or synthetic accessibility. Several approaches have been introduced to steer pretrained models toward better downstream properties, either through test-time guidance or fine-tuning. In particular, online reward alignment methods have demonstrated promising results by directly using gradients computed from generated samples to update the model. However, applying them to ligand generation introduces additional challenges: molecular generators can produce chemically invalid samples with undefined molecular rewards. Excluding these outputs from the alignment objective provides no direct feedback, and improvement in the average reward among valid molecules therefore does not imply improvement in the overall quality of generated samples. Furthermore, molecular rewards capture different aspects of molecular quality, and optimizing for one property can degrade others. In online reward alignment, each model

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.

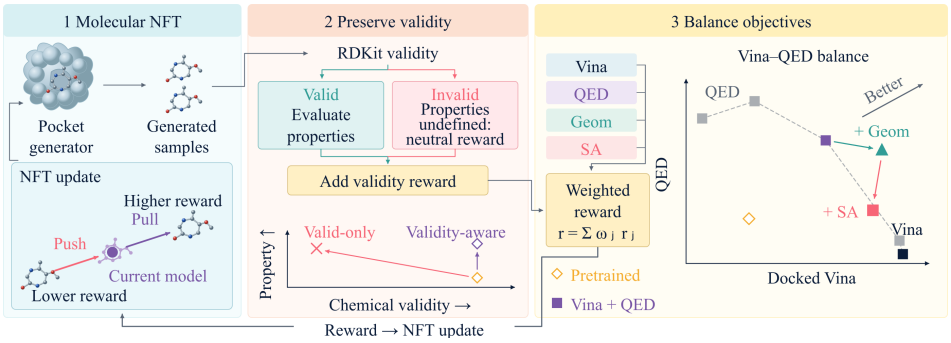

Figure 1: **Overview of FlowR.NFT.** (1) We adapt DiffusionNFT to jointly align continuous coordinates and categorical molecular features. (2) Excluding chemically invalid samples from fine-tuning can cause validity collapse. We retain these samples, assigning neutral rewards to undefined properties so that only chemical validity provides an alignment signal. (3) We vary the reward composition to balance molecular properties and geometry, identifying configurations that improve docking scores, QED, and SA relative to the pretrained generator.

update changes the distribution of samples used in subsequent training. Reward composition therefore shapes both the current optimization update and the samples on which later updates depend. This makes the treatment of invalid outputs and the balance between complementary molecular objectives central to the alignment process. This raises the central question of this work: *How should rewards be constructed to improve desired molecular properties while preserving chemical validity and balancing other aspects of molecular quality?*

To investigate this question, we adapt DiffusionNFT (Zheng et al., 2026) to fine-tune a pretrained pocket-conditioned molecular generator, Flowr.root (Cremer et al., 2025b). We study how invalid molecules should be handled during online reward alignment and develop a validity-aware reward formulation. We further systematically investigate different compositions of molecular-property and geometry rewards and show that a careful balance enables joint improvements in molecular properties while largely preserving molecular validity. Evaluation across SPINDR and CrossDocked2020 reveals state-of-the-art docking performance and, with the exception of generated poses, strong cross-dataset transfer of our results. **Our key contributions are:**

1. **Validity-aware molecular alignment.** We develop FlowR.NFT by adapting DiffusionNFT to the joint continuous and categorical representations in molecular generation. We demonstrate validity collapse when excluding chemically invalid samples and, guided by a decomposition of the alignment gradient, design a validity-aware reward that incorporates invalid samples and largely preserves chemical validity during fine-tuning.
2. **Characterizing molecular reward trade-offs.** We systematically compare combinations of Vina, QED, synthetic accessibility, and PoseBusters geometry rewards and identify configurations that jointly improve all objectives over the pretrained generator while largely preserving chemical validity and physical plausibility.
3. **Transfer across evaluation datasets.** We benchmark SPINDR-finetuned FlowR.NFT models against existing molecular generators on CrossDocked2020 without any further training or checkpoint selection. FlowR.NFT$_{\text{multi}}$ achieves a lower mean redocked Vina than all prior baselines in our benchmark, alongside strong SA and QED scores.

2 Related Work

For de-novo ligand generation, SBDD models generate molecules conditioned on the 3D-structure of a protein pocket. Pocket2Mol constructs ligands autoregressively (Peng et al., 2022), while TargetDiff (Guan et al., 2023) and DiffSBDD (Schneuing et al., 2024) use equivariant diffusion to generate the continuous 3D coordinates and categorical atom and bond features of ligands. More recently, flow matching-based approaches show strong performance on pocket-conditioned ligands

generation (Schneuing et al., 2025; Zhang & Zhang, 2025; Cremer et al., 2025a). Flowr.root (Cremer et al., 2025b) builds on top of FlowR with an extended training pipeline and an auxiliary affinity prediction objective. Due to its good performance on the high-quality SPINDR dataset, it represents a strong baseline for our investigation of online reward alignment for ligand generation.

Steering molecular generators. Among the methods to optimize certain properties of the outputs of generative models, *inference-time methods* modify the sampling process without updating the generator parameters. In this category, classifier guidance uses the gradient of a noise-aware classifier or property predictor to steer each denoising step (Dhariwal & Nichol, 2021), and is applied to molecular generation in Qiu et al. (2025). Classifier-free guidance avoids the need for a separate classifier and instead requires a model with both conditional and unconditional inputs to steer the sampling in the desired direction (Ho & Salimans, 2022). In contrast, *offline preference alignment methods* such as DiffusionDPO update the model parameters based on the signal from ranked sample pairs (Wallace et al., 2024). This approach has also been applied to molecular generation (Gu et al., 2024; Cheng et al., 2024; Zhang & Zhang, 2025; Lee et al., 2025), but it depends on a fixed preference dataset that is often difficult to curate and has limited coverage. *Online reward fine-tuning* avoids a fixed preference dataset by evaluating samples from the current policy. Policy-gradient methods formulate the denoising trajectory as a Markov decision process (Black et al., 2023) and use algorithms such as PPO or GRPO to optimize the sampling process (Schulman et al., 2017; Shao et al., 2024). FlowGRPO (Liu et al., 2026) extends this approach to flow models, while FTDiff applies GRPO-style fine-tuning to molecular diffusion models (Lin et al., 2026). These methods require storing and training on complete sampling trajectories rather than only the final samples. In contrast, diffusion negative aware fine-tuning (DiffusionNFT) (Zheng et al., 2026) requires only final samples and their rewards and optimizes a reward-weighted, flow-matching-style objective on the forward process. This makes DiffusionNFT a suitable candidate for molecular fine-tuning with rewards such as drug-likeness, synthetic accessibility, and Vina affinity.

Multi-objective molecular reward alignment. Prior work combines molecular objectives through both preference construction and composite online rewards. DecompDPO (Cheng et al., 2024) uses the sum of QED, SA, and rescaled Vina to score molecules and build preference pairs. Lee et al. (2025) combine different objectives, including affinity, drug-likeness, and strain energy, in Bayesian flow network preference optimization. For online fine-tuning, FTDiff (Lin et al., 2026) employs equally weighted affinity, QED, and SA components in the optimization reward. These studies establish multi-objective reward compositions for molecule generation. We build on this work by examining how reward components enter the molecular NFT update, including when a property is undefined, and by varying the allocation of reward to molecular properties and geometric plausibility.

3 METHOD

3.1 MOLECULAR NEGATIVE-AWARE FINE-TUNING

Diffusion Negative-aware Fine-Tuning (NFT) (Zheng et al., 2026) softly partitions generated samples into positive and negative components according to their rewards. These define an improvement direction toward higher-reward samples and away from lower-reward ones. NFT implements this update through implicit positive and negative prediction branches in a reward-weighted denoising objective. Unlike trajectory-based GRPO methods, it requires only generated endpoints and their rewards for optimization. We apply this construction to molecular coordinates and extend it to categorical atom, charge, and bond predictions. Here, a clean ligand is $z_1 = (\mathbf{x}_1, \mathbf{y}_{\text{atom}}, \mathbf{y}_{\text{charge}}, \mathbf{y}_{\text{bond}})$, where $\mathbf{x}_1$ contains its three-dimensional coordinates and $\mathbf{y}_c$ denotes the one-hot categorical targets. For each pocket P, the rollout policy generates ligands, each assigned a reward $r \in [0, 1]$. After forward corruption, the trainable and frozen rollout models predict clean endpoints from (z_t, t, P); we suppress these arguments below.

Continuous coordinates. Although Zheng et al. (2026) derive NFT in velocity space, the coordinate improvement direction follows directly by taking conditional expectations of its posterior-mixture identity (Appendix A). With trainable and rollout predictions $\hat{\mathbf{x}}_{1,\theta}$ and $\hat{\mathbf{x}}_{1,\text{old}}$, we use

$$\hat{\mathbf{x}}^{\pm}_{1,\theta} = \hat{\mathbf{x}}_{1,\text{old}} \pm \beta_c \left(\hat{\mathbf{x}}_{1,\theta} - \hat{\mathbf{x}}_{1,\text{old}}\right), \tag{1}$$

$$\ell_{\text{coord}}(r) = r \left\| \hat{\mathbf{x}}^{+}_{1,\theta} - \mathbf{x}_1 \right\|_2^2 + (1 - r) \left\| \hat{\mathbf{x}}^{-}_{1,\theta} - \mathbf{x}_1 \right\|_2^2 . \tag{2}$$

where $\beta_c > 0$ controls guidance strength along the update direction from lower to higher reward samples.

Categorical endpoints. For each channel $c \in \{\text{atom}, \text{charge}, \text{bond}\}$, let $\mathbf{a}_{\theta,c}$ and $\mathbf{a}_{\text{old},c}$ denote the trainable and rollout logits. We construct the implicit branches in logit space:

$$p_c^{\pm} = \text{softmax}(\mathbf{a}_{\text{old},c} \pm \beta_d \left(\mathbf{a}_{\theta,c} - \mathbf{a}_{\text{old},c}\right)), \tag{3}$$

ensuring that both branches define valid categorical distributions. We train them using

$$\ell_c(r) = r \ \text{CE}(\mathbf{y}_c, p_c^+) + (1 - r) \ \text{CE}(\mathbf{y}_c, p_c^-). \tag{4}$$

The complete training objective is

$$\mathcal{L}(\theta) = \mathbb{E}\left[\lambda_{\text{coord}} \ell_{\text{coord}}(r) + \lambda_{\text{disc}} \sum_c \ell_c(r)\right]. \tag{5}$$

The same molecular reward weights both modalities. After training on each rollout batch, we replace the rollout policy with the updated trainable policy. Algorithm 1 summarizes the pseudocode for the training loop.

3.2 Molecular reward design

The generator can produce chemically invalid ligands for which the molecular properties used as reward are undefined. Reward design must therefore specify both which samples enter the objective and how undefined properties contribute to their rewards. Excluding such samples is a straightforward choice, but removes all direct training feedback on those outputs. We instead retain all generated samples and combine explicit chemical-validity feedback with the available property rewards.

Validity-aware composite rewards. Let $r_{j,i} \in [0, 1]$ denote the normalized reward for property j of ligand i, as defined below. For an undefined property, we assign a neutral value in the normalized reward space:

$$\bar{r}_{j,i} = \begin{cases} r_{j,i}, & \text{if property } j \text{ is evaluable,} \\ 1/2, & \text{otherwise.} \end{cases} \tag{6}$$

We combine these components with a normalized chemical-validity reward $r_{v,i}$ derived from RDKit sanitization:

$$r_i = \omega_v r_{v,i} + \sum_{j \in \mathcal{P}} \omega_j \bar{r}_{j,i}, \qquad \omega_v + \sum_{j \in \mathcal{P}} \omega_j = 1, \quad \omega_j \geq 0, \ \omega_v > 0. \tag{7}$$

In our experiments, $\mathcal{P}$ contains Vina, QED, SA, and PoseBusters geometry objectives. The validity component provides explicit negative feedback.

Reward alignment and drift regularization. The NFT gradient explains the choice of $1/2$ for unavailable properties and distinguishes it from sample exclusion. Let $\mathbf{u}$ denote the difference between the trainable and frozen rollout models' coordinate predictions, and $\mathbf{d}$ the difference between the frozen rollout model's prediction and the generated endpoint: $\mathbf{u} = \hat{\mathbf{x}}_{1,\theta} - \hat{\mathbf{x}}_{1,\text{old}}$, and $\mathbf{d} = \hat{\mathbf{x}}_{1,\text{old}} - \mathbf{x}_1$. In terms of these differences, the gradient of the NFT loss in Eq. 2 can be written as

$$\nabla_{\hat{\mathbf{x}}_{1,\theta}} \ell_{\text{coord}}(r) = \underbrace{2\beta_c^2 \mathbf{u}}_{\text{drift regularization}} + \underbrace{2\beta_c(2r - 1)\mathbf{d}}_{\text{reward alignment}}. \tag{8}$$

The first term penalizes deviation from the frozen rollout prediction, while the second depends on the signed reward $2r - 1$. Because the NFT loss is affine in r, the composite reward in Eq. 7 produces a weighted sum of component losses and gradients. Its signed coefficient is

$$2r_i - 1 = \omega_v(2r_{v,i} - 1) + \sum_{j \in \mathcal{P}} \omega_j(2\bar{r}_{j,i} - 1).$$

Thus, completing an unavailable component with $1/2$ removes its contribution to reward alignment. For an invalid ligand whose property components are all unavailable, this reduces to $2r_i - 1 =$

$\omega_v(2r_{v,i}-1)$: only chemical validity supplies the signed alignment signal, which is negative whenever $r_{v,i} < 1/2$. Neutral completion does not remove the sample from training. Assigning a total reward of $1/2$ retains drift regularization, whereas excluding a sample removes both gradient terms. The categorical objective has the same neutrality property: at $r = 1/2$, its parameter-dependent part is a target-independent regularizer (Appendix A.1). These regularizers constrain deviation from the current rollout model, which is refreshed during training.

Rewarded properties. We use QED to assess drug-likeness (Bickerton et al., 2012), SA (Ertl & Schuffenhauer, 2009) to assess synthetic accessibility, and re-docked Vina score (Eberhardt et al., 2021) as a proxy for binding affinity. To address potential degradation in geometric plausibility during property optimization, we also define an intramolecular geometry reward based on PoseBusters criteria (Buttenschoen et al., 2024), similar to Kazeminia et al. (2026). For each molecule i, the geometry score is

$$s_{\text{geom},i} = \frac{1}{5}\left(f_{\text{length},i} + f_{\text{angle},i} + f_{\text{clash},i} + f_{\text{double},i} + f_{\text{ring},i}\right), \tag{9}$$

where the components denote the fractions of tested bond lengths, bond angles, nonbonded atom pairs, double-bond groups, and rings passing their respective PoseBusters checks. Here, $f_{\text{clash},i}$ measures the fraction of tested pairs free of internal steric clashes, while $f_{\text{double},i}$ and $f_{\text{ring},i}$ measure compliance with the corresponding flatness criteria.

Normalizing molecular property scores. Following Zheng et al. (2026), we normalize each score within the pocket's rollout group:

$$r_{p,i} = \frac{1}{2} + \frac{1}{2}\,\text{clip}\left(\frac{s_i - \mu_P}{\sigma_P}, -1, 1\right) \tag{10}$$

where μ_P and σ_P are the mean and standard deviation of defined property scores among RDKit-valid molecules in the rollout group. For chemical validity reward $r_{v,i}$, we normalize binary RDKit sanitization scores (valid: 1, invalid: 0) over all samples in the group. The total reward equals $r_{p,i}$ in the property-only setup or combines components according to Eq. 7. We negate Vina scores before normalization so that higher values consistently indicate better outcomes.

4 Results

4.1 Experimental setup

Training We initialize all models from the released Flowr.root v2.1 checkpoint and fine-tune on the SPINDR training split (Cremer et al., 2025a). We train separate models, rewarding Vina (Eberhardt et al., 2021; Yu et al., 2023) as a proxy for binding affinity, drug-likeness (QED) (Bickerton et al., 2012), and synthetic accessibility (SA) (Ertl & Schuffenhauer, 2009). We further investigate weighted combinations of these rewards and a PoseBusters geometry reward (see section 3.2). Each NFT rollout contains $K = 8$ ligands for each of 128 pockets for a total of 1,024 generated ligands per online iteration. We train with two optimizer steps per rollout for three epochs, which corresponds to 828 rollouts and 1,656 optimizer updates. We compute validation scores every 40 steps and select the best checkpoint based on a weighted average of normalised validation scores, using the same weights as for the respective reward composition. For main results, we train with three independent seeds per configuration.

Evaluation We sample 100 molecules per pocket without any filtering, e.g. for validity or uniqueness. The main experiments use the SPINDR (Cremer et al., 2025a) test set, while comparisons to prior methods use the standard 100 CrossDocked2020 test pockets (Francoeur et al., 2020). We report chemical validity (RDKit-sanitizable), PoseBusters validity (Buttenschoen et al., 2024), score-only and docked Vina energies, and molecule diversity based on Tanimoto similarity of fingerprints. RDKit validity is computed over all generations; all other metrics are computed over the subset of RDKit-valid molecules. Metric definitions and baselines are listed in Appendix B.

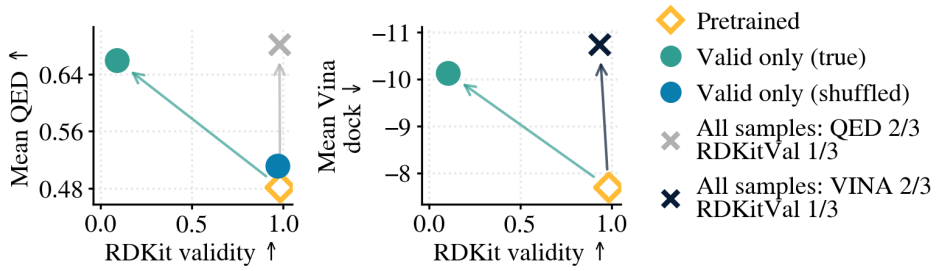

Figure 2: **Effect of invalid molecule handling in reward design on alignment and RDKit validity:** We report RDKit validity and mean target property among valid molecules generated for SPINDR test set pockets, arrows indicating the change from the pretrained model.

4.2 Validity-aware rewards mitigate chemical-validity collapse

We evaluate whether the validity-aware reward in Eq. 7 preserves chemical validity while improving the targeted property. We compare it with valid-only alignment, which excludes RDKit-invalid samples from the NFT loss and assigns valid samples only the normalized property reward. For both formulations, we measure chemical validity across all generated molecules and the targeted property among valid outputs.

First, we consider the simple *valid-only* setup. Invalid samples are excluded from the NFT losses in Eq. 2 and Eq. 4, while valid samples receive the property reward from Eq. 10. As shown in Fig. 2, this improves each target property among valid molecules, but drastically reduces RDKit validity from approximately 0.98 to 0.09 and 0.1 for QED and Vina, respectively. Because invalid samples do not contribute to the loss, there is no direct signal to preserve validity. We hypothesize that this creates a reward-hacking loophole: to increase the average reward over valid molecules, probability mass may move from lower-reward valid molecules to invalid outputs rather than toward higher-reward valid molecules. Some validity loss could also result from training on self-generated samples (Alemohammad et al., 2026). To test this, we randomly shuffle rewards among the valid molecules generated for each pocket (Fig 2). Shuffled-reward training almost maintains chemical validity and only slightly improves the rewarded property. This indicates that molecule–reward association drives most of the observed collapse. In contrast, the *validity-aware* reward in Eq. 7 improves the targeted properties while largely preserving chemical validity. Together, these results show that the validity-aware formulation improves the target property without the severe validity collapse observed when invalid samples are excluded. We therefore use this formulation in the remaining experiments to investigate combinations of molecular-property rewards.

4.3 Balancing molecular properties through reward design

The validity-aware formulation enables targeted single-property improvements while largely preserving chemical validity. However, optimizing one property can degrade others. For example, alignment using only Vina and chemical-validity rewards ($w_{\text{Vina}} = 2/3$, $w_v = 1/3$) improves docked Vina scores but lowers QED, PoseBusters validity, and SA relative to the pretrained model (Fig. 3, upper). We therefore investigate how to combine these objectives to improve docking scores while maintaining or improving the other molecular properties.

Throughout these experiments, we fix the chemical-validity reward weight at $w_v = 1/3$ and allocate the remaining weight across Vina, QED, Geom, and SA. We first vary the relative weights of QED and Vina and find that selected reward configurations improve both properties relative to the pretrained model (Fig. 3, lower left). We next vary the relative weights of Vina and geometry reward based on Eq. 9. Increasing the geometry reward weight generally improves the geometry pass rate, but does not improve on the pretrained model, whose pass rate is already close to one (Fig. 3, lower right). For both $w_{\text{Geom}} = 2/3$ and $(w_{\text{Vina}}, w_{\text{Geom}}) = (1/6, 1/2)$, the validation-score based checkpoint selection (see Section 4.1) chooses the checkpoint at step 0, corresponding to the pretrained model. To combine improvements in docking scores and drug-likeness with explicit geometry feedback, we start from the configuration with $w_{\text{Vina}} = w_{\text{QED}} = 1/3$. Reallocating half of the QED weight to

Configuration	Reward weights				
	RDKit	Vina	QED	Geom	SA
Vina	1/3	2/3	0	0	0
Vina + QED	1/3	1/3	1/3	0	0
Vina + QED + Geom	1/3	1/3	1/6	1/6	0
FlowR.NFT$_{\text{multi}}$ Vina + QED + Geom + SA	1/3	1/3	1/6	1/12	1/12

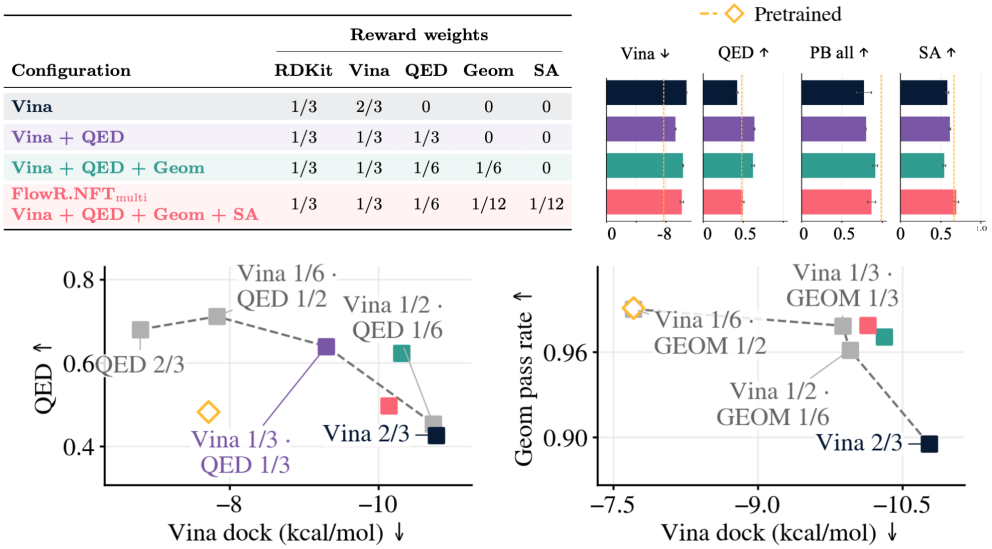

Figure 3: **Balancing molecular rewards on SPINDR.** *Upper*: Molecular properties for selected reward configurations compared with the pretrained model. *Lower left*: QED and docked Vina scores for different allocations of QED and Vina reward weights. *Lower right*: Geometry pass rates and docked Vina scores for different allocations of geometry and Vina reward weights. The chemical-validity reward weight is fixed at $w_v = 1/3$ throughout. Results are averaged across pockets and training seeds, with pockets weighted equally; error bars show variability across seeds. For $w_{\text{Geom}} = 2/3$ and $(w_{\text{Vina}}, w_{\text{Geom}}) = (1/6, 1/2)$, validation selects the checkpoint at step 0.

Geom gives $(w_{\text{Vina}}, w_{\text{QED}}, w_{\text{Geom}}) = (1/3, 1/6, 1/6)$. Compared to the Vina+QED configuration, this improves both PoseBusters validity and docked Vina scores, with only a small reduction in QED. The improvement in docking scores is consistent with the reduced emphasis on QED observed in the Vina–QED weight sweep. Finally, we reallocate half of the geometry weight to SA, yielding our final FlowR.NFT$_{\text{multi}}$ model with $(w_{\text{Vina}}, w_{\text{QED}}, w_{\text{Geom}}, w_{\text{SA}}) = (1/3, 1/6, 1/12, 1/12)$. Compared with the Vina+QED+Geom configuration, this substantially improves SA while reducing QED and overall PoseBusters validity. Relative to the pretrained baseline, the resulting model achieves higher SA and slightly higher QED, but lower PoseBusters validity. These results show that reward composition controls the balance between docking scores, molecular properties, and geometry. We carry forward the Vina, Vina+QED, Vina+QED+Geom, and Vina+QED+Geom+SA configurations, which place different emphases on these objectives, to assess how their improvements transfer to CrossDocked2020. Additional Vina metrics are reported in Appendix Table 2.

4.4 Property gains transfer to CrossDocked2020

We evaluate the selected Flowr.NFT configurations (Fig. 3, top panel) on CrossDocked2020 and compare them with the pretrained Flowr.root model, unguided molecular generators, inference-time guidance methods (MolJo Qiu et al. (2025)), and offline reward-alignment methods[1] (DrugFlow DPO Schneuing et al. (2025) and AliDiff Gu et al. (2024)) in Tab. 1. The FlowR.NFT configurations selected above progressively incorporate additional molecular objectives, allowing us to assess which improvements transfer to CrossDocked2020. The other aligned methods in the benchmark optimize different combinations of molecular properties, and we compare the resulting property profiles. We point out that Flowr.NFT is fine-tuned exclusively on SPINDR, whereas DrugFlow-DPO, MolJO, and AliDiff use generators trained on CrossDocked2020. This difference in training data limits how directly performance differences can be attributed to the alignment method.

[1]We do not include FTDiff Lin et al. (2026), an online reward-alignment method, because there are no publicly available model checkpoints or generated samples, and the paper does not report PoseBusters results,

Table 1: **Comparison with pocket-conditional molecular generators on CrossDocked2020** grouped into unguided generation, inference-time guidance, offline preference alignment, and our online reward alignment. Flowr.NFT is trained on SPINDR and evaluated on CrossDocked2020 without further fine-tuning or checkpoint selection. Other preference-aligned methods optimize different objectives: MolJO uses SA and affinity, DrugFlow-DPO aligns QED, SA, Vina, and REOS, and AliDiff aligns Vina. For all prior methods, we recompute metrics from the authors' provided samples, except for Flowr.root and DrugFlow-DPO where we generate samples ourselves from published checkpoints. RDKit validity is not reported for Pocket2Mol because its sampling procedure generates molecules until 100 valid outputs are obtained, limiting comparability. Values are mean ± standard deviation across checkpoints trained with different seeds, with best value shown in bold.

	RDKit-Val↑	PB all↑	PB intra↑	PB inter↑	Vina Score↓	Vina Dock↓	SA↑	QED↑	Diversity↑
Pocket2Mol	NA	0.85	0.92	0.92	−4.24	−6.11	0.81	0.59	**0.84**
TargetDiff	0.88	0.51	0.57	0.88	−6.13	−7.66	0.58	0.48	0.72
DiffSBDD	**1.00**	0.38	0.48	0.70	−2.31	−7.24	0.58	0.47	0.73
DrugFlow	0.85	0.74	0.82	0.90	−5.02	−6.85	0.71	0.51	0.73
FlowR.root	0.98	**0.91**	**0.97**	0.93	−5.96	−7.34	0.71	0.50	0.60
MolJO	0.97	0.88	0.89	**0.98**	**−8.25**	−9.07	0.77	0.55	0.75
DrugFlow DPO	0.70	0.71	0.75	0.94	−5.77	−7.51	**0.85**	**0.66**	0.79
AliDiff	0.92	0.26	0.27	0.94	−7.62	−8.41	0.57	0.50	0.74
VINA	$0.84_{\pm 0.11}$	$0.60_{\pm 0.11}$	$0.76_{\pm 0.11}$	$0.76_{\pm 0.04}$	$-4.69_{\pm 0.95}$	$\mathbf{-9.50}_{\pm 0.09}$	$0.62_{\pm 0.03}$	$0.44_{\pm 0.01}$	$0.58_{\pm 0.00}$
VINA+QED	$0.96_{\pm 0.01}$	$0.53_{\pm 0.01}$	$0.78_{\pm 0.03}$	$0.66_{\pm 0.03}$	$-2.68_{\pm 0.50}$	$-8.58_{\pm 0.05}$	$0.65_{\pm 0.01}$	$0.63_{\pm 0.01}$	$0.67_{\pm 0.00}$
VINA+QED+Geom	$0.99_{\pm 0.01}$	$0.67_{\pm 0.04}$	$0.92_{\pm 0.02}$	$0.72_{\pm 0.06}$	$-3.50_{\pm 1.55}$	$-9.15_{\pm 0.03}$	$0.61_{\pm 0.01}$	$0.60_{\pm 0.01}$	$0.63_{\pm 0.01}$
VINA+QED+Geom+SA	$0.98_{\pm 0.01}$	$0.56_{\pm 0.05}$	$0.95_{\pm 0.00}$	$0.58_{\pm 0.05}$	$-1.14_{\pm 1.09}$	$-9.10_{\pm 0.09}$	$0.74_{\pm 0.03}$	$0.52_{\pm 0.01}$	$0.65_{\pm 0.01}$
Test set	1.00	0.94	0.96	0.98	−6.54	−7.19	0.73	0.48	0.00

Molecular-property improvements. The Vina-only configuration achieves the lowest mean docked Vina score among the evaluated methods, improving from -7.34 to -9.50 kcal/mol relative to the pretrained model, compared with -9.07 kcal/mol for MolJO. However, this configuration also has lower chemical validity, QED, and SA than the pretrained model. The effects of reward composition follow the trends observed on SPINDR. Allocating reward weight to QED increases QED but weakens the improvement in docking scores. Reallocating part of that weight to Geom improves both intramolecular validity and docking scores: Vina+QED+Geom achieves a docked Vina score of -9.15 kcal/mol and intramolecular PB validity of 0.92, compared with -8.58 kcal/mol and 0.78 for Vina+QED. Replacing half of the geometry weight with SA increases SA from 0.61 to 0.74, with a small deterioration in docked Vina and lower QED. The resulting Vina+QED+Geom+SA configuration improves docked Vina, SA, and QED relative to the pretrained model while maintaining chemical validity at 0.98. Its docked Vina score is similar to MolJO's, although its SA and QED are slightly lower. DrugFlow-DPO achieves higher SA and QED than either model, but has less favorable docked Vina scores. These observed trade-offs between different models motivate evaluating how many molecules satisfy additional quality requirements and whether docking scores remain favorable after filtering.

Property tradeoffs across models We filter generated molecules by intramolecular PoseBusters validity, SA> 0.59, and QED> 0.25 (SA and QED thresholds following the convention in Guan et al. (2024)), both individually and jointly (Fig. 4). For each filter, we report the fraction of molecules retained, their mean docked Vina score, and the change in that score relative to the unfiltered samples on matched pockets. Filtering based on PoseBusters' intra-molecular validity criteria assesses the generated molecule's internal geometry separately from its placement in the pocket. Intramolecular PB filtering retains 95.3% of molecules generated with FlowR.NFT$_{\text{multi}}$ with essentially unchanged mean docked Vina scores. Pass rates are lower for MolJO (89.0%), DrugFlow-DPO (74.4%), and AliDiff (26.4%), with progressively larger deteriorations in docking scores compared to the unfiltered molecule sets. SA filtering instead favours DrugFlow-DPO and MolJO, with pass rates of 97.2% and 92.0%, respectively. All models except for AliDiff show little to no effect of SA-based filtering on average docking scores. QED filtering is comparatively permissive, retaining at least 92.9% of molecules across all aligned models and changing mean Vina docking scores by at most 0.05 kcal/mol. When jointly applying all three filters, MolJO retains slightly more molecules than FlowR.NFT$_{\text{multi}}$, but FlowR.NFT$_{\text{multi}}$ retains a slightly more favorable mean docked Vina score (-8.95 versus -8.88 kcal/mol).

which prevents a comparison in terms of all molecular qualities considered here - geometric plausibility, Vina, QED, and SA.

Generated-pose plausibility remains a limitation. The observed improvements in molecular properties do not extend consistently to the initially generated poses. For FlowR.NFT$_{\text{multi}}$, intramolecular PB validity remains close to the pretrained baseline (0.95 versus 0.97), but intermolecular validity decreases from 0.93 to 0.58 (Tab. 1). This deterioration is consistent with score-only Vina worsening from -5.96 to -1.14 kcal/mol, despite the improvement in docked Vina. PB validity of redocked pose is good again. The models were trained with docked Vina as reward which give no signal for good pocket conditioning. This limitation is more pronounced on CrossDocked2020 than on SPINDR, consistent with the pretrained model's lower intermolecular PB validity on CrossDocked2020 (0.93 versus 1.00). Alignment further reduces intermolecular validity on CrossDocked2020, widening this existing gap in generated-pose plausibility.

These results show that molecular-property gains achieved on SPINDR transfer to CrossDocked2020. Among the evaluated aligned methods, FlowR.NFT$_{\text{multi}}$ shows the smallest deterioration in mean docked Vina score after jointly filtering for intramolecular PoseBusters validity, QED, and SA. Its docking gains are therefore largely retained among molecules satisfying these additional quality requirements. However, preserving plausible placement of generated ligands within the pocket remains a limitation.

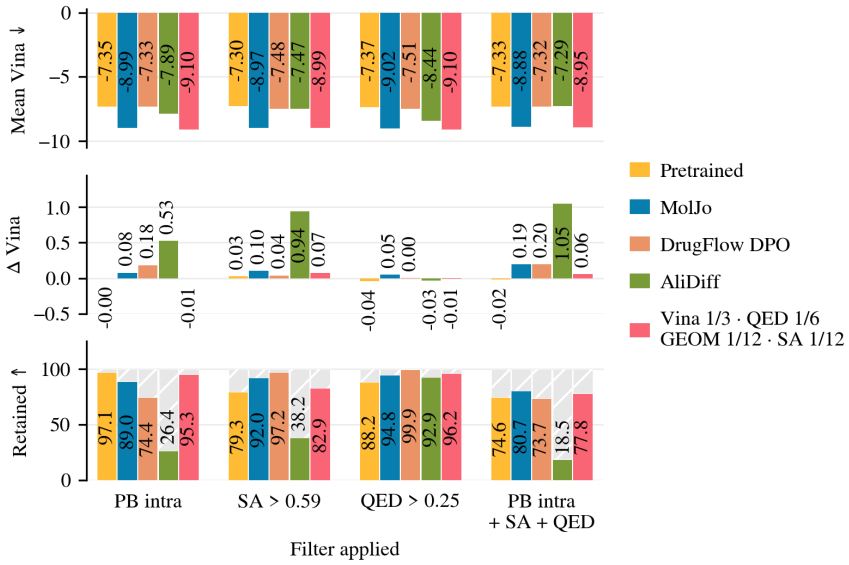

Figure 4: **Molecular quality after filtering on CrossDocked2020.** We compare the pretrained Flowr.root model, Flowr.NFT (Vina+QED+Geom+SA), MolJO, DrugFlow-DPO, and AliDiff. Filters require intramolecular PoseBusters validity, SA>0.59, QED>0.25, or all three. *Upper*: Mean docked Vina score among retained molecules. *Middle*: Change from the unfiltered mean on matched pockets; positive values indicate worse docking scores. *Lower*: Percentage of molecules retained.

5 Conclusion

In this study we adapt DiffusionNFT to fine-tune Flowr.root for molecular properties and study how chemically invalid samples should enter the fine-tuning objective. We demonstrate that, with a well-balanced multi-objective reward, our FlowR.NFT model achieves a highly competitive Vina docking affinity along with strong QED, SA, and PoseBusters validity scores. With the exception of the ligand positioning within pockets, these results generalise well to CrossDocked2020 and compare favorably against other tested models when filtering generated molecules for intramolecular PoseBusters validity, QED, and SA. **Limitations:** In contrast to the other reward-aligned models included in our comparison, the FlowR.NFT models have not been trained on CrossDocked2020, hence we cannot directly separate dataset and methodological effects in our results. However, our benchmark is conservative and might hide additional methodological advantages behind a dataset disadvantage.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the prose of this manuscript, for literature research, and for aiding code development. We have manually reviewed all LLM-generated code and text and verified results of any LLM-assisted literature review. We have not used generative AI tools for drafting full paper sections, and generating synthetic datasets as well as proving mathematical claims are not applicable to this work.

REPRODUCIBILITY STATEMENT

Code, checkpoints and samples generated from our models will be made publicly available upon publication.

REFERENCES

Sina Alemohammad, Zhangyang Wang, and Richard Baraniuk. Neon: Negative extrapolation from self-training improves image generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kpLRYtPGt3.

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. doi: 10.1038/nchem.1243.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Martin Buttenschoen, Garrett M Morris, and Charlotte M Deane. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical science*, 15(9):3130–3139, 2024.

Xiwei Cheng, Xiangxin Zhou, Yuwei Yang, Yu Bao, and Quanquan Gu. Decomposed direct preference optimization for structure-based drug design. *arXiv preprint arXiv:2407.13981*, 2024. URL https://arxiv.org/abs/2407.13981.

Julian Cremer, Ross Irwin, Alessandro Tibo, Jon Paul Janet, Simon Olsson, and Djork-Arné Clevert. Flowr: Flow matching for structure-aware de novo, interaction-and fragment-based ligand generation. *arXiv preprint arXiv:2504.10564*, 2025a.

Julian Cremer, Tuan Le, Mohammad M Ghahremanpour, Emilia Sługocka, Filipe Menezes, and Djork-Arné Clevert. Flowr. root: A flow matching based foundation model for joint multi-purpose structure-aware 3d ligand generation and affinity prediction. *arXiv preprint arXiv:2510.02578*, 2025b.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

Vineeth Dorna, D. Subhalingam, Keshav Kolluru, Shreshth Tuli, Mrityunjay Singh, Saurabh Singal, N. M. Anoop Krishnan, and Sayan Ranu. TAGMol: Target-aware gradient-guided molecule generation. *arXiv preprint arXiv:2406.01650*, 2024.

Jerome Eberhardt, Diogo Santos-Martins, Andreas F. Tillack, and Stefano Forli. AutoDock Vina 1.2.0: New docking methods, expanded force field, and python bindings. *Journal of Chemical Information and Modeling*, 61(8):3891–3898, 2021. doi: 10.1021/acs.jcim.1c00203.

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):8, 2009. doi: 10.1186/1758-2946-1-8.

Paul G. Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B. Iovanisci, Ian Snyder, and David R. Koes. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *Journal of Chemical Information and Modeling*, 60(9):4200–4215, 2020. doi: 10.1021/acs.jcim.0c00411.

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.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Sanaz Kazeminia, Lewis R. Vidler, Pushkar G. Ghanekar, Nele P Quast, and Garrett M Morris. PRISM: A hybrid diffusion-reinforcement learning framework for 3d structure-based de novo design. In *ICLR 2026 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2026. URL https://openreview.net/forum?id=TAeVNZ77sH.

Seungbeom Lee, Munsun Jo, Jungseul Ok, and Dongwoo Kim. Enhancing ligand validity and affinity in structure-based drug design with multi-reward optimization. In *ICML*, 2025.

Guang Lin, Shikui Tu, and Lei Xu. Fine-tuning diffusion models for molecular generation via reinforcement learning and fast sampling. *arXiv preprint arXiv:2606.01220*, 2026. URL https://arxiv.org/abs/2606.01220.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di ZHANG, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. *Advances in neural information processing systems*, 38:40783–40818, 2026.

Xingang Peng, Shitong Luo, Jiaqi Guan, Qi Xie, Jian Peng, and Jianzhu Ma. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17644–17655. PMLR, 2022.

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 *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Arne Schneuing, Charles Harris, Yuanqi Du, Kieran Didi, Arian Jamasb, Ilia Igashov, Weitao Du, Carla Gomes, Tom L Blundell, Pietro Lio, et al. Structure-based drug design with equivariant diffusion models. *Nature Computational Science*, 4(12):899–909, 2024.

Arne Schneuing, Ilia Igashov, Adrian W Dobbelstein, Thomas Castiglione, Michael Bronstein, and Bruno Correia. Multi-domain distribution learning for de novo drug design. *arXiv preprint arXiv:2508.17815*, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238. IEEE, 2024.

Kunyu Wang, Jon Paul Janet, and Alessandro Tibo. Controllable molecular generation with fine-tuned flow-matching model. *ChemRxiv*, 2025(1125), 2025. doi: 10.26434/chemrxiv-2025-db0m9. URL https://chemrxiv.org/doi/abs/10.26434/chemrxiv-2025-db0m9.

Yuejiang Yu, Chun Cai, Jiayue Wang, Zonghua Bo, Zhengdan Zhu, and Hang Zheng. Uni-dock: Gpu-accelerated docking enables ultralarge virtual screening. *Journal of Chemical Theory and Computation*, 19(11):3336–3345, 2023.

Daiheng Zhang and Zhao Zhang. MolFORM: Preference-aligned multimodal flow matching for structure-based drug design. *arXiv preprint arXiv:2507.05503*, 2025. URL https://arxiv.org/abs/2507.05503.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, volume 2026, pp. 134129–134150, 2026.

A DIFFUSIONNFT FOR COORDINATE AND CATEGORICAL ENDPOINTS

Here, we give more details on the DiffusionNFT loss for coordinate and categorical endpoints. In the following, we represent a clean ligand by $z_1 = (\mathbf{x}_1, \mathbf{y}_{\text{atom}}, \mathbf{y}_{\text{charge}}, \mathbf{y}_{\text{bond}})$, and z_t is its noised state. This corresponds to x_0 and x_t, respectively, in Zheng et al. (2026).

Positive and negative endpoint distributions. At each online update, the rollout policy π_{old} generates K ligands for each pocket P. The property score of each ligand is converted into an optimality probability $r(z_1, P) \in [0, 1]$. Following Lemma A.1 of Zheng et al. (2026), define

$$\rho(P) = \mathbb{E}_{\pi_{\text{old}}(z_1|P)}[r(z_1, P)]. \tag{11}$$

The reward induces the endpoint distributions

$$\pi^+(z_1 \mid P) = \frac{r(z_1, P)}{\rho(P)}\pi_{\text{old}}(z_1 \mid P), \qquad \pi^-(z_1 \mid P) = \frac{1 - r(z_1, P)}{1 - \rho(P)}\pi_{\text{old}}(z_1 \mid P). \tag{12}$$

Consequently,

$$\pi_{\text{old}}(z_1 \mid P) = \rho(P)\pi^+(z_1 \mid P) + [1 - \rho(P)]\pi^-(z_1 \mid P). \tag{13}$$

Thus, high-reward ligands receive more probability under π^+, while low-reward ligands receive more probability under π^-. DiffusionNFT uses both distributions to define an improvement direction rather than fitting only the positive samples.

Coordinate-space improvement direction. Applying the same forward noising process to π^+, π^-, and π_{old}, Lemma A.2 of Zheng et al. (2026) gives

$$\pi_{\text{old}}(z_1 \mid z_t, P) = \alpha(z_t, P)\pi^+(z_1 \mid z_t, P) + [1 - \alpha(z_t, P)]\pi^-(z_1 \mid z_t, P), \tag{14}$$

where

$$\alpha(z_t, P) = \rho(P)\frac{\pi_t^+(z_t \mid P)}{\pi_t^{\text{old}}(z_t \mid P)}. \tag{15}$$

For $a \in \{+, -, \text{old}\}$, let the optimal clean-coordinate predictor be the corresponding posterior mean,

$$\hat{\mathbf{x}}_1^a(z_t, t, P) = \mathbb{E}_{\pi^a(\mathbf{x}_1|z_t,P)}[\mathbf{x}_1]. \tag{16}$$

Taking the coordinate expectation of Eq. 14 gives

$$\hat{\mathbf{x}}_1^{\text{old}} = \alpha\hat{\mathbf{x}}_1^+ + (1 - \alpha)\hat{\mathbf{x}}_1^-. \tag{17}$$

Rearranging yields

$$\underbrace{(1 - \alpha)\big(\hat{\mathbf{x}}_1^{\text{old}} - \hat{\mathbf{x}}_1^-\big)}_{\boldsymbol{\Delta}_x} = \alpha\big(\hat{\mathbf{x}}_1^+ - \hat{\mathbf{x}}_1^{\text{old}}\big). \tag{18}$$

Hence, the negative, old, and positive coordinate predictors are collinear. The direction from the negative predictor to the old predictor is the same as the direction from the old predictor to the positive predictor. This is the clean-coordinate analogue of the velocity-space improvement direction in Theorem A.3 of Zheng et al. (2026).

Coordinate-space NFT objective. We want to fine-tune a molecular generator that predicts clean coordinates. We therefore define the implicit positive and negative coordinate predictors as

$$\hat{\mathbf{x}}_{1,\theta}^+ = (1 - \beta_c)\hat{\mathbf{x}}_{1,\text{old}} + \beta_c\hat{\mathbf{x}}_{1,\theta}, \tag{19}$$

$$\hat{\mathbf{x}}_{1,\theta}^- = (1 + \beta_c)\hat{\mathbf{x}}_{1,\text{old}} - \beta_c\hat{\mathbf{x}}_{1,\theta}. \tag{20}$$

The coordinate objective is

$$\mathcal{L}_{\text{coord}} = \mathbb{E}_{P, z_1 \sim \pi_{\text{old}}, t}\left[r\left\|\hat{\mathbf{x}}_{1,\theta}^+ - \mathbf{x}_1\right\|_2^2 + (1 - r)\left\|\hat{\mathbf{x}}_{1,\theta}^- - \mathbf{x}_1\right\|_2^2\right], \tag{21}$$

where $r = r(z_1, P)$. The proof of Theorem A.4 in Zheng et al. (2026) relies on the posterior-mean relation and squared-error objective, and therefore applies directly to the clean-coordinate predictor. Under unlimited data and model capacity, its optimum satisfies

$$\hat{\mathbf{x}}_{1,\theta^*} = \hat{\mathbf{x}}_{1,\text{old}} + \frac{2}{\beta_c}\boldsymbol{\Delta}_x. \tag{22}$$

Thus, the NFT objective moves the coordinate predictor away from the rollout predictor along the positive-over-negative improvement direction.

Algorithm 1 Training loop.

Require: Reference policy π_{ref}, pockets $\{P\}$, group size K, score function s, reward rule

$\pi_{\text{old}} \leftarrow \pi_{\text{ref}}$; $\pi_\theta \leftarrow \pi_{\text{old}}$
for each online update **do**
  $\mathcal{D}_{rollout} \leftarrow \varnothing$
  for each pocket P **do**
    Generate K ligands $\{z_{1,i}\}_{i=1}^{K} \sim \pi_{\text{old}}(\cdot \mid P)$
    Compute scores s_i and normalized rewards r_i using the chosen reward rule
    $\mathcal{D}_{rollout} \leftarrow \mathcal{D}_{rollout} \cup \{(P, z_{1,i}, r_i)\}$
  end for
  for each minibatch $\{(P, z_{1,i}, r_i)\}$ in $\mathcal{D}_{rollout}$ **do**
    Sample noise times and corrupt each $z_{1,i}$ to obtain $z_{t,i}$
    Evaluate current and frozen rollout predictors on each $(z_{t,i}, t, P)$
    Form implicit predictions $\hat{\mathbf{x}}^{\pm}_{1,\theta}$ and $p_c^{\pm}$ using Eqs. 2–4
    Compute losses and update current policy π_θ
  end for
  Update old policy: $\pi_{\text{old}} \leftarrow \pi_\theta$ ▷ full replacement
end for

Categorical endpoints. Our molecular sampler also predicts categorical atom, formal-charge, and bond endpoints. For each channel $c \in \{\text{atom}, \text{charge}, \text{bond}\}$, let $l_{\theta,c}$ and $l_{\text{old},c}$ denote the current- and rollout-policy logits. Directly extrapolating probabilities can leave the probability simplex, so we apply the NFT construction in unconstrained logit space:

$$l^+_{\theta,c} = (1-\beta_d) l_{\text{old},c} + \beta_d l_{\theta,c}, \tag{23}$$

$$l^-_{\theta,c} = (1+\beta_d) l_{\text{old},c} - \beta_d l_{\theta,c}. \tag{24}$$

For the one-hot clean endpoint $\mathbf{y}_c$, we use

$$\mathcal{L}_{\text{disc}} = \mathbb{E}_{P,z_1\sim\pi_{\text{old}},t} \sum_c \left[r\,\text{CE}\Big(\mathbf{y}_c, \text{softmax}(l^+_{\theta,c})\Big) + (1-r)\,\text{CE}\Big(\mathbf{y}_c, \text{softmax}(l^-_{\theta,c})\Big)\right]. \tag{25}$$

The final objective is

$$\mathcal{L} = w_{\text{coord}}\mathcal{L}_{\text{coord}} + w_{\text{disc}}\mathcal{L}_{\text{disc}}. \tag{26}$$

Algorithm 1 summarizes the pseudocode for the NFT training loop.

A.1 Reward components, drift, and invalid-sample updates

We characterize how the reward components enter the molecular NFT losses and how masking, neutral completion, and rewards below $1/2$ change their gradients. Throughout this subsection, we condition on a fixed rollout batch and a realization of the forward corruption. The rollout policy, clean targets, reward values, normalization statistics, and component weights are held fixed when differentiating with respect to θ. As in the main text, r denotes the normalized total reward and ℓ a single-sample loss; averaging over samples and corruptions recovers the corresponding objective $\mathcal{L}$. The identities below concern loss gradients before gradient clipping or adaptive optimizer transformations.

Additive rewards combine NFT losses exactly. For either the coordinate or a categorical objective, write the positive and negative branch losses as ℓ^+_θ and ℓ^-_θ. Then

$$\ell_\theta(r) = r\ell^+_\theta + (1-r)\ell^-_\theta. \tag{27}$$

Suppose that component rewards $r_j \in [0,1]$ are combined with fixed weights $\omega_j \geq 0$, where $\sum_j \omega_j = 1$ and $r = \sum_j \omega_j r_j$. Here, j indexes reward components, i indexes samples, and c indexes categorical channels. For a fixed sample, we omit i, so r_v and r_p denote the components $r_{v,i}$ and $r_{p,i}$ in Eqs. 7 and 10. The reward weights ω_j are distinct from the modality weights w_{coord} and w_{disc}. Since $1 - r = \sum_j \omega_j (1 - r_j)$,

$$\ell_\theta(r) = \sum_j \omega_j \ell_\theta(r_j), \qquad \nabla_\theta \ell_\theta(r) = \sum_j \omega_j \nabla_\theta \ell_\theta(r_j). \tag{28}$$

Thus, the combined reward produces a weighted sum of component-specific NFT gradients. This identity assumes the same branch predictors, guidance strengths, and sample inclusion for every component. It applies to component rewards after their individual normalization; an additional nonlinear transformation of the combined reward generally changes the identity. Because the same r weights both coordinate and categorical losses, a graph-only property can influence coordinate predictions even though the property itself does not depend on coordinates.

Coordinate loss and gradient. For a fixed noised input, define

$$\mathbf{d} = \hat{\mathbf{x}}_{1,\text{old}} - \mathbf{x}_1, \qquad \mathbf{u} = \hat{\mathbf{x}}_{1,\theta} - \hat{\mathbf{x}}_{1,\text{old}}, \qquad \tilde{r} = 2r - 1. \tag{29}$$

The branch residuals are $\mathbf{d} + \beta_c \mathbf{u}$ and $\mathbf{d} - \beta_c \mathbf{u}$. Expanding their weighted squared errors gives

$$\ell_{\text{coord}}(r) = \underbrace{\|\mathbf{d}\|_2^2}_{\text{base}} + \underbrace{\beta_c^2 \|\mathbf{u}\|_2^2}_{\text{drift}} + \underbrace{2\beta_c \tilde{r}\, \mathbf{d}^\mathsf{T}\mathbf{u}}_{\text{reward alignment}}. \tag{30}$$

Let $J_x = \partial \hat{\mathbf{x}}_{1,\theta}/\partial\theta$. With the rollout predictor frozen, the parameter gradient is

$$\nabla_\theta \ell_{\text{coord}}(r) = 2\beta_c J_x^\mathsf{T} \left[\beta_c \mathbf{u} + \tilde{r}\mathbf{d}\right]. \tag{31}$$

The base term has zero gradient. The drift term penalizes deviation from the rollout predictor, while the signed reward controls the remaining gradient. In particular,

$$\tilde{r} = \sum_j \omega_j (2r_j - 1). \tag{32}$$

The weights therefore combine signed alignment coefficients while leaving the coordinate drift coefficient β_c^2 unchanged. When the current and rollout predictions coincide, $\mathbf{u} = 0$, a positive signed reward gives the usual endpoint-reconstruction gradient direction, and a negative signed reward reverses that direction.

Masking and neutral rewards are different operations. For a sample mask $m_i \in \{0, 1\}$, a masked batch loss has the form

$$\mathcal{L}_{\text{mask}} = \frac{1}{Z} \sum_i m_i \ell_i(r_i), \tag{33}$$

where $Z > 0$ is the chosen batch normalizer. An excluded sample contributes neither a drift penalty nor an alignment term. By contrast, assigning the total reward $r_i = 1/2$ retains its drift penalty and gives gradient $2\beta_c^2 J_{x,i}^\mathsf{T} \mathbf{u}_i$. At coincident current and rollout predictions, both treatments give zero direct gradient for that sample. Once updates from other samples change its prediction, the neutral sample resists this change while the excluded sample remains absent.

For comparison, setting the total reward to $r_i = 0$ retains both terms, with signed coefficient -1, the most negative value allowed by $r_i \in [0, 1]$. It is therefore an active negative alignment signal rather than an exclusion rule. The drift penalty in all these expressions refers to the current rollout policy, which is replaced during training; it does not permanently anchor the model to the pretrained policy.

Completing an undefined reward component. Let $m_j(z_1)$ indicate whether component j can be evaluated for a generated ligand, and let $\kappa_j \in [0, 1]$ be its value when undefined. Define

$$r_j^{(\kappa_j)}(z_1) = \begin{cases} r_j(z_1), & m_j(z_1) = 1, \\ \kappa_j, & m_j(z_1) = 0. \end{cases} \tag{34}$$

Here, m_j indicates evaluability, not success: an evaluable but failed physical-validity check still receives its defined failure reward. Completion is applied in the normalized component-reward space. Inserting a raw default score before group normalization would also change the normalized rewards of other samples and is a different procedure.

For $r = \sum_j \omega_j r_j^{(\kappa_j)}$, the signed coefficient is

$$\tilde{r} = \sum_{j:m_j=1} \omega_j (2r_j - 1) + \sum_{j:m_j=0} \omega_j (2\kappa_j - 1). \tag{35}$$

For any component with positive weight, $\kappa_j = 1/2$ is the unique constant completion that supplies no signed alignment signal when that component is undefined. The component still contributes its share of the drift penalty through Eq. 28. For an undefined component j, changing only its completion from $1/2$ to κ_j changes the coordinate gradient by

$$\nabla_\theta \ell_{\mathrm{coord}}(r^{(\kappa_j)}) - \nabla_\theta \ell_{\mathrm{coord}}(r^{(1/2)}) = 4\beta_c \omega_j (\kappa_j - 1/2) J_x^{\mathsf{T}} \mathbf{d}. \tag{36}$$

Here, $r^{(\kappa_j)}$ and $r^{(1/2)}$ denote total rewards differing only in the completion of component j. Neutral completion therefore avoids adding an arbitrary signed signal for the unavailable property; it does not assert that the molecule itself is desirable or that its total reward is neutral.

For example, suppose chemical validity is evaluable, while all remaining components are undefined and completed with $1/2$. Then

$$r^{\mathrm{invalid}} = \omega_v r_v + \frac{1-\omega_v}{2}, \qquad 2r^{\mathrm{invalid}} - 1 = \omega_v (2r_v - 1). \tag{37}$$

Only the chemical component supplies the signed signal. If its normalized value is $r_v = 0$ and its weight is $1/2$, the total reward is $1/4$ and the signed coefficient is $-1/2$. This numerical example is conditional on $r_v = 0$: a group-normalized binary validity score need not assign zero to every invalid sample. The normalization convention and treatment of groups with no score variance determine the actual values. Moreover, the additive reward may also assign different rewards to valid samples, so their full training difference cannot be attributed solely to the invalid-sample coefficient.

Categorical gradients and an exact neutral regularizer. For a categorical channel c, define the logit displacement $\mathbf{h}_c = l_{\theta,c} - l_{\mathrm{old},c}$ and

$$p_c^{\pm} = \mathrm{softmax}(l_{\mathrm{old},c} \pm \beta_d \mathbf{h}_c), \qquad p_c^{\mathrm{old}} = \mathrm{softmax}(l_{\mathrm{old},c}). \tag{38}$$

Let $\ell_c^{\pm} = \mathrm{CE}(\mathbf{y}_c, p_c^{\pm})$ and $\ell_c^{\mathrm{old}} = \mathrm{CE}(\mathbf{y}_c, p_c^{\mathrm{old}})$ for the one-hot endpoint $\mathbf{y}_c$. Differentiation with respect to the displacement gives

$$\nabla_{\mathbf{h}_c} \ell_c(r) = \beta_d \left[r(p_c^{+} - \mathbf{y}_c) - (1-r)(p_c^{-} - \mathbf{y}_c) \right]. \tag{39}$$

The corresponding parameter gradient is $J_{l,c}^{\mathsf{T}} \nabla_{\mathbf{h}_c} \ell_c(r)$, where $J_{l,c} = \partial l_{\theta,c} / \partial \theta$. At coincident current and rollout logits,

$$\nabla_{\mathbf{h}_c} \ell_c(r)|_{\mathbf{h}_c = 0} = \beta_d \tilde{r} (p_c^{\mathrm{old}} - \mathbf{y}_c). \tag{40}$$

Thus, the same signed coefficient scales or reverses the usual cross-entropy gradient at this point.

To characterize neutrality for arbitrary logit displacements, define

$$D_c(\mathbf{h}_c) = \tfrac{1}{2}(\ell_c^{+} + \ell_c^{-}) - \ell_c^{\mathrm{old}}. \tag{41}$$

Writing $A(l) = \log \sum_k \exp(l_k)$, the identity $\mathrm{CE}(\mathbf{y}, \mathrm{softmax}(l)) = A(l) - \mathbf{y}^{\mathsf{T}} l$ shows that the terms linear in $\mathbf{y}_c$ cancel. Consequently,

$$\begin{aligned} D_c(\mathbf{h}_c) &= \tfrac{1}{2}\left[A(l_{\mathrm{old},c} + \beta_d \mathbf{h}_c) + A(l_{\mathrm{old},c} - \beta_d \mathbf{h}_c) \right] - A(l_{\mathrm{old},c}) \\ &= \tfrac{1}{2}\mathrm{KL}(p_c^{\mathrm{old}} \| p_c^{+}) + \tfrac{1}{2}\mathrm{KL}(p_c^{\mathrm{old}} \| p_c^{-}) \geq 0. \end{aligned} \tag{42}$$

The second equality follows by expanding each KL divergence: the terms $\pm\beta_d (p_c^{\mathrm{old}})^{\mathsf{T}} \mathbf{h}_c$ cancel in their average. The full categorical loss therefore decomposes exactly as

$$\ell_c(r) = \underbrace{\ell_c^{\mathrm{old}}}_{\text{base}} + \underbrace{D_c(\mathbf{h}_c)}_{\text{drift}} + \underbrace{\tfrac{1}{2}\tilde{r}(\ell_c^{+} - \ell_c^{-})}_{\text{reward alignment}}. \tag{43}$$

For $r = 1/2$, only a constant and the nonnegative regularizer remain. The regularizer is independent of the target class and is minimized when the branch probabilities equal the old probabilities, allowing the usual invariance to a common additive logit shift. Hence, neutral completion has a precise interpretation for both coordinates and categorical endpoints. These identities concern the logit-space objective used here; they do not require a categorical analogue of the coordinate posterior-mean improvement theorem.

Separating feasibility feedback from weaker alignment. The decomposition suggests a control for adding an explicit physical-validity component. Let r^{base} be a baseline reward, such as the chemical-validity and Vina combination, and let r_{PB} be a PB-derived component, completed with $1/2$ when undefined. For a fixed $\lambda \in [0, 1]$, compare

$$r^{\text{aug}} = (1 - \lambda)r^{\text{base}} + \lambda r_{\text{PB}}, \qquad r^{\text{ctrl}} = (1 - \lambda)r^{\text{base}} + \lambda/2. \tag{44}$$

Both constructions give the baseline components the same weights, but only the first supplies physical-validity information. In particular,

$$2r^{\text{ctrl}} - 1 = (1 - \lambda)(2r^{\text{base}} - 1). \tag{45}$$

The neutral component attenuates the signed alignment term while retaining the coordinate drift coefficient and categorical regularizer. At fixed predictors, targets, and normalized component rewards, the difference in coordinate gradients is

$$\nabla_\theta \ell_{\text{coord}}(r^{\text{aug}}) - \nabla_\theta \ell_{\text{coord}}(r^{\text{ctrl}}) = 2\beta_c \lambda (2r_{\text{PB}} - 1) J_x^{\mathsf{T}} \mathbf{d}. \tag{46}$$

This comparison separates the signal supplied by PB from the attenuation caused by introducing another reward component. It is an analytical motivation for a matched control, not evidence that PB augmentation improves the reported models.

Scope of the derivation. The analysis establishes neutrality and gradient effects for the specified objectives. It does not prove that neutral completion is optimal for downstream performance, that rewards below $1/2$ increase valid yield, or that an additive feasibility reward enforces a hard constraint. In a weighted sum, favorable property components can outweigh the negative signed contribution of a physical-validity component. Nor does the negative coefficient at $r = 0$ by itself explain validity collapse. Completing the square gives

$$\ell_{\text{coord}}(r) = \beta_c^2 \left\| \mathbf{u} + \frac{\tilde{r}}{\beta_c} \mathbf{d} \right\|_2^2 + (1 - \tilde{r}^2) \|\mathbf{d}\|_2^2, \tag{47}$$

which remains nonnegative and convex in the prediction displacement for every $r \in [0, 1]$ and $\beta_c > 0$, including $r = 0$. This does not imply convexity in the shared network parameters. The observed training behavior also depends on reward normalization, interactions among sample gradients, optimizer dynamics, and subsequent rollouts.

B Experimental details

Training details. We initialize from the Flowr.root v2.1 checkpoint and retain its architecture, data statistics, molecular size prior, and forward noising processes. Training uses four NVIDIA B200 GPUs. Each device generates 256 ligands per rollout, giving a global rollout size of 1,024. The group size is $K = 8$, corresponding to 32 unique pockets per device and 128 pockets globally. Each rollout is replayed once with a global batch size of 512. The old policy is fully replaced by the updated policy after each rollout.

We optimize all parameters using AdamW with learning rate 3×10^{-6}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 10^{-4}, and a maximum gradient norm of 5. We set the continuous and discrete NFT guidance strength to $\beta_c = \beta_d = 1$ and weight both losses equally. We selected learning rate and guidance strength β through a coarse QED-fine-tuning sweep based on improvement in rollout QED while maintaining rollout RDKit validity. We reuse these settings for all other reward configurations without further tuning. Rollouts use SDE sampling with 100 Euler integration steps. The reported checkpoint was chosen based on property scores on validation pockets rollout, aggregated with the same weights as the reward. For the QED experiment, training for three epochs required approximately 47 hours on the four NVIDIA B200 GPUs.

Datasets and evaluation. SPINDR (Cremer et al., 2025a) is a curated dataset of refined ligand–pocket cocrystal structures derived from PLINDER. We use its predefined training and test splits. CrossDocked2020 (Francoeur et al., 2020) contains ligand poses cross-docked into structurally similar protein pockets. We evaluate the SPINDR-trained checkpoints on the standard 100 CrossDocked2020 test pockets without additional fine-tuning or CrossDocked-based checkpoint selection.

We generate 100 molecules per pocket and do not remove invalid, duplicate, or low-diversity samples before evaluation. RDKit validity is computed over all generations; all other metrics are computed over the subset of RDKit-valid molecules. We report QED and SA as chemical-property metrics; PoseBusters as physical-validity metric; score-only and docked Vina energies as pocket-compatibility metrics; Tanimoto similarity-based diversity as distributional metric; and Wasserstein distances between generated and test-set bond-angle and bond-length distributions.

Baselines. The CrossDocked2020 comparison covers the unguided generators Pocket2Mol (Peng et al., 2022), DiffSBDD (Schneuing et al., 2024), TargetDiff (Guan et al., 2023), DrugFlow (Schneuing et al., 2025), FLOWR (Cremer et al., 2025a), and Flowr.root (Cremer et al., 2025b); the inference-time guidance methods TAGMol (Dorna et al., 2024) and MolJO (Qiu et al., 2025); and the parameter-updating methods conditional SEMLA with RWS (Wang et al., 2025), DrugFlow with DPO (Schneuing et al., 2025), MolFORM with DPO (Zhang & Zhang, 2025), and AliDiff with E2PO (Gu et al., 2024).

B.1 Extended benchmark results

Fig. 5 and Fig. 6 show a detailed breakdown of the PoseBusters validity of the SPINDR-pretrained Flowr.root baseline and the SPINDR-finetuned FlowR.NFT models, for CrossDocked2020 we also show results for previous methods. Tables 2, 3, 4 , 5 provide additional results on the geometry of ligands generated for the SPINDR and CrossDocked2020 datasets, including Wasserstein distances for bond-angle and bond-length distributions, with standard deviation across training run seeds and pockets.

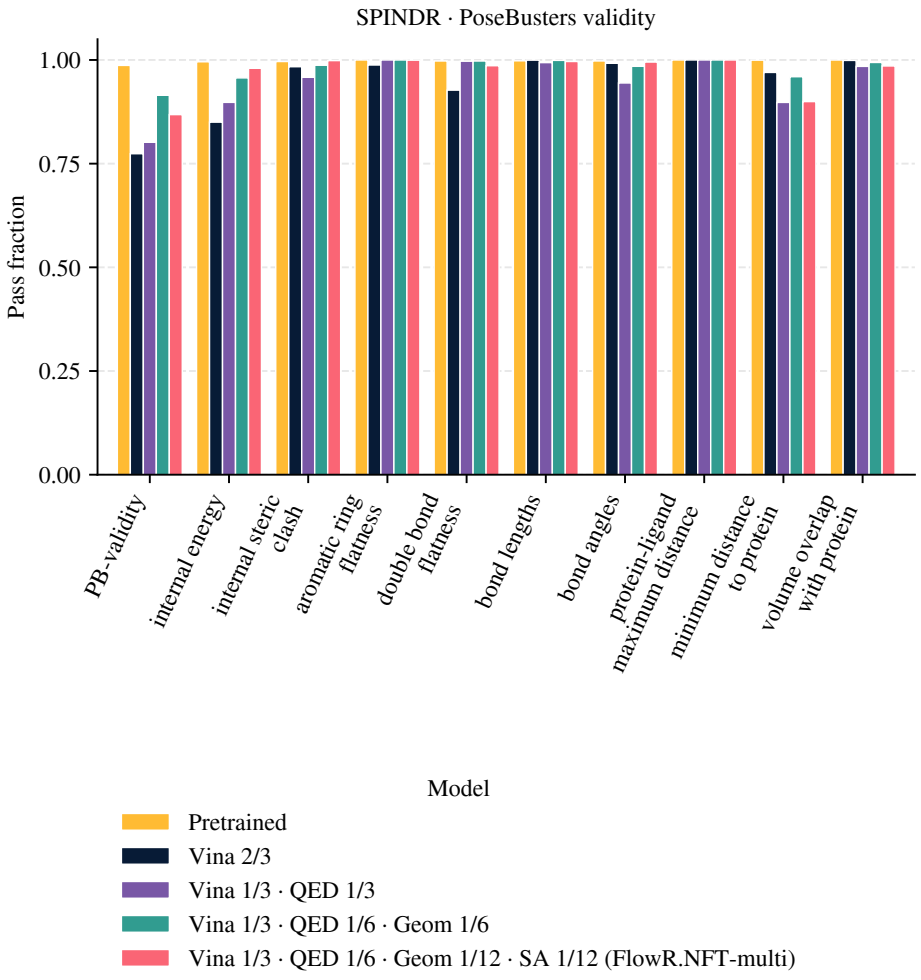

Figure 5: **PoseBusters validity and its breakdown by individual PoseBusters criteria on SPINDR:** Bars compare mean per-pocket pass rates, computed over RDKit-valid samples, for the pretrained Flowr.root model and aligned FlowR.NFT. All these models have only been pretrained and finetuned on SPINDR.

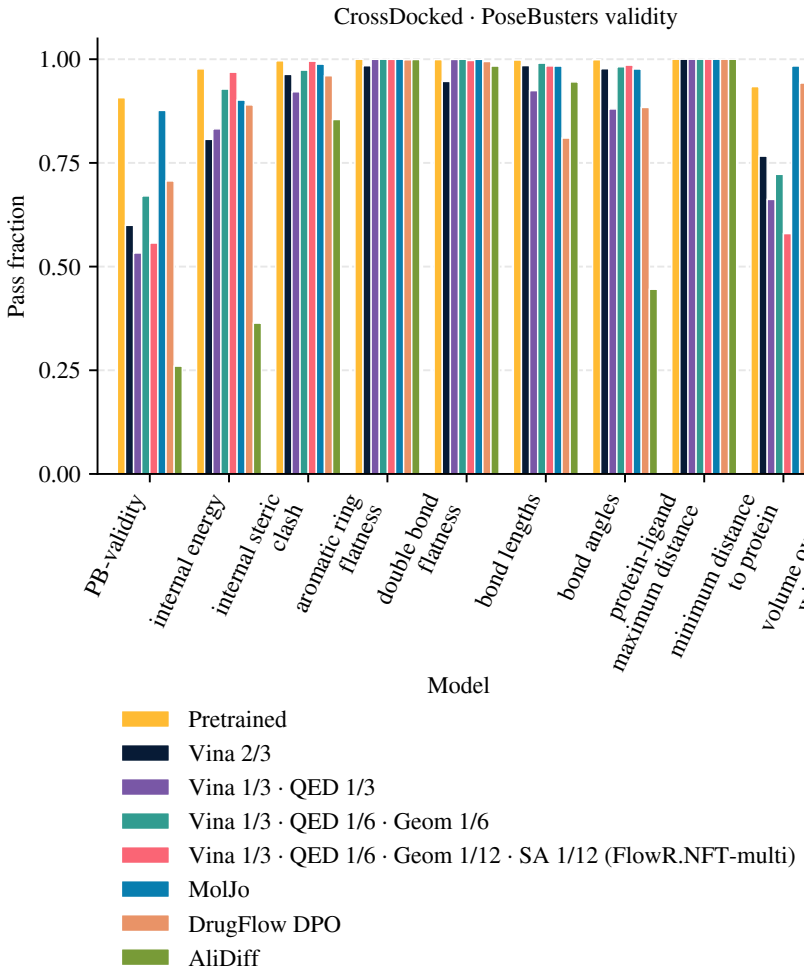

Figure 6: **PoseBusters validity and its breakdown by individual PoseBusters criteria on CrossDocked2020:** Bars compare mean per-pocket pass rates, computed over RDKit-valid samples, for the pretrained Flowr.root model and aligned FlowR.NFT. All these models have only been pretrained and finetuned on SPINDR. Additionally, we show results for MolJo, DrugFlow DPO and AliDiff.

Table 2: **Reward alignment results on the SPINDR test set** with additional metrics. We compare the pretrained Flowr.root base model with FlowR.NFT models fine-tuned for QED, SA, or Vina using the validity-aware reward. Values are mean ± standard deviation across seeds.

	RDKit-Val↑	PB all↑	PB intra↑	PB inter↑	Vina Score↓	Vina Dock↓	SA↑	QED↑	Diversity↑	AnglesW1↓	BondLengthsW1 (10^{-2})↓
Pretrained	0.98	**0.99**	**0.99**	**1.00**	−7.35	−7.71	0.67	0.48	0.53	**0.50**	**0.06**
Vina 2/3	$0.91_{\pm 0.05}$	$0.77_{\pm 0.09}$	$0.80_{\pm 0.09}$	$0.97_{\pm 0.02}$	$\mathbf{-9.52}_{\pm 0.39}$	$\mathbf{-10.78}_{\pm 0.05}$	$0.58_{\pm 0.02}$	$0.43_{\pm 0.01}$	$0.55_{\pm 0.01}$	$4.41_{\pm 0.47}$	$0.33_{\pm 0.05}$
Vina 1/3 · QED 1/3	$0.99_{\pm 0.00}$	$0.80_{\pm 0.00}$	$0.88_{\pm 0.01}$	$0.90_{\pm 0.01}$	$-6.74_{\pm 0.35}$	$-9.30_{\pm 0.05}$	$0.61_{\pm 0.01}$	$\mathbf{0.64}_{\pm 0.01}$	$\mathbf{0.63}_{\pm 0.01}$	$4.07_{\pm 0.23}$	$0.44_{\pm 0.04}$
Vina 1/3 · QED 1/6 · Geom 1/6	$\mathbf{1.00}_{\pm 0.00}$	$0.91_{\pm 0.03}$	$0.95_{\pm 0.01}$	$0.96_{\pm 0.04}$	$-8.85_{\pm 0.98}$	$-10.31_{\pm 0.06}$	$0.55_{\pm 0.02}$	$0.62_{\pm 0.02}$	$0.58_{\pm 0.01}$	$2.94_{\pm 0.67}$	$0.32_{\pm 0.02}$
FlowR.NFT-multi	$0.99_{\pm 0.00}$	$0.87_{\pm 0.05}$	$0.96_{\pm 0.00}$	$0.90_{\pm 0.05}$	$-7.43_{\pm 1.20}$	$-10.14_{\pm 0.22}$	$\mathbf{0.70}_{\pm 0.03}$	$0.50_{\pm 0.01}$	$\mathbf{0.63}_{\pm 0.02}$	$3.20_{\pm 0.16}$	$0.32_{\pm 0.02}$
Test set	1.00	0.99	1.00	1.00	−7.63	−7.88	0.67	0.49	0.00	0.00	0.00

Table 3: **Reward alignment results on the SPINDR test set** with additional metrics. We compare the pretrained Flowr.root base model with FlowR.NFT models fine-tuned for QED, SA, or Vina using the validity-aware reward. Values are mean ± standard deviation across pockets.

	RDKit-Val↑	PB all↑	PB intra↑	PB inter↑	Vina Score↓	Vina Dock↓	SA↑	QED↑	Diversity↑	AnglesW1↓	BondLengthsW1 (10^{-2})↓
Pretrained	$0.98_{\pm 0.03}$	$\mathbf{0.99}_{\pm 0.02}$	$\mathbf{0.99}_{\pm 0.02}$	$\mathbf{1.00}_{\pm 0.00}$	$-7.35_{\pm 1.93}$	$-7.71_{\pm 1.77}$	$0.67_{\pm 0.11}$	$0.48_{\pm 0.19}$	$0.53_{\pm 0.18}$	**0.50**	**0.06**
Vina 2/3	$0.91_{\pm 0.09}$	$0.77_{\pm 0.16}$	$0.80_{\pm 0.14}$	$0.97_{\pm 0.09}$	$\mathbf{-9.52}_{\pm 3.76}$	$\mathbf{-10.78}_{\pm 3.16}$	$0.58_{\pm 0.12}$	$0.43_{\pm 0.12}$	$0.55_{\pm 0.10}$	4.41	0.33
Vina 1/3 · QED 1/3	$0.99_{\pm 0.01}$	$0.80_{\pm 0.20}$	$0.88_{\pm 0.09}$	$0.90_{\pm 0.19}$	$-6.74_{\pm 6.33}$	$-9.30_{\pm 2.37}$	$0.61_{\pm 0.12}$	$\mathbf{0.64}_{\pm 0.11}$	$\mathbf{0.63}_{\pm 0.08}$	4.07	0.44
Vina 1/3 · QED 1/6 · Geom 1/6	$\mathbf{1.00}_{\pm 0.01}$	$0.92_{\pm 0.12}$	$0.95_{\pm 0.04}$	$0.97_{\pm 0.11}$	$-9.07_{\pm 4.52}$	$-10.32_{\pm 2.88}$	$0.54_{\pm 0.11}$	$0.63_{\pm 0.10}$	$0.57_{\pm 0.06}$	2.94	0.32
FlowR.NFT-multi	$0.99_{\pm 0.04}$	$0.87_{\pm 0.18}$	$0.96_{\pm 0.03}$	$0.90_{\pm 0.18}$	$-7.43_{\pm 5.58}$	$-10.14_{\pm 2.76}$	$\mathbf{0.70}_{\pm 0.11}$	$0.50_{\pm 0.13}$	$\mathbf{0.63}_{\pm 0.07}$	3.20	0.32
Test set	$1.00_{\pm 0.00}$	$0.99_{\pm 0.09}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$-7.63_{\pm 2.18}$	$-7.88_{\pm 2.10}$	$0.67_{\pm 0.12}$	$0.49_{\pm 0.22}$	$0.00_{\pm 0.00}$	0.00	0.00

Table 4: **Comparison with pocket-conditional molecular generators on CrossDocked2020** grouped into unguided generation, inference-time guidance, offline preference alignment, and our online reward alignment with additional metrics. Flowr.NFT is fine-tuned on SPINDR and evaluated on CrossDocked2020 without further fine-tuning or checkpoint selection. Other preference-aligned methods optimize different objectives: MolJO uses SA and affinity, DrugFlow-DPO aligns QED, SA, Vina, and REOS, and AliDiff aligns Vina. We recompute all metrics with our evaluation pipeline. For prior methods, we use the authors' provided samples, except for Flowr.root and DrugFlow-DPO we generate samples ourselves. RDKit validity is not reported for Pocket2Mol because its sampling procedure generates molecules until 100 valid outputs are obtained, limiting comparability. Values are mean ± standard deviation across seeds.

	RDKit-Val↑	PB all↑	PB intra↑	PB inter↑	Vina Score↓	Vina Dock↓	SA↑	QED↑	Diversity↑	AnglesW1↓	BondLengthsW1 (10^{-2})↓
Pocket2Mol	NA	0.85	0.92	0.92	−4.24	−6.11	0.81	0.59	**0.84**	2.28	0.47
TargetDiff	0.88	0.51	0.57	0.88	−6.13	−7.66	0.58	0.48	0.72	3.19	0.21
DiffSBDD	**1.00**	0.38	0.48	0.70	−2.31	−7.24	0.58	0.47	0.73	6.99	0.33
DrugFlow	0.85	0.74	0.82	0.90	−5.02	−6.85	0.71	0.51	0.73	2.06	0.13
FlowR.root	0.98	**0.91**	**0.97**	0.93	−5.96	−7.34	0.71	0.50	0.60	**0.59**	**0.10**
MolJO	0.97	0.88	0.89	**0.98**	**−8.25**	−9.07	0.77	0.55	0.75	3.79	0.66
DrugFlow DPO	0.70	0.71	0.75	0.94	−5.77	−7.51	**0.85**	**0.66**	0.79	4.12	0.54
AliDiff	0.92	0.26	0.27	0.94	−7.62	−8.41	0.57	0.50	0.74	6.46	0.13
Vina 2/3	$0.84_{\pm 0.11}$	$0.60_{\pm 0.11}$	$0.76_{\pm 0.11}$	$0.76_{\pm 0.04}$	$-4.69_{\pm 0.95}$	$\mathbf{-9.50}_{\pm 0.09}$	$0.62_{\pm 0.03}$	$0.44_{\pm 0.01}$	$0.58_{\pm 0.00}$	$4.44_{\pm 0.18}$	$0.39_{\pm 0.09}$
Vina 1/3 · QED 1/3	$0.96_{\pm 0.01}$	$0.53_{\pm 0.01}$	$0.78_{\pm 0.03}$	$0.66_{\pm 0.03}$	$-2.68_{\pm 0.50}$	$-8.58_{\pm 0.05}$	$0.65_{\pm 0.01}$	$0.63_{\pm 0.01}$	$0.67_{\pm 0.00}$	$5.12_{\pm 0.34}$	$0.74_{\pm 0.06}$
Vina 1/3 · QED 1/6 · Geom 1/6	$0.99_{\pm 0.01}$	$0.67_{\pm 0.04}$	$0.92_{\pm 0.02}$	$0.72_{\pm 0.06}$	$-3.50_{\pm 1.55}$	$-9.15_{\pm 0.03}$	$0.61_{\pm 0.01}$	$0.60_{\pm 0.01}$	$0.63_{\pm 0.01}$	$4.18_{\pm 0.07}$	$0.48_{\pm 0.03}$
FlowR.NFT-multi	$0.98_{\pm 0.01}$	$0.56_{\pm 0.05}$	$0.95_{\pm 0.00}$	$0.58_{\pm 0.05}$	$-1.14_{\pm 1.09}$	$-9.10_{\pm 0.09}$	$0.74_{\pm 0.03}$	$0.52_{\pm 0.01}$	$0.65_{\pm 0.01}$	$4.95_{\pm 0.04}$	$0.48_{\pm 0.04}$
Test set	1.00	0.94	0.96	0.98	−6.54	−7.19	0.73	0.48	NA	NA	NA

Table 5: **Comparison with pocket-conditional molecular generators on CrossDocked2020** grouped into unguided generation, inference-time guidance, offline preference alignment, and our online reward alignment with additional metrics. Flowr.NFT is fine-tuned on SPINDR and evaluated on CrossDocked2020 without further fine-tuning or checkpoint selection. Other preference-aligned methods optimize different objectives: MolJO uses SA and affinity, DrugFlow-DPO aligns QED, SA, Vina, and REOS, and AliDiff aligns Vina. We recompute all metrics with our evaluation pipeline. For prior methods, we use the authors' provided samples, except for Flowr.root and DrugFlow-DPO we generate samples ourselves. RDKit validity is not reported for Pocket2Mol because its sampling procedure generates molecules until 100 valid outputs are obtained, limiting comparability. Values are mean ± standard deviation across pockets.

	RDKit-Val↑	PB all↑	PB intra↑	PB inter↑	Vina Score↓	Vina Dock↓	SA↑	QED↑	Diversity↑	AnglesW1↓	BondLengthsW1 (10^{-2})↓
Pocket2Mol	NA	$0.85_{\pm 0.09}$	$0.92_{\pm 0.05}$	$0.92_{\pm 0.09}$	$-4.24_{\pm 1.30}$	$-6.11_{\pm 1.33}$	$0.81_{\pm 0.05}$	$0.59_{\pm 0.09}$	$\mathbf{0.84}_{\pm 0.05}$	2.28	0.47
TargetDiff	$0.88_{\pm 0.10}$	$0.51_{\pm 0.16}$	$0.57_{\pm 0.14}$	$0.88_{\pm 0.15}$	$-6.13_{\pm 2.17}$	$-7.66_{\pm 1.75}$	$0.58_{\pm 0.07}$	$0.48_{\pm 0.11}$	$0.72_{\pm 0.09}$	3.19	0.21
DiffSBDD	$\mathbf{1.00}_{\pm 0.00}$	$0.38_{\pm 0.13}$	$0.48_{\pm 0.11}$	$0.70_{\pm 0.23}$	$-2.31_{\pm 5.84}$	$-7.24_{\pm 1.43}$	$0.58_{\pm 0.05}$	$0.47_{\pm 0.09}$	$0.73_{\pm 0.07}$	6.99	0.33
DrugFlow	$0.85_{\pm 0.12}$	$0.74_{\pm 0.16}$	$0.82_{\pm 0.09}$	$0.90_{\pm 0.16}$	$-5.02_{\pm 2.50}$	$-6.85_{\pm 1.73}$	$0.71_{\pm 0.09}$	$0.51_{\pm 0.15}$	$0.73_{\pm 0.12}$	2.06	0.13
FlowR.root	$0.98_{\pm 0.03}$	$\mathbf{0.91}_{\pm 0.12}$	$\mathbf{0.97}_{\pm 0.05}$	$0.93_{\pm 0.11}$	$-5.96_{\pm 2.27}$	$-7.34_{\pm 2.04}$	$0.71_{\pm 0.11}$	$0.50_{\pm 0.16}$	$0.60_{\pm 0.18}$	**0.59**	**0.10**
MolJO	$0.97_{\pm 0.07}$	$0.88_{\pm 0.10}$	$0.89_{\pm 0.09}$	$\mathbf{0.98}_{\pm 0.03}$	$\mathbf{-8.25}_{\pm 2.45}$	$-9.07_{\pm 2.40}$	$0.77_{\pm 0.06}$	$0.55_{\pm 0.08}$	$0.75_{\pm 0.09}$	3.79	0.66
DrugFlow DPO	$0.70_{\pm 0.10}$	$0.71_{\pm 0.16}$	$0.75_{\pm 0.12}$	$0.94_{\pm 0.14}$	$-5.77_{\pm 2.36}$	$-7.51_{\pm 1.82}$	$\mathbf{0.85}_{\pm 0.05}$	$\mathbf{0.66}_{\pm 0.07}$	$0.79_{\pm 0.05}$	4.12	0.54
AliDiff	$0.92_{\pm 0.14}$	$0.26_{\pm 0.21}$	$0.27_{\pm 0.21}$	$0.94_{\pm 0.10}$	$-7.62_{\pm 2.59}$	$-8.41_{\pm 2.13}$	$0.57_{\pm 0.08}$	$0.50_{\pm 0.10}$	$0.74_{\pm 0.07}$	6.46	0.13
Vina 2/3	$0.84_{\pm 0.12}$	$0.60_{\pm 0.28}$	$0.76_{\pm 0.18}$	$0.76_{\pm 0.31}$	$-4.69_{\pm 7.57}$	$\mathbf{-9.50}_{\pm 3.28}$	$0.62_{\pm 0.15}$	$0.44_{\pm 0.11}$	$0.58_{\pm 0.11}$	4.44	0.39
Vina 1/3 · QED 1/3	$0.96_{\pm 0.12}$	$0.53_{\pm 0.30}$	$0.78_{\pm 0.20}$	$0.66_{\pm 0.31}$	$-2.68_{\pm 7.61}$	$-8.58_{\pm 2.49}$	$0.65_{\pm 0.13}$	$0.63_{\pm 0.12}$	$0.67_{\pm 0.08}$	5.12	0.74
Vina 1/3 · QED 1/6 · Geom 1/6	$0.99_{\pm 0.05}$	$0.67_{\pm 0.31}$	$0.92_{\pm 0.07}$	$0.72_{\pm 0.32}$	$-3.50_{\pm 9.27}$	$-9.15_{\pm 2.93}$	$0.61_{\pm 0.16}$	$0.60_{\pm 0.10}$	$0.63_{\pm 0.07}$	4.18	0.48
FlowR.NFT-multi	$0.98_{\pm 0.08}$	$0.56_{\pm 0.34}$	$0.95_{\pm 0.07}$	$0.58_{\pm 0.34}$	$-1.14_{\pm 11.12}$	$-9.10_{\pm 2.73}$	$0.74_{\pm 0.13}$	$0.52_{\pm 0.11}$	$0.65_{\pm 0.11}$	4.95	0.48
Test set	$1.00_{\pm 0.00}$	$0.94_{\pm 0.24}$	$0.96_{\pm 0.20}$	$0.98_{\pm 0.14}$	$-6.54_{\pm 2.74}$	$-7.19_{\pm 2.20}$	$0.73_{\pm 0.14}$	$0.48_{\pm 0.21}$	NA	NA	NA