FLOWR: flow matching for structure-aware de novo, interaction- and fragment-based ligand generation

nature computational science

Brief Communication

https://doi.org/10.1038/s43588-026-00998-8

FLOWR: flow matching for structure-aware de novo, interaction- and fragment-based ligand generation

Received: 12 May 2025

Accepted: 27 April 2026

Published online: 28 May 2026

Check for updates

Julian Cremer[1,4] ✉, Ross Irwin[2,3,4] ✉, Alessandro Tibo[2], Jon Paul Janet[2], Simon Olsson[3] & Djork-Arné Clevert[1]

Here we introduce FLOWR, a structure-based framework for the generation and optimization of three-dimensional ligands. FLOWR integrates continuous and categorical flow matching with equivariant optimal transport, enhanced by an efficient protein pocket conditioning. Alongside FLOWR, we present SPINDR, a curated dataset comprising ligand–pocket cocrystal complexes specifically designed to address existing data quality issues. Empirical evaluations demonstrate that FLOWR surpasses current state-of-the-art diffusion- and flow-based methods in terms of PoseBusters-validity, pose accuracy and interaction recovery, while offering an inference speed-up, achieving up to 70-fold faster performance. In addition, we introduce FLOWR.MULTI, a highly accurate multi-purpose model allowing for the targeted sampling of ligands that adhere to predefined interaction profiles and chemical substructures for fragment-based design without the need of retraining or any resampling strategies. Collectively, our results indicate that FLOWR and FLOWR.MULTI represent an advancement in artificial intelligence-driven structure-based drug design, substantially enhancing the reliability and applicability of de novo, interaction- and fragment-based ligand generation in real-world drug discovery settings.

Structure-based drug discovery (SBDD) is an integrated computational and experimental approach that leverages the three-dimensional structures of biological macromolecules to guide the rational design and optimization of bioactive compounds. By analyzing protein or nucleic acid binding sites, SBDD aims to identify ligands capable of effectively modulating biological functions[1,2]. Commonly employed techniques within this paradigm include molecular docking, virtual screening and structure-guided ligand optimization[3,4]. Despite notable successes, traditional SBDD methods face substantial challenges, such as the inherent complexity of molecular interactions, the vastness of chemical space and difficulties in accurately predicting ligand binding poses and affinities[5,6].

Recent advances in deep learning have provided promising avenues to overcome these limitations. Classical computational methods, including molecular docking and virtual screening, typically rely on simplified approximations of molecular interactions and struggle to efficiently explore extensive chemical spaces. By contrast, data-driven deep learning approaches, particularly generative models, have demonstrated potential in capturing complex relationships inherent in distributions of experimentally determined ligand–protein complexes[7–9].

Among generative modeling techniques, diffusion models have emerged as particularly promising tools for de novo ligand design. These models employ iterative stochastic processes to progressively refine molecular structures from initial random noise into chemically

[1]Machine Learning and Computational Sciences, Pfizer Worldwide R&D, Berlin, Germany. [2]Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg, Sweden. [3]Department of Computer Science and Engineering, Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden. [4]These authors contributed equally: Julian Cremer, Ross Irwin. ✉e-mail: julian.cremer@pfizer.com; rossir@chalmers.se

https://doi.org/10.1038/s43588-026-00998-8

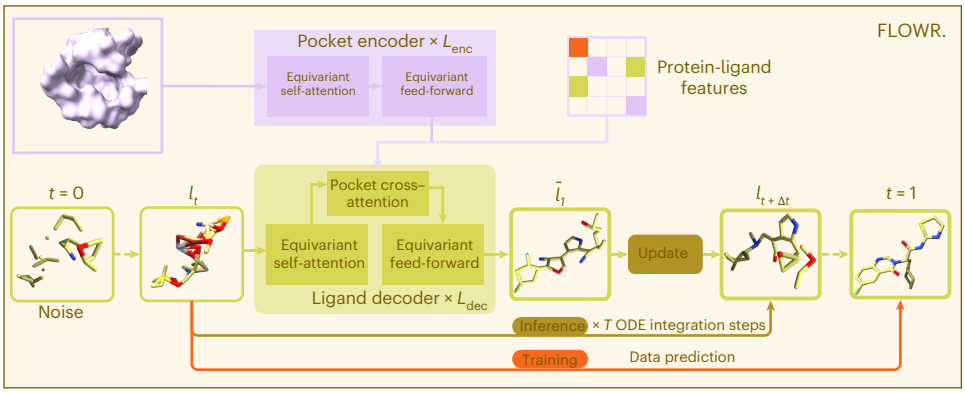


Fig. 1 | Overview of FLOWR. A schematic overview of the FLOWR model for 3D ligand generation. A protein pocket is encoded and passed, along with the noisy ligand l_t, into the ligand decoder, which is trained to produce a denoised ligand $\tilde{l}_t$. Optionally, a set of desired pocket–ligand features can be incorporated. A mixed continuous and categorical flow matching integration scheme is then used to push l_t toward the data distribution and generate a sample $\tilde{l}_1$ by solving the probability-flow ordinary differential equation (ODE) from $t = 0$ to $t = 1$ with a forward Euler solver using T uniform steps. The FLOWR model takes as input pocket coordinates along with atom, bond and residue types, as well as ligand coordinates (with added noise), atom types and bond types. Pocket features are processed through L_{enc} sequential blocks consisting of equivariant self-attention and equivariant feed-forward layers, resulting in a pocket encoding. This pocket encoding is subsequently integrated via equivariant cross-attention into L_{dec} blocks of equivariant self-attention that process ligand features. Finally, FLOWR outputs denoised ligand coordinates, atom types, bond types and charges. During inference, the pocket encoding is computed only once and reused for all ligand generation steps. Atom colors: C (yellow), N (blue), O (red), Cl (green).

valid conformations[10–12]. By incorporating pocket-specific constraints during generation, diffusion models effectively capture the geometric and chemical subtleties of protein–ligand interactions, addressing the challenge of accurately predicting binding poses while generating diverse sets of ligands[12–17].

Nevertheless, existing diffusion-based approaches are not without drawbacks. Their reliance on iterative stochastic sampling can result in molecules exhibiting strained conformations, uncommon substructures and reduced drug-likeness[17]. In addition, these methods typically suffer from prolonged sampling times compared to alternative generative frameworks[18].

Recently, generative flow matching models have emerged as an alternative paradigm, offering substantial improvements in generation efficiency[19]. Notably, flow matching approaches employing mini-batch[20] and equivariant optimal transport[21] have been proposed, with the latter demonstrating particular efficacy in molecular generation tasks[18].

Building upon these insights, we introduce FLOWR, a flow matching model specifically designed for the de novo generation of three-dimensional ligands explicitly conditioned on structural constraints. Our framework enables the efficient generation of ligands informed by the geometry of a protein pocket by using a dedicated pocket encoding scheme in contrast to prior works. In addition, we propose FLOWR.MULTI, a versatile extension capable of multi-purpose conditional generation. This model can efficiently and accurately generate ligands adhering to predefined interaction profiles between ligand atoms and pocket residues and can design ligands around specific chemical substructures such as scaffolds and functional groups, facilitating scaffold elaboration, scaffold hopping and fragment-based ligand design—all without requiring model retraining or computationally expensive stabilization techniques during inference as in prior work[16].

However, the evaluation of SBDD methodologies remains challenging, primarily due to inherent data quality concerns and prevalent data leakage issues in widely utilized benchmark datasets[22,23]. In particular, the commonly employed CROSSDOCKED2020 dataset[24] exhibits substantial limitations for practical drug discovery applications, stemming from its reliance on rigid-pocket cross-docking protocols. Consequently, ligands are artificially constrained into noncognate binding pockets, causing models trained on such data to internalize biased, flawed and unrealistic distributions of ligand–pocket interactions.

To address these critical issues, we introduce SPINDR, a high-quality benchmark dataset specifically developed for SBDD, derived from the recently presented PLINDER dataset[23]. In constructing SPINDR, we implemented an extensive filtering and structural refinement pipeline designed to correct structural defects prevalent in existing datasets[25], accurately infer protonation states and atomic-resolution protein–ligand interaction profiles, and minimize potential data leakage between training and test sets by maintaining the PLINDER dataset split.

In summary, we propose the FLOWR model that improves upon existing approaches in both generative quality and computational efficiency. Moreover, our multi-purpose approach, FLOWR.MULTI, enables the generation of ligands conditioned on specific interaction profiles or chemical substructures, substantially increasing the proportion of ligands closely aligned with reference complexes and enhancing applicability in downstream tasks such as hit expansion, hit-to-lead and lead optimization. Finally, our SPINDR dataset provides a high-quality resource for training and evaluating three-dimensional (3D) generative models, addressing limitations—particularly regarding pose quality and data leakage—in currently available datasets.

Results

We present FLOWR—a flow-based generative model for de novo ligand generation conditioned on a protein pocket and desired pocket–ligand features. We assume access to a dataset containing tuples of a ligand l, a protein pocket $\mathcal{P}$ to which the ligand binds and optionally a matrix $\mathcal{I} \in \mathbb{N}^{M \times N}$ of atomic protein–ligand features, where M and N refer to the number of atoms in the protein and ligand, respectively. In Fig. 1, we show an overview of how our model generates ligands based on protein pocket and pocket–ligand feature conditioning.

The neural network architecture for FLOWR is based on the recently proposed SEMLA architecture[18], an $E(3)$-equivariant message passing framework with latent attention that achieves state-of-the-art results on unconditional 3D molecular generation. We extend SEMLA to allow conditional generation by incorporating a separate pocket encoder and adding an equivariant cross-attention module within the ligand decoder, enabling structural conditioning on the protein pocket and desired protein–ligand features. Critically, the pocket encoder

does not depend on the flow time t or the noisy ligand l_t, meaning only one forward pass through the encoder is required when generating ligands, amortizing the encoding cost over many samples. We further improve the base architecture by introducing a gated equivariant feed-forward module and passing bond embeddings into every self-attention layer, yielding improved validity and efficiency. Full architectural details and hyperparameters are provided in the Methods.

FLOWR jointly models continuous (coordinates) and discrete (atom types and bond orders) molecular features using a combination of continuous flow matching[19,20] for coordinates and discrete flow models[26] for categorical properties, with equivariant optimal transport[21] to reduce transport costs. Ligand formal charges are directly predicted. The model learns to recover the clean ligand l_1 from a noisy interpolant l_t via $p^{\theta}_{1|t}(l_1|l_t, t; \mathcal{P}, \mathcal{I})$, minimizing mean-squared error for coordinates and cross-entropy for categorical features. Given $\mathcal{P}$ and optionally $\mathcal{I}$, ligands are generated by iteratively refining an initial noisy ligand $l_0 \approx p_{\text{noise}}$ through Euler integration of the learned vector field. Full training and sampling details are provided in the 'Training and inference' section in the Methods.

We extend the FLOWR model with FLOWR.MULTI—a multi-purpose training and inference framework that simultaneously supports both de novo generation and any form of fragment-based sampling, similar to scaffold hopping, scaffold elaboration, fragment linking and fragment-based generation, which is highly relevant from hit expansion to lead optimization campaigns. As before, we consider a protein pocket $\mathcal{P}$ and a protein–ligand feature matrix $\mathcal{I}$, while assuming a (set of) predefined fragmentation(s) applied onto a ligand l. Let the ligand consist of N atoms with $\mathbf{l_1} \in \mathbb{R}^{N\times 3}$ denoting its coordinates, and, for simplicity, assume that it is split into two fragments containing n_1 and n_2 atoms, respectively, with $n_1 + n_2 = N$. Thus, we have $\mathbf{l_{t_1t_2}} = \begin{pmatrix} \mathbf{l}^1_{\mathbf{t}_1} \\ \mathbf{l}^2_{\mathbf{t}_2} \end{pmatrix} \in \mathbb{R}^{(n_1+n_2)\times 3}$ with t_1 and t_2 sampled independently from a uniform distribution. Setting $\mathbf{t_{12}} = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$, the linear interpolation reads $\mathbf{l_{t_1t_2}} = \mathbf{t_{12}} \odot \mathbf{l_1} + (\mathbf{1} - \mathbf{t_{12}}) \odot \mathbf{l_{00}} = \begin{pmatrix} t_1\mathbf{l}^1_{\mathbf{t}_1} + (1-t_1)\mathbf{l}^1_{\mathbf{0}} \\ t_2\mathbf{l}^2_{\mathbf{t}_2} + (1-t_2)\mathbf{l}^2_{\mathbf{0}} \end{pmatrix}$ where $\odot$ denotes elementwise multiplication, $\mathbf{1}$ is the all-ones vector and $\mathbf{l_{00}} \approx p_{\text{noise}}$ is the initial noise sample.

The goal is to learn a joint probability distribution $p^{\theta}_{1|t_1t_2}(\mathbf{l_1}|\mathbf{l_{t_1t_2}}, \mathbf{t_{12}}; \mathcal{P}, \mathcal{I})$, from which at inference we sample $\tilde{\mathbf{l}}_1 = \begin{pmatrix} \tilde{\mathbf{l}}^1_1 \\ \tilde{\mathbf{l}}^2_1 \end{pmatrix} \in \mathbb{R}^{N\times 3}$ to retrieve the joint vector field $f(\mathbf{l_{t_1t_2}}, \mathbf{t_{12}}; \mathcal{P}, \mathcal{I}) = \tilde{\mathbf{l}}_1 - \mathbf{l_{00}}$.
Denoting the per-fragment step sizes by $\Delta t_1 = t_1 + s_1$ and $\Delta t_2 = t_2 + s_2$, where s_i is derived from the number of inference steps, and defining $\Delta t_{12} = \begin{pmatrix} \Delta t_1 \\ \Delta t_2 \end{pmatrix}$, the Euler update step reads

$$\mathbf{l_{t_1+\Delta t_1 t_2+\Delta t_2}} = \mathbf{l_{t_1t_2}} + \Delta t_{12} \odot f(\mathbf{l_{t_1t_2}}, \mathbf{t_{12}}; \mathcal{P}, \mathcal{I}) = \begin{pmatrix} \mathbf{l}^1_{\mathbf{t}_1} + \Delta t_1 \cdot (\tilde{\mathbf{l}}^1_1 - \mathbf{l}^1_{\mathbf{0}}) \\ \mathbf{l}^2_{\mathbf{t}_2} + \Delta t_2 \cdot (\tilde{\mathbf{l}}^2_1 - \mathbf{l}^2_{\mathbf{0}}) \end{pmatrix}. \quad (1)$$

Notably, when setting, for example, $t_1 = 1$ and $\mathbf{l}^1_{\mathbf{0}} = \mathbf{l}^1_{\mathbf{t}_1} = \mathbf{l}^1_{\mathbf{1}}$, we have $\Delta t_1 = 1$, as s_1 becomes 0 and the update becomes

$$\mathbf{l_{1t_2+\Delta t_2}} = \begin{pmatrix} \tilde{\mathbf{l}}^1_1 \\ \mathbf{l}^2_{\mathbf{t}_2} + \Delta t_2 \cdot (\tilde{\mathbf{l}}^2_1 - \mathbf{l}^2_{\mathbf{0}}) \end{pmatrix}. \quad (2)$$

In this scenario, the atoms corresponding to $t_1 = 1$ remain fixed to be the predictions of the model at each inference step. Assuming the model has successfully learned the identity mapping $\tilde{\mathbf{l}}^1_1 = \mathbf{l}^1_{\mathbf{1}}$ for the conditional distribution $p^{\theta}_{1|1t2}(\mathbf{l_1}|\mathbf{l_{1t_2}}, \begin{pmatrix} 1 \\ t_2 \end{pmatrix}; \mathcal{P}, \mathcal{I})$, this approach effectively resembles the concept of so-called inpainting. Originally proposed in computer vision[27], inpainting has already been adopted for molecular generation tasks[16]. However, unlike ref. 16, which requires costly resampling steps, and ref. 28, which suffers from reduced structural fidelity, FLOWR.MULTI avoids both limitations while maintaining or even enhancing quality across all generation modes. Specifically, since FLOWR.MULTI is explicitly trained on a diverse set of inpainting tasks, we anticipate substantial improvements in validity rates, structural accuracy and inference efficiency, considerably broadening its downstream applicability.

Modeling interactions between protein pockets and ligands has recently been gaining attention as a method for evaluating the quality of binding poses and designing better small molecule drug candidates[29,30]. At the same time, questions have been raised about the quality of existing benchmark datasets—PDBBind[31] has been found to contain covalently bound ligands, missing atoms in pockets and steric clashes between the pocket and the ligand[25]. CROSSDOCKED2020[24], another commonly used dataset for pocket-conditioned ligand generative models, is based on the PDBBind General set and is also likely to share many of these structural defects. Moreover, questions have also been raised as to how well temporal data splits, which are commonly used to create benchmark test sets, are able to assess models' abilities to generalize to unseen data, as there are often close structural similarities between complexes in the training and test sets.

To address the issues of data quality and information leakage and to provide rich, fine-grained information on the interactions between protein pockets and small molecule ligands, we present SPINDR (Small molecule Protein Interaction Dataset, Refined). Using the recently proposed PLINDER dataset[23] as a starting point we apply an extensive filtering and processing pipeline to produce a refined set of high-quality structures. Specifically, to create the SPINDR dataset, we took the PLINDER dataset release June 2024 (PLINDER version 2) and applied the following processing pipeline:

(1) Initial filtering: We remove all PLINDER systems which contain more than one ligand or have more than one protein chain in the pocket. We then remove all systems where the ligand is marked as one or more of the following: 'oligo', 'ion', 'cofactor', 'artifact', 'fragment', 'covalent' or 'other'.
(2) Structure refinement: We use Schrodinger protein preparation wizard (which uses the OPLS 2005 forcefield[32]) to refine the structure of the remaining systems. These tools perform the following:
   - Add missing atoms to partially filled residues in the protein
   - Convert some nonstandard residue types to standard ones
   - Assign protonation states to heavy atoms and add hydrogen atoms to both the protein and ligand
   - Infer bonds and formal charges for both the protein and ligand
   - Apply local energy minimization to the protein–ligand complex
(3) Infer protein–ligand interactions: We use ProLIF[33] to infer the interactions between the protein and ligand at an atomic resolution, creating a binary matrix of shape $N_{\text{prot}} \times N_{\text{lig}} \times |S|$, where N_{prot} is the number of atoms in the protein, N_{lig} is the number of atoms in the ligand and S is the set of possible interaction types. We apply ProLIF with the default settings and infer all supported interaction types, $|S| = 13$.
(4) Quality filtering: We apply a final filtering step and accumulate the processed systems into train, validation and testing splits. Here, we ensure that all systems contain RDKit-valid ligands. We also filter out any system which contains atoms other than {H, C, N, O, F, P, S, Cl, Se, Br} and any system with fewer than five residues in the pocket. In addition, we filter out all systems containing NAG ligands because we found these were highly overrepresented, which would probably create an unwanted bias for

https://doi.org/10.1038/s43588-026-00998-8

Table 1 | Feature comparison of SPINDR with commonly used datasets

Dataset	Crystal structure	Energy minimized	Explicit	Protein–ligand
	Complexes	Conformations	Hydrogens	Interactions
CROSSDOCKED	✗	✗	✗	✗
PDBBIND	✓	✗	✗	✗
SPINDR	✓	✓	✓	✓

An overview of the additional features provided by the SPINDR dataset compared to datasets commonly used for training generative models for structure-based drug design and docking tasks. The checkmarks indicate presence, and the crosses indicate absence of each feature.

generative models. We also filter out all systems derived from the PDB complex '1mvm' because it contains many small DNA fragments and was not originally filtered by PLINDER.

(5) Data deduplication: As existing datasets often contain substantial structural redundancy, we experiment with two data deduplication strategies presented in Supplementary Information section 1. In practice, we found little empirical difference between the original and deduplicated datasets, despite their substantially smaller size (10,000–15,000 fewer complexes), suggesting there is a significant amount of redundancy in the original data. However, we use the nondeduplicated version of SPINDR for the remainder of this study, though we make all three versions publicly available to enable further investigation of deduplication strategies in ligand generation tasks.

Our final dataset contains 35,666 protein–ligand complexes, making SPINDR the largest dataset of high-quality, refined structures derived directly from crystallographic data. In Table 1, we compare some of the features of SPINDR to other commonly used dataset for SBDD and docking. Notably, in addition to the features in Table 1, we maintain the same data splits as PLINDER. The PLINDER splits were carefully selected to minimize data leakage between train and test sets and to ensure test systems were always of high quality. This careful curation enables realistic assessment of models' generalizability to unseen data, unlike many existing benchmarks which contain substantial train-test data leakage[23].

We compare our model against recently published generative methods on the commonly used CROSSDOCKED2020 dataset as an initial benchmark. As can be seen in Extended Data Table 1, the proposed FLOWR model substantially outperforms all baseline methods (POCKET2MOL[14], DIFFSBDD[16], TARGETDIFF[15], DRUGFLOW[34] and PILOT[17]) across the evaluated metrics on the CROSSDOCKED2020 test dataset. Specifically, FLOWR achieves the highest PoseBusters-validity (PB-validity) (0.92 ± 0.22), lowest strain energy (87.83 ± 74.30), best AutoDock-Vina scores (mean −6.29 ± 1.56, minimized −6.48 ± 1.45) and lowest Wasserstein distances for bond angles (0.96) and bond lengths (0.27). In addition, FLOWR demonstrates substantially faster inference time (12.05 ± 8.01 s) compared to other methods. These results indicate that FLOWR generates ligand conformations closest to the test set distribution and with superior computational efficiency. We note that the second best model, PILOT, also shows substantially better results compared to all other methods, especially in terms of PB-validity (0.83 ± 0.33). Thus, we selected PILOT as our main competitor for the remaining of this study.

In Extended Data Fig. 1, we compare PILOT and FLOWR trained on the SPINDR training dataset in terms of RDKit-validity, PB-validity and inference speed on the SPINDR test set. Our results indicate that FLOWR generates ligands with substantially higher validity on average. Although RDKit-validity is a two-dimensional ligand-centric measure, the PoseBusters suite[35] evaluates ligand conformations using well-established 3D ligand–pocket-based metrics, providing a more comprehensive assessment of pose accuracy. FLOWR achieves a substantial improvement over PILOT in both metrics, with an average RDKit-validity of 0.94 ± 0.24 versus 0.79 ± 0.39 and an average PB-validity of 0.88 ± 0.21 versus 0.71 ± 0.18, respectively. Notably, FLOWR substantially improves inference speed, outperforming PILOT by a factor of 20 when using 100 inference steps, as shown in Extended Data Fig. 1, right. This efficiency gain is primarily attributed to FLOWR's model architecture and the protein pocket encoder requiring only a single forward when integrating the vector field. By contrast, prior models[12,15–17] often recompute protein pocket embeddings at every sampling step. Notably, the number of integration steps can be reduced as low as 20, achieving a 70-fold speed-up over PILOT while impacting model performance comparably little.

In Extended Data Table 2, we compare PILOT and FLOWR in terms of strain energy (calculated using GenBench3D[36]), AutoDock-Vina score (used as an approximate measure of pose quality and binding affinity[37]) and their ability to generalize to the test set distribution based on Wasserstein distance measures for bond angles, bond lengths and dihedral angles following[11,12,17]. A more comprehensive overview of results is given in Supplementary Information section 2. As flow matching allows for setting the number of inference steps, we also report the same results for different number of steps, namely 20, 50 and 100 (default).

In terms of strain energy values, FLOWR substantially outperforms PILOT (90.05 ± 52.18 versus 120 ± 71.61). However, we note that, on average, the strain energies of generated ligands do not align well with those of the test set (43.27 ± 41.85), as illustrated in Extended Data Fig. 2, top left. We hypothesize that this discrepancy arises primarily from limited coverage of chemical and conformational space in the training data, due to the relatively low availability of cocrystal structures. Notably, these elevated strain energies reflect subtle deviations in bond angles and lengths rather than gross structural defects-simple MMFF94s-based relaxation with fixed protein pockets reduces strain energies to 28.05 ± 28.25 kcal mol^{-1}, below the test set average. In addition, a mean r.m.s.d. of 0.775 ± 0.144 Å; confirms that generated molecules readily relax to low-energy conformations with minimal structural perturbation (Supplementary Fig. 3), whereas mean PB-validity increases to 0.95 ± 0.08 and mean Vina score decreases to −6.97 ± 0.89. In addition, Extended Data Fig. 2, top right, shows the relaxation energy distribution of generated ligands calculated using the GFN2-XTB method[38] with implicit solvation using the ALPB model[39] (46.37 ± 64.05 kcal mol^{-1} on the test set; 100.37 ± 59.85 for FLOWR versus 107.89 ± 66.37 for PILOT). Another commonly reported metric in this context is the clash count between ligand and pocket atoms. Using PoseCheck[30], we observe a clear improvement for FLOWR (5.25 ± 2.21) compared to the PILOT model (6.28 ± 2.61), with FLOWR more closely resembling the test set distribution (4.24).

FLOWR outperforms PILOT in docking assessments, suggesting a higher pose accuracy (−6.93 ± 0.92 versus −6.30 ± 0.96). We not only use Vina's scoring function with no redocking applied but also report the minimized Vina score, where local energy minimization is applied to the ligand (−7.22 ± 0.92 versus −6.68 ± 1.07). In Extended Data Fig. 3, left, we compare the Vina score distribution across targets and the mean Vina score per target (Extended Data Fig. 3, right, log scale). As can be seen, FLOWR shows a 12.8% increase in success rate (number of ligands per target that are either equal or better than the reference with respect to Vina scoring) with an average success of 29.5%.

Moreover, we measure distribution learning capabilities in terms of bond angle and bond length Wasserstein distances to the test set.

Here, FLOWR demonstrates substantially better generalization compared to PILOT with a mean bond angles distance of 1.08 versus 1.82, mean bond lengths distance of 0.35 versus 0.42 and mean dihedral angles distribution distance of 5.51 versus 3.45. We observe similar results when comparing both models on a set of relevant molecular properties such as lipophilicity (log*P*), topological polar surface area (TPSA) and number of aromatic rings and the synthesizability of generated molecules against the test set shown in Extended Data Fig. 2, bottom. Although PILOT shows similar distributions for both log*P* and TPSA, it is substantially worse in reproducing the number of aromatic rings and similarly synthesizable compounds compared to the test set. In Supplementary Fig. 2, we provide additional results comparing FLOWR and PILOT on the SPINDR test set using key drug-likeness filters proposed by ref. 40 (inspired by ref. 41). These findings demonstrate that ligands generated by PILOT exhibit up to 40% lower pass-through rates compared to those generated by FLOWR, with FLOWR's results aligning substantially more closely with the SPINDR test data.

In Supplementary Table 3, we repeat the same experiments while incorporating explicit hydrogens in the ligands for both training and inference. Under these conditions, PILOT exhibits a clear decrease in performance, whereas FLOWR maintains comparable results. However, for both models validity drops substantially, with RDKit-validity decreasing to 0.64 ± 0.48 for FLOWR and 0.52 ± 0.50 for PILOT, whereas PB-validity declines to 0.60 ± 0.22 and 0.47 ± 0.14, respectively. Notably, strain energy metrics improve substantially with explicit hydrogen modeling for both models. Although heavy-atom-only approaches require post hoc hydrogen addition potentially leading to artificially inflated strain energies, analysis of individual PoseBusters metrics reveals that reduced validity stems predominantly from ligand–protein clashes rather than internal molecular geometry issues. As SPINDR provides limited coverage of both chemical and conformational space, we anticipate these clashes will diminish with increased training data and, critically, by modeling explicit hydrogens also in protein pockets alongside ligands.

Overall, the proposed FLOWR model consistently outperforms PILOT across all evaluated metrics, demonstrating substantially improved capability in modeling and generalizing ligand–pocket complex distributions. Specifically, we observe an average increase of approximately 15% in ligand and ligand–pocket validity metrics, along with substantially improved AutoDock-Vina scores, indicating higher-quality generated poses. Interestingly, using FLOWR with only 50 inference steps consistently yields better results and yields comparable or slightly worse results with 20 inference steps compared to the PILOT model with 500 steps. Thus, FLOWR achieves substantial performance gains while reducing inference time by up to a factor of 70.

Nevertheless, there remains room for improvement, particularly in reducing strain energies of generated ligands. Moreover, accurately modeling ligand–pocket complexes with explicit hydrogens continues to be challenging, especially in scenarios with limited training data. We encourage the scientific community to evaluate generative models incorporating explicit hydrogen atoms as a challenging benchmark in future research. In this context, the proposed SPINDR dataset represents a valuable resource, providing a robust and comprehensive benchmark for evaluating and comparing ligand generation models.

In SBDD, understanding how a ligand interacts with its target binding site at the atomic level is essential for optimizing potency, selectivity and pharmacological properties[33,42,43]. Ligand–pocket interactions—including hydrogen bonds, hydrophobic contacts, π–π and π–cation stacking, salt bridges and electrostatic or van der Waals interactions—collectively determine binding affinity and specificity. Consequently, these protein–ligand interactions or, more precisely, the ligand's binding pose are crucial for assessing biological relevance and activity[29]. To systematically identify such interactions, protein–ligand interaction fingerprints (PLIFs) are commonly employed[29,33]. PLIFs encode key interaction features, including the interacting protein residue, interaction type and, optionally, the ligand atom involved[29,33].

In contrast to previous studies, we closely investigate our proposed model's capability to reproduce interactions observed in reference ligands. Extended Data Fig. 4 illustrates the distribution of interaction recovery rates for PILOT and FLOWR across the SPINDR test set targets, both with and without explicit hydrogen modeling, using the same sampling settings as before. We also report the success rate, defined as the proportion of RDKit- and PoseBusters-valid ligands for which interactions could be identified. As shown, FLOWR consistently outperforms PILOT (47.1% versus 43.2%, with success rates of 90.4% versus 75.5%), particularly when explicitly modeling hydrogens (42.5% versus 26.5%, with success rates of 67.9% versus 49.8%).

However, these results suggest that a purely de novo generation approach may be less suitable for targeted ligand generation tasks commonly employed in hit expansion and optimization campaigns. To address this, we propose using FLOWR.MULTI (see above), a multi-purpose model capable of interaction-conditional generation. We present detailed results for this approach in the following section.

To improve interaction recovery, we propose to use FLOWR.MULTI with an interaction-based fragmentation for training and inference, which ensures that in inpainting-mode ligand atoms involved in pocket interactions are kept fixed. Let $\mathbf{X}_\mathrm{p} = \{\mathbf{x}_{\mathrm{p},j} \in \mathbb{R}^3 : j = 1, \ldots, n_\mathrm{p}\}$ denote the 3D coordinates of the n_p pocket atoms, $\mathbf{X}_\mathrm{l} = \{\mathbf{x}_{\mathrm{l},i} \in \mathbb{R}^3 : i = 1, \ldots, n_l\}$ denote the ground-truth (native) 3D coordinates of the n_l ligand atoms, $I \in \{0,1\}^{n_\mathrm{p} \times n_\mathrm{l} \times d_I}$ be an interaction tensor, where the entry $I_{j,i,k}$ indicates whether pocket atom j and ligand atom i participate in an interaction of type k (with d_I possible interaction channels). We define a binary mask $M \in \{0,1\}^{n_\mathrm{l}}$ by

$$M_i = \mathbb{I}\left\{\sum_{j=1}^{n_\mathrm{p}} \sum_{k=1}^{d_I} I_{j,i,k} > 0\right\}, \quad i = 1, \ldots, n_\mathrm{l}, \tag{3}$$

where $\mathbb{I}\{\cdot\}$ denotes the indicator function. This mask partitions the ligand atoms into a set of interacting atoms (superscript I) for which we fix the flow time to $t_I = 1$ and set the noise to the ground truth $\mathbf{I}_0^I = \mathbf{I}_1^I$, and a set of remaining atoms that are generated unconstrained, as described above.

Using FLOWR.MULTI, we achieve an average interaction recovery rate of 76.1%; the distribution compared to the FLOWR model is shown in Extended Data Fig. 5, left. Notably, despite the conditional generation process, the model maintains its ability to explore chemical space, achieving an average molecular diversity of 0.83 compared to 0.86 for FLOWR. As illustrated in Extended Data Fig. 5, right, FLOWR.MULTI also substantially improves predicted binding affinity, as indicated by a lower average Vina score (−7.18 versus −6.93), while interaction Tanimoto similarity nearly doubles.

Thus, we observe that FLOWR.MULTI effectively generates ligands adhering to predefined interaction profiles, improving pose accuracy (as measured by Vina scoring) without substantially compromising chemical diversity or exploration compared to the purely de novo FLOWR model. For a more comprehensive overview of the performance of FLOWR.MULTI on the SPINDR test dataset for interaction-conditional and multi-purpose generation, we refer to Supplementary Information section 2 for detailed results. In the next section, we further investigate the multi-purpose capabilities of FLOWR.MULTI in more details using two test set targets.

To evaluate the multi-purpose generative capabilities of the FLOWR.MULTI model, we randomly selected two targets (PDB ID: 5YEA and 4MPE) from the test set and generated ligands under three distinct conditions: interaction-conditional, scaffold-conditional and functional group-conditional generation. Note that FLOWR.MULTI can also be applied to tasks such as fragment linking and fragment growing; however, for clarity, we leave the evaluation of these additional applications to future work.

The selected crystal structure 5YEA represents lipoprotein-associated phospholipase A2 (Lp-PLA2), a validated therapeutic target

implicated in atherosclerosis, Alzheimer's disease and diabetic macular edema[44]. Potent inhibitors has been found through fragment screening, molecular docking and structure-guided optimization, achieving substantial potency improvements from micromolar to single-digit nanomolar inhibitors[44]. Given the proven effectiveness of structure-based approaches for this target, applying FLOWR.MULTI to Lp-PLA2 (5YEA) is particularly interesting. The other selected protein target, 4MPE, corresponds to pyruvate dehydrogenase kinase (PDK), an enzyme family (isoforms 1–4) that negatively regulates mitochondrial pyruvate dehydrogenase complex activity through phosphorylation[45]. PDK isoforms are clinically relevant, as their overexpression is associated with obesity, diabetes, heart failure and cancer, making them attractive therapeutic targets. Previous works explored a structure-guided approach to design selective inhibitors targeting the conserved ATP-binding pocket of PDK isoforms, resulting in the potent inhibitor PS10[45], making it an interesting reference point for our FLOWR.MULTI model.

For each target and condition, we generated 100 ligands and compared the resulting ligand distributions to the respective reference ligand in terms of PB-validity, Vina docking score, interaction recovery rate and synthetic accessibility. The results are summarized in Supplementary Tables 6 and 7. We consistently observe high PB-validity across targets, suggesting that FLOWR.MULTI effectively learned to generate accurate ligand poses independent of the conditioning mode. Although the mean Vina scores across generated ligands does not match those of the reference ligands, selecting the top-10 ligands based on Vina scores consistently yielded ligands with superior docking scores compared to the references (with slightly worse results for functional group-conditional generation). Interaction recovery rates are generally close to 1, indicating that the generated ligands closely reproduce the interaction profiles of the reference ligands. Finally, the mean SA scores, indicative of synthesizability, are consistently around or above 0.80, comparable to the reference ligands. This suggests that the generated ligands not only satisfy relevant physicochemical criteria but also are likely to be synthetically accessible. In Extended Data Fig. 6, we show the chemical space coverage of generated ligands per generation mode, evaluate the sample diversity and the diversity toward the reference compound. Notably, we find a strong dependence of ligand diversity and reference similarity on the condition-mode. Although in the de novo setting we get the most diverse set of ligands, as to be expected, the interaction-conditional setting also shows a strong chemical space coverage although interaction recovery is substantially enhanced reaching almost 90%. However, especially the functional-group-conditional setting allows for a close resemblance of the reference's chemical space. This is interesting, as this shows that practitioners can use different conditional setups of FLOWR.MULTI for controlled chemical space exploration. In Extended Data Fig. 7, we visualize a randomly selected ligand for 5YEA per conditioning mode and compare to the reference ligand. Extended Data Fig. 8 shows the corresponding visualization for 4MPE. More examples and visualizations for both targets are provided in Supplementary Information section 2.

Discussion

Although structure-based generative models have the potential to accelerate early-stage drug discovery by serving as ideation tools from hit exploration to lead optimization, they must be viewed as complementary to—rather than replacements for—medicinal chemistry expertise and experimental validation. The generated molecules represent hypotheses that require rigorous downstream evaluation, including synthesis feasibility assessment, in vitro activity profiling and binding pose confirmation through cocrystallography or cryo-electron microscopy.

Several limitations of the current approach should be acknowledged. First, ligand validity degrades substantially when modeling explicit hydrogens, which we attribute to limited training data coverage and the absence of explicit hydrogen modeling on the protein side. Addressing this will probably require both larger datasets and joint protein–ligand hydrogen modeling during training. Second, although the SPINDR dataset provides high-quality cocrystal structures with careful split design, it offers limited chemical and conformational space coverage compared to the full drug-like molecule landscape; models trained on it may not generalize well to underrepresented chemotypes or binding site topologies. Third, the model operates on static protein structures and does not account for conformational flexibility, induced fit effects or allosteric modulations, which are often critical for accurate binding mode prediction in real drug discovery campaigns. Finally, although strain energies of generated ligands are substantially lower than those of competing approaches, they remain elevated relative to cocrystal reference structures, indicating that further improvements in conformational accuracy are needed.

Several future directions could address these limitations and expand the applicability of flow-based generative models for structure-based drug design. Scaling to larger and more chemically diverse training sets, potentially incorporating data from predicted protein–ligand complexes, could improve coverage of chemical space and enhance explicit hydrogen modeling. Incorporating protein pocket flexibility—for example, by conditioning on ensembles of conformations or integrating molecular dynamics snapshots—would better capture the dynamic nature of protein–ligand interactions. Direct integration of pharmacokinetic property predictors, synthetic accessibility constraints and selectivity objectives into the generative process or as guidance signals during inference would enhance practical utility for medicinal chemistry programs. Prospective experimental validation campaigns, in which generated ligands are synthesized and profiled against their intended targets, are essential to assess real-world performance and identify failure modes not captured by computational benchmarks. In addition, systematic evaluation of FLOWR.MULTI on further fragment-based design tasks, including fragment linking and fragment growing, would provide a more comprehensive assessment of its multi-purpose capabilities.

Methods

Here, we provide the methodological foundations and notation for the flow matching framework used in FLOWR. Our approach builds on continuous normalizing flows (CNFs) trained via conditional flow matching (CFM)[19,46], which combines the stable regression objectives of diffusion models with efficient deterministic inference in a simulation-free framework. We specifically employ optimal transport CFM (OT-CFM)[20] to construct simpler, more stable flows by minimizing transport costs between source and target distributions. Extending this with equivariant flow matching[21] allows us to exploit the rotational and translational symmetries inherent to molecular systems, yielding flows with shorter integration paths, improved sampling efficiency and natural incorporation of physical symmetries—essential for generating geometrically valid protein–ligand complexes. Below, we detail each of these components and describe how they are integrated into our model.

Flow matching for continuous data

Flow matching[19,47,48] is a generative modeling framework based on CNFs. A CNF defines a time-dependent flow $\phi_t : \mathbb{R}^d \to \mathbb{R}^d$, where d is the data dimensionality, through the ordinary differential equation (ODE)

$$\frac{d}{dt}\phi_t(x) = v_t(\phi_t(x)), \qquad \phi_0(x) = x, \tag{4}$$

where $v_t : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ is a time-dependent vector field. The flow ϕ_t pushes forward an initial (prior) distribution p_0 to a target (data) distribution p_1 through the induced time-dependent density path p_t. Samples from p_1 are obtained by drawing $x_0 \approx p_0$ and integrating the ODE forward from $t = 0$ to $t = 1$.

Training a CNF via maximum likelihood requires simulating the ODE trajectory and computing the divergence of v_t, which is computationally expensive. Flow matching provides a simulation-free regression objective by regressing a parameterized vector field $v_t^\theta(x_t)$, where θ denotes the learnable model parameters, against a target vector field $u_t(x_t)$

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],x_t\sim p_t(x_t)}||v_t^\theta(x_t) - u_t(x_t)||^2. \quad (5)$$

In practice, both the marginal vector field $u_t(x_t)$ and the marginal probability path $p_t(x_t)$ are intractable. The crucial insight of CFM[19,20] is that equivalent gradients can be obtained by conditioning on data samples $x_1 \approx p_1$. One defines a conditional probability path $p_{t|1}(x_t|x_1)$ with an associated conditional vector field $u_t(x_t|x_1)$, leading to the CFM objective

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],x_1\sim p_1,x_t\sim p_{t|1}(x_t|x_1)}||v_t^\theta(x_t) - u_t(x_t|x_1)||^2. \quad (6)$$

A common choice of conditional path is the Gaussian interpolation

$$p_{t|1}(x_t|x_1) = \mathcal{N}(x_t|t\,x_1 + (1-t)\,x_0,\ \sigma^2 I), \quad (7)$$

where $x_0 \approx p_0$ is a sample from the prior distribution, $\sigma > 0$ is a small noise parameter, and I denotes the identity matrix. The corresponding conditional vector field is $u_t(x_t|x_1) = x_1 - x_0$, defining straight conditional paths between noise and data that can be efficiently integrated during sampling.

Mini-batch optimal transport

The choice of coupling between prior samples x_0 and data samples x_1 substantially affects the geometry of the learned flows. When x_0 and x_1 are coupled independently, that is, $(x_0, x_1) \sim p_0(x_0)\,p_1(x_1)$, the resulting marginal vector field can exhibit crossing paths and unnecessarily complex trajectories. OT-CFM[20] addresses this by replacing the independent coupling with an approximate optimal transport plan $\pi(x_0, x_1)$ that minimizes the expected squared Euclidean transport cost.

Computing the exact OT plan between continuous distributions is intractable in general. OT-CFM therefore approximates the OT coupling at the mini-batch level: given a batch of prior samples $(x_0^1, \ldots, x_0^B)$ and data samples $(x_1^1, \ldots, x_1^B)$, one computes the pairwise cost matrix $M_{ij} = \| x_0^i - x_1^j \|^2$ and solves the resulting discrete OT problem to obtain an optimal assignment within the batch. Training pairs are then formed according to this assignment. This mini-batch OT coupling produces straighter conditional paths, more stable training and faster inference since the learned vector fields require fewer integration steps[20].

Equivariant optimal transport

For molecular systems that exhibit symmetries under the Euclidean group $E(3)$ and the permutation group S_N, standard OT-CFM can yield suboptimal transport plans because it does not account for the invariance of the target distribution under rotations, reflections and atom permutations. Equivariant optimal transport flow matching[21] addresses this by replacing the standard squared Euclidean cost with a symmetry-aware cost function

$$\tilde{c}(x_0, x_1) = \min_{g\in G} \| x_0 - \rho(g)\,x_1 \|^2, \quad (8)$$

where G is the relevant symmetry group, and $\rho(g)$ denotes its action on the molecular configuration. For molecules, G comprises rotations and reflections $O(D)$ combined with atom permutations $S(N)$, where $D = 3$ is the spatial dimension, and N is the number of atoms. The cost function aligns each pair of samples along their symmetry orbits before computing the transport cost.

In practice, jointly optimizing over all rotations and permutations is computationally intractable, so the minimization is approximated sequentially[21]: first, the optimal permutation $\tilde{s} = \arg\ \min_{s\in S_N} \| x_0 - \rho(s)\,x_1 \|^2$ is found using the Hungarian algorithm; then, the optimal rotation $R^* = \arg\ \min_{R\in O(D)} \| x_0 - R\,\rho(\tilde{s})\,x_1 \|^2$ is computed via the Kabsch algorithm. The modified cost matrix is then used within the standard mini-batch OT solver. This equivariant OT procedure produces nearly optimal integration paths even for small batch sizes, leading to shorter paths, reduced integration errors and improved sampling fidelity for molecular systems.

Discrete flow models

For categorical molecular features such as atom types and bond orders, we adopt the discrete flow model framework[26,49], which extends flow matching to discrete state spaces via continuous-time Markov chains (CTMCs). Analogously to continuous flow matching, discrete flow model defines a conditional probability path $p_{t|1}(\cdot|x_1)$ that interpolates between a uniform prior and the data distribution. For a categorical variable x_1 with K possible states, the conditional probability at time t is given by

$$p_{t|1}(x_t|x_1) = t\,\delta(x_t, x_1) + (1-t)\,\frac{1}{K}, \quad (9)$$

where $\delta(x_t, x_1)$ is the Kronecker delta. At $t = 0$, this reduces to a uniform distribution over all K states, whereas at $t = 1$ it concentrates on the data x_1. A neural network learns a data denoiser $p_{1|t}^\theta(\cdot|x_t)$ that predicts the clean data from the noisy sample, trained by minimizing the cross-entropy between the predicted and true posterior distributions. During inference, the denoiser is used within a CTMC integration scheme that progressively drives x_t from the uniform prior toward the data distribution, analogously to the Euler integration used for continuous variables.

The SEMLA architecture and FLOWR extensions

The neural network architecture for FLOWR is based on SEMLA[18], a scalable $E(3)$-equivariant message passing architecture originally proposed in the SEMLAFLOW framework for unconditional 3D molecular generation. SEMLA represents each atom i with invariant features $\mathbf{h}_i \in \mathbb{R}^{d_{\mathrm{inv}}}$ and equivariant features $\mathbf{x}_i \in \mathbb{R}^{3\times d_{\mathrm{equi}}}$, where d_{inv} and d_{equi} denote the invariant and equivariant feature dimensions, respectively. Translation invariance is enforced through zero-centering of equivariant features; combined with equivariant updates throughout the network, the learned density is $E(3)$-invariant.

A key innovation in SEMLA is latent attention: invariant node features are projected into a smaller latent space of dimension $d_l \ll d_{\mathrm{inv}}$ before computing pairwise messages, reducing the computational complexity of the attention mechanism from $\mathcal{O}(N^2 d_{\mathrm{inv}}^2)$ to $\mathcal{O}(N^2 d_l^2)$, where N is the number of atoms. Pairwise messages are computed using a two-layer multilayer perceptron that combines latent invariant features with dot products of equivariant features and are then split into separate invariant and equivariant attention scores. Softmax-normalized attention weights aggregate node features with a variance-preserving scheme.

We extend SEMLA for FLOWR in several ways. First, we incorporate a separate pocket encoder that processes the protein pocket $\mathcal{P}$ independently of the flow time t and the noisy ligand l_t. This encoder uses the same SEMLA layer design and outputs pocket embeddings that are reused across all integration steps during ligand generation, substantially reducing computational cost. Second, we add a cross-attention module within the ligand decoder that takes invariant and equivariant embeddings of $\mathcal{P}$, l_t and optionally the interaction matrix $\mathcal{I}$, following the same latent attention design for efficiency.

Third, we replace the equivariant feed-forward module in SEMLA with a gated variant. For atom i with invariant features $\mathbf{h}_i \in \mathbb{R}^{d_{\mathrm{inv}}}$ and equivariant features $\mathbf{x}_i \in \mathbb{R}^{3\times d_{\mathrm{equi}}}$, the gated feed-forward output is

$$\mathbf{x}_i^{\mathrm{out}} = \mathbf{W}_\theta^2 \hat{\mathbf{x}}_i,\ \text{where}\ \hat{\mathbf{x}}_i = \sigma(\Phi_\theta(\mathbf{h}_i, \| \mathbf{x}_i \|)) \odot \mathbf{W}_\theta^1 \mathbf{x}_i, \quad (10)$$

where σ denotes the elementwise sigmoid function, $\odot$ is elementwise multiplication, $\mathbf{W}_\theta^1, \mathbf{W}_\theta^2 \in \mathbb{R}^{d_{\text{equi}} \times d_{\text{equi}}}$ are learnable weight matrices and Φ_θ is a two-layer multilayer perceptron mapping invariant features and equivariant norms to gating coefficients. This gated module is substantially faster than the original equivariant feed-forward block. Fourth, we pass bond embeddings into the self-attention module on every layer, as opposed to only the first layer, improving molecular validity with negligible impact on inference time.

We parameterize FLOWR with a 4-layer pocket encoder using $d_{\text{inv}}^{\text{enc}} = 256$ and a 12-layer ligand decoder using $d_{\text{inv}}^{\text{dec}} = 384$. Both encoder and decoder use $d_{\text{equi}} = 64$, a latent attention dimension of $d_l = 64$ and $n_{\text{heads}} = 32$ attention heads.

Training and Inference

We train FLOWR to generate ligands conditioned on a given structure. As 3D molecular graphs contain a mixture of continuous and categorical data types, FLOWR jointly generates continuous and discrete distributions. Our approach follows a similar setup to ref. 18. Specifically, we apply the continuous flow matching framework from ref. 20 to learn ligand coordinates and the discrete flow models framework from ref. 26 to learn atom types and bond orders. Ligand formal charges are not learned through a flow but simply predicted by the model.

Model training

Training proceeds by sampling ligand noise $l_0 \approx p_{\text{noise}}$, a ligand, pocket and interaction tuple $(l_1, \mathcal{P}, \mathcal{I}) \sim p_{\text{data}}$, and a time $t \in [0, 1]$. We use Gaussian noise for coordinates and uniform distributions for atom and bond types to create p_{noise}. Writing $l_t = (\mathbf{x}_t, \mathbf{a}_t, \mathbf{b}_t)$ for the coordinate, atom type and bond order components of the noisy ligand, we sample from the conditional probability path $l_t \approx p_{t|1}(l_t|l_1)$ used in ref. 18, defined as

$$t \sim \text{Beta}(\alpha, \beta) \mathbf{x}_t \sim \mathcal{N}(t\mathbf{x}_1 + (1-t)\mathbf{x}_0, \sigma^2) \tag{11}$$

$$\mathbf{a}_t \sim \text{Cat}\left(t\delta(\mathbf{a}_1) + (1-t)\frac{1}{|A|}\right); \mathbf{b}_t \sim \text{Cat}\left(t\delta(\mathbf{b}_1) + (1-t)\frac{1}{|B|}\right). \tag{12}$$

Here Cat$(\cdot)$ denotes the categorical distribution, A and B are the sets of possible values for atom types and bond orders, respectively, and $\delta(\cdot)$ is the one-hot encoding operation applied to each item in a sequence individually. We use values $\alpha = 2.0$, $\beta = 1.0$ and $\sigma = 0.2$ for all FLOWR models.

Following refs. 11,12,17 we train FLOWR to predict l_1 directly by learning the distribution $p_{1|t}^\theta(l_1|l_t, \mathcal{P}, \mathcal{I})$. This leads to the same loss function as SEMLAFLOW[18]—we apply a mean-squared error loss for ligand coordinates and cross-entropy losses for atom types, bond orders and formal charges. When the model is conditioned on both $\mathcal{P}$ and $\mathcal{I}$, the interaction features are provided as additional input to the cross-attention module, enabling the model to attend to both structural pocket information and desired interaction patterns simultaneously. Further implementation details and ablation studies are provided in Supplementary Information section 2.

In addition, during training we apply self-conditioning[50] to improve generation quality. In self-conditioned training, half of the training batches are processed normally, whereas the other half first generate a preliminary prediction of l_1 from the model, which is then detached from the computation graph and provided as additional conditioning input in the subsequent forward pass. For atom and bond types, the conditioning inputs are softmax-normalized probability distributions over the predicted categorical types.

We also apply equivariant optimal transport[21] during training to reduce the transport cost between p_{noise} and p_{data}. For each training pair (l_0, l_1), the coordinate noise $\mathbf{x}_0$ is transformed via $\hat{\mathbf{x}}_0 = f_\pi(\mathbf{x}_0, \mathbf{x}_1)$, where f_π applies the permutation and rotation that minimize the squared error between $\mathbf{x}_0$ and $\mathbf{x}_1$, as described in the 'Equivariant optimal transport' section above. This alignment reduces transport costs and yields straighter, more efficient integration paths during sampling.

Fragment extraction

We extract molecular scaffolds using RDKit's `GetScaffoldForMol` from the `MurckoScaffold` implementation, defining functional groups as all atoms not part of the scaffold and linkers as nonring scaffold atoms. To enable diverse fragment-based learning, we additionally employ RDKit's matched molecular pairs analysis via `FragmentMol` to decompose molecules into chemically meaningful fragments, randomly sampling from these at each training batch. This strategy allows the model to learn scaffold hopping and fragment growing patterns naturally from the data. For interaction-conditional training, we leverage `ProLIF`-derived interaction fingerprints precalculated for all complexes in the SPINDR dataset.

Ligand generation

Given a protein pocket $\mathcal{P}$ and, optionally, a desired interaction matrix $\mathcal{I}$, we can generate samples from the learned data distribution by setting $l_t \leftarrow l_0$, where $l_0 \approx p_{\text{noise}}$ and pushing l_t toward the data distribution by following the learned vector field. Specifically, for molecular coordinates $\mathbf{x}_t$ we follow the vector field $v_t^\theta(\mathbf{x}_t) = \frac{1}{1-t}(\tilde{\mathbf{x}}_1 - \mathbf{x}_t)$, where $\tilde{\mathbf{x}}_1$ is the coordinate component of $\tilde{l}_1 \sim p_{1|t}^\theta(l_1|l_t, \mathcal{P}, \mathcal{I})$. We then integrate the vector field using an Euler solver with step size Δt as follows: $\tilde{\mathbf{x}}_{t+\Delta t} = \mathbf{x}_t + \Delta t v_t^\theta(\mathbf{x}_t)$. For discrete atom and bond types, integration proceeds analogously: at each step, the model predicts the posterior distribution $p_{1|t}^\theta(x_1|x_t)$ over categorical types, and x_t is updated toward the data distribution according to the CTMC integration scheme, where the transition rates are derived from the predicted posteriors and the conditional probability path defined in the 'Discrete flow models' section.

Evaluation. To maintain consistency across models, we used identical random seeds for training, inference and data loading. Moreover, we applied the same sampling and evaluation scripts across all models. For each of the 225 test set targets, we generated 100 ligand samples using a standardized size sampling approach. Specifically, we determined native ligand sizes and applied a uniform sampling scheme, allowing for a size deviation of −25% to +10%. This procedure was performed using the same seed across all models to ensure direct comparability.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The CrossDocked2020 dataset[24] with precomputed data splits can be downloaded via GitHub at https://github.com/pengxingang/Pocket2Mol/tree/main/data. The SPINDR dataset can be downloaded via Zenodo at https://zenodo.org/records/15257565 (ref. 51). Source data for Extended Data Figs. 1–8 is available with this manuscript. The protein-ligand complexes used as case studies were obtained from the Protein Data Bank under accession codes 5YEA and 4EMP. Source data are provided with this paper.

Code availability

The source code used in this study is available under the MIT License via GitHub at https://github.com/jule-c/flowr and via Zenodo at https://doi.org/10.5281/zenodo.18668261 (ref. 52). Model weights can be downloaded via Zenodo at https://zenodo.org/records/15257565 (ref. 53).

References

1. Anderson, A. C. The process of structure-based drug design. *Chem. Biol.* **10**, 787–797 (2003).
2. Anderson, A. C. Structure-based functional design of drugs: from target to lead compound. in *Molecular Profiling* Methods in Molecular Biology (eds Espina, V. & Liotta, L.) 359–366 (Humana Press, 2012); https://doi.org/10.1007/978-1-60327-216-2_23

3. Kitchen, D. B., Decornez, H., Furr, J. R. & Bajorath, J. Docking and scoring in virtual screening for drug discovery: methods and applications. *Nat. Rev. Drug Discov.* **3**, 935–949 (2004).
4. Wang, L. et al. Accurate and reliable prediction of relative ligand binding potency in prospective drug discovery by way of a modern free-energy calculation protocol and force field. *J. Am. Chem. Soc.* **137**, 2695–2703 (2015).
5. Ferreira, L. G., Dos Santos, R. N., Oliva, G. & Andricopulo, A. D. Molecular docking and structure-based drug design strategies. *Molecules* **20**, 13384–13421 (2015).
6. Shoichet, B. K. Virtual screening of chemical libraries. *Nature* **432**, 862–865 (2004).
7. Jumper, J. et al. Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
8. Baek, M. et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871–876 (2021).
9. Abramson, J. et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature* **630**, 493–500 (2024).
10. Hoogeboom, E., Satorras, V. G., Vignac, C. & Welling, M. Equivariant diffusion for molecule generation in 3D. In *Proc. 39th International Conference on Machine Learning* (eds Chaudhuri, K. et al.) https://proceedings.mlr.press/v162/hoogeboom22a.html (PMLR, 2022).
11. Vignac, C. et al. MiDi: mixed graph and 3D denoising diffusion for molecule generation. In *Machine Learning and Knowledge Discovery in Databases: Research Track — European Conference, ECML PKDD 2023* Lecture Notes in Computer Science (eds Koutra, D., Plant, C., Gomez Rodriguez, M., Baralis, E. & Bonchi, F.) 560–576 (Springer, 2023); https://doi.org/10.1007/978-3-031-43415-0_33
12. Le, T., Cremer, J., Noé, F., Clevert, D.-A. & Schütt, K. T. Navigating the design space of equivariant diffusion-based generative models for de novo 3D molecule generation. In *The Twelfth International Conference on Learning Representations* https://openreview.net/forum?id=kzGuiRXZrQ (2024).
13. Luo, S., Guan, J., Ma, J. & Peng, J. A 3d generative model for structure-based drug design. In *35th Conference on Neural Information Processing Systems (NeurIPS 2021)* https://proceedings.neurips.cc/paper_files/paper/2021/file/314450613369e0ee72d0da7f6fee773c-Paper.pdf (2021).
14. Peng, X. et al. Pocket2Mol: efficient molecular sampling based on 3D protein pockets. In *Proc. 39th International Conference on Machine Learning* (eds Chaudhuri, K. et al.) 17644–17655 (2022).
15. Guan, J. et al. 3D equivariant diffusion for target-aware molecule generation and affinity prediction. In *The Eleventh International Conference on Learning Representations* https://openreview.net/forum?id=kJqXEPXMsE0 (2023).
16. Schneuing, A. et al. Structure-based drug design with equivariant diffusion models. *Nat. Comput. Sci.* **4**, 899–909 (2024).
17. Cremer, J., Le, T., Noé, F., Clevert, D.-A. & Schütt, K. T. PILOT: equivariant diffusion for pocket-conditioned de novo ligand generation with multi-objective guidance via importance sampling. *Chem. Sci.* **15**, 14954–14967 (2024).
18. Irwin, R., Tibo, A., Janet, J. P. & Olsson, S. SemlaFlow – efficient 3D molecular generation with latent attention and equivariant flow matching. In *The 28th International Conference on Artificial Intelligence and Statistics* https://openreview.net/forum?id=bee2G6pEh0 (2025).
19. Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M. & Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations* https://openreview.net/forum?id=PqvMRDCJT9t (2023).
20. Tong, A. et al. Improving and generalizing flow-based generative models with minibatch optimal transport. In *Transactions in Machine Learning Ressearch* https://openreview.net/forum?id=CD9Snc73AW (2024).
21. Klein, L., Krämer, A. & Noé, F. Equivariant flow matching. In *37th Conference on Neural Information Processing Systems (NeurIPS 2023)* https://openreview.net/pdf?id=eLH2NFOO1B (2023).
22. Škrinjar, P., Eberhardt, J., Durairaj, J. & Schwede, T. Have protein–ligand co-folding methods moved beyond memorisation? Preprint at *bioRxiv* https://doi.org/10.1101/2025.02.03.636309 (2025).
23. Durairaj, J. et al. Plinder: the protein–ligand interactions dataset and evaluation resource. Preprint at *bioRxiv* https://doi.org/10.1101/2024.07.17.603955 (2024).
24. Francoeur, P. G. et al. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *J. Chem. Inf. Model.* **60**, 4200–4215 (2020).
25. Wang, Y. et al. A workflow to create a high-quality protein–ligand binding dataset for training, validation, and prediction tasks. *Digit. Discov.* **4**, 1209–1220 (2025).
26. Campbell, A., Yim, J., Barzilay, R., Rainforth, T. & Jaakkola, T. Generative flows on discrete state-spaces: enabling multimodal flows with applications to protein co-design. In *Proc. 41st International Conference on Machine Learning* (eds Salakhutdinov, R. et al.) 5453–5512 (PMLR, 2024).
27. Lugmayr, A. et al. RePaint: inpainting using denoising diffusion probabilistic models. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 11461–11471 (IEEE, 2022).
28. Ziv, Y., Imrie, F., Marsden, B. & Deane, C. M. Molsnapper: conditioning diffusion for structure-based drug design. *J. Chem. Inf. Model.* **65**, 4263–4273 (2025).
29. Errington, D., Schneider, C., Bouysset, C. & Dreyer, F. A. Assessing interaction recovery of predicted protein–ligand poses. *J. Cheminform.* **17**, 67 (2025).
30. Harris, C. et al. Benchmarking generated poses: How rational is structure-based drug design with generative models? Preprint at https://arxiv.org/abs/2308.07413 (2023).
31. Wang, R., Fang, X., Lu, Y., Yang, C.-Y. & Wang, S. The PDBbind database: methodologies and updates. *J. Med. Chem.* **48**, 4111–4119 (2005).
32. Banks, J. L. et al. Integrated modeling program, applied chemical theory (impact). *J. Comput. Chem.* **26**, 1752–1780 (2005).
33. Bouysset, C. & Fiorucci, S. Prolif: a library to encode molecular interactions as fingerprints. *J. Cheminform.* **13**, 72 (2021).
34. Schneuing, A. et al. Multi-domain distribution learning for de novo drug design. In *The Thirteenth International Conference on Learning Representations* https://openreview.net/forum?id=g3VCIM94ke (2025).
35. Buttenschoen, M., Morris, G. M. & Deane, C. M. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chem. Sci.* **15**, 3130–3139 (2024).
36. Baillif, B., Cole, J., McCabe, P. & Bender, A. Benchmarking structure-based three-dimensional molecular generative models using genbench3D: ligand conformation quality matters. Preprint at https://arxiv.org/abs/2407.04424 (2024).
37. Eberhardt, J., Santos-Martins, D., Tillack, A. F. & Forli, S. Autodock vina 1.2.0: new docking methods, expanded force field, and python bindings. *J. Chem. Inf. Model.* **61**, 3891–3898 (2021).
38. Bannwarth, C., Ehlert, S. & Grimme, S. Gfn2-xtb—an accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions. *J. Chem. Theory Comput.* **15**, 1652–1671 (2019).

39. Ehlert, S., Stahn, M., Spicher, S. & Grimme, S. Robust and efficient implicit solvation model for fast semiempirical methods. *J. Chem. Theory Comput.* **17**, 4250–4261 (2021).
40. Walters, W. P., Murcko, A. A. & Murcko, M. A. Recognizing molecules with drug-like properties. *Curr. Opin. Chem. Biol.* **3**, 384–387 (1999).
41. Walters, P. Generative molecular design isn't as easy as people make it look. *Practical Cheminformatics Blog* https://practicalcheminformatics.blogspot.com/2024/05/generative-molecular-design-isnt-as.html (2024).
42. Salentin, S., Schreiber, S., Haupt, V. J., Adasme, M. F. & Schroeder, M. PLIP: fully automated protein–ligand interaction profiler. *Nucleic Acids Res.* **43**, W443–7 (2015).
43. Jubb, H. C. et al. Arpeggio: a web server for calculating and visualising interatomic interactions in protein structures. *J. Mol. Biol.* **429**, 365–371 (2016).
44. Liu, Q. et al. Structure-guided discovery of novel, potent, and orally bioavailable inhibitors of lipoprotein-associated phospholipase a2. *J. Med. Chem.* **60**, 10231–10244 (2017).
45. Tso, S.-C. et al. Structure-guided development of specific pyruvate dehydrogenase kinase inhibitors targeting the atp-binding pocket*. *J. Biol. Chem.* **289**, 4432–4443 (2014).
46. Shaul, N. et al. Flow matching with general discrete paths: a kinetic-optimal perspective. In *The Thirteenth International Conference on Learning Representations* https://openreview.net/forum?id=tcvMzR2NrP (2025).
47. Albergo, M. S. & Vanden-Eijnden, E. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations* https://openreview.net/forum?id=li7qeBbCR1t (2023).
48. Liu, X., Gong, C. & Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations* https://openreview.net/forum?id=XVjTT1nw5z (2023).
49. Gat, I. et al. Discrete flow matching. In *38th Conference on Neural Information Processing Systems (NeurIPS 2024)* https://proceedings.neurips.cc/paper_files/paper/2024/file/f0d629a734b56a642701bba7bc8bb3ed-Paper-Conference.pdf (2024).
50. Chen, T., Zhang, R. & Hinton, G. Analog bits: generating discrete data using diffusion models with self-conditioning. In *The Eleventh International Conference on Learning Representations* https://openreview.net/forum?id=3itjR9QxFw (2023).
51. Cremer, J. & Irwin, R. SPINDR (v1.0). *Zenodo* https://doi.org/10.5281/zenodo.15212509 (2025).
52. Cremer, J. & Irwin, R. FLOWR (v1.0). *Zenodo* https://doi.org/10.5281/zenodo.18668261 (2025).
53. Cremer, J. & Irwin, R. Model weights (v1.0). *Zenodo* https://doi.org/10.5281/zenodo.15212509 (2025).

Acknowledgements

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. R.I. acknowledges computational resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725, specifically compute and data allocations 2024/22-1297, 2025/22-1514 and 2025/23-635.

Author contributions

J.C. and R.I. conceptualized the study, analyzed the data, developed the methodology, implemented the computer code and ran the computational experiments. D.-A.C. provided the computational resources required to conduct this study. A.T., J.P.J., S.O. and D.-A.C. supervised and oversaw the research planning and execution. All authors contributed to writing and reviewing the study.

Competing interests

The authors declare no competing interests.

Additional information

Extended data is available for this paper at https://doi.org/10.1038/s43588-026-00998-8.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43588-026-00998-8.

Correspondence and requests for materials should be addressed to Julian Cremer or Ross Irwin.

Peer review information *Nature Computational Science* thanks the anonymous reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Kaitlin McCardle, in collaboration with the *Nature Computational Science* team. Peer reviewer reports are available.

 https://doi.org/10.1038/s43588-026-00998-8

Extended Data Table 1 | Evaluation and comparison of FLOWR on CrossDocked2020

Model	PB-valid↑	Strain↓	Vina score↓	$\text{Vinascore}^{\text{min}}$ ↓	BondA.W1↓	BondL.W1 [10^{-2}]↓	Size	Time (s)↓
POCKET2MOL	0.76 ± 0.39	147.22 ± 61.41	-4.72 ± 1.47	-5.80 ± 1.26	2.04	0.66	17.04 ± 4.11	2320 ± 45
DIFFSBDD	0.38 ± 0.46	519.03 ± 251.32	-2.97 ± 5.21	-4.71 ± 3.30	7.00	0.51	24.85 ± 8.94	160.31 ± 73.30
TARGETDIFF	0.57 ± 0.46	294.89 ± 136.32	-5.20 ± 1.79	-5.82 ± 1.60	7.76	0.42	22.79 ± 9.46	3228 ± 121
DRUGFLOW	0.75 ± 0.39	120.21 ± 73.28	-5.66 ± 1.78	-6.10 ± 1.62	2.11	0.38	21.14 ± 6.81	-
PILOT	0.83 ± 0.33	110.48 ± 87.47	-5.73 ± 1.72	-6.21 ± 1.65	1.75	0.33	22.58 ± 9.77	295.42 ± 117.35
FLOWR	0.92 ± 0.22	87.83 ± 74.30	-6.29 ± 1.56	-6.48 ± 1.45	0.96	0.27	22.28 ± 9.78	12.05 ± 8.01
Test set	0.95 ± 0.21	75.62 ± 57.29	-6.44 ± 2.74	-6.46 ± 2.61	-	-	22.75 ± 9.90	-

Benchmark comparison of the proposed FLOWR model against Pocket2Mol, TargetDiff, DiffSBDD, PILOT and DrugFlow on the CrossDocked2020 test dataset. We follow the conventions in this field and sample 100 ligands per test target, of which there are 100. We evaluate the most expressive metrics, namely PoseBusters-validity, GenBench3D strain energy, AutoDock-Vina scores and the Wasserstein distance of the generated ligands' bond angles (BondA.W1) and bond lengths (BondL.W1) distributions relative to the test set. For all values, we report the mean across ligands and targets and the average standard deviation across targets as subscripts. For all models, we ran all evaluations on the subset of RDKit-valid ligands. Arrows (up, down) indicate that higher or lower values are preferred, respectively.

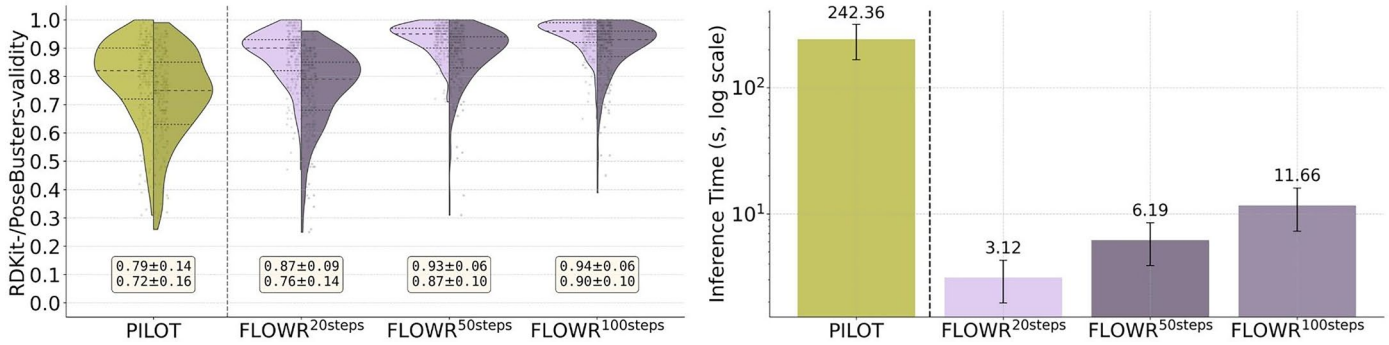
https://doi.org/10.1038/s43588-026-00998-8

Extended Data Table 2 | Evaluation and comparison of PILOT and FLOWR on SPINDR

Model	Strain energy ↓	Vina score ↓	$\text{Vinascore}^{\text{min}}$ ↓	BondA.W1 ↓	BondL.W1 [10^{-2}] ↓	DihedralW1 ↓
PILOT	120.10 ± 71.61	-6.30 ± 0.96	-6.68 ± 1.07	1.82	0.42	5.52
$\text{FLOWR}^{\text{20 steps}}$	134.70 ± 77.58	-6.61 ± 0.98	-6.92 ± 0.96	1.55	0.74	4.67
$\text{FLOWR}^{\text{50 steps}}$	98.47 ± 56.64	-6.83 ± 0.93	-7.13 ± 0.93	1.18	0.51	4.04
$\text{FLOWR}^{\text{100 steps}}$	90.05 ± 52.18	-6.93 ± 0.92	-7.22 ± 0.92	1.08	0.35	3.88
Test set	43.27 ± 41.85	-7.69 ± 2.00	-7.88 ± 2.00	-	-	-

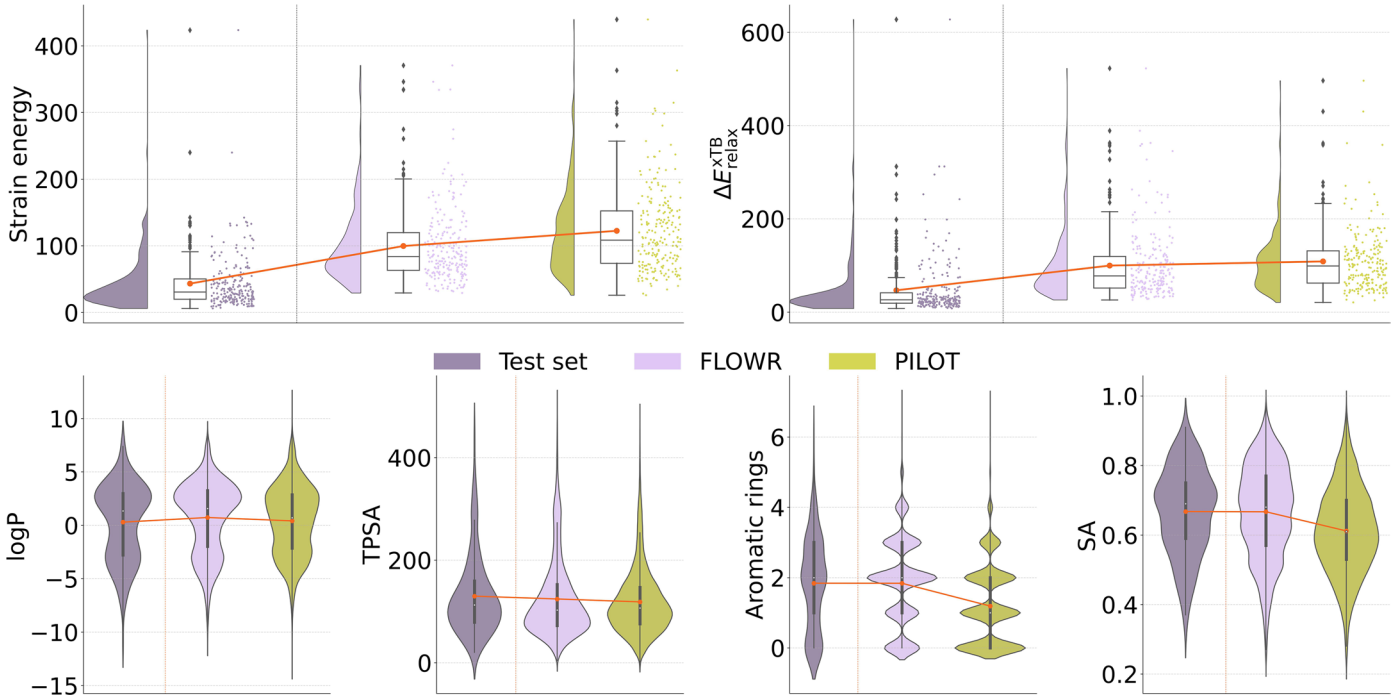
Benchmark comparison of the proposed FLOWR model against the PILOT model using the SPINDR test dataset, which consists of 225 targets. For FLOWR results are reported for inference steps of 20, 50, and 100. For both models, 100 ligands were sampled per target. The evaluation includes strain energy calculated with GenBench3D and AutoDock-Vina scores (kcal/mol). For all values, we report the mean across ligands and targets and the average standard deviation across targets as subscripts. Additionally, we report the Wasserstein distance of the generated ligands' bond angles (BondA.W1), bond lengths (BondL.W1) and dihedral angles (DihedralW1) distributions relative to those in the SPINDR test set. Ligand sizes for all models are sampled uniformly with a -10%/+10% margin around the respective reference ligand size. Arrows (up, down) indicate that higher or lower values are preferred, respectively.

https://doi.org/10.1038/s43588-026-00998-8



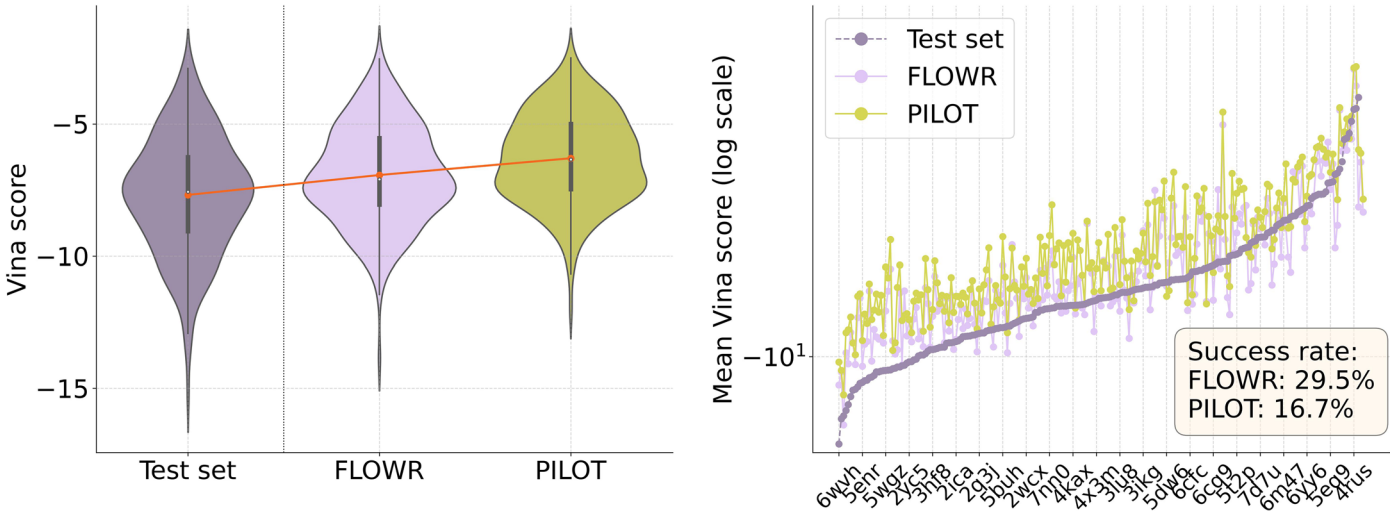
Extended Data Fig. 1 | Comparison of PILOT and FLOWR on RDKit-PoseBusters-validity and inference speed on the SPINDR test set. **(Left)** Violin plots depicting the distribution of per-target RDKit validity (left half) and PoseBusters (PB) validity (right half) rates across all 225 targets in the SPINDR test set for each model. Each target's validity rate is computed as the fraction of valid molecules out of 100 generated ligands. Dashed horizontal lines indicate the lower quartile, median, and upper quartile of the distribution. Individual per-target rates are overlaid as scatter points. Text annotations below each violin report the mean +- standard deviation of per-target validity rates (RDkit, PB). FLOWR is evaluated at 20, 50, and 100 integration steps. Note, both RDKit- and PoseBusters-validity are evaluated on the full set of generated ligands per target. **(Right)** Bar plot comparing mean inference times (in seconds, log scale) for generating 100 ligands per target across models. Bar heights represent the mean wall-clock time, with error bars in black indicating the standard deviation. Exact mean values are annotated above each bar. FLOWR timings are evaluated at 20, 50, and 100 integration steps. All models are evaluated using a single Nvidia H100 GPU.

https://doi.org/10.1038/s43588-026-00998-8



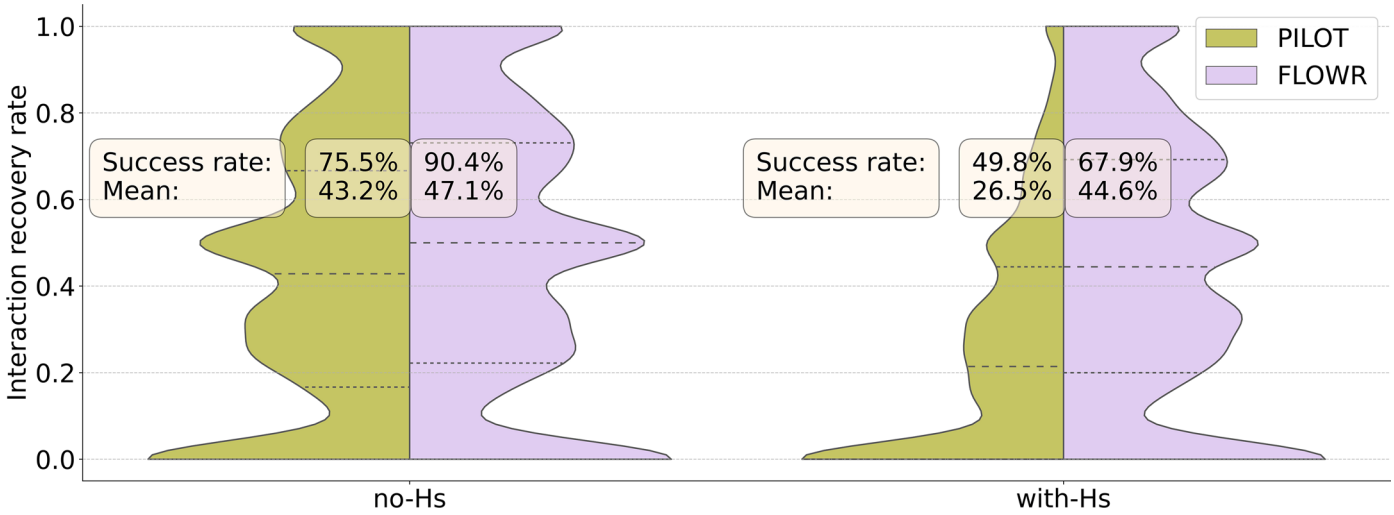
Extended Data Fig. 2 | Comparison of PILOT and FLOWR on molecular properties on SPINDR. (Top left) Raincloud distribution and box-and-whisker plot showing the distribution of per-target mean strain energy across SPINDR test set targets and 100 sampled ligands per target; red connected dots indicate per-model means. **(Top right)** Raincloud distribution and box-and-whisker plot showing xTB relaxation energies (E_{relax}^{xTB}, kcal/mol) using GFN2-xTB with implicit solvation and the Analytical Linearized Poisson–Boltzmann (ALPB) solvation model; red connected dots indicate per-model means. Per-target values are computed by averaging over all ligands generated for each target. **(Bottom row)** Violin plots of per-ligand molecular property distributions for octanol–water partition coefficient (logP), topological polar surface area (TPSA), number of aromatic rings, and synthetic accessibility (SA) score; red connected dots indicate per-model means. Across all panels, test set distributions are shown in purple, FLOWR in pink, and PILOT in green. In all box-and-whisker plots, the centre line represents the median, box bounds represent the 25th and 75th percentiles (interquartile range, IQR), and whiskers extend to 1.5x IQR from the box bounds. Data are presented as mean values +- standard deviation.

https://doi.org/10.1038/s43588-026-00998-8



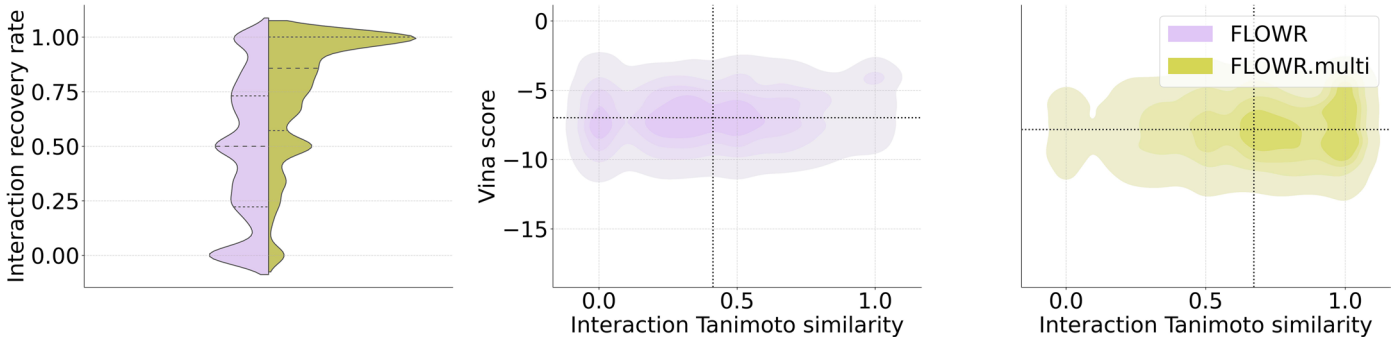
Extended Data Fig. 3 | Comparison of PILOT and FLOWR on AutoDock-Vina scores on SPINDR. (Left) Violin plots of per-target mean Vina scores across the SPINDR test set targets sampling 100 ligands per target with embedded box plots; red connected dots indicate per-model means. In box plots, the center line represents the median, box bounds represent the 25th and 75th percentiles (interquartile range, IQR), and whiskers extend to 1.5x IQR from the box bounds. **(Right)** Per-target mean Vina scores (symmetric log scale) sorted in ascending order by test set score (purple), with targets labeled every 10th position along the x-axis. FLOWR sample scores are denoted by pink, PILOT scores by green scatter points. The inset text box reports the mean per-target success rate, defined as the fraction of generated ligands achieving a Vina score equal to or lower than the corresponding test set mean.

https://doi.org/10.1038/s43588-026-00998-8



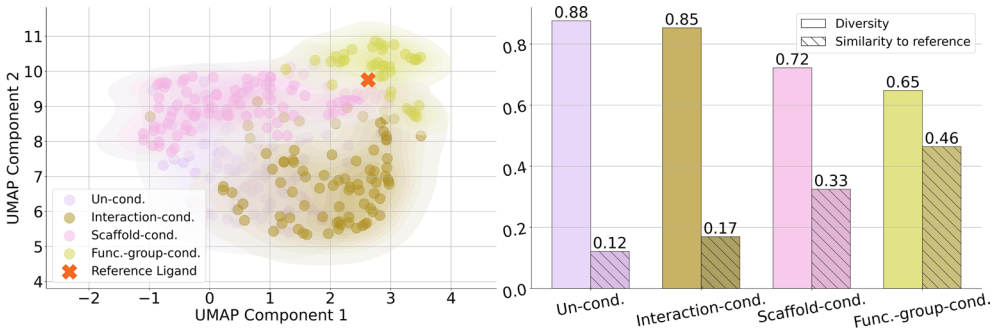
Extended Data Fig. 4 | Comparison of PILOT and FLOWR on interaction recovery on SPINDR. Comparison of PILOT and FLOWR in terms of interaction recovery rate distributions across the SPINDR test set targets sampling 100 ligands per target, shown as violin plots with embedded box plots. Both models are either trained without explicit hydrogens (no-Hs) or with explicit hydrogens (with-Hs). The success rate is the percentage of ligands for which interaction fingerprints could be retrieved. In the embedded box plots, the center line represents the median, box bounds represent the 25th and 75th percentiles (interquartile range, IQR), and whiskers extend to 1.5x IQR from the box bounds. Green color denotes PILOT and pink FLOWR distributions. Interaction fingerprints were calculated using ProLIF.

https://doi.org/10.1038/s43588-026-00998-8



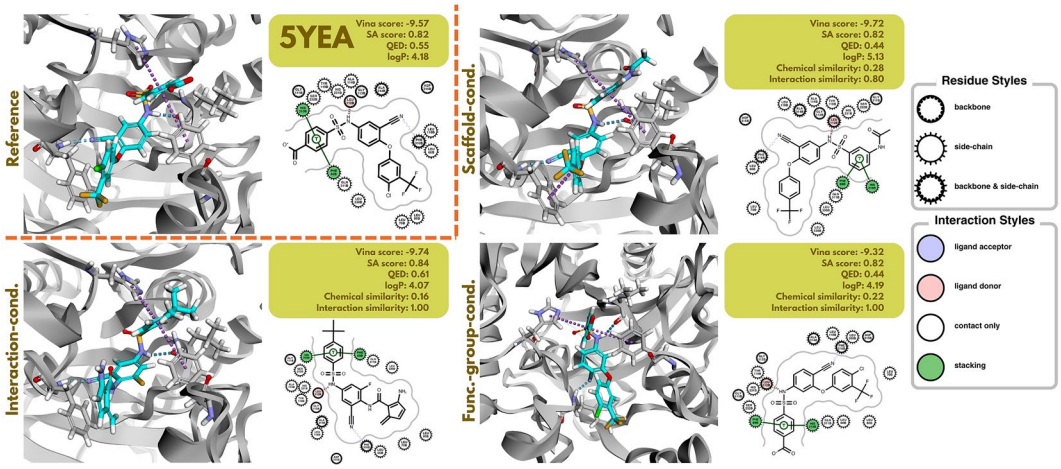
Extended Data Fig. 5 | Comparison of FLOWR and FLOWR.multi on interaction similarity and Vina scores on SPINDR. (Left) Split violin plot of per-ligand interaction recovery rates across the SPINDR test set targets with 100 sampled ligands per target. Dashed lines indicate the lower quartile (25th percentile), median, and upper quartile (75th percentile); the left half corresponds to FLOWR (pink) and the right half to FLOWR.multi (green). **(Right)** Two-dimensional (2D) kernel density estimates of per-ligand protein–ligand interaction fingerprint (PLIF) Tanimoto similarity versus AutoDock Vina docking score for FLOWR and FLOWR.multi, respectively. Dotted lines indicate the respective mean values. Interaction fingerprints were calculated using ProLIF.

https://doi.org/10.1038/s43588-026-00998-8



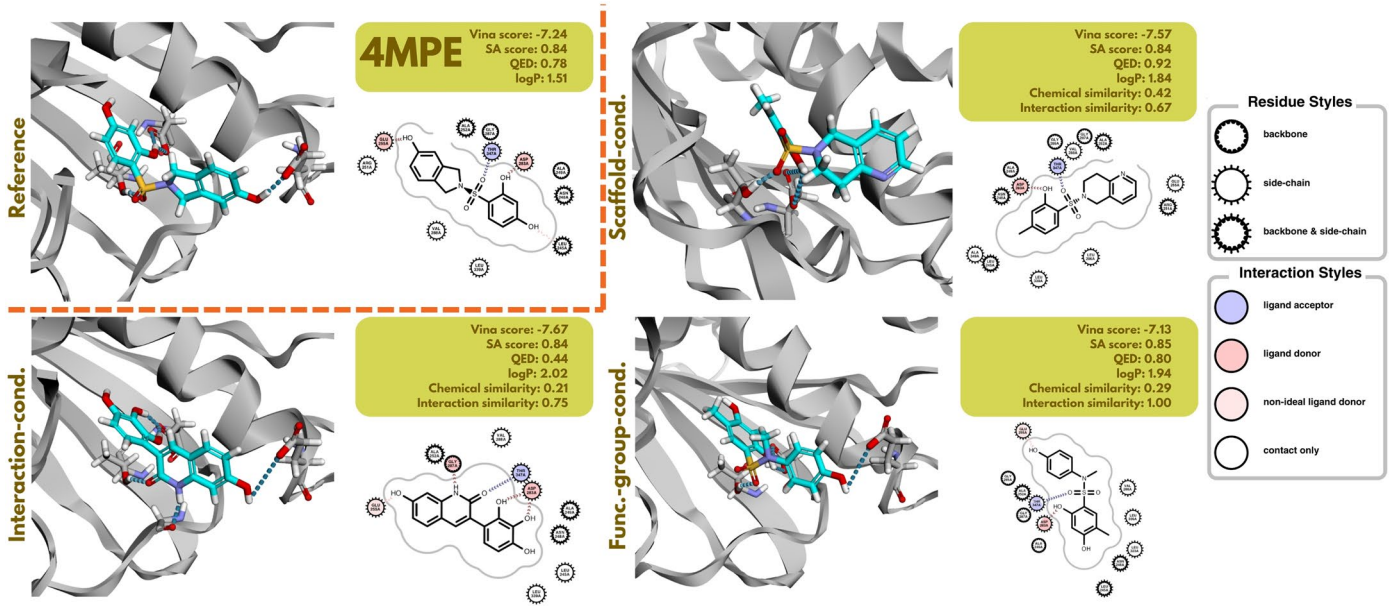
Extended Data Fig. 6 | Evaluation of chemical space coverage on 5YEA with FLOWR.multi. Using the unconditional, interaction-, scaffold-, and functional group-conditional generation modes of FLOWR.multi, 100 ligands were sampled for a randomly selected target from the SPINDR test set (here: PDB-ID 5YEA). Chemical space coverage is visualized with respect to the reference ligand using Morgan fingerprints and Uniform Manifold Approximation and Projection (UMAP). Average diversity of the sampled ligand sets and average similarity to the reference are reported. Each generation mode is represented by a distinct color as indicated in the legend.

https://doi.org/10.1038/s43588-026-00998-8



Extended Data Fig. 7 | Visualization of conditional generation on 5YEA with FLOWR.multi. Using the interaction-, scaffold-, and functional group-conditional generation modes of FLOWR.multi, 100 ligands for a randomly selected target from the SPINDR test set (here: PDB-ID 5YEA) were sampled. Three ligands were selected at random and compared to the reference compound based on Vina score, synthetic accessibility (SA) score, quantitative estimate of drug-likeness (QED), octanol–water partition coefficient (logP), chemical similarity, and interaction similarity. Atom colors: C (cyan/gray), N (blue), O (red), S (yellow), F (ochre), Cl (green), H (white). Interaction fingerprints were calculated using ProLIF. Interaction diagrams were generated using the OpenEye Python Toolkit.

https://doi.org/10.1038/s43588-026-00998-8



Extended Data Fig. 8 | Evaluation of conditional generation on 4MPE with FLOWR.multi. Using the interaction-, scaffold-, and functional group-conditional generation modes of FLOWR.multi, 100 ligands for a randomly selected target from the SPINDR test set (here: PDB-ID 4MPE) were sampled. Three ligands were selected at random and compared to the reference compound based on Vina score, synthetic accessibility (SA) score, quantitative estimate of drug-likeness (QED), octanol–water partition coefficient (logP), chemical similarity, and interaction similarity. Atom colors: C (cyan/gray), N (blue), O (red), S (yellow), F (ochre), Cl (green), H (white). Interaction fingerprints were calculated using ProLIF. Interaction diagrams were generated using the OpenEye Python Toolkit.

nature portfolio

Corresponding author(s): Julian Cremer

Last updated by author(s): Mar 5, 2026

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐ A description of all covariates tested
☒	☐ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Plinder dataset and code (https://github.com/plinder-org/plinder), Schrödinger Maestro, RDKit, Openbabel, Biotite, ProLIF (https://prolif.readthedocs.io/en/stable), xTB (https://xtb-docs.readthedocs.io/en/latest/)
Data analysis	The source code is publicly available at https://github.com/jule-c/flowr. Model weights can be downloaded from Zenodo at https://zenodo.org/records/15257565

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

The CrossDocked2020 dataset with pre-computed data splits can be downloaded from GitHub at https://github.com/pengxingang/Pocket2Mol/tree/main/data. The SPINDR dataset can be downloaded from Zenodo at https://zenodo.org/records/15257565.

Research involving human participants, their data, or biological material

Policy information about studies with human participants or human data. See also policy information about sex, gender (identity/presentation), and sexual orientation and race, ethnicity and racism.

Reporting on sex and gender	Not applicable
Reporting on race, ethnicity, or other socially relevant groupings	Not applicable
Population characteristics	Not applicable
Recruitment	Not applicable
Ethics oversight	Not applicable

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Sample sizes were taken over from previous peer-reviewed studies in this field to ensure comparability
Data exclusions	Data exclusion based on data quality criteria
Replication	The source code to reproduce this study is publicly available at https://github.com/jule-c/flowr. Model weights can be downloaded from Zenodo at https://zenodo.org/records/15257565
Randomization	Not applicable
Blinding	Not applicable

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern
☒	☐ Plants

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Plants

Seed stocks	Not applicable
Novel plant genotypes	Not applicable
Authentication	Not applicable