Supplementary Information (SI) for Digital Discovery.

Supplementary Information (SI)

to the manuscript

Coupled Fragment-Based Generative Modeling with Stochastic Interpolants

Tuan Le*[a], Yanfei Guan[b], Djork-Arné Clevert[a], and Kristof T. Schütt[a]

[a] *Pfizer Research and Development, Machine Learning and Computational Sciences, Friedrichstraße 110, 10177 Berlin, Germany.*

[b] *Pfizer Research and Development, Medicine Design Computational Chemistry, 1 Portland St, Cambridge MA 02139, United States.*

*Correspondence to tuan.le@pfizer.com

A Supplementary Information (SI)

B Dataset

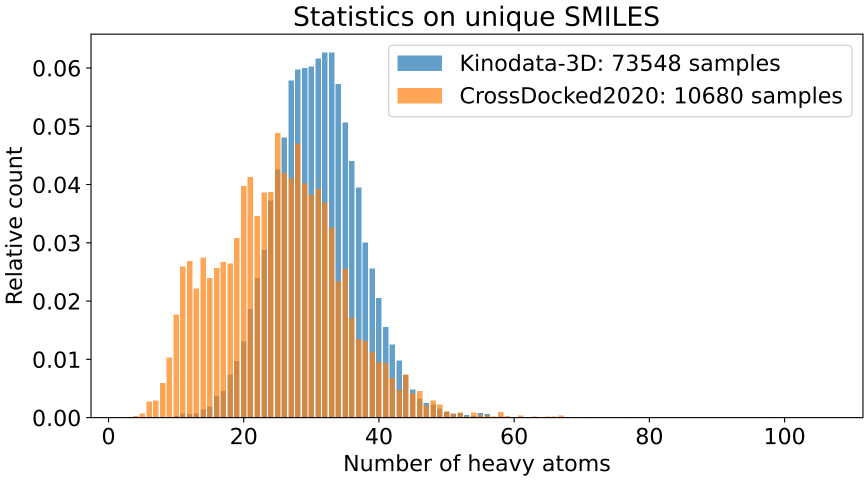


Fig. 1 Heavy atoms count distribution on CrossDocked2020 and Kinodata-3D. The legend also indicates how many unique compounds (SMILES) are available in the respective datasets.

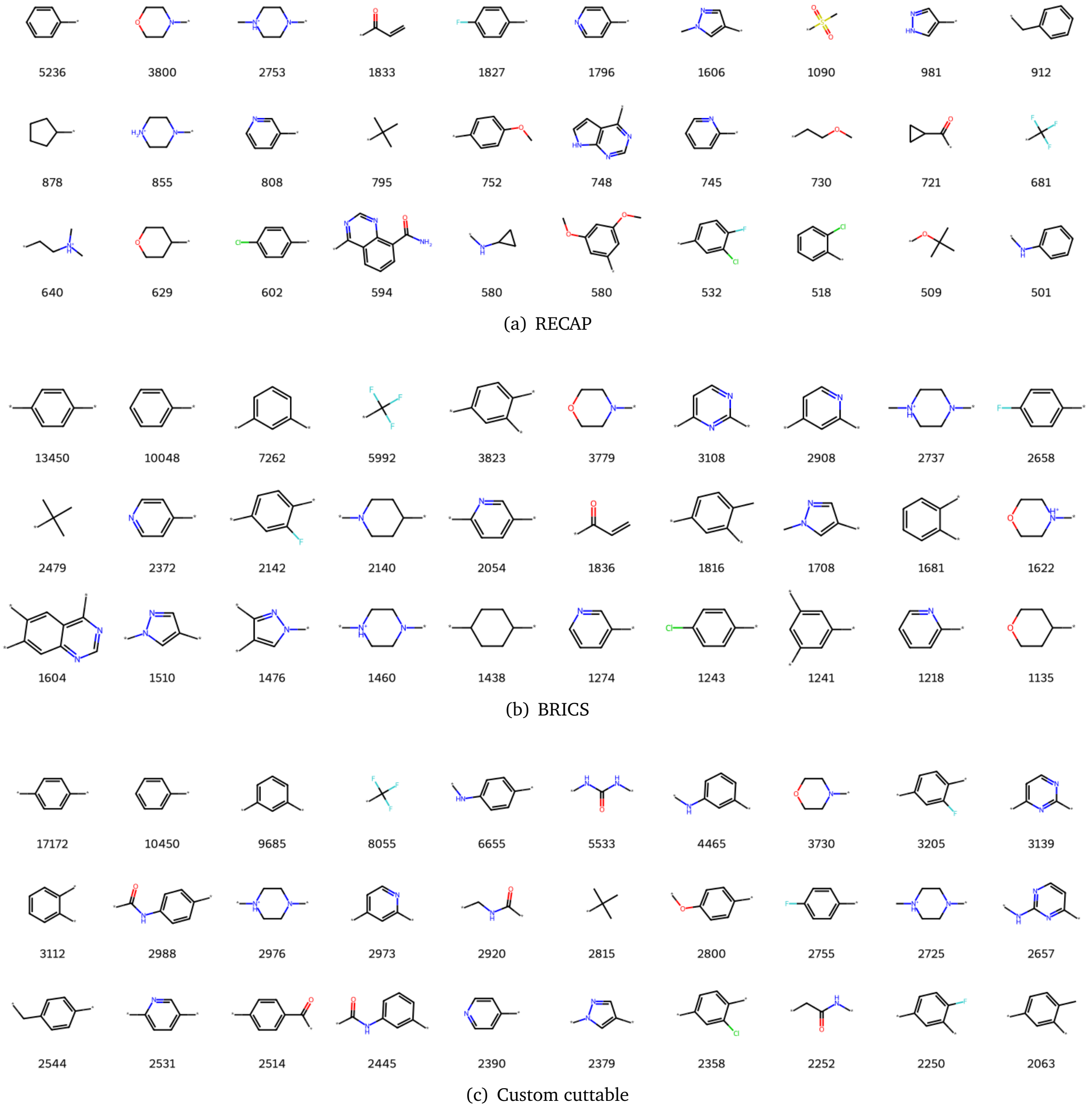
We evaluate our fragment-based generative models on two established protein-ligand datasets: CrossDocked2020 and Kinodata-3D. CrossDocked2020[1] is a widely-used benchmark dataset containing approximately 22.5 million protein-ligand complexes derived from cross-docking experiments, providing diverse binding poses across multiple protein targets. Kinodata-3D[2] represents a more recent and chemically diverse dataset focused on kinase targets, containing high-quality experimental structures with improved template-docking protocols. Both datasets provide essential training and evaluation data for structure-based drug design, with complementary strengths: CrossDocked2020 offers scale and diversity across protein families, while Kinodata-3D provides higher chemical space coverage and more sophisticated structural annotations within the kinase domain. Figure 1 illustrates the molecular size distributions and unique compound counts for both datasets, highlighting their distinct characteristics in terms of ligand complexity and chemical diversity.

B.1 Molecular Fragments

Figure 2 shows the top 30 most frequently occurring fragments from Kinodata-3D according to each fragmentation algorithm. The fragmentation patterns reveal distinct characteristics across methods. RECAP fragments contain only single anchor points (*), indicating that molecules are exclusively fragmented into two disjoint subgraphs. This pattern is particularly suitable for fragment replacement tasks where one molecular region is substituted while maintaining the remaining scaffold.

In contrast, BRICS generates fragments with multiple anchor points, enabling fragmentation into up to four disjoint subgraphs. This capability supports core replacement scenarios where central linkers or scaffolds can be substituted while preserving peripheral fragments. For example, the fragment in the first element of the third row in Figure 2b contains multiple anchor points (*), making it suitable for core replacement applications.

The custom cuttable algorithm produces the most diverse fragment library, generating both single and multiple anchor point fragments with significantly higher chemical diversity. Table 1 quantifies these differences, showing that the custom algorithm generates substantially more total fragments (2,259,910 vs. 360,892 for RECAP) and unique fragments (781,201 vs. 125,962 for RECAP) on Kinodata-3D. The average number of



(a) RECAP

(b) BRICS

(c) Custom cuttable

Fig. 2 Top 30 occuring fragments of Kinodata-3D ligands according to different fragmentation algorithms. The label below each fragment indicates the count how often the particular fragment appears in the dataset. The asteriks (*) indicate the position where the fragment was cut and therefore shows the anchor points for possible attachment(s).

unique fragments per molecule is comparable between RECAP and custom methods (11.01 vs. 11.37), while BRICS produces fewer unique fragments per molecule (5.42), indicating more conservative fragmentation patterns.

This analysis demonstrates that fragmentation algorithm choice significantly impacts the diversity and type of molecular building blocks available for generative modeling, with the custom cuttable approach providing the broadest chemical space coverage for fragment-based drug design applications.

C Stochastic Interpolants

Stochastic interpolants offer a general framework for generative modeling that unifies diffusion and flow matching methods[3]. The approach works by creating probability paths that smoothly transform samples from a simple prior distribution (such as Gaussian noise) to the target data distribution (molecular structures) over continuous time. This eliminates the need for expensive likelihood calculations or complex ODE solving during training. Instead, the method learns directly from pairs of source and target samples.

The framework handles both deterministic flows and stochastic diffusion by controlling whether noise is added during the interpolation process. This flexibility makes it useful for molecular generation, where both approaches have been successful. The interpolation schedule can be adjusted depending on the specific task, which is valuable for applications like fragment-based drug design where different transport strategies may be more effective.

C.1 Distribution

C.1.0.1 Linear Interpolation without Noise For linear interpolation between source $X_0 \sim q(X_0)$ and target $X_1 \sim q(X_1)$, the deterministic interpolant is:

$$I_t(X_0, X_1) = (1-t)X_0 + tX_1, \quad t \in [0,1], \tag{1}$$

with conditional probability path:

$$\rho_t(X_t | X_0, X_1) = (1-t)\delta_{X_0} + t\delta_{X_1}, \tag{2}$$

where δ denotes the Dirac delta function.

C.1.0.2 Adding Gaussian Noise Including a noise term $Z \sim \mathrm{N}(0, I)$ gives the stochastic interpolant:

$$X_t | X_0, X_1 = (1-t)X_0 + tX_1 + \sqrt{\sigma^2 t(1-t)} Z. \tag{3}$$

Table 1 Fragmentation algorithm statistics for different datasets.

Dataset	**Fragmentation Algorithm**		
	RECAP	**BRICS**	**Custom**
Kinodata3D			
Total Fragments	360,892	346,129	2,259,910
Unique Fragments	125,962	13,037	781,201
Avg. Unique per Molecule	11.01	5.42	11.37
CrossDocked2020			
Total Fragments	25,745	28,571	218,121
Unique Fragments	15,796	6,543	131,312
Avg. Unique per Molecule	8.89	5.27	9.16

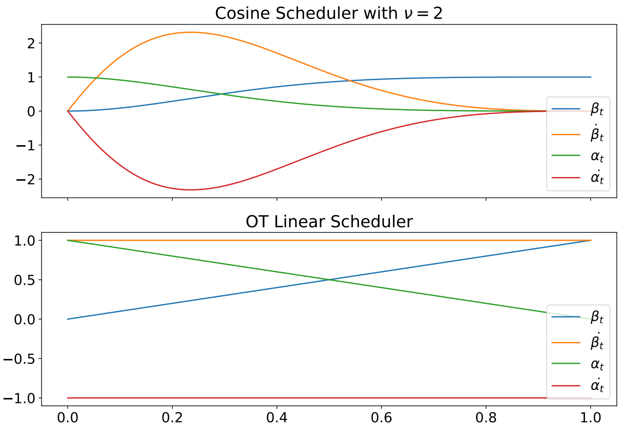


Fig. 3 Interpolation schedulers for $X_t = \alpha_t X_0 + \beta_t X_1$. Cosine scheduler (top) for atomic coordinates; linear scheduler (bottom) for discrete variables (atom types, bonds, charges, hybridization). Time derivatives $\dot{\alpha}_t, \dot{\beta}_t$ show the rate of change during interpolation.

Since X_0 and X_1 are fixed given the conditioning, the mean and variance are:

$$\mathbb{E}[X_t|X_0, X_1] = (1-t)X_0 + tX_1 \tag{4}$$

$$\mathrm{Var}(X_t|X_0, X_1) = \sigma^2 t(1-t)I \tag{5}$$

Therefore, $X_t|X_0, X_1 \sim \mathrm{N}(\mu_t, \Sigma_t)$ where $\mu_t = (1-t)X_0 + tX_1$ and $\Sigma_t = \sigma^2 t(1-t)I$.

Figure 3 shows the linear and cosine schedulers. The cosine scheduler (top) is used for continuous coordinates, while the linear scheduler (bottom) is used for discrete variables in the proposed PILOT-Flow model. The time derivatives $\dot{\alpha}_t$ and $\dot{\beta}_t$ indicate the rate of change during the interpolation process.

C.2 Training and Sampling

For training and sampling on CrossDocked2020 and Kinodata-3D, we use batch size 32 on an H100 80GB GPU on a slurm cluster. We employ $L = 12$ message-passing layers using the denoising architecture presented in Cremer *et al.*[4] with approximately 12.4M trainable parameters. We present the training and sampling procedures in Algorithms 1 and 2. A complete molecular representation includes atomic coordinates, atomic elements, formal charges, hybridization states, and bond types that define atomic connectivity. While our implementation trains on and generates all these modalities to construct complete ligands, we focus the algorithmic description on atomic elements (discrete) and atomic coordinates (continuous) for clarity and brevity.

Algorithm 1 Training PILOT-Flow with Mixed Conditional and De Novo Generation

```
Require: Training dataset $\mathscr{D} = \{(M_i, P_i)\}_{i=1}^{N}$ where $M_i$ are molecules and $P_i$ are corresponding pockets
Require: Flow matching model $f_\theta(M_t, t, P, \tilde{M})$ parameterized by $\theta$
Require: Learning rate $\alpha$, number of epochs $E$
Initialize model parameters $\theta$
for epoch $= 1, 2, \ldots, E$ do
    for batch $(M_1, P) \in \mathscr{D}$ do
        $N_M = |M_1|$  ▷ Number of atoms in molecule
        Sample coordinate noise $X_0 \sim \mathsf{N}(0, I_{N_M \times 3})$
        Sample atom type noise $H_0 \sim \mathsf{U}(1, F)^{N_M}$  ▷ Uniform over $F$ atom types
        Sample conditioning indicator $u \sim \mathsf{U}(0,1)$
        Sample time $t \sim \mathsf{U}(0,1)$
        Broadcast time to nodes $\vec{t} = (t, \ldots, t) \in (0,1)^{N_M}$
        if $u > 0.5$ then
            $\tilde{M} = \emptyset$  ▷ Unconditional generation
        else
            $\tilde{M} = \text{fragment}(M_1)$  ▷ Conditional generation with molecular fragment
        end if
        Set $t_i = 1$ for atoms $i \in \tilde{M}$  ▷ Fix fragment atoms
        Interpolate atomic coordinates:
        $X_t = \vec{\alpha}_t \odot X_0 + \vec{\beta}_t \odot X_1 + \vec{\gamma}_t \odot Z$
        where $Z \sim \mathsf{N}(0, I)$ and $(\vec{\alpha}_t, \vec{\beta}_t)$ follow cosine schedule
        Interpolate atomic elements:
        $H_t = \vec{\alpha}_t \odot H_0 + \vec{\beta}_t \odot H_1$  ▷ Linear interpolation of one-hot vectors
        $H_t \leftarrow \text{Multinomial}(H_t)$  ▷ Sample discrete atom types
        $(\hat{X}_1, \hat{H}_1, \hat{Z}) = f_\theta(X_t, H_t, \vec{t}, P, \tilde{M})$  ▷ Model prediction
        Compute weighted loss:
        $w(\vec{t}) = \text{clamp}\left(\frac{\dot{\vec{\beta}}_t}{\vec{\alpha}_t}, 0.05, 1.5\right)$
        $\mathscr{L} = w(\vec{t}) \odot \left(\lambda_X \|\hat{X}_1 - X_1\|^2 + \lambda_H \text{CE}(\hat{H}_1, H_1)\right) + \|\vec{\gamma}_t \odot \hat{Z} + Z\|^2$
        where $\lambda_X = 3.0, \lambda_H = 1.0$
        Update parameters:
        $\theta \leftarrow \theta - \alpha \nabla_\theta \mathscr{L}$
    end for
end for
return Trained parameters $\theta$
```

Algorithm 2 PILOT-Flow Inference for Conditional and De-novo Molecule Generation

```
Require: Trained flow matching model $f_\theta(M_t, t, P, \tilde{M})$ with parameters $\theta$
Require: Target pocket $P$
Require: Optional molecular fragment $\tilde{M}$ (empty for de-novo generation)
Require: Number of atoms $N_M$ to generate
Require: Integration steps $T$, step size $\Delta t = \frac{1}{T}$
Initialize from noise:
Sample initial coordinates $X_0 \sim \mathsf{N}(0, I)$ of size $N_M \times 3$
Sample initial atom types $H_0 \sim \mathsf{U}(1, F)$ of size $N_M$
Convert $H_0$ to one-hot encoding
Initialize time vector $\vec{t} = (0, \ldots, 0)$ for all atoms
Set up conditional generation:
if $\tilde{M} \neq \emptyset$ then  ▷ Conditional generation
    Fix coordinates and atom types for atoms in $\tilde{M}$
    Set $t_i = 1$ for atoms $i$ belonging to $\tilde{M}$
else  ▷ De-novo generation
    $\tilde{M} = \emptyset$
end if
Euler ODE/SDE Integration:
for $k = 0, 1, \ldots, T-1$ do
    $t = k \cdot \Delta t$
    Update time vector $\vec{t} = (t, \ldots, t)$ for non-fixed atoms
    Predict:
    $(\hat{X}_1, \hat{H}_1, \hat{Z}) = f_\theta(X_t, H_t, \vec{t}, P, \tilde{M})$
    Compute velocity field:
    $\dot{X}_t = \frac{\dot{\vec{\beta}}_t}{\vec{\alpha}_t} \odot (\hat{X}_1 - X_t) + \frac{1}{2} \vec{\gamma}_t \odot \hat{Z}$
    $\dot{H}_t = \frac{\dot{\vec{\beta}}_t}{\vec{\alpha}_t} \odot (\hat{H}_1 - H_t)$
    Euler step:
    if stochastic sampling then
        $X_{t+\Delta t} = X_t + \Delta t \odot \dot{X}_t + \sqrt{\Delta t \odot \vec{\gamma}_t^2} \odot \varepsilon$, where $\varepsilon \sim \mathsf{N}(0, I)$
    else
        $X_{t+\Delta t} = X_t + \Delta t \odot \dot{X}_t$  ▷ ODE
    end if
    $H_{t+\Delta t} \sim \text{Multinomial}(H_t + \Delta t \odot \dot{H}_t)$
    Handle fixed atoms:
    if $\tilde{M} \neq \emptyset$ then
        Keep $X_{t+\Delta t}$ and $H_{t+\Delta t}$ unchanged for atoms in $\tilde{M}$
    end if
end for
return Generated molecule $M = (X_1, H_1)$
```

D Further Evaluations

This section provides additional evaluation metrics and detailed comparisons that support the main findings presented in the manuscript. We include comprehensive training curves, extended molecular descriptor analyses, and supplementary conditional generation results that demonstrate the superior performance of flow matching over diffusion models across diverse evaluation criteria. These extended metrics provide deeper insights into model behavior, convergence properties, and the quality of generated molecular structures beyond the core results discussed in the main text.

D.1 Diffusion vs. Flow Matching: Unconditional de-novo generation

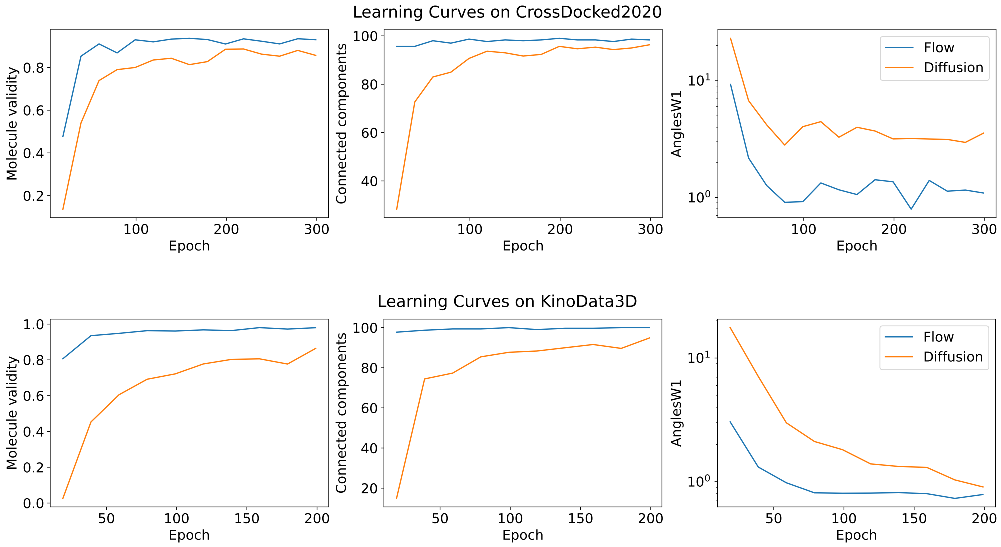


Fig. 4 Training curves comparing flow matching and diffusion models on CrossDocked2020 (top) and Kinodata-3D (bottom) datasets. Left: Molecule validity showing the flow model achieves higher validity rates and faster convergence on both datasets. Center: Connected components indicating the flow matching model generates more structurally coherent molecules with fewer disconnected fragments. Right: Angular divergence (AnglesW1) demonstrating flow matching's superior geometric accuracy with lower divergence from reference bond angles. Flow matching model consistently outperforms Diffusion across all evaluation metrics while requiring fewer training epochs for convergence.

D.1.0.1 3D Metrics

D.2 Additional Metrics on Kinodata-3D Generated Samples

We evaluate steric clashes using PoseCheck[5] and find that the Kinodata-3D training set (104k complexes) averages 4.88 clashes per complex, while the test set shows 3.42 steric clashes. Generated ligands from the flow model achieve fewer steric clashes (3.68), closely matching test set statistics, whereas the diffusion model produces significantly more clashes (6.64 on average). This difference aligns with the inferior PoseBusters validity observed for diffusion models (Table **??**).

Additionally, ligands generated by the flow model show better shape similarity to reference ligands, with a lower shape Tanimoto distance (0.5042) compared to the diffusion model (0.5473). Figure 5 presents a 2D kernel density plot comparing Vina scores and shape Tanimoto distances, demonstrating the superiority of the flow model across both metrics.

D.2.0.1 2D Metrics In Figure 6, we present comprehensive 2D molecular descriptor distributions comparing flow matching and diffusion models across different sampling strategies. The analysis reveals that flow matching with deterministic ODE sampling (Flow-ODE) produces molecular structures that more closely

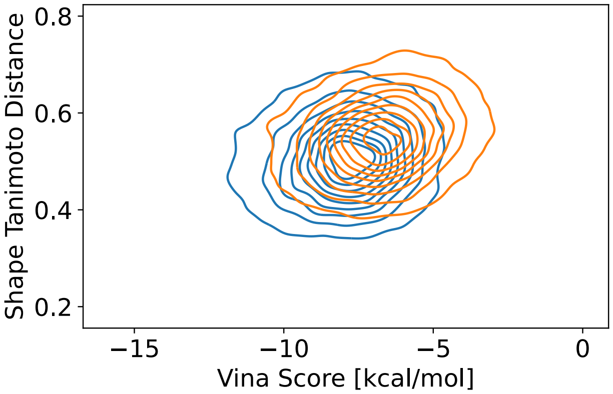


Fig. 5 2D density plot of Vina scores vs. shape Tanimoto distances for diffusion and flow models, showing flow matching achieves lower shape tanimoto distance and better (lower) Vina pose scores.

match the training set statistics compared to diffusion models. However, when stochastic sampling is applied to flow matching (Flow-SDE variants), the generated molecules reveal characteristics that converge toward those produced by diffusion models.

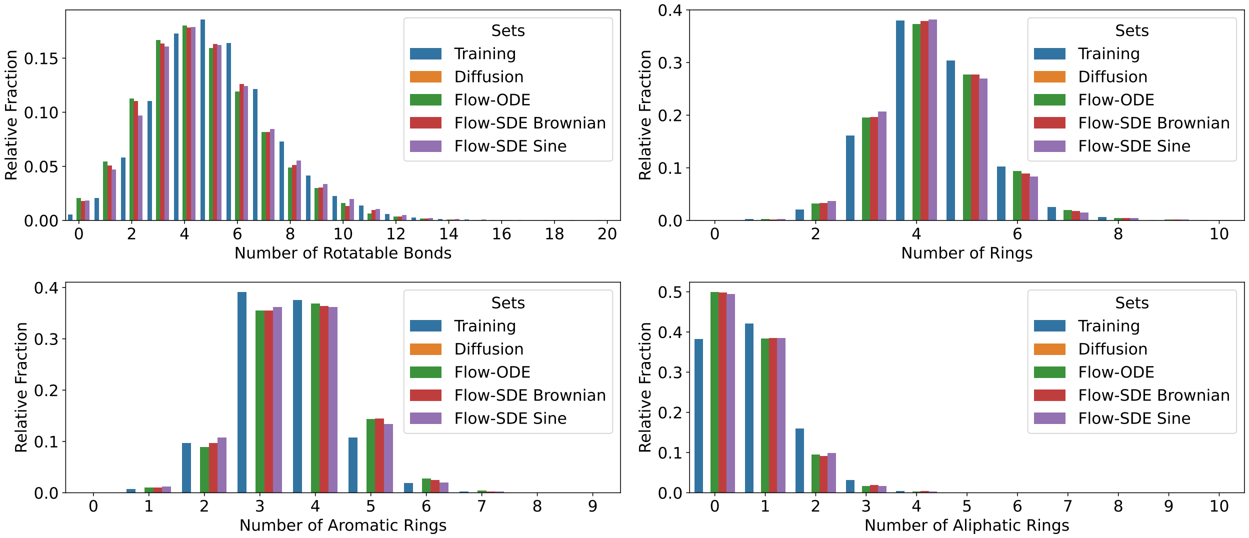


Fig. 6 Empirical distributions of bond and ring features for different sets, indicating that sets from the Flow model resemble more the samples from the Diffusion model when SDE sampling is applied. By including SDE sampling, stochasticity is included for atomic coordinates, such that descriptors such that number of rotatable bonds increase (Flow-SDE-* vs. Flow-ODE) leading to overall less rings in the molecules.

The stochastic sampling variants (Flow-SDE Brownian and Flow-SDE Sine) introduce noise during coordinate generation, resulting in increased rotatable bonds and reduced ring counts compared to deterministic ODE sampling. This effect is quantified in Table 2, which shows Jensen-Shannon divergences (JSD) for various molecular descriptors. Flow-ODE achieves the lowest divergences for ring counts (JSD = 0.0524) and aromatic rings (JSD = 0.0511), indicating preservation of training set characteristics. The stochastic variants show progressively better performance, with Flow-SDE Sine achieving the best overall balance across descriptors (JSD = 0.1195 for rotatable bonds, 0.0687 for rings).

Notably, diffusion models consistently generate molecules with fewer rings (mean = 3.62) and more rotatable bonds (mean = 6.68) compared to the training set (4.4 rings, 5.42 rotatable bonds), reflecting

Table 2 Statistics for Models/Sets.

Model/Set	# of Rotatable Bonds		# of Rings		# of Aromatic Rings		# of Aliphatic Rings	
	Mean	JSD ↓	Mean	JSD ↓	Mean	JSD ↓	Mean	JSD ↓
Training Set	5.42	0.0	4.4	0.0	3.54	0.0	0.86	0.0
Diffusion	6.68	0.1817	3.62	0.2567	2.63	0.3242	0.99	0.0775
Flow ODE	4.54	0.1464	4.3	0.0524	3.65	0.0511	0.65	0.0995
Flow SDE Brownian	4.6	0.1356	4.27	0.0551	3.62	0.0485	0.65	0.0981
Flow SDE Sine	4.75	0.1195	4.22	0.0687	3.57	0.0391	0.66	0.0945

the inherent stochasticity in the diffusion process. The flow matching variants demonstrate more controlled generation, with deterministic sampling preserving structural complexity while stochastic sampling provides a tunable balance between diversity and fidelity to training data.

D.2.0.2 Different Diffusion Coefficients We ablated different stochastic sampling by comparing various diffusion coefficients in the *Flow SDE-Brownian* and *Flow SDE-Sine* cases as listed in Figure 6 and Table 2. The diffusion coefficients for the two variants are defined as

$$\begin{aligned}\gamma_{\text{Brownian}}(t) &= t(1-t)\\ \gamma_{\text{Sine}}(t) &= \sin^2(t\pi)\end{aligned}$$

In the ODE setting, we set $\gamma(t) = 1$ for the deterministic updates (line 20) including the score function, while stochastic updates (line 24) are set to $\gamma(t) = 0$, see Algorithm 2 for details.

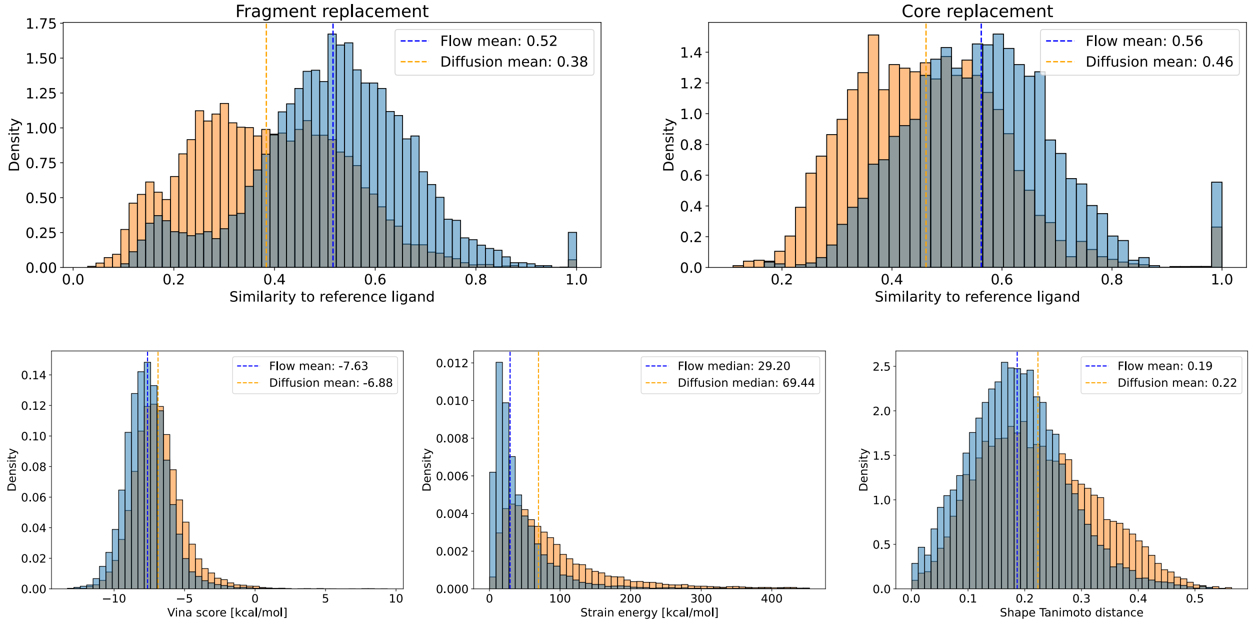


Fig. 7 Conditional generation comparison between flow matching and diffusion models. Top row: 2D Tanimoto similarity distributions to reference ligands for fragment replacement (left) and core replacement (right), showing flow matching achieves higher similarity scores in both tasks. Bottom row: Evaluation metrics including Vina scores (left), strain energies (center), and shape Tanimoto distances (right). Flow matching consistently produces better Vina scores (-7.63 vs. -6.88 kcal/mol), lower strain energies (median 29.20 vs. 69.44 kcal/mol), and improved shape similarity (mean 0.19 vs. 0.22 distance). All metrics demonstrate the superior performance of flow matching for conditional molecular generation tasks.

D.3 Diffusion vs. Flow Matching: Conditional Inpainting Generation on Kinodata-3D

For the conditional inpainting tasks (fragment and core replacement), we compare PILOT-Diffusion against PILOT-Flow by generating ligands for the protein-ligand complexes within the test set.

While both models are conditional models explicitly trained to perform inpainting, we observe that PILOT-Flow is superior to PILOT-Diffusion across multiple 2D and 3D evaluation metrics as shown in Figure 7. For fragment replacement tasks, PILOT-Flow achieves higher similarity to reference ligands (mean = 0.52 vs. 0.38 for diffusion). Similarly, for core replacement tasks, flow matching demonstrates better performance with a mean similarity of 0.56 compared to 0.46 for diffusion models.

The flow matching model also shows significantly improved PoseBusters validity (93.45% vs. 79.29%), which correlates with lower strain energies (29.20 kcal/mol vs. 69.44 kcal/mol) and fewer steric clashes (6.6 vs. 8.84 on average). These results demonstrate that flow matching consistently outperforms diffusion for conditional generation tasks, producing more physically plausible molecular poses with better similarity to reference structures.

E PLK3 Case Study: Detailed Molecular Descriptor Analysis

We evaluated PILOT-Flow on PLK3. This case study uses real experimental data from around 100 internally tested and synthesized compounds to assess whether our generative approach can produce fragments competitive with MedChem-designed molecules and database mining tools like BROOD.

The following analysis examines both basic molecular properties (molecular weight, rotatable bonds, heteroatoms) and drug-like characteristics (TPSA, hydrogen bonding capacity, synthetic accessibility) to assess whether generative approaches can produce chemically reasonable fragments that complement traditional methods. The descriptor statistics are retrieved from each of the 5,000 generated ligands per method.

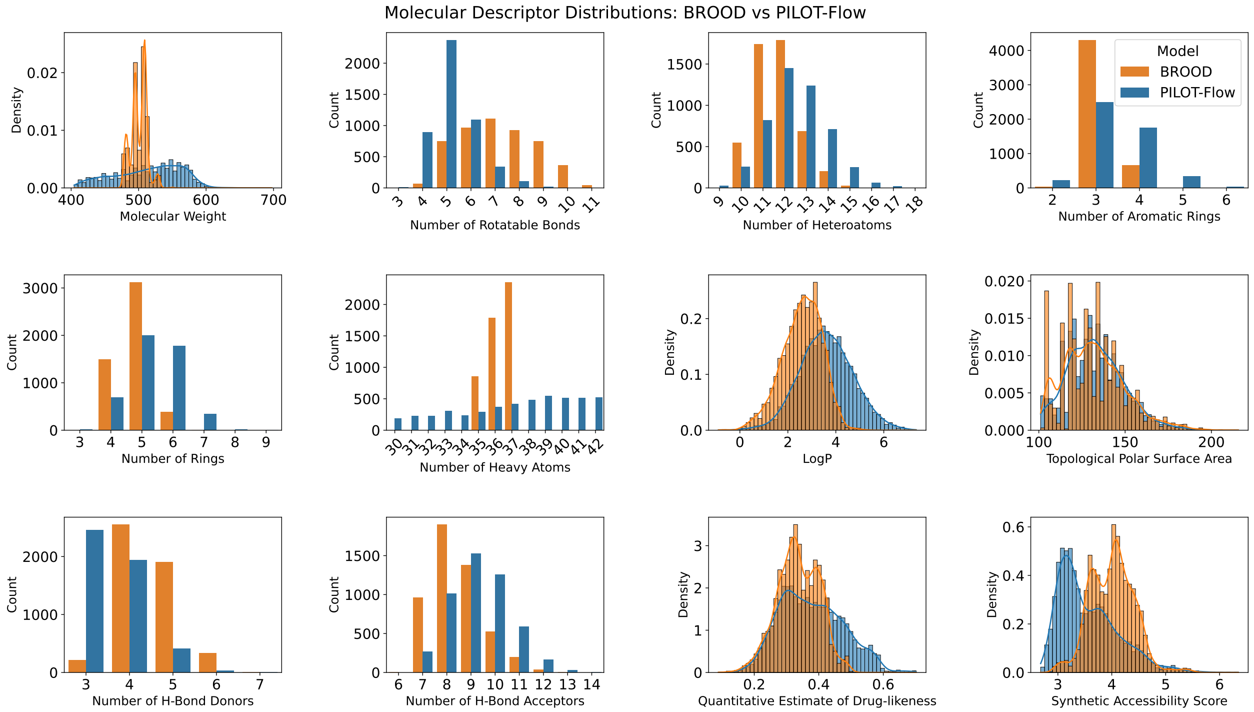


Fig. 8 Distribution of 2d molecular descriptors from the generated sets obtained by BROOD and PILOT-Flow for the PLK3 target.

Table 3 Comparison of BROOD and PILOT-Flow drug-like properties on generated sets for the PLK3 target

Model	MW	Rot. Bonds	Heteroatoms	Arom. Rings	Heavy Atoms	LogP
BROOD	$500.75_{\pm 11.81}$	$7.21_{\pm 1.57}$	$11.67_{\pm 1.01}$	$3.13_{\pm 0.35}$	$36.30_{\pm 0.74}$	$2.59_{\pm 0.83}$
PILOT-Flow	$513.90_{\pm 49.03}$	$5.27_{\pm 0.96}$	$12.49_{\pm 1.35}$	$3.48_{\pm 0.72}$	$37.22_{\pm 3.52}$	$3.60_{\pm 1.11}$

Table 4 Comparison of BROOD and PILOT-Flow drug-like properties on generated sets for the PLK3 target

Model	TPSA	H-Donors	H-Acceptors	Rings	QED	SA
BROOD	$131.98_{\pm 17.54}$	$4.47_{\pm 0.69}$	$8.44_{\pm 1.08}$	$4.78_{\pm 0.58}$	$0.34_{\pm 0.06}$	$4.01_{\pm 0.40}$
PILOT-Flow	$132.94_{\pm 16.42}$	$3.59_{\pm 0.67}$	$9.31_{\pm 1.22}$	$5.37_{\pm 0.84}$	$0.38_{\pm 0.10}$	$3.56_{\pm 0.55}$

Figure 8 shows the descriptor distributions for both methods. PILOT-Flow generates a broader range of molecular weights and fewer rotatable bonds compared to BROOD. We list numerical descriptor values molecular topology and druglikeness in Table 3 and Table 4.

Notes and references

[1] P. G. Francoeur, T. Masuda, J. Sunseri, A. Jia, R. B. Iovanisci, I. Snyder and D. R. Koes, *Journal of Chemical Information and Modeling*, 2020, **60**, 4200–4215.

[2] M. Backenköhler, J. Groß, V. Wolf and A. Volkamer, *Journal of Chemical Information and Modeling*, 2024, **64**, 4009–4020.

[3] M. S. Albergo and E. Vanden-Eijnden, The Eleventh International Conference on Learning Representations, 2023.

[4] J. Cremer, T. Le, F. Noé, D.-A. Clevert and K. T. Schütt, *Chem. Sci.*, 2024, **15**, 14954–14967.

[5] C. Harris, K. Didi, A. R. Jamasb, C. K. Joshi, S. V. Mathis, P. Lio and T. Blundell, *Benchmarking Generated Poses: How Rational is Structure-based Drug Design with Generative Models?*, 2023, `https://arxiv.org/abs/2308.07413`.