Refine Drugs, Don't Complete Them: Uniform-Source Discrete Flows for Fragment-Based Drug Discovery

Benno Kaech* **Luis Wyss†** **Karsten Bogwardt†** **Gianvito Grasso***

Abstract

We present a versatile molecular generative model for drug discovery, supporting de novo generation, fragment-constrained design, and property optimization. Our approach employs a discrete flow model that gradually transforms a uniform source distribution into the target molecular distribution. Unlike continuous-time discrete models or autoregressive approaches, our method decouples the number of sampling steps from the sequence length. This allows us to increase the sampling time resolution independently of the molecular representation, leading to improved generation quality without modifying the model architecture. As a result, our model achieves a new Pareto frontier in the quality–diversity trade-off for de novo generation and establishes a new state of the art in fragment-constrained generation. Finally, we combine a genetic algorithm, where our model performs the crossover step, with proximal policy optimization adapted to our discrete flow setting, achieving state-of-the-art results on the Practical Molecular Optimization benchmark. We further contribute to open science by releasing pretrained checkpoints and code, making our results fully reproducible[3].

1 Introduction

Fragment-based drug discovery (FBDD) is widely used in both academia and industry for its efficient exploration of chemical space [Kirkman et al., 2024]. FBDD relies on fragment-constrained design, in which new candidate molecules are generated by preserving specific substructures, such as active scaffolds or pharmacophores, and modifying surrounding regions to tune properties [Kirsch et al., 2019]. However, FBDD is typically guided by expert-defined heuristics based on chemical intuition, which may limit the exploration of the vast chemical space.

In contrast, *in silico* molecular generation seeks to formalize and automate this intuition, leveraging data-driven generative models to navigate chemical space more systematically. While many such models have emerged as promising candidates to accelerate the drug discovery pipeline [Zeng et al., 2022], their adoption in practice remains limited. A key barrier is that their molecular representations are often poorly aligned with medicinal chemistry workflows, making them difficult to integrate into existing pipelines.

Although graphs provide a natural representation for molecules, generative frameworks tailored to graph-structured data remain limited in performance. For instance, the state-of-the-art AUTOGRAPH Chen et al. [2025] sidesteps direct graph generation by linearizing graphs into sequences via depth-first traversal and then applying next-token prediction. Similarly, the widely used Simplified Molecular Input Line Entry System (SMILES) [Weininger, 1988] encodes chemical graphs as ASCII strings through a graph traversal, typically a depth-first walk of a spanning tree, with

*InVirtuoLabs

†Max Planck Institute of Biochemistry

[3] `https://github.com/invirtuolabs/InVirtuoGen_results`

The NeurIPS 2025 Workshop on AI Virtual Cells and Instruments: A New Era in Drug Discovery and Development (AI4D3 2025), San Diego, California, USA, 2025.

additional annotations for branches and cycles. However, SMILES was not originally designed with generative modeling in mind. However, both approaches disrupt chemically meaningful substructures during linearization, making them poorly suited for scaffold- or fragment-constrained design in practical drug discovery. To alleviate these issues, fragment-based representations have been proposed, which preserve chemically meaningful substructures and enable more direct control over scaffold retention and fragment assembly [Jinsong et al., 2024]. We build on this motivation with InVirtuoGen, a continuous-time discrete flow model [Campbell et al., 2024, Gat et al., 2024] designed for fragment-based molecular generation.

Our main contributions are:

- A discrete flow model for fragmented SMILES that, unlike masked or autoregressive training, is optimized over all sequence positions simultaneously. This decouples the number of sampling steps from sequence length, providing fine-grained control over time discretization.
- Competitive performance on both de novo and fragment-constrained molecule generation tasks.
- We combine GA with PPO adapted to discrete flows for the Practical Molecular Optimization benchmark (PMO), outperforming previous baselines [Gao et al., 2022a, Lee et al., 2024, 2025a, Jensen, 2019, Kim et al., 2024].

2 Related Work

While numerous generative models have been proposed for small molecule drug design [Jensen, 2019, Olivecrona et al., 2017, Morrison et al., 2024, Gao et al., 2022b, Nigam et al., 2020, Bou et al., 2024], few are explicitly designed for fragment-level control. In this work, we focus on approaches that operate on sequential fragment-based representations. Several alternative approaches construct molecules via graph-based operations on substructures, for example by adding or deleting fragments through Markov Chain Monte Carlo sampling [Xie et al., 2021], using graph-based VAEs conditioned on identified substructures [Jin et al., 2020, 2018, Maziarz et al., 2024], or applying graph-based genetic algorithms [Jensen, 2019, Tripp and Hernández-Lobato, 2023] (GA). Although these models encode domain-specific priors, they often suffer from poor scalability and limited generalization beyond known chemical regions, in part due to their reliance on graph operations and discrete mutation strategies. By contrast, generative models operating on linear sequential representations of fragments, such as SAFE-GPT [Noutahi et al., 2023] and GenMol [Lee et al., 2025a], offer a more scalable and expressive alternative, and form the primary baselines for our work.

Autoregressive Models Autoregressive approaches, such as SAFE-GPT [Noutahi et al., 2023], generate molecular sequences token by token in a fixed left-to-right order. This ordering is arbitrary with respect to molecular structure, which is inherently unordered.

Masked Diffusion Models Masked discrete diffusion models, such as GenMol [Lee et al., 2025a], iteratively unmask tokens starting from a fully masked input. While predictions are produced for the entire sequence at each step, the training objective evaluates errors only on the masked positions. As a consequence, once a token is unmasked during sampling, it is treated as fixed and no longer updated. This introduces a fundamental limitation: the number of sampling steps is bounded by the number of initially masked tokens. Without remasking heuristics, this tightly couples the sampling schedule to sequence length.

Our Approach Unlike autoregressive models, which are trained to predict only the next token given previous tokens, or masked diffusion models, which are trained to predict only masked positions, our approach is trained to accurately predict all positions simultaneously at every denoising step. This training objective allows the model to learn coordinated refinements across the entire molecule rather than merely completing partial structures. Hence, aligning with our core principle: *refine drugs, don't just complete them*.

2.1 Discrete Flow Models

We adopt the discrete flow model framework of Gat et al. [2024], where the goal is to transform samples from a source distribution $X_0 \sim p$ into samples from a target distribution $X_1 \sim q$. Training data consists of interpolation pairs (X_0, X_1), sampled independently from the source and target. We choose the linear scheduler κ_j^t of Gat et al. [2024], resulting in following the probability path:

$$p_t(x^i \mid x_0, x_1) = (1-t)\,\delta_{x_0}(x^i) + t\,\delta_{x_1}(x^i), \quad t \in [0,1].$$

During sampling, each token X_t^i is updated independently according to the discrete-time Markov update

$$X_{t+h}^i \sim \delta_{X_t^i}(\cdot) + h\,u_t^i(\cdot, X_t), \tag{1}$$

where u_t is the *probability velocity* and $h > 0$ is the step size. Following Gat et al. [2024], u_t must satisfy the validity constraints

$$\sum_{x^i \in [d]} u_t^i(x^i, z) = 0, \quad u_t^i(x^i, z) \geq 0 \text{ for all } i \text{ and } x^i \neq z.$$

For our scheduler choice, the training objective becomes

$$\mathcal{L}(\theta) = -\mathbb{E}_{t\sim U(0,1),(X_0,X_1),X_t} \frac{1}{1-t^2} \sum_i \log p_{1|T}(X_1^i \mid X_t), \tag{2}$$

where $p_{1|T}$ denotes the model prediction and the sum is over the sequence. The time dependent loss weighting term was inspired by Sahoo et al. [2024] and places greater emphasis on later timesteps, encouraging higher accuracy near the end of the trajectory. As a backbone model, we use a diffusion transformer [Peebles and Xie, 2022] to parameterize $p_{1|T}$, leveraging its bidirectional self-attention to capture long-range dependencies between fragments while predicting the target token distribution at each position. The training hyperparameters are listed in App. A.1.

2.2 Fragmented SMILES Notation & Preprocessing

Our representation is based on and extends the Sequential Attachment-Based Fragment Embedding (SAFE) framework of Noutahi et al. [2023] by encoding molecules as sequences of fragment blocks with explicit attachment points, improving readability and direct control over molecular substructures. A limitation is that stereochemical information is discarded, since stereochemistry is a property of the molecule and is therefore not defined for individual fragments.

To produced chemically meaningful fragments, the molecules are decomposed using the revised BRICS algorithm [Degen et al., 2008], with bond break locations marked by attachment-points of the form $[i*]$, where i enumerates the broken bonds. In Fig. 1, we illustrate the difference between the SMILES, SAFE and our notation.

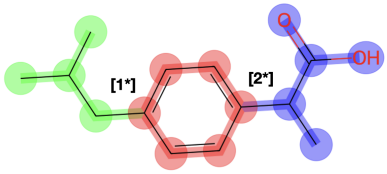
To remove any implicit structural bias, the resulting fragments are randomly shuffled rather than ordered by their attachment point in the original molecule.

The number of fragments per molecule directly affects sequence length because each attachment point and inter-fragment space introduces extra tokens. We set the maximum number of fragments per molecule to seven. The resulting fragmented SMILES strings are tokenized at the character level, yielding a vocabulary of 204 tokens.

3 Experiments

We evaluate InVirtuoGen on three central tasks in molecular design:

- *De novo generation*, where the goal is to generate molecules from scratch that are diverse, synthesizable, and drug-like;
- *Fragment-constrained design*, which requires constructing molecules that respect predefined substructures such as scaffolds or pharmacophores;
- *Target property optimization*, evaluated on the PMO benchmark [Gao et al., 2022a], which assesses both sample efficiency and achieved oracle score.



SMILES: CC(Cc1ccc(cc1)C(C(=O)O)C)C
SAFE: c12ccc3cc1.C3(C)C(=O)O.CC(C)C2
Ours: [1*]c1ccc([2*])cc1 [2*]C(C)C(=O)O [1*]CC(C)C

Figure 1: Comparison between SMILES, SAFE, and our notation for the same molecule. Our notation preserves fragment integrity while providing explicit attachment point numbering that facilitates bidirectional modeling of molecular structure.

For comparability, we train on the same datasets as GenMol [Lee et al., 2025b] and SAFE-GPT [Noutahi et al., 2023]: ZINC [Sterling and Irwin, 2015] and UniChem [Chambers et al., 2013], containing roughly one billion molecules.

Similarly to GenMol [Lee et al., 2025a], the non-autoregressive formulation enables bidirectional attention, making the model well suited for the inherently unordered molecular data representation. However, this also implies that we explicitly decouple sequence length from the generation process by factorizing the output distribution as

$$p_\theta(\mathbf{x}) = p(n)\, p_\theta(\mathbf{x} \mid n), \tag{3}$$

where $p(n)$ models the sequence length. Our base distribution is chosen to be the empirical length distribution of ZINC250k, a curated subset of ZINC [Sterling and Irwin, 2015] containing synthesizable, drug-like compounds.

3.1 De Novo Generation

For generated molecules to be relevant in drug discovery, they must be diverse, synthesizable, and drug-like. We evaluate these properties using the following metrics, consistent with prior work on fragmented SMILES generation:

- **Validity**: fraction of generated sequences that correspond to valid SMILES.
- **Uniqueness**: fraction of valid molecules that are unique.
- **Diversity**: average Tanimoto distance between Morgan fingerprints of generated molecules.
- **Quality** (introduced by Lee et al. [2025a]): fraction of valid, unique, drug-like, and synthesizable molecules. Drug-likeness is defined through a cut on the quantitative estimate of drug-likeness [Bickerton et al., 2012] (QED) ≥ 0.6, and synthesizability via the synthetic accessibility [Ertl and Schuffenhauer, 2009] (SA) with SA ≤ 4.

Quality and diversity form the primary criteria for evaluating generative performance but are often in tension: higher diversity typically lowers validity, synthesizability or drug-likeness, all reflected by a lower quality metric. We control this trade-off using two parameters: the softmax temperature T and a noise scaling parameter r, which modulates the Gumbel noise added to predicted logits. During generation, r is damped over time by a factor $(1 - t)$, encouraging exploration early in the trajectory and more deterministic refinement toward the end. Similarly, the temperature is annealed during the sampling trajectory to a minimum of 0.25.

Empirically, we find that sampling directly from the model's predicted token-wise distribution

$$X^i_{t+h} \sim \hat{p}^i_t(X_t) \tag{4}$$

yields substantially better results on these metrics than sampling according to Eq. 1.

Results In Figure 2 we present and compare our results to other state-of-the-art fragment-based generative models. Increasing the simulation granularity (i.e., using smaller timestep sizes h) consistently improves both quality and diversity, pushing the Pareto frontier beyond the one achieved by GenMol. This demonstrates the strength of our model in generating diverse yet high-quality molecules from scratch. We provide non-curated samples in the Appendix C.1.

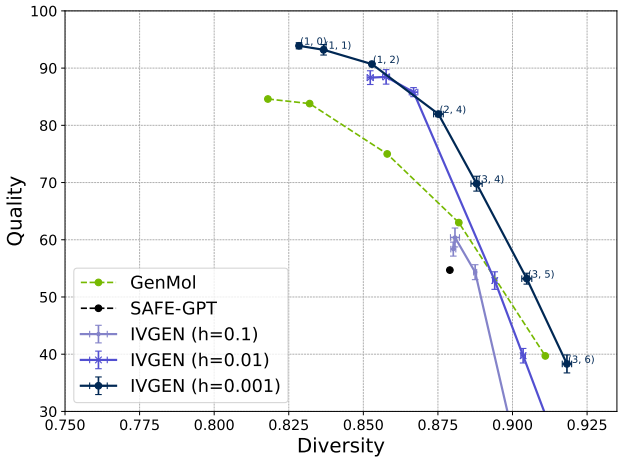


Figure 2: Quality–diversity trade-off for GenMol, SAFE-GPT (single point, as no quality–diversity scan data is available), and our model (IVGEN) at different simulation time granularities ($h \in \{0.1, 0.01, 0.001\}$). Curves correspond to varying sampling noise (T, r), where T is the softmax temperature and r is the Gumbel noise scale. InVirtuoGen consistently achieves a superior Pareto frontier, with the largest gains at fine granularity ($h = 0.001$), outperforming all baselines across the full quality–diversity spectrum.

3.2 Fragment-Constrained Generation

A common task in drug discovery is to complete a molecule given one or more fragments it should contain. We follow the benchmark of Noutahi et al. [2023], which uses fragments from ten known drugs, and evaluate five subtasks: linker design (connecting two or more terminal fragments with a feasible linker), scaffold morphing (modifying the core scaffold while preserving pharmacophoric features), motif extension (growing a fixed motif with new substituents), scaffold decoration (attaching functional groups at predefined positions), and superstructure generation (assembling multiple fragments into a coherent larger molecule). In our notation, as in SAFE, linker design and scaffold morphing yield identical prompts and thus identical results. Note that both next-token prediction models and masked discrete diffusion models such as GenMol can be prompted in a straightforward manner by using a prefix. In contrast, our model starts from a uniform source and predicts a complete new sequence at every timestep. Therefore, fragment-constrained prompts must be explicitly enforced throughout the trajectory by overwriting the corresponding positions at each step, ensuring all generated molecules remain consistent with the provided fragments.

Results In Table 1 we report results for the fragment-constrained generation tasks. On average across all tasks, our model produces more diverse, drug-like and synthesizable molecules. Additionally, for GenMol, the generation parameters were tuned separately for each task, whereas we used a single fixed parameter setting across all tasks. . We provide non-curated samples in the Appendix C.2.

3.3 Target Property Optimization

Target property optimization leverages our model's ability to refine partially correct input sequences. Our approach combines a genetic algorithm and reinforcement learning based on Proximal Policy Optimization [Schulman et al., 2017], adapted to the discrete flow setting.

Table 1: Performance across five fragment-constrained generation tasks, averaged over three random seeds: motif extension, linker design, superstructure generation, scaffold morphing, and scaffold decoration. The results for GenMol [Lee et al., 2025a] and SAFE-GPT [Noutahi et al., 2023] are taken from Lee et al. [2025a].

Task	Method	Diversity	Quality	Uniqueness	Validity
Motif Extension	SAFE-GPT	0.56 ± 0.003	18.60 ± 2.100	66.80 ± 1.200	$\mathbf{96.10 \pm 1.900}$
	GenMol	$\mathbf{0.62 \pm 0.002}$	30.10 ± 0.400	77.50 ± 0.100	82.90 ± 0.100
	InVirtuoGen	$\mathbf{0.62 \pm 0.001}$	$\mathbf{39.17 \pm 0.665}$	$\mathbf{96.17 \pm 0.308}$	68.43 ± 1.212
Linker Design	SAFE-GPT	$\mathbf{0.55 \pm 0.007}$	$\mathbf{21.70 \pm 1.100}$	$\mathbf{82.50 \pm 1.900}$	76.60 ± 5.100
	GenMol	$\mathbf{0.55 \pm 0.002}$	$\mathbf{21.90 \pm 0.400}$	$\mathbf{83.70 \pm 0.500}$	$\mathbf{100.00}$
	InVirtuoGen	0.51 ± 0.002	$\mathbf{21.97 \pm 1.053}$	$\mathbf{83.89 \pm 1.781}$	60.67 ± 1.642
Scaffold Morphing	SAFE-GPT	0.51 ± 0.011	16.70 ± 2.300	70.40 ± 5.700	58.90 ± 6.800
	GenMol	$\mathbf{0.55 \pm 0.002}$	$\mathbf{21.90 \pm 0.400}$	$\mathbf{83.70 \pm 0.500}$	$\mathbf{100.00}$
	InVirtuoGen	0.51 ± 0.002	$\mathbf{21.97 \pm 1.053}$	$\mathbf{83.89 \pm 1.781}$	60.67 ± 1.642
Superstructure Design	SAFE-GPT	0.57 ± 0.028	14.30 ± 3.700	83.00 ± 5.900	95.70 ± 2.000
	GenMol	0.60 ± 0.009	$\mathbf{34.80 \pm 1.000}$	83.60 ± 1.000	$\mathbf{97.50 \pm 0.900}$
	InVirtuoGen	$\mathbf{0.73 \pm 0.001}$	28.63 ± 1.226	$\mathbf{99.67 \pm 0.055}$	74.90 ± 0.432
Scaffold Decoration	SAFE-GPT	0.57 ± 0.008	10.00 ± 1.400	74.70 ± 2.500	$\mathbf{97.70 \pm 0.300}$
	GenMol	$\mathbf{0.59 \pm 0.001}$	31.80 ± 0.500	82.70 ± 1.800	96.60 ± 0.800
	InVirtuoGen	0.56 ± 0.004	$\mathbf{35.90 \pm 0.942}$	$\mathbf{88.21 \pm 0.546}$	90.47 ± 0.754
Average	SAFE-GPT	0.55 ± 0.006	16.26 ± 1.031	75.48 ± 1.773	85.00 ± 1.788
	GenMol	0.58 ± 0.002	28.10 ± 0.263	82.24 ± 0.436	$\mathbf{95.40 \pm 0.242}$
	InVirtuoGen	$\mathbf{0.59 \pm 0.001}$	$\mathbf{29.53 \pm 0.600}$	$\mathbf{90.37 \pm 0.432}$	71.03 ± 0.539

Genetic Algorithm We follow the standard crossover–mutation scheme, with mutation operations adapted from Jensen [2019] and crossover handled by our model. Both parents are first decomposed using our fragmentation rule and the offspring is formed by taking all but the last fragments from the former parent, which are then concatenated with the last fragment of the second parent to form an initial, most-likely invalid, sequence. This sequence is then refined by simulating the trajectory from t_{start} with step size $h = 0.01$.

Reinforcement Learning We adapt Proximal Policy Optimization [Schulman et al., 2017] to fine-tune the discrete flow policy. At each update, we: (i) sample partial trajectories by selecting a small set of timesteps t for each molecule x_1, apply the noise schedule to obtain x_t and a corresponding noise mask, and approximate the joint likelihood by averaging over these timesteps; (ii) compute the policy log-probability $\log \pi_\theta(x_0 \mid x_t, t)$ and sequence entropy using only the noised positions; (iii) form the advantage $A = (r - \bar{r})/(\sigma_r + \epsilon)$ by standard scaling the oracle score r, multiplying negative advantages by c_{neg}; (iv) maximize the clipped PPO surrogate with entropy regularization, where $\rho(\theta) = \exp(\log \pi_\theta - \log \pi_{\text{old}})$; (v) backpropagate only through noised tokens.

Since our model requires a sequence length as input during generation, we use an online bandit to bias sampling toward lengths with higher expected rewards while preserving prior distributions for exploration (see App. B). To summarize, pseudocode for the combined genetic algorithm and PPO fine-tuning is shown in Alg. 1.

Results The PMO benchmark comprises 23 single-objective molecular optimization tasks. Evaluation considers both the best achieved score and sample efficiency, summarized by the *TOP-10 AUC*: the area under the curve of the mean score of the top ten molecules as a function of oracle calls. Scores are normalized to $[0, 1]$, and each run is limited to 10,000 oracle evaluations. GenMol [Lee et al., 2025a] and f-RAG [Lee et al., 2024] initialize their population by screening the entire ZINC250k dataset, i.e., making 250,000 oracle calls before optimization begins. This implicitly uses oracle information at initialization and complicates direct comparison with methods such as REINVENT [Olivecrona et al., 2017], Mol GA, Genetic GFN, and others that start that do not

Algorithm 1 Target-Property Optimization (GA + PPO)

Require: Model π, frozen prior π_{old}, oracle O, population $\mathcal{P}$, bandit B, maximum oracle calls $N_{\max}$, fragmentation rule $\mathcal{R}$, mutation op, PPO params $(\epsilon, c_{\text{neg}}, \beta)$

```
while oracle calls < N_max do
    sample parent pairs from P; draw lengths ℓ ~ B
    P_off ← {X ~ π(· | crossover(R(p_1), R(p_2)), ℓ)}
    P_off ← P_off ∪ {mutate(x) : x ∈ top_N(P)}
    r ← O(P_off)
    for k = 1 to K do
        θ ← θ − ∇_θ L(θ; r, π_θ, π_old)
    end for
    update B with (ℓ, r);    P ← top{P ∪ P_off}
    π_old ← π
end while
return top molecules
```

prescreen any data. To match prior protocols, we also report results using a ZINC250k-screened initial population (Table 2), but emphasize that this setup is not suitable for fair benchmarking across all methods. To compare to latter methods, Table 3 reports results with an initial population generated by our model. Due to space constraints, we only quoted the best four methods that were published on this benchmark. In both setups, InVirtuoGen consistently achieves superior performance when considering the sum over all tasks as a summary metric.

Table 2: Comparison of models on the PMO benchmark that screen ZINC250k before initialization. We report the AUC-top10 scores, averaged over three runs with standard deviations. Best results and those within one standard deviation of the best are indicated in bold. The scores for f-RAG [Lee et al., 2024] and GenMol [Lee et al., 2025a] are taken from the respective publications.

Oracle	InVirtuoGen	GenMol	f-RAG
albuterol similarity	**0.975** (± 0.016)	0.937 (± 0.010)	**0.977** (± 0.002)
amlodipine mpo	**0.836** (± 0.031)	**0.810** (± 0.012)	0.749 (± 0.019)
celecoxib rediscovery	**0.839** (± 0.013)	0.826 (± 0.018)	0.778 (± 0.007)
deco hop	**0.968** (± 0.012)	**0.960** (± 0.010)	0.936 (± 0.011)
drd2	**0.995** (± 0.000)	**0.995** (± 0.000)	0.992 (± 0.000)
fexofenadine mpo	**0.904** (± 0.000)	0.894 (± 0.028)	0.856 (± 0.016)
gsk3b	**0.988** (± 0.001)	0.986 (± 0.003)	0.969 (± 0.003)
isomers c7h8n2o2	**0.988** (± 0.002)	0.942 (± 0.004)	0.955 (± 0.008)
isomers c9h10n2o2pf2cl	**0.898** (± 0.018)	0.833 (± 0.014)	0.850 (± 0.005)
jnk3	**0.898** (± 0.031)	**0.906** (± 0.023)	**0.904** (± 0.004)
median1	0.386 (± 0.003)	**0.398** (± 0.000)	0.340 (± 0.007)
median2	**0.377** (± 0.006)	0.359 (± 0.004)	0.323 (± 0.005)
mestranol similarity	**0.991** (± 0.002)	0.982 (± 0.000)	0.671 (± 0.021)
osimertinib mpo	**0.881** (± 0.012)	**0.876** (± 0.008)	0.866 (± 0.009)
perindopril mpo	**0.753** (± 0.019)	0.718 (± 0.012)	0.681 (± 0.017)
qed	**0.943** (± 0.000)	0.942 (± 0.000)	0.939 (± 0.001)
ranolazine mpo	**0.854** (± 0.012)	0.821 (± 0.011)	0.820 (± 0.016)
scaffold hop	**0.711** (± 0.081)	0.628 (± 0.008)	0.576 (± 0.014)
sitagliptin mpo	**0.743** (± 0.022)	0.584 (± 0.034)	0.601 (± 0.011)
thiothixene rediscovery	**0.652** (± 0.024)	**0.692** (± 0.123)	**0.584** (± 0.009)
troglitazone rediscovery	**0.853** (± 0.003)	**0.867** (± 0.022)	0.448 (± 0.017)
valsartan smarts	**0.935** (± 0.012)	0.822 (± 0.042)	0.627 (± 0.058)
zaleplon mpo	**0.624** (± 0.040)	**0.584** (± 0.011)	0.486 (± 0.004)
Sum	**18.993** (± 0.219)	18.362	16.928

Table 3: The results of the best performing models on the PMO benchmark, where we quote the AUC-top10 averaged over 3 runs with standard deviations. The best results are highlighted in bold. Values within one standard deviation of the best are also marked in bold. The results for Genetic GFN [Kim et al., 2024] and Mol GA [Tripp and Hernández-Lobato, 2023] are taken from the respective papers. The other results are taken from the original PMO benchmark paper by [Gao et al., 2022a].

Oracle	InVirtuoGen (no prescreen)	Gen. GFN	Mol GA	REINVENT
albuterol similarity	**0.950** (± 0.017)	**0.949** (± 0.010)	0.896 (± 0.035)	0.882 (± 0.006)
amlodipine mpo	0.733 (± 0.043)	**0.761** (± 0.019)	0.688 (± 0.039)	0.635 (± 0.035)
celecoxib rediscovery	**0.798** (± 0.028)	**0.802** (± 0.029)	0.567 (± 0.083)	0.713 (± 0.067)
deco hop	**0.748** (± 0.109)	**0.733** (± 0.109)	**0.649** (± 0.025)	**0.666** (± 0.044)
drd2	**0.985** (± 0.002)	0.974 (± 0.006)	0.936 (± 0.016)	0.945 (± 0.007)
fexofenadine mpo	**0.845** (± 0.016)	**0.856** (± 0.039)	**0.825** (± 0.019)	0.784 (± 0.006)
gsk3b	**0.952** (± 0.016)	0.881 (± 0.042)	0.843 (± 0.039)	0.865 (± 0.043)
isomers c7h8n2o2	**0.968** (± 0.005)	**0.969** (± 0.003)	0.878 (± 0.026)	0.852 (± 0.036)
isomers c9h10n2o2pf2cl	0.874 (± 0.013)	**0.897** (± 0.007)	0.865 (± 0.012)	0.642 (± 0.054)
jnk3	**0.825** (± 0.016)	0.764 (± 0.069)	0.702 (± 0.123)	0.783 (± 0.023)
median1	0.342 (± 0.008)	**0.379** (± 0.010)	0.257 (± 0.009)	0.356 (± 0.009)
median2	**0.288** (± 0.008)	**0.294** (± 0.007)	**0.301** (± 0.021)	0.276 (± 0.008)
mestranol similarity	**0.797** (± 0.033)	0.708 (± 0.057)	0.591 (± 0.053)	0.618 (± 0.048)
osimertinib mpo	**0.870** (± 0.005)	0.860 (± 0.008)	0.844 (± 0.015)	0.837 (± 0.009)
perindopril mpo	**0.645** (± 0.032)	0.595 (± 0.014)	0.547 (± 0.022)	0.537 (± 0.016)
qed	**0.942** (± 0.000)	**0.942** (± 0.000)	0.941 (± 0.001)	0.941 (± 0.000)
ranolazine mpo	**0.848** (± 0.010)	0.819 (± 0.018)	0.804 (± 0.011)	0.760 (± 0.009)
scaffold hop	**0.589** (± 0.021)	**0.615** (± 0.100)	**0.527** (± 0.025)	**0.560** (± 0.019)
sitagliptin mpo	**0.709** (± 0.029)	0.634 (± 0.039)	0.582 (± 0.040)	0.021 (± 0.003)
thiothixene rediscovery	**0.625** (± 0.014)	0.583 (± 0.034)	0.519 (± 0.041)	0.534 (± 0.013)
troglitazone rediscovery	**0.595** (± 0.053)	0.511 (± 0.054)	0.427 (± 0.031)	0.441 (± 0.032)
valsartan smarts	**0.210** (± 0.297)	**0.135** (± 0.271)	**0.000** (± 0.000)	**0.178** (± 0.358)
zaleplon mpo	**0.536** (± 0.006)	**0.552** (± 0.033)	**0.519** (± 0.029)	0.358 (± 0.062)
Sum	**16.676** (± 0.256)	16.213	14.708	14.184

4 Conclusion

We have presented a discrete flow model with a uniform source, which generates fragmented SMILES, enabling flexible, fragment-level control and bidirectional context in molecular generation. By decoupling sequence length from token generation, we can show that a finer granularity during the simulation trajectory leads to more diverse and drug-like molecules. The uniform-source formulation also enables seamless integration with an genetic algorithm and PPO-based fine-tuning, yielding strong gains in performance and sample efficiency. Across de novo generation, fragment-constrained design, and target property optimization, our approach advances the state-of-the-art, achieving a new Pareto frontier in quality–diversity trade-offs, higher average quality and diversity in fragment-constrained tasks and superior performance on the PMO benchmark.

References

R. Bickerton, G. Paolini, J. Besnard, S. Muresan, and A. Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4:90–8, 02 2012. doi: 10.1038/nchem.1243.

A. Bou, M. Thomas, S. Dittert, C. N. Ramírez, M. Majewski, Y. Wang, S. Patel, G. Tresadern, M. Ahmad, V. Moens, W. Sherman, S. Sciabola, and G. D. Fabritiis. Acegen: Reinforcement learning of generative chemical agents for drug discovery, 2024. URL https://arxiv.org/abs/2405.04657.

A. Campbell, J. Yim, R. Barzilay, T. Rainforth, and T. Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design, 2024. URL https://arxiv.org/abs/2402.04997.

J. Chambers, M. Davies, A. Gaulton, A. Hersey, S. Velankar, R. Petryszak, J. Hastings, L. Bellis, S. McGlinchey, and J. P. Overington. Unichem: a unified chemical structure cross-referencing and identifier tracking system. *Journal of Cheminformatics*, 5(1):3, 2013. doi: 10.1186/1758-2946-5-3. URL https://doi.org/10.1186/1758-2946-5-3.

D. Chen, M. Krimmel, and K. Borgwardt. Flatten graphs as sequences: Transformers are scalable graph generators, 2025. URL https://arxiv.org/abs/2502.02216.

J. Degen, C. Wegscheid-Gerlach, A. Zaliani, and M. Rarey. On the art of compiling and using 'drug-like' chemical fragment spaces. *ChemMedChem: Chemistry Enabling Drug Discovery*, 3 (10):1503–1507, 2008.

P. Ertl and A. Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1:8, 06 2009. doi: 10.1186/1758-2946-1-8.

W. Gao, T. Fu, J. Sun, and C. W. Coley. Sample efficiency matters: A benchmark for practical molecular optimization, 2022a. URL https://arxiv.org/abs/2206.12411.

W. Gao, R. Mercado, and C. W. Coley. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design, 2022b. URL https://arxiv.org/abs/2110.06389.

I. Gat, T. Remez, N. Shaul, F. Kreuk, R. T. Q. Chen, G. Synnaeve, Y. Adi, and Y. Lipman. Discrete flow matching. *arXiv preprint arXiv:2407.15595*, 2024.

J. H. Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chem. Sci.*, 10:3567–3572, 2019. doi: 10.1039/C8SC05372C. URL http://dx.doi.org/10.1039/C8SC05372C.

W. Jin, R. Barzilay, and T. Jaakkola. Junction tree variational autoencoder for molecular graph generation. *Proceedings of the 35th International Conference on Machine Learning*, 80:2323–2332, 2018.

W. Jin, D. Barzilay, and T. Jaakkola. Multi-objective molecule generation using interpretable substructures. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4849–4859. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/jin20b.html.

S. Jinsong, J. Qifeng, C. Xing, Y. Hao, and L. Wang. Molecular fragmentation as a crucial step in the ai-based drug development pathway. *Communications Chemistry*, 7(1):20, 2024. doi: 10.1038/s42004-024-01109-2. URL https://doi.org/10.1038/s42004-024-01109-2.

H. Kim, M. Kim, S. Choi, and J. Park. Genetic-guided gflownets for sample efficient molecular optimization, 2024. URL https://arxiv.org/abs/2402.05961.

T. Kirkman, C. Silva, M. Tosin, and M. Dias. How to find a fragment: Methods for screening and validation in fragment-based drug discovery. *ChemMedChem*, 19, 11 2024. doi: 10.1002/cmdc.202400342.

P. Kirsch, A. M. Hartman, A. K. H. Hirsch, and M. Empting. Concepts and core principles of fragment-based drug design. *Molecules*, 24(23), 2019. ISSN 1420-3049. doi: 10.3390/molecules24234309. URL https://www.mdpi.com/1420-3049/24/23/4309.

S. Lee, K. Kreis, S. P. Veccham, M. Liu, D. Reidenbach, S. Paliwal, A. Vahdat, and W. Nie. Molecule generation with fragment retrieval augmentation, 2024. URL https://arxiv.org/abs/2411.12078.

S. Lee, K. Kreis, S. P. Veccham, M. Liu, D. Reidenbach, Y. Peng, S. Paliwal, W. Nie, and A. Vahdat. Genmol: A drug discovery generalist with discrete diffusion, 2025a. URL https://arxiv.org/abs/2501.06158.

S. Lee, K. Kreis, S. P. Veccham, M. Liu, D. Reidenbach, Y. Peng, S. Paliwal, W. Nie, and A. Vahdat. Genmol: A drug discovery generalist with discrete diffusion. *arXiv preprint arXiv:2501.06158*, 2025b.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

K. Maziarz, H. Jackson-Flux, P. Cameron, F. Sirockin, N. Schneider, N. Stiefl, M. Segler, and M. Brockschmidt. Learning to extend molecular scaffolds with structural motifs, 2024. URL https://arxiv.org/abs/2103.03864.

O. M. Morrison, F. Pichi, and J. S. Hesthaven. Gfn: A graph feedforward network for resolution-invariant reduced operator learning in multifidelity applications, 2024. URL https://arxiv.org/abs/2406.03569.

A. Nigam, P. Friederich, M. Krenn, and A. Aspuru-Guzik. Augmenting genetic algorithms with deep neural networks for exploring the chemical space, 2020. URL https://arxiv.org/abs/1909.11655.

E. Noutahi, C. Gabellini, M. Craig, J. S. C. Lim, and P. Tossou. Gotta be safe: A new framework for molecular design, 2023. URL https://arxiv.org/abs/2310.10773.

M. Olivecrona, T. Blaschke, O. Engkvist, and H. Chen. Molecular de novo design through deep reinforcement learning, 2017. URL https://arxiv.org/abs/1704.07555.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

W. Peebles and S. Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

S. S. Sahoo, M. Arriola, Y. Schiff, A. Gokaslan, E. Marroquin, J. T. Chiu, A. Rush, and V. Kuleshov. Simple and effective masked diffusion language models, 2024. URL https://arxiv.org/abs/2406.07524.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

T. Sterling and J. Irwin. Zinc 15 - ligand discovery for everyone. *Journal of chemical information and modeling*, 55, 10 2015. doi: 10.1021/acs.jcim.5b00559.

J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

A. Tripp and J. M. Hernández-Lobato. Genetic algorithms are strong baselines for molecule generation, 2023. URL https://arxiv.org/abs/2310.09267.

D. Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988. doi: 10.1021/ci00057a005.

Y. Xie, C. Shi, H. Zhou, Y. Yang, W. Zhang, Y. Yu, and L. Li. Mars: Markov molecular sampling for multi-objective drug discovery, 2021. URL https://arxiv.org/abs/2103.10432.

X. Zeng, F. Wang, Y. Luo, S. gu Kang, J. Tang, F. C. Lightstone, E. F. Fang, W. Cornell, R. Nussinov, and F. Cheng. Deep generative molecular design reshapes drug discovery. *Cell Reports Medicine*, 3(12):100794, 2022.

A Experimental Details

A.1 Implementation Details

The backbone model, adapted from Peebles and Xie [2022], has 36 layers[4], 12 heads, the hidden dimension is chosen as 768, and uses rotary positional embedding [Su et al., 2023]. Training is performed for 1 epoch with batch size of 300. Sequences of similar length are bucketed together to reduce the padding per batch. The maximal number of tokens per batch is limited to $24'000$. The AdamW [Loshchilov and Hutter, 2019] optimizer, with $(\beta_1 = 0.99, \beta_2 = 0.999)$ is used with a learning rate of 10^{-4}. Additionally, the learning rate is varied according to a linear warmup cosine annealing scheduler. All training is performed on a single NVIDIA RTX 4090 GPU using PyTorch's automatic mixed precision (AMP) in `float16` mode [Paszke et al., 2019].

B Algorithms

B.1 Softmax Bandit

Algorithm 2 Softmax Bandit for Adaptive Length Sampling

Require: Candidate lengths $\{n_k\}$, initial prior $\pi^{(0)}$, Q-values $Q_k \leftarrow 0$, learning rate α, inertia β, temperature τ

while generating molecules **do**
  Compute policy: $\pi_k \propto \exp(\log \pi_k^{(t)} + Q_k/\tau)$
  Sample sequence length $L \sim \text{Categorical}(\pi)$
  Generate molecules of length L and obtain reward r
  Update Q-values: $Q_k \leftarrow Q_k + \alpha(r - Q_k)$
  Update prior: $\pi^{(t+1)} \leftarrow \beta\pi^{(t)} + (1 - \beta)\pi$
  Normalize $\pi^{(t+1)}$
end while

[4]For the results in the target property optimization a smaller model with 12 layers was used to improve speed and reduce memory use.

C Additional Experimental Results

C.1 De Novo Generation

In Fig. 3 we provide non-curated samples from de novo generation with temperature $T = 1$ and randomness $r = 0$. Additionally, Fig. 4 shows the distributions of QED and SA for different

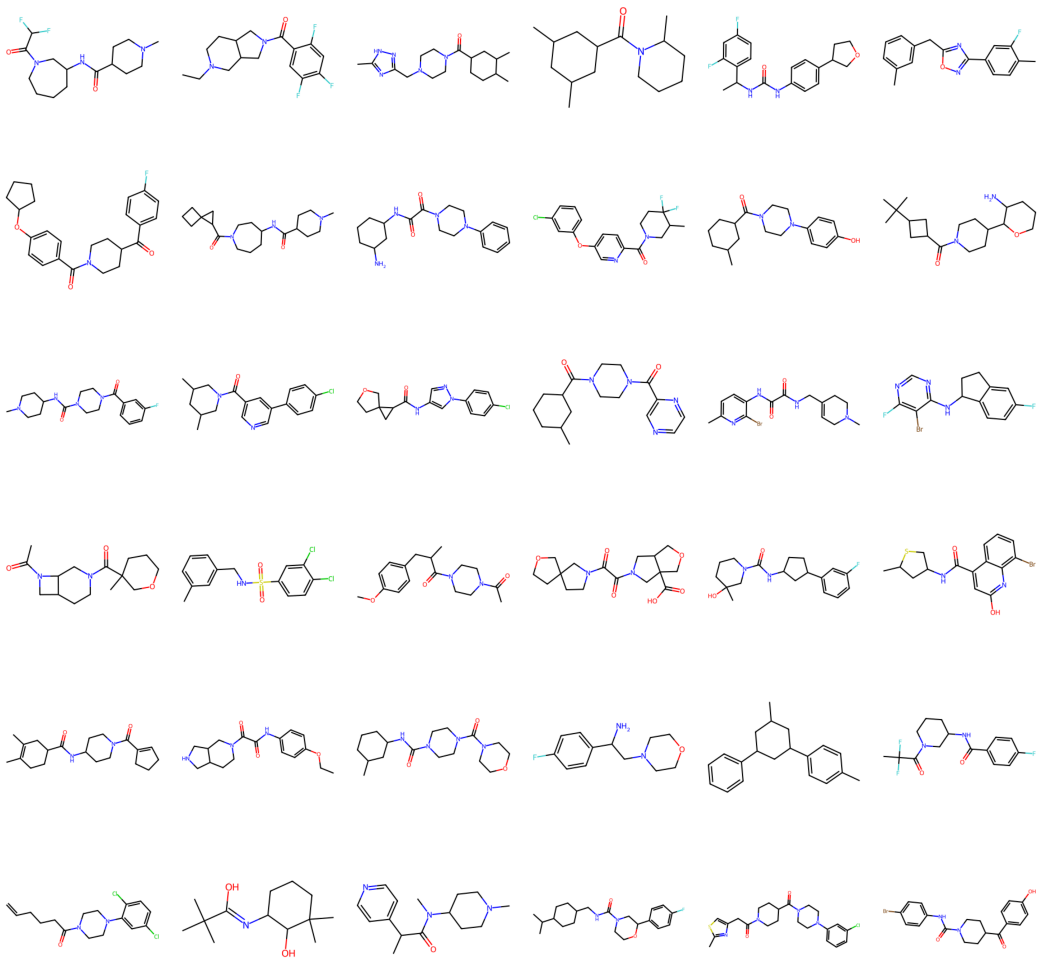


Figure 3: Non curated samples for de novo generation

time resolutions under our proposed sampling method (Eq. 4). As a reference, the corresponding distributions from ZINC250k are also shown. For comparison, Fig. 5 reports the same distributions when using the standard discrete flow sampling update (Eq. 1). Our method produces molecules with substantially higher QED, while the baseline more closely follows the ZINC250k distribution. The reason for this shift in QED under our sampling scheme remains unclear, and a more detailed analysis of the underlying dynamics is an interesting direction for future work.

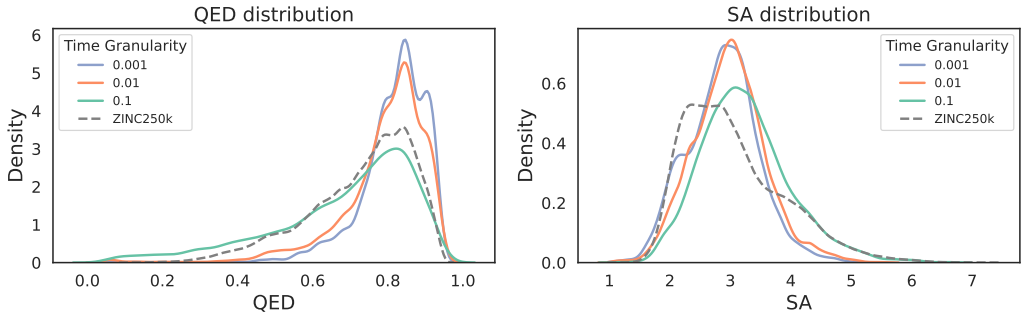


Figure 4: Distributions of QED and SA for molecules generated with our sampling method (Eq. 4) at different time resolutions. ZINC250k serves as a reference.

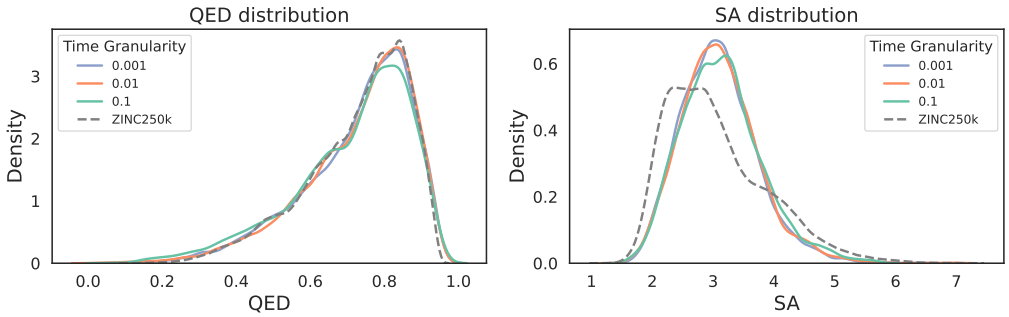
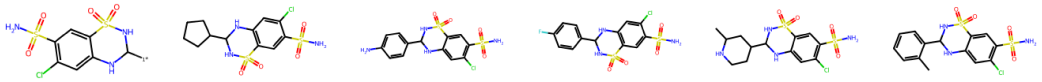


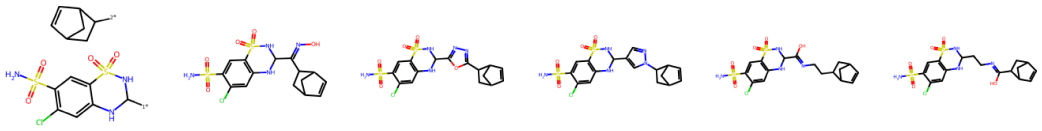
Figure 5: Distributions of QED and SA for molecules generated with the standard discrete flow sampling update (Eq. 1).

C.2 Fragment-Constrained Generation

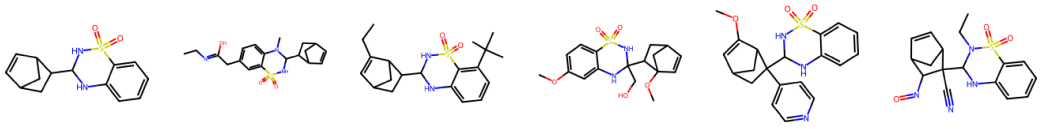
In Fig. 6 we provide non-cherry picked samples for the fragment-constrained generation task.



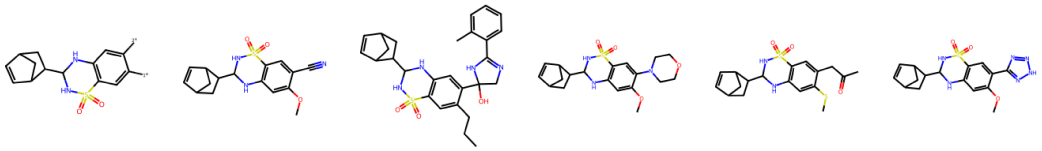
(a) Motif Extension



(b) Linker Design/Scaffold Morphing



(c) Superstructure Generation



(d) Scaffold Decoration

Figure 6: Non cherry-picked samples generated by InVirtuoGen on fragment-constrained design tasks.