RETRO SYNFLOW: Discrete Flow Matching for Accurate and Diverse Single-Step Retrosynthesis

Robin Yadav[1]
robiny12@student.ubc.ca

Qi Yan[1,2]
qi.yan@ece.ubc.ca

Guy Wolf[3,4,5]
wolfguy@mila.quebec

Avishek Joey Bose[3,6]
joey.bose@mail.mcgill.ca

Renjie Liao[1,2,5]
rjliao@ece.ubc.ca

[1]UBC; [2]Vector Institute; [3]Mila; [4]Université de Montréal;
[5]Canada CIFAR AI Chair; [6]University of Oxford

Abstract

A fundamental problem in organic chemistry is identifying and predicting the series of reactions that synthesize a desired target product molecule. Due to the combinatorial nature of the chemical search space, single-step reactant prediction—*i.e.* single-step retrosynthesis—remains challenging even for existing state-of-the-art template-free generative approaches to produce an accurate yet diverse set of feasible reactions. In this paper, we model single-step retrosynthesis planning and introduce RETRO SYNFLOW (RSF) a discrete flow-matching framework that builds a Markov bridge between the prescribed target product molecule and the reactant molecule. In contrast to past approaches, RSF employs a reaction center identification step to produce intermediate structures known as synthons as a more informative source distribution for the discrete flow. To further enhance diversity and feasibility of generated samples, we employ Feynman-Kac steering with Sequential Monte Carlo based resampling to steer promising generations at inference using a new reward oracle that relies on a forward-synthesis model. Empirically, we demonstrate RSF achieves 60.0% top-1 accuracy, which outperforms the previous SOTA by 20%. We also substantiate the benefits of steering at inference and demonstrate that FK-steering improves top-5 round-trip accuracy by 19% over prior template-free SOTA methods, all while preserving competitive top-k accuracy results.

1 Introduction

Retrosynthesis planning is a fundamental problem in chemistry that involves decomposing a complex target molecule (the product) into simpler, commercially available structures (the reactants) to establish synthesis routes [8, 45]. This process is crucial for verifying the synthesizability of proposed molecules with desirable properties, particularly in drug discovery [43]. For instance, retrosynthesis is critical for lead optimization in medicinal chemistry, which requires designing efficient synthetic routes to modify chemical structures to enhance a compound's potency, selectivity, and pharmacokinetic properties [25]. Traditionally, chemists manually identify and validate reactants and pathways, a labor-intensive process exacerbated by the vast search space of transformations from reactant to product molecules. This enduring challenge has driven decades of research in computer-assisted retrosynthesis [8], with recent advances in machine learning (ML) enabling more effective exploration of the combinatorial reactant space [3, 6, 48, 9]. Such methods are promising in significantly accelerate the drug discovery pipeline.

The current dominant paradigm for ML-based retrosynthesis planning consists of two main components: a *single-step retrosynthesis* model and a multi-step planning algorithm. These ML-based approaches can be broadly categorized as *template-based* and *template-free* methods. Template-based

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

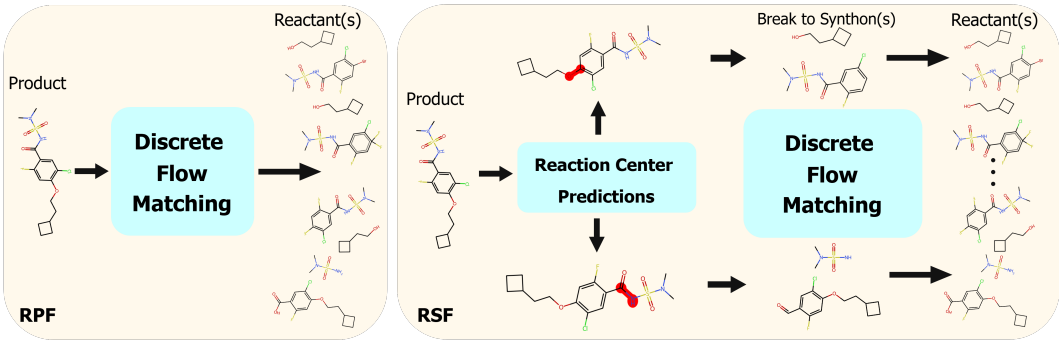


Figure 1: An overview of our RETRO PRODFLOW (RPF) and RETRO SYNFLOW (RSF) framework. RPF directly maps a product molecule to reactants via discrete flow. RSF first predicts synthons from the product using a reaction center predictor, then maps these synthons to reactants via discrete flow.

methods rely on a predefined database of reaction templates with hand-crafted specificity [36], which ensures syntactic and chemical validity but often limits diversity and generalization to novel reaction types. In contrast, template-free methods and *semi-template* methods are more flexible and capable of predicting reactions not seen in existing databases [50], offering improved generalization. More recently, template-free and semi-template approaches have emerged as a natural fit for generative modeling techniques, enabling retrosynthesis prediction to be formulated as a *conditional generation problem*: generating reactants conditioned on a given product molecule. Semi-template methods increase interpretability by breaking the generation process into two steps by first identifying intermediate molecular structures called synthons and completing synthons to form reactants. While promising, many existing generative methods rely on sequential molecular representations like SMILES [39], and as a result, fail to capture the rich chemical contexts encoded in molecular attributed graphs.

Current work. In this paper, we frame single-step retrosynthesis as learning a transport map from a source distribution to an intractable target data distribution using finite paired samples. We represent molecules as attributed graphs and propose two template-free/semi-template generative models— RETRO PRODFLOW (RPF) and RETRO SYNFLOW (RSF)–based on recent advances in discrete flow matching [14, 2]. As discrete flows are flexible in choosing the source distribution, we explore two options as shown in Figure 1. First, in RPF, we use the product distribution directly as the source and learn a flow that transforms products to reactants. Second, we leverage pretrained reaction center prediction models to construct an informative source distribution over *synthons*—intermediate molecular fragments obtained by decomposing the product at its reaction center. This reduces the original problem to a simpler conditional generation task, *i.e.*, mapping from synthons to reactants, and improves performance. Both models learn continuous-time Markov chains to stochastically transport product molecules to reactants, benefiting from fast inference and high sample quality. Furhtermore, they naturally support scoring and ranking of generated reactants via its stochastic formulation.

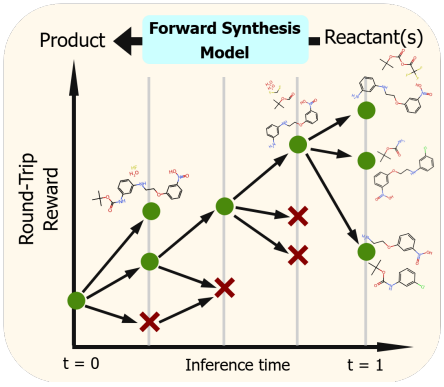


Figure 2: Inference time steering with a forward-synthesis reward model.

As the space of possible reactants that synthesize into valid products is combinatorially large, a fundamental design goal in retrosynthesis is to generate diverse candidate reactants that simultaneously achieve high accuracy. To address this, we leverage Feynman-Kac (FK) steering [40], an inference-time steering method that guides sampling of flow models towards more feasible and diverse outputs. FK steering employs sequential Monte-Carlo (SMC), a particle-based method that resamples promising intermediate candidates throughout the generation process based on a reward function. We define this reward using a forward-synthesis model to enforce round-trip consistency—a standard measure of diversity and feasibility. We highlight the benefits of reward-based inference time steering, demonstrating a 13% improvement in top-5 round-trip accuracy over prior SOTA methods.

In summary, our main contributions are listed below,

1. We propose the first flow matching framework for retrosynthesis, introducing two variants: RETRO PRODFLOW (RPF), which maps products directly to reactants, and RETRO SYNFLOW (RSF), which leverages *synthons* (intermediate molecular fragments) to simplify the generation task.
2. We improve the diversity and feasibility of generated reactants using FK-steering, an inference-time technique guided by a forward-synthesis reward. This yields a 19% gain in top-5 round-trip accuracy over prior template-free methods.
3. We show that using synthons as an inductive prior significantly enhances performance. RSF achieves 60% top-1 accuracy on the USPTO-50k benchmark, outperforming state-of-the-art template-free and semi-template methods.

2 Background and preliminaries

Notations. Given a vocabulary set $\mathcal{X}$ with d elements, we establish a bijection between $\mathcal{X}$ and the index set $[d] = \{1, \dots, d\}$. Accordingly, any discrete data x drawn from $\mathcal{X}$ can be represented as an integer index in $[d]$. A categorical distribution over $\mathcal{X}$, denoted $\text{Cat}(x; p)$, is given by $p(x = i) = p^i$, where $\sum_{i=1}^{d} p^i = 1$ and $p^i \geq 0, \forall i$. A sequence $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^n)$ of n tokens is defined over the product space $\mathcal{X}^n$. We assume a dataset of such sequences is sampled from a target data distribution p_{data}. Discrete flow matching models, like their continuous counterparts, aim to transport a source distribution $p_{\text{source}} := p_0$ defined at time $t = 0$ to the data distribution $p_{\text{data}} := p_1$ at time $t = 1$.

2.1 Discrete Flow Matching

Discrete flow matching operates directly on discrete data, mirroring the construction of flow matching models over continuous spaces. Analogously, our goal is to construct a generative probability path, p_t that interpolates between the source and target distributions. The key insight of flow matching is to construct p_t by marginalizing simpler probability paths conditioned on samples from the source and data distributions. A conditional probability path, $p_t(\cdot|\mathbf{x}_0, \mathbf{x}_1)$ is a time-evolving distribution satisfying $p_0(\mathbf{x}^i|\mathbf{x}_0, \mathbf{x}_1) = \delta(\mathbf{x}_0^i)$ and $p_1(\mathbf{x}^i|\mathbf{x}_0, \mathbf{x}_1) = \delta(\mathbf{x}_1^i)$ where $\mathbf{x}_0 \sim p_0$ and $\mathbf{x}_1 \sim p_1$. Conditional probability paths are independent across each dimension of the sequence, with the simplest choice being a convex combination of $p_0(\mathbf{x}^i|\mathbf{x}_0, \mathbf{x}_1)$ and $p_1(\mathbf{x}^i|\mathbf{x}_0, \mathbf{x}_1)$,

$$p_t(\mathbf{x}_t^i|\mathbf{x}_0, \mathbf{x}_1) = \text{Cat}(\mathbf{x}_t^i; (1-t)\delta(\mathbf{x}_0^i) + t\delta(\mathbf{x}_1^i)), \quad \text{where } p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_1) = \prod_{i=1}^{n} p_t(\mathbf{x}_t^i|\mathbf{x}_0, \mathbf{x}_1). \tag{1}$$

Similar to the continuous setting, discrete flow matching constructs a generating *probability velocity* $u_t(\cdot, \mathbf{x}_t) \in \mathbb{R}^n$, which models the rate of probability mass change of the sample $\mathbf{x}_t$ in each of its n positions. Specifically, we view $(\mathbf{x}_t)_{0 \leq t \leq 1}$ as a collection of random variables that form a continuous-time Markov Chain (CTMC), jumping between states in $\mathcal{X}^n$. Each position of $\mathbf{x}_t$ can be simulated by the following probability transition kernel $p_{t+h|t}(\mathbf{x}_{t+h}^i|\mathbf{x}_t) = \text{Cat}(\mathbf{x}_{t+h}^i; \delta(\mathbf{x}_t^i) + hu_t^i(\mathbf{x}_{t+h}^i, \mathbf{x}_t))$. Thus, sampling from the CTMC and simulating a trajectory from p_t given its velocity u_t is straightforward. We start from a sample $\mathbf{x}_0 \sim p_0$ from the source distribution and update each dimension with the transition kernel $\mathbf{x}_{t+h}^i \sim p_{t+h|t}(\mathbf{x}_{t+h}^i|\mathbf{x}_t)$. This results in samples from the desired data distribution p_1. Analogous to continuous flow matching, u_t^i is constructed by marginalizing the conditional probability velocities that generate the conditional probability paths. For the simple conditional probability paths given by Eq. 1, the marginal probability velocity is $u_t^i(\mathbf{x}^i, \mathbf{x}_t) = \left(p_{1|t}(\mathbf{x}^i|\mathbf{x}_t) - \delta(\mathbf{x}_t^i)\right)/(1-t)$. The intractable posterior distribution, $p_{1|t}$, known as the probability *denoiser*, predicts a clean sample $\mathbf{x}_1$ from an intermediate noisy sample $\mathbf{x}_t$. We can approximate the denoiser with a neural network $p_\theta(\mathbf{x}_1^i|\mathbf{x}_t)$ and train it by minimizing a cross-entropy loss that forms a weighted evidence lower bound (ELBO) on $\log p_{1,\theta}(\mathbf{x}_1)$ [12].

2.2 Feynman-Kac Steering

Given a trained flow model whose marginal distribution at time $t = 1$ is denoted by $p_\theta(\mathbf{x}_1)$. We are interested in sampling from a target distribution that tilts $p_\theta(\mathbf{x}_1)$ using a terminal (parametrized) reward function $r : \mathcal{X}^n \to [0, 1]$, that consumes fully denoised sequences $\mathbf{x}_1$,

$$p_{\text{target}}(\mathbf{x}_1) = \frac{1}{\mathcal{Z}} p_\theta(\mathbf{x}_1) \exp(\lambda r(\mathbf{x}_1)), \tag{2}$$

where λ controls the steering intensity, and $\mathcal{Z}$ is a normalization constant. Sampling trajectories in flow models involves discretizing the interval $[0, 1]$ into a grid of timesteps $\{0, h, \cdots, 1-h, 1\}$ and sampling from the learned transition kernel $p_{t+h|t}$. To steer the sampling process toward high-reward outcomes, we employ *Feynman-Kac (FK) steering* [40], which modifies the transition kernels using potential functions that favor trajectories $\tau(\mathbf{x}_{0:1})$ ending with high-reward $\mathbf{x}_1$ samples. The FK process begins from a reweighted initial distribution $p_{\text{FK},0}(\mathbf{x}_0) \propto p_{\text{source}}(\mathbf{x}_0)U_0(\mathbf{x}_0)$ and iteratively build $p_{\text{FK},t+h}$ by tilting the transition kernel with a potential $U_t(\mathbf{x}_0, \cdots, \mathbf{x}_{t-h}, \mathbf{x}_t)$:

$$p_{\text{FK},t+h}(\mathbf{x}_{t+h}) = \frac{1}{\mathcal{Z}_t} p_{t+h|t}(\mathbf{x}_{t+h}|\mathbf{x}_t)U_t(\mathbf{x}_{0:t}) \underbrace{p_\theta(\mathbf{x}_{0:t}) \prod_{s\in\{0,\cdots,t-h,t\}} U_s(\mathbf{x}_{0:s})}_{\propto p_{\text{FK},t}}.$$

Since direct sampling from $p_{\text{FK},1}$ is intractible, we employ Sequential Monte Carlo (SMC) methods. SMC begins with K particles $\{\mathbf{x}_0^m\}_{m=0}^K$ sampled from the source distribution. At each transition step, it updates their importance weights and resamples the particles accordingly. The importance weight for particle m at time $t+h$ is given by $w_{t+h}^m = p_{t+h|t}(\mathbf{x}_{t+h}^m|\mathbf{x}_t)U_t(\mathbf{x}_{0:t}^m)$.

3 Discrete Flow Matching for Retrosynthesis

We model single-step retrosynthesis using discrete flow matching by representing product and reactant molecules as a pair of molecular graphs (G^p, G^r). Our focus is on the single product setting, *i.e.*, a single product molecule corresponds to a set of reactant molecules. The reactants are represented as a single disconnected graph to account for multiple molecules. A molecular graph $G = (\mathbf{v}, \boldsymbol{E})$ with N atoms consists of: a node feature vector $\mathbf{v} \in [K_n]^N$, where $\mathbf{v}^i$ encodes the atom type of atom i (out of K_n possible types), and an edge feature matrix $\boldsymbol{E} \in [K_e]^{N\times N}$, where $\boldsymbol{E}^{i,j}$ indicates the bond type (among K_e possible types) between atoms i and j.

In this formulation, both $\mathbf{v}$ and $\boldsymbol{E}$ can be viewed as collections of discrete random variables. We aim to learn a generative probability path p_t that interpolates between a source and the data distributions over graphs of product and reactant pairs. Correspondingly, the design of the conditional probability path of a graph, $p_t(G_t|G_0, G_1)$ factorizes over the nodes and edges as follows:

$$p_t(\mathbf{v}_t^i|\mathbf{v}_0, \mathbf{v}_1) = (1-t)\delta(\mathbf{v}_0^i) + t\delta(\mathbf{v}_1^i), \quad p_t(\boldsymbol{E}_t^{i,j}|\boldsymbol{E}_0, \boldsymbol{E}_1) = (1-t)\delta(\boldsymbol{E}_0^{i,j}) + t\delta(\boldsymbol{E}_1^{i,j}).$$

To complete the specification of a discrete flow matching model, we must also define both the source and data distributions. For retrosynthesis, the data distribution, p_{data} is simply the distribution of reactant molecules. We explore two choices for the source distribution. The most natural option is to set the source distribution, p_{source} to the empirical distribution of product molecules. This leads to our first model, RETRO PRODFLOW (RPF), which directly transports product molecules to their corresponding reactants. The second option, which we discuss in detail in the following section, sets the source distribution to the space of intermediate synthons which acts as a more informative structural prior. Since some atoms present in the product may not appear in the reactants, we follow RetroBridge [18] and append dummy atoms to the product to ensure alignment of node dimensions. More details are discussed in Appendix A.

The probabilistic formulation of flow matching provides a natural way to select the most likely reactants for a product molecule out of all possible generations. For a set of N samples, $\{\mathbf{x}_1^i\}_{i=1}^N$ generated by the flow matching model for $\mathbf{x}_0 \sim p_{\text{source}}$, the score of sample $\hat{\mathbf{x}}_1$ is the empirical frequency. This score is an estimation of the probability of $\hat{\mathbf{x}}_1$ given $\mathbf{x}_0$, i.e.,

$$p_\theta(\hat{\mathbf{x}}_1|\mathbf{x}_0) = \mathbb{E}_{\mathbf{z}\sim p_\theta(\cdot|\mathbf{x}_0)} \mathbb{1}[\mathbf{z} = \hat{\mathbf{x}}_1] \approx \frac{1}{N}\sum_{i=1}^N \mathbb{1}[\mathbf{x}_1^i = \hat{\mathbf{x}}_1].$$

3.1 RETRO SYNFLOW

In our approach, we inject more valuable structural information into the flow matching process by setting the distribution of synthons as a more informative source. This is motivated by the two-stage formulation of single-step retrosynthesis: reaction center identification and synthon completion. The two-stage approach improves interpretability by closely mirroring the way expert chemists reason about retrosynthesis. Synthons are hypothetical intermediate molecules representing potential

reactants [39]. Although a synthon may not correspond to a chemically valid molecule, it can be transformed into one by adding suitable leaving groups that account for reactivity.

Synthon generation begins by identifying a reaction center in the product molecule. A reaction center is defined as a pair of atoms (i, j) in the product satisfying two criteria: 1) atoms i and j are connected by a bond in the product molecule, and 2) there is no bond between atoms i and j in the reactant(s). We can derive the synthon molecule(s) by deleting the bond that connects the atoms in the reaction center.

Our primary focus lies in the second step of retrosynthesis: applying flow matching for synthon completion. In this case, the source distribution p_{source} is the distribution of synthons. Given a product molecule G^p, we use a reaction center prediction model to output M potential reaction-center candidates. This results in M predictions for the set of synthon(s). Each set of synthon(s) is treated as a single disconnected graph, G^s. To train the discrete flow model, we once again decompose the conditional probability path $p_t(G_t|G_0, G_1)$ over the nodes and edges where $G_0 := G^s$ and $G_1 := G^r$. Therefore, we can model the generating probability velocity and update the tokens of the nodes and edges separately according to their respective transition kernels:

$$u^i_{\text{nodes},t}(\mathbf{v}^i, \mathbf{v}_t) = \frac{1}{1-t}\left[p_\theta(\mathbf{v}^i|\mathbf{v}_t, \boldsymbol{E}_t) - \delta(\mathbf{v}^i_t)\right],$$
$$u^{i,j}_{\text{edges},t}(\boldsymbol{E}^{i,j}, \mathbf{v}_t) = \frac{1}{1-t}\left[p_\theta(\boldsymbol{E}^{i,j}|\mathbf{v}_t, \boldsymbol{E}_t) - \delta(\boldsymbol{E}^{i,j}_t)\right].$$

The denoiser model $p_\theta(G_1|G_t)$ outputs probabilities of a "clean" (reactant) graph over nodes and edges conditioned on a noisy state graph G_t, an interpolation between synthons and reactant graphs. The stochastic processes over nodes and edges are coupled due to the denoiser model taking the noisy graph as input. We can express the outputs of the denoiser model separately over the nodes and edges $p_\theta(\mathbf{v}|\mathbf{v}_t, \boldsymbol{E}_t)$ and $p_\theta(\boldsymbol{E}|\mathbf{v}_t, \boldsymbol{E}_t)$. This leads to a natural objective for discrete flow matching where we minimize the weighted combination of the cross-entropy loss over nodes and edges:

$$\mathcal{L}(\theta) = -\mathbb{E}_{p(t),p_0(G_0),p_1(G_1),p_t(G_t|G_0,G_1)}\left[\sum_i \log p_\theta(\mathbf{v}^i|\mathbf{v}_t, \boldsymbol{E}_t) + \lambda \sum_{i,j} \log p_\theta(\boldsymbol{E}^{i,j}|\mathbf{v}_t, \boldsymbol{E}_t)\right],$$

where the distribution over time $p(t)$ is sampled uniformly, i.e. $t \sim \mathcal{U}(0, 1)$.

As in the previous case, we append "dummy" nodes (atoms) to the synthon molecule graph since the reactant molecule graph can have atoms that are not present in the synthon graph. For every synthon graph G^s_i, we generate N_i sets of reactants. The N_i's and M are hyperparameters. Given a budget of N set of reactants per product, we constrain $\sum_{i=1}^M N_i = N$. We demonstrate in our experiment section that these hyperparameters do not require much tuning, and $M = 2$ provides SOTA performance.

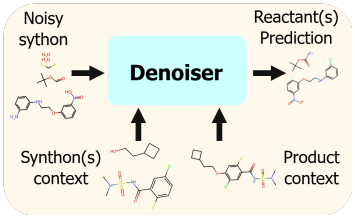


Figure 3: Overview of flow matching denoiser p_θ.

We adopt the reaction center prediction model from Shi et al. [39] to identify synthons. We do note however, that any model that outputs synthons could be equivalently used. Prior works such as [54, 39, 41] typically treat reaction center identification as a bond-level classification task. They use graph neural networks to classify each bond in the product as reactive or not by predicting bond-level reactivity scores or edit scores. During inference, we select the top-M bonds above a certain score threshold as candidate reaction centers, yielding M synthon predictions.

3.2 Reward-based steering

Given the discrete flow matching framework for retrosynthesis above, we can specify a potential function U_t and reward $r(\mathbf{x}_1)$ to perform inference time steering. For intermediate steps $t < 1$, we define:

$$U_t(\mathbf{x}_{0:t}) = \exp\left(\sum_{s=0}^t r_\phi(\mathbf{x}_s)\right), \text{ and } \quad U_1 = \exp(\lambda r_\phi(\mathbf{x}_1))\left(\prod_{t\in\{0,\cdots,1-h\}} U_t\right)^{-1}.$$

This design ensures that $p_{\text{FK},1} \propto p_{\text{target}}$ while steering intermediate particles toward high-reward trajectories. Furthermore, it selects particles that have the highest accumulated reward.

To perform SMC resampling, we need access to a reward oracle, r_ϕ that models the distribution of rewards $p_\theta(r(\mathbf{x}_1)|\mathbf{x}_t)$ generated from the intermediate state $\mathbf{x}_t$. Fortunately, we can still obtain high-quality estimates from this reward distribution without training a separate reward model by querying the flow matching denoiser model $p_\theta(\mathbf{x}_1|\mathbf{x}_t)$ instead. Specifically, the intermediate reward is defined as $r_\phi(\mathbf{x}_t) := r(\hat{\mathbf{x}}_1)$ where $\hat{\mathbf{x}}_1 = \mathbb{E}_{p_\theta(\mathbf{x}_1|\mathbf{x}_t)}[\mathbf{x}_1|\mathbf{x}_t]$ is the expected $\mathbf{x}_1$ given $\mathbf{x}_t$. This choice of intermediate reward can be evaluated efficiently without significant additional computational cost.

Our reward function $r_\phi(G_1)$ is inspired by round-trip accuracy, which measures the ability of a retrosynthesis model to recover diverse and feasible reactants. There may be many different sets of reactants that can synthesize the same product. Top-k round-trip accuracy aims to capture this characteristic of retrosynthesis by quantifying the proportion of feasible reactants(s) among the top-k predictions from the retrosynthesis model. We can assess whether a reaction is feasible by using a forward-synthesis model, which predicts the product molecule produced by a set of reactants. A reaction is deemed feasible if the forward model satisfies $F(\hat{G}^r) = G^p$, where $\hat{G}^r$ is the predicted reactants. The reward function serves as a proxy for round-trip accuracy, encouraging the generation of chemically valid and synthetically feasible reactants. Suppose (G^p, G^r) is a pair of product and reactant molecule graphs. The reward for the intermediate state G_t for the flow matching process on graphs is defined as $r_\phi(G_t) = \mathbb{1}[F(\hat{G}_1) = G^p]$ where $\hat{G}_1$ is the expected G_1 (reactants) given G_t. With this formulation of the reward, we introduce RETRO PRODFLOW-RS, a reward-steered version for RETRO PRODFLOW. Finally, we use Molecular Transformer [35] for the forward-synthesis model.

4 Experiments and Results

We evaluate our proposed methods against state-of-the-art template-free and template-based models on standard single-step retrosynthesis benchmarks. Through RETRO SYNFLOW, we aim to highlight the effectiveness of synthons as an inductive prior for generating reactants using flow matching. Additionally, we evaluate RETRO PRODFLOW-RS to show that inference-time reward-based steering enhances the diversity and feasibility of predicted reactants. We conduct various ablation studies assessing the performance of FK-steering and SMC-based resampling on standard metrics. Unless otherwise stated, we generate $N = 100$ sets of reactants for each input product. Our methods discretize the time interval $[0, 1]$ into $T = 50$ steps. For SMC resampling, RPF-RS uses $K = 4$ particles. For RSF, we use $M = 2$ synthon predictions with $N_1 = 70$ and $N_2 = 30$ for the top-2 synthon predictions respectively. Code is available at: https://github.com/DSL-Lab/RetroSynFlow.

4.1 Experimental Setup

Dataset. We trained and evaluated our methods on the USPTO-50K dataset [34], a standard benchmark for retrosynthesis modelling containing 50k atom-mapped reactions extracted from US patents. We follow the same train/evaluation/test split used by RetroBridge [18] and GLN [9]. As done in Retrobridge, we randomly permute the graph nodes as a pre-processing step before input to the flow matching model.

Baselines. We evaluate our methods against both template and template-free baselines. On the template-free side, we compare against graph-based approaches such as RetroBridge [18] and G2G [39]. We compare against SMILE string translation approaches such as Tied-Transformer [20], Augmented-Transformer [48], and SCROP [57]. Our baselines also include methods that combine graph and SMILE representations: GTA [38], MEGAN [33], Dual-TF [46], and Graph2SMILES [49]. On the template-based side, we compare against state-of-the-art approaches such as GLN [9], GraphRetro [41], LocalRetro [4], and RetroGFN [13]. We also compare against Chimera [27], a framework that ensembles multiple different SMILE and graph based models across template and template-free approaches.

Evaluation. We report top-k exact match accuracy, which measures the proportion of reactions where the ground-truth set of reactants is in the top k set of reactants predicted by the model. Following standard practices in prior works, we report $k = 1, 3, 5, 10$. Additionally, we report top-k round-trip accuracy and round-trip coverage to measure reaction feasibility. Given product and reactant molecular graphs (G^p, G^r), let $\mathcal{R} = \{\hat{G}_i^r\}_{i=1}^k$ be the top-k predicted sets of reactants for G^p. Let $\mathcal{P} = \{F(\hat{G}_i^r)\}_{i=1}^k$ be the predicted products from the forward-synthesis model. Then

top-k exact-match accuracy, round-trip accuracy and coverage for the product $\mathbf{x}$ are computed as follows:$\mathbb{1}[G^r \in \mathcal{R}]$, $\frac{1}{k}\sum_{i=1}^{k} \mathbb{1}[G^p = F(\hat{G}_i^r)]$, and $\mathbb{1}[G^p \in \mathcal{P}]$.

4.2 Main Results

Table 1: Top-k accuracy (exact match) on the USPTO-50k test dataset.

		Top-k Accuracy			
	Model	$k=1$	$k=3$	$k=5$	$k=10$
TB	GLN	52.5	74.7	81.2	87.9
	GraphRetro	**53.7**	68.3	72.2	75.5
	LocalRetro	52.6	**76.0**	**84.4**	**90.6**
	RetroGFN	49.2	73.3	81.1	88.0
TF	SCROP	43.7	60.0	65.2	68.7
	G2G	48.9	67.6	72.5	75.5
	Aug. Transformer	48.3	—	73.4	77.4
	DualTF$_{\text{aug}}$	53.6	70.7	74.6	77.0
	MEGAN	48.0	70.9	78.1	85.4
	Tied Transformer	47.1	67.1	73.1	76.3
	GTA$_{\text{aug}}$	51.1	67.0	74.8	81.6
	Graph2SMILES	52.9	68.5	70.0	75.2
	Retroformer$_{\text{aug}}$	52.9	68.2	72.5	76.4
	Chimera [1]	59.6	**82.8**	**89.2**	**94.2**
	RetroBridge	50.8	74.1	80.6	85.6
	RETRO PRODFLOW	50.0 ± 0.15	74.3 ± 0.30	81.2 ± 0.08	85.8 ± 0.04
	RETRO SYNFLOW	$\mathbf{60.0 \pm 0.22}$	77.9 ± 0.13	82.7 ± 0.15	85.3 ± 0.19

Our main results are presented in Table 1, comparing RETRO SYNFLOW to several previous SOTA models for retrosynthesis on top-k accuracy, and Table 2 compares RETRO PRODFLOW-RS on top-k round-trip accuracy. In particular, RETRO PRODFLOW-RS achieves a top-1 accuracy of 60%, approximately 20 % higher than other template-free methods that do not perform ensembling. It outperforms RetroBridge and RETRO PRODFLOW, which use a source distribution of product molecules to build the Markov Bridge/CTMC [2]. This demonstrates that synthons encode valuable structural information and serve as a more informative source of distribution for generating reactants. Additionally, RETRO SYNFLOW also produces notable gains in top-3 and top-5 exact match accuracy. Our approach is competitive with Chimera [27], a framework that ensembles many different retrosynthesis models across both graph and SMILE string-based methods. As a result, our method is complementary to their framework and may be less resource intensive.

Table 2: Top-k Round-trip coverage and accuracy on USPTO-50k test dataset.

	Model	**Round-Trip Coverage**				**Round-Trip Accuracy**			
		k=1	k=3	k=5	k=10	k=1	k=3	k=5	k=10
TB	GLN	**82.5**	92.0	94.0	–	**82.5**	**71.0**	66.2	–
	LocalRetro	82.1	**92.3**	**94.7**	–	82.1	**71.0**	**66.7**	–
	RetroGFN	–	–	–	–	76.7	69.1	65.5	60.8
TF	MEGAN	78.1	88.6	91.3	–	78.1	67.3	61.7	–
	Graph2SMILES	–	–	–	–	76.7	56.0	46.4	–
	Retroformer$_{\text{aug}}$	–	–	–	–	78.6	71.8	67.1	–
	RetroBridge	85.1	95.7	97.1	97.7	85.1	73.6	67.8	56.3
	RETRO SYNFLOW-RS	88.9	97.3	98.4	98.9	88.9	73.9	69.1	61.8
	RETRO PRODFLOW-RS	**91.4**	**97.6**	**98.7**	**99.3**	**91.4**	**84.1**	**80.1**	**75.1**

[1] Chimera [27] an ensemble method combining multiple retrosynthesis models (TF + TB)

[2] RetroBridge was evaluated using $T = 500$ sampling steps as done in Igashov et al. [18]

However, exact match accuracy is limited because it does not capture the fact that multiple reactants (some of which may not exist in the dataset) can synthesize the same product molecule. Therefore, we evaluate RETRO PRODFLOW-RS on round-trip accuracy and compare it to the baselines present in [18] in addition to [13]. As shown in Table 2, RETRO PRODFLOW-RS is capable of generating diverse and feasible reactants, outperforming state-of-the-art methods on round-trip coverage and accuracy. Additionally, RETRO SYNFLOW-RS also achieves competitive results on round-trip coverage and accuracy.

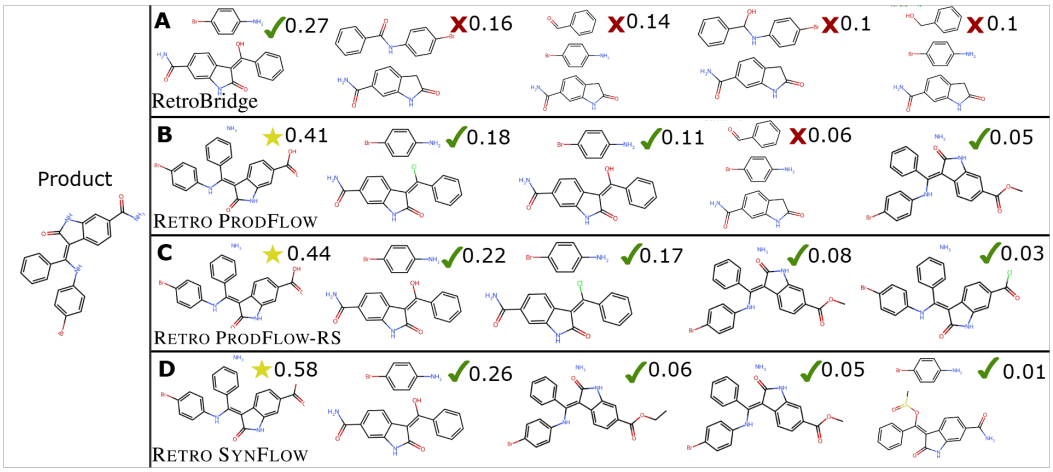


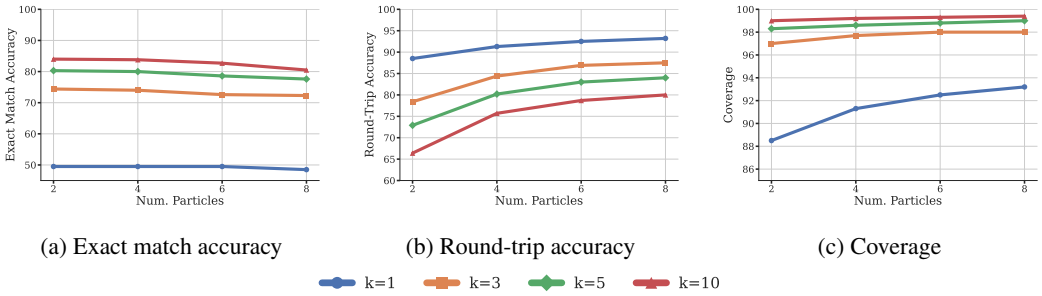
Figure 4: Top-5 reactants selected by each method. A star indicates an exact match, a checkmark indicates a round-trip match but not an exact match, and a cross means neither.

4.3 Ablation Studies

In Table 3, we show the performance of RETRO PRODFLOW-RS for the synthons to reactants generation task. Here, top-k accuracy refers to the proportion of synthons where the true set of reactants exists in the top-k predicted sets of reactants. Furthermore, we evaluate the effects of providing the product molecule as additional context to the flow matching model p_θ.

Table 3: Top-k accuracy for synthon completion task on USPTO-50k test dataset.

Model	1	3	5	10
RSF (w/o product)	59.7	75.5	79.1	82.0
RSF (w product)	**67.7**	**82.9**	**85.7**	**87.5**



(a) Exact match accuracy (b) Round-trip accuracy (c) Coverage

Figure 5: Performance of RETRO PRODFLOW-RS on the USPTO-50k validation set as we vary the number of particles for SMC resampling. We sample $N = 50$ reactants per product.

In Figure 5, we study the performance of RETRO PRODFLOW-RS by varying the number of particles in the SMC resampling procedure for FK-steering. We find that $K = 4$ particles already provides significant gains in round-trip accuracy and coverage with negligible reduction in exact match

accuracy. As we increase the number of particles further, we see noticeable increases in round-trip accuracy and coverage with a slight decrease in exact match accuracy.

Table 4: Comparing RETRO PRODFLOW-RS against baselines on the USPTO-50k test set.

	Round-Trip Coverage				Round-Trip Accuracy			
Model	$k=1$	$k=3$	$k=5$	$k=10$	$k=1$	$k=3$	$k=5$	$k=10$
RPF	84.4	95.3	96.9	97.7	84.4	72.8	66.8	57.6
RPF-RS	**91.4**	**97.6**	**98.7**	**99.3**	**91.4**	**84.1**	**80.1**	**75.1**
Greedy Sampling	89.4	97.0	98.4	99.2	89.4	79.0	73.0	66.2
RPF (400 reactants)	84.2	95.3	97.1	98.1	84.2	73.1	68.5	60.7

Next, we further demonstrate the benefits of reward-based steering with SMC resampling by comparing RETRO PRODFLOW-RS against two baselines. The greedy sampling approach does not perform any steering or resampling and instead selects the particle with the highest reward at the end of the generation process. Table 4 shows that RETRO PRODFLOW-RS outperforms greedy sampling, highlighting the superiority of SMC resampling. Furthermore, we also scale the computational budget of RETRO PRODFLOW by generating $N = 400$ reactants per product molecule. This has the same computational cost as RETRO PRODFLOW-RS, which uses $K = 4$ particles and samples 100 reactants per product molecule. Again, Table 4 shows that increasing the number of reactants sampled per product does not improve round-trip accuracy and coverage by a significant amount, highlighting the need for reward-based SMC steering. Additional ablation studies are available in Appendix B.

5 Related Works

Template-based. The approaches for single-step retrosynthesis can be divided into two main categories: template-based and template-free. Reaction templates are molecular subgraph patterns that encode pre-defined reaction rules to transform a target product into simpler reactants. They can be handcrafted by experts [15, 47] or extracted algorithmically from large databases [5]. The main challenge for template-based methods, such as Segler and Waller [36], Coley et al. [5], Dai et al. [9], is ranking and selecting the correct templates for a target molecule. More recently, Gaiński et al. [13] utilizes the recent GFlowNet framework to build RetroGFN, a model capable of composing existing reaction templates to explore the solution space of reactants beyond the dataset to increase feasibility and diversity.

Template-free. Templates provide a strong inductive bias, and template-based methods offer greater interpretability at the expense of generalization. On the other hand, template-free approaches directly transform products to reactants without pre-defined rules, providing more flexibility. Many works in this area frame the task as a sequence-to-sequence modelling problem on SMILES string representation of molecules [24, 57, 46, 48]. Another approach is to use a graph representation of molecules and transform product molecule graphs to reactant graphs [33, 39]. The recent work by Igashov et al. [18] builds a Markov Bridge model between the space of products and reactants. Also, Laabid et al. [22] employs absorbing state diffusion and builds a graph diffusion model to generate reactants. Other works leverage a combination of graph-based and SMILE representations of molecules [38, 49, 52]. In the context of multi-step retrosynthesis, prior works have used a forward-synthesis model to select for promising reactants [7, 37, 57].

Discrete Diffusion and FM. Discrete diffusion and flow matching are a powerful class of generative models that have demonstrated impressive results across various tasks, *e.g.* language modelling [28], symmetric group learning [56], and biological applications such as protein synthesis[1, 17], 3D molecule generation [10, 42], and DNA sequence design [44]. Traditionally, continuous-state diffusion models have also been employed for discrete data generation tasks such as graph synthesis [55, 53, 19, 29], despite relying on hard-coded post-processing steps.

6 Conclusion

In this work, we approached single-step retrosynthesis as learning the transport map between two intractable distributions and explored two different options for the source. We first introduced RETRO

PRODFLOW, a discrete flow matching model that transforms products into reactants. Next, we proposed RETRO SYNFLOW, the first flow matching model that transforms products to reactants through intermediary structures known as synthons. We demonstrated that synthons serve as a more informative source of distribution for generating reactants with RETRO SYNFLOW achieving 60% top-1 accuracy, beating previous SOTA methods for retrosynthesis modelling. Furthermore, we enhanced the diversity and feasibility of predictions by leveraging Feynman-Kac steering, an inference-time, reward-based steering method. We define the reward function using a forward-synthesis model motivated by round-trip accuracy. A reward-steered version of RETRO PRODFLOW achieves 80.1% top-5 round-trip accuracy, beating template-free SOTA methods by up to 19%.

Although template-based planning is constrained, it may remain preferable to many chemists due to its alignment with established reactants, reagents, and reaction types. Future work could explore incorporating template information to guide the generation process toward specific compound sets, enhancing the interpretability of our method.

7 Acknowledgements

This work was funded, in part, by the NSERC DG Grant (No. RGPIN-2022-04636), the Vector Institute for AI, Canada CIFAR AI Chairs, a Google Gift Fund, and the CIFAR Pan-Canadian AI Strategy through a Catalyst award. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through the Digital Research Alliance of Canada alliance.can.ca, and companies sponsoring the Vector Institute www.vectorinstitute.ai/#partners, and Advanced Research Computing at the University of British Columbia. Additional hardware support was provided by John R. Evans Leaders Fund CFI grant. AJB is partially supported by an NSERC Postdoc fellowship and supported by the EPSRC Turing AI World-Leading Research Fellowship No. EP/X040062/1 and EPSRC AI Hub No. EP/Y028872/1. QY is supported by UBC Four Year Doctoral Fellowships.

References

[1] J. Bose, T. Akhound-Sadegh, G. Huguet, K. Fatras, J. Rector-Brooks, C.-H. Liu, A. C. Nica, M. Korablyov, M. M. Bronstein, and A. Tong. SE(3)-Stochastic Flow Matching for Protein Backbone Generation. Oct. 2023. URL https://openreview.net/forum?id=kJFIH23hXb. (Cited on page 9)

[2] 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. *arXiv preprint arXiv:2402.04997*, 2024. (Cited on page 2)

[3] B. Chen, T. Shen, T. S. Jaakkola, and R. Barzilay. Learning to Make Generalizable and Diverse Predictions for Retrosynthesis, Oct. 2019. URL http://arxiv.org/abs/1910.09688. arXiv:1910.09688 [cs]. (Cited on page 1)

[4] S. Chen and Y. Jung. Deep Retrosynthetic Reaction Prediction using Local Reactivity and Global Attention. *JACS Au*, 1(10):1612–1620, Oct. 2021. doi: 10.1021/jacsau.1c00246. URL https://doi.org/10.1021/jacsau.1c00246. Publisher: American Chemical Society. (Cited on page 6)

[5] C. W. Coley, R. Barzilay, T. S. Jaakkola, W. H. Green, and K. F. Jensen. Prediction of Organic Reaction Outcomes Using Machine Learning. *ACS Central Science*, 3(5):434–443, May 2017. ISSN 2374-7943. doi: 10.1021/acscentsci.7b00064. URL https://doi.org/10.1021/acscentsci.7b00064. Publisher: American Chemical Society. (Cited on page 9)

[6] C. W. Coley, L. Rogers, W. H. Green, and K. F. Jensen. Computer-Assisted Retrosynthesis Based on Molecular Similarity. *ACS Central Science*, 3(12):1237–1245, Dec. 2017. ISSN 2374-7943. doi: 10.1021/acscentsci.7b00355. URL https://doi.org/10.1021/acscentsci.7b00355. Publisher: American Chemical Society. (Cited on page 1)

[7] C. W. Coley, D. A. ThomasIII, J. A. M. Lummiss, J. N. Jaworski, C. P. Breen, V. Schultz, T. Hart, J. S. Fishman, L. Rogers, H. Gao, R. W. Hicklin, P. P. Plehiers, J. Byington, J. S. Piotti, W. H. Green, A. J. Hart, T. F. Jamison, and K. F. Jensen. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science*, Aug. 2019. doi: 10.1126/science.aax1566. URL https://www.science.org/doi/10.1126/science.aax1566. Publisher: American Association for the Advancement of Science. (Cited on page 9)

[8] E. J. Corey and W. T. Wipke. Computer-Assisted Design of Complex Organic Syntheses. *Science*, 166(3902):178–192, Oct. 1969. doi: 10.1126/science.166.3902.178. URL https://www.science.org/doi/10.1126/science.166.3902.178. Publisher: American Association for the Advancement of Science. (Cited on page 1)

[9] H. Dai, C. Li, C. Coley, B. Dai, and L. Song. Retrosynthesis Prediction with Conditional Graph Logic Network. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html. (Cited on pages 1, 6, and 9)

[10] I. Dunn and D. R. Koes. Mixed Continuous and Categorical Flow Matching for 3D De Novo Molecule Generation, Apr. 2024. URL http://arxiv.org/abs/2404.19739. arXiv:2404.19739 [q-bio]. (Cited on page 9)

[11] V. P. Dwivedi and X. Bresson. A Generalization of Transformer Networks to Graphs, Jan. 2021. URL http://arxiv.org/abs/2012.09699. arXiv:2012.09699 [cs]. (Cited on page 16)

[12] F. Eijkelboom, G. Bartosh, C. A. Naesseth, M. Welling, and J.-W. van de Meent. Variational Flow Matching for Graph Generation. *arXiv preprint arXiv:2406.04843*, 2024. (Cited on page 3)

[13] P. Gaiński, M. Koziarski, K. Maziarz, M. Segler, J. Tabor, and M. Śmieja. RetroGFN: Diverse and Feasible Retrosynthesis using GFlowNets, Apr. 2025. URL http://arxiv.org/abs/2406.18739. arXiv:2406.18739 [cs]. (Cited on pages 6, 8, and 9)

[14] I. Gat, T. Remez, N. Shaul, F. Kreuk, R. T. Chen, G. Synnaeve, Y. Adi, and Y. Lipman. Discrete Flow Matching. *arXiv preprint arXiv:2407.15595*, 2024. (Cited on page 2)

[15] M. Hartenfeller, M. Eberle, P. Meier, C. Nieto-Oberhuber, K.-H. Altmann, G. Schneider, E. Jacoby, and S. Renner. A Collection of Robust Organic Synthesis Reactions for In Silico Molecule Design. *Journal of Chemical Information and Modeling*, 51(12):3093–3098, Dec. 2011. ISSN 1549-9596. doi: 10.1021/ci200379p. URL https://doi.org/10.1021/ci200379p. Publisher: American Chemical Society. (Cited on page 9)

[16] P. Holderrieth, M. S. Albergo, and T. Jaakkola. LEAPS: A discrete neural sampler via locally equivariant networks, Feb. 2025. URL http://arxiv.org/abs/2502.10843. arXiv:2502.10843 [cs]. (Cited on page 21)

[17] G. Huguet, J. Vuckovic, K. Fatras, E. Thibodeau-Laufer, P. Lemos, R. Islam, C.-H. Liu, J. Rector-Brooks, T. Akhound-Sadegh, M. Bronstein, A. Tong, and A. J. Bose. Sequence-Augmented SE(3)-Flow Matching For Conditional Protein Backbone Generation, Dec. 2024. URL http://arxiv.org/abs/2405.20313. arXiv:2405.20313 [cs]. (Cited on page 9)

[18] I. Igashov, A. Schneuing, M. Segler, M. M. Bronstein, and B. Correia. RetroBridge: Modeling Retrosynthesis with Markov Bridges. Oct. 2023. URL https://openreview.net/forum?id=770DetV8He. (Cited on pages 4, 6, 7, 8, 9, 16, and 17)

[19] J. Jo, S. Lee, and S. J. Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International conference on machine learning*, pages 10362–10383. PMLR, 2022. (Cited on page 9)

[20] E. Kim, D. Lee, Y. Kwon, M. S. Park, and Y.-S. Choi. Valid, Plausible, and Diverse Retrosynthesis Using Tied Two-Way Transformers with Latent Variables. *Journal of Chemical Information and Modeling*, 61(1):123–133, Jan. 2021. ISSN 1549-9596. doi: 10.1021/acs.jcim.0c01074. URL https://doi.org/10.1021/acs.jcim.0c01074. Publisher: American Chemical Society. (Cited on page 6)

[21] J. Kim, K. Shah, V. Kontonis, S. Kakade, and S. Chen. Train for the Worst, Plan for the Best: Understanding Token Ordering in Masked Diffusions, Mar. 2025. URL http://arxiv.org/abs/2502.06768. arXiv:2502.06768 [cs]. (Cited on page 21)

[22] N. Laabid, S. Rissanen, M. Heinonen, A. Solin, and V. Garg. Equivariant Denoisers Cannot Copy Graphs: Align Your Graph Diffusion Models, June 2025. URL http://arxiv.org/abs/2405.17656. arXiv:2405.17656 [cs]. (Cited on page 9)

[23] G. Landrum. RDKit: Open-Source Cheminformatics Software | BibSonomy. URL https://www.bibsonomy.org/bibtex/28d01fceeccd6bf2486e47d7c4207b108/salotz. (Cited on page 16)

[24] B. Liu, B. Ramsundar, P. Kawthekar, J. Shi, J. Gomes, Q. Luu Nguyen, S. Ho, J. Sloane, P. Wender, and V. Pande. Retrosynthetic Reaction Prediction Using Neural Sequence-to-Sequence Models. *ACS Central Science*, 3(10):1103–1113, Oct. 2017. ISSN 2374-7943. doi: 10.1021/acscentsci.7b00303. URL https://doi.org/10.1021/acscentsci.7b00303. Publisher: American Chemical Society. (Cited on page 9)

[25] L. Long, R. Li, and J. Zhang. Artificial Intelligence in Retrosynthesis Prediction and its Applications in Medicinal Chemistry. *Journal of Medicinal Chemistry*, 68(3):2333–2355, Feb. 2025. ISSN 0022-2623. doi: 10.1021/acs.jmedchem.4c02749. URL https://doi.org/10.1021/acs.jmedchem.4c02749. Publisher: American Chemical Society. (Cited on page 1)

[26] I. Loshchilov and F. Hutter. Decoupled Weight Decay Regularization, Jan. 2019. URL http://arxiv.org/abs/1711.05101. arXiv:1711.05101 [cs]. (Cited on page 17)

[27] K. Maziarz, G. Liu, H. Misztela, A. Tripp, J. Li, A. Kornev, P. Gaiński, H. Hoefling, M. Fortunato, R. Gupta, and M. Segler. Chemist-aligned retrosynthesis by ensembling diverse inductive bias models, Aug. 2025. URL http://arxiv.org/abs/2412.05269. arXiv:2412.05269 [cs]. (Cited on pages 6 and 7)

[28] S. Nie, F. Zhu, Z. You, X. Zhang, J. Ou, J. Hu, J. Zhou, Y. Lin, J.-R. Wen, and C. Li. Large Language Diffusion Models, Feb. 2025. URL http://arxiv.org/abs/2502.09992. arXiv:2502.09992 [cs]. (Cited on page 9)

[29] C. Niu, Y. Song, J. Song, S. Zhao, A. Grover, and S. Ermon. Permutation invariant graph generation via score-based generative modeling. In *International conference on artificial intelligence and statistics*, pages 4474–4484. PMLR, 2020. (Cited on page 9)

[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html. (Cited on page 16)

[31] F. Z. Peng, Z. Bezemek, S. Patel, J. Rector-Brooks, S. Yao, A. Tong, and P. Chatterjee. Path Planning for Masked Diffusion Model Sampling, Feb. 2025. URL http://arxiv.org/abs/2502.03540. arXiv:2502.03540 [cs]. (Cited on page 21)

[32] Y. Ren, H. Chen, Y. Zhu, W. Guo, Y. Chen, G. M. Rotskoff, M. Tao, and L. Ying. Fast Solvers for Discrete Diffusion Models: Theory and Applications of High-Order Algorithms, Feb. 2025. URL http://arxiv.org/abs/2502.00234. arXiv:2502.00234 [cs]. (Cited on page 21)

[33] M. Sacha, M. Błaż, P. Byrski, P. Dąbrowski-Tumański, M. Chromiński, R. Loska, P. Włodarczyk-Pruszyński, and S. Jastrzębski. Molecule Edit Graph Attention Network: Modeling Chemical Reactions as Sequences of Graph Edits. *Journal of Chemical Information and Modeling*, 61 (7):3273–3284, July 2021. ISSN 1549-9596. doi: 10.1021/acs.jcim.1c00537. URL https://doi.org/10.1021/acs.jcim.1c00537. Publisher: American Chemical Society. (Cited on pages 6 and 9)

[34] N. Schneider, N. Stiefl, and G. A. Landrum. What's What: The (Nearly) Definitive Guide to Reaction Role Assignment. *Journal of Chemical Information and Modeling*, 56(12):2336–2346, Dec. 2016. ISSN 1549-9596. doi: 10.1021/acs.jcim.6b00564. URL https://doi.org/10.1021/acs.jcim.6b00564. (Cited on page 6)

[35] P. Schwaller, R. Petraglia, V. Zullo, V. H. Nair, R. Andreas Haeuselmann, R. Pisoni, C. Bekas, A. Iuliano, and T. Laino. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chemical Science*, 11(12):3316–3325, 2020. doi: 10.1039/C9SC05704H. URL https://pubs.rsc.org/en/content/articlelanding/2020/sc/c9sc05704h. Publisher: Royal Society of Chemistry. (Cited on page 6)

[36] M. H. S. Segler and M. P. Waller. Neural-Symbolic Machine Learning for Retrosynthesis and Reaction Prediction. *Chemistry – A European Journal*, 23(25): 5966–5971, 2017. ISSN 1521-3765. doi: 10.1002/chem.201605499. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/chem.201605499. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/chem.201605499. (Cited on pages 2 and 9)

[37] M. H. S. Segler, M. Preuss, and M. P. Waller. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature*, 555(7698):604–610, Mar. 2018. ISSN 1476-4687. doi: 10.1038/nature25978. URL https://www.nature.com/articles/nature25978. Publisher: Nature Publishing Group. (Cited on page 9)

[38] S.-W. Seo, Y. Y. Song, J. Y. Yang, S. Bae, H. Lee, J. Shin, S. J. Hwang, and E. Yang. GTA: Graph Truncated Attention for Retrosynthesis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):531–539, May 2021. ISSN 2374-3468. doi: 10.1609/aaai.v35i1.16131. URL https://ojs.aaai.org/index.php/AAAI/article/view/16131. Number: 1. (Cited on pages 6 and 9)

[39] C. Shi, M. Xu, H. Guo, M. Zhang, and J. Tang. A Graph to Graphs Framework for Retrosynthesis Prediction. In *Proceedings of the 37th International Conference on Machine Learning*, pages 8818–8827. PMLR, Nov. 2020. URL https://proceedings.mlr.press/v119/shi20d.html. ISSN: 2640-3498. (Cited on pages 2, 5, 6, and 9)

[40] R. Singhal, Z. Horvitz, R. Teehan, M. Ren, Z. Yu, K. McKeown, and R. Ranganath. A General Framework for Inference-time Scaling and Steering of Diffusion Models, Jan. 2025. URL http://arxiv.org/abs/2501.06848. arXiv:2501.06848 [cs]. (Cited on pages 2 and 4)

[41] V. R. Somnath, C. Bunne, C. Coley, A. Krause, and R. Barzilay. Learning Graph Models for Retrosynthesis Prediction. In *Advances in Neural Information Processing Systems*, volume 34, pages 9405–9415. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/4e2a6330465c8ffcaa696a5a16639176-Abstract.html. (Cited on pages 5 and 6)

[42] Y. Song, J. Gong, M. Xu, Z. Cao, Y. Lan, S. Ermon, H. Zhou, and W.-Y. Ma. Equivariant Flow Matching with Hybrid Probability Transport for 3D Molecule Generation. In *Neural Information Processing Systems 2023*, Nov. 2023. URL https://openreview.net/forum?id=hHUZ5V9XFu. (Cited on page 9)

[43] M. Stanley and M. Segler. Fake it until you make it? Generative de novo design and virtual screening of synthesizable molecules. *Current Opinion in Structural Biology*, 82: 102658, Oct. 2023. ISSN 0959-440X. doi: 10.1016/j.sbi.2023.102658. URL https://www.sciencedirect.com/science/article/pii/S0959440X2300132X. (Cited on page 1)

[44] H. Stark, B. Jing, C. Wang, G. Corso, B. Berger, R. Barzilay, and T. Jaakkola. Dirichlet Flow Matching with Applications to DNA Sequence Design, May 2024. URL http://arxiv.org/abs/2402.05841. arXiv:2402.05841 [q-bio]. (Cited on page 9)

[45] F. Strieth-Kalthoff, F. Sandfort, M. H. Segler, and F. Glorius. Machine learning the ropes: principles, applications and directions in synthetic chemistry. *Chemical Society Reviews*, 49 (17):6154–6168, 2020. (Cited on page 1)

[46] R. Sun, H. Dai, L. Li, S. Kearnes, and B. Dai. Towards understanding retrosynthesis by energy-based models. In *Advances in Neural Information Processing Systems*, volume 34, pages 10186–10194. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/5470abe68052c72afb19be45bb418d02-Abstract.html. (Cited on pages 6 and 9)

[47] S. Szymkuć, E. P. Gajewska, T. Klucznik, K. Molga, P. Dittwald, M. Startek, M. Bajczyk, and B. A. Grzybowski. Computer-Assisted Synthetic Planning: The End of the Beginning. *Angewandte Chemie International Edition*, 55(20):5904–5937, 2016. ISSN 1521-3773. doi: 10.1002/anie.201506101. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.201506101. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/anie.201506101. (Cited on page 9)

[48] I. V. Tetko, P. Karpov, R. Van Deursen, and G. Godin. State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. *Nature Communications*, 11(1): 5575, Nov. 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-19266-y. URL https://www.nature.com/articles/s41467-020-19266-y. Publisher: Nature Publishing Group. (Cited on pages 1, 6, and 9)

[49] Z. Tu and C. W. Coley. Permutation Invariant Graph-to-Sequence Model for Template-Free Retrosynthesis and Reaction Prediction. *Journal of Chemical Information and Modeling*, 62(15):3503–3513, Aug. 2022. ISSN 1549-9596. doi: 10.1021/acs.jcim.2c00321. URL https://doi.org/10.1021/acs.jcim.2c00321. Publisher: American Chemical Society. (Cited on pages 6 and 9)

[50] U. V. Ucak, I. Ashyrmamatov, J. Ko, and J. Lee. Retrosynthetic reaction pathway prediction through neural machine translation of atomic environments. *Nature Communications*, 13(1): 1186, Mar. 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-28857-w. URL https://www.nature.com/articles/s41467-022-28857-w. Publisher: Nature Publishing Group. (Cited on page 2)

[51] C. Vignac, I. Krawczuk, A. Siraudin, B. Wang, V. Cevher, and P. Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022. (Cited on pages 16 and 17)

[52] Y. Wan, C.-Y. Hsieh, B. Liao, and S. Zhang. Retroformer: Pushing the Limits of End-to-end Retrosynthesis Transformer. In *Proceedings of the 39th International Conference on Machine Learning*, pages 22475–22490. PMLR, June 2022. URL https://proceedings.mlr.press/v162/wan22a.html. ISSN: 2640-3498. (Cited on page 9)

[53] B. Xu, Q. Yan, R. Liao, L. Wang, and L. Sigal. Joint generative modeling of scene graphs and images via diffusion models. *arXiv preprint arXiv:2401.01130*, 2024. (Cited on page 9)

[54] C. Yan, Q. Ding, P. Zhao, S. Zheng, J. YANG, Y. Yu, and J. Huang. RetroXpert: Decompose Retrosynthesis Prediction Like A Chemist. In *Advances in Neural Information Processing Systems*, volume 33, pages 11248–11258. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/819f46e52c25763a55cc642422644317-Abstract.html. (Cited on page 5)

[55] Q. Yan, Z. Liang, Y. Song, R. Liao, and L. Wang. Swingnn: Rethinking permutation invariance in diffusion models for graph generation. *arXiv preprint arXiv:2307.01646*, 2023. (Cited on page 9)

[56] Y. Zhang, D. Yang, and R. Liao. Symmetricdiffusers: Learning discrete diffusion on finite symmetric groups. *arXiv preprint arXiv:2410.02942*, 2024. (Cited on page 9)

[57] S. Zheng, J. Rao, Z. Zhang, J. Xu, and Y. Yang. Predicting Retrosynthetic Reactions Using Self-Corrected Transformer Neural Networks. *Journal of Chemical Information and Modeling*, 60(1):47–55, Jan. 2020. ISSN 1549-9596. doi: 10.1021/acs.jcim.9b00949. URL https://doi.org/10.1021/acs.jcim.9b00949. Publisher: American Chemical Society. (Cited on pages 6 and 9)

Supplementary Materials

Table of Contents

A Experimental details

In this section, we elaborate on our experimental setup. We view a set of molecules as a single potentially disconnected graph. We follow the procedure outlined in Igashov et al. [18] and introduce a "dummy" node as an atom type. The graph of reactant molecules has at least as many nodes as the product molecule graph. During training and inference for RETRO PRODFLOW, we append ten dummy nodes to each product molecule. This covers 99.4% of the reactions in the USPTO-50k test dataset. Following Igashov et al. [18], the remaining reactions are removed from the test data. During inference, these dummy nodes are potentially transformed into true atom nodes. Similarly, we also append ten dummy nodes to the synthon molecule graphs when using RETRO SYNFLOW. There are 16 atom types (not including dummy atoms) and 4 bond types (not including no bond). Our methods are implemented in PyTorch [30], and we also use an open-source software RDKit [23], for operations involving chemical reactions and molecular graphs.

A.1 Neural Network Model

We use a graph transformer network [11, 51] also used by Igashov et al. [18] to model the denoiser p_θ of the flow matching process. The denoiser model takes a noisy graph $(\mathbf{v}, \boldsymbol{E})$ and graph-level features $\mathbf{y}$ as input and outputs probabilities of graphs over the data distribution. In the case of RETRO PRODFLOW, the product molecule graph is also provided as input to the denoiser model. This is done by appending the product molecule graph's node feature vector and adjacency matrix to the node feature vector and adjacency matrix of the noisy graph. For RETRO SYNFLOW, both the product molecule graph and synthon molecule graph are provided as input.

The graph transformer network is similar to the standard transformer architecture and consists of a graph attention module depicted in Figure 6. The graph attention module takes in input node features $\mathbf{v}$, edge features $\boldsymbol{E}$, and graph-level features $\mathbf{y}$. The FiLM is defined as $\text{FiLM}(\boldsymbol{M}_1, \boldsymbol{M}_2) = \boldsymbol{M}_1\boldsymbol{W}_1 + (\boldsymbol{M}_1\boldsymbol{W}_2) \odot \boldsymbol{M}_2 + \boldsymbol{M}_2$ where $\boldsymbol{W}_1, \boldsymbol{W}_2$ are learnable weights. Also,

PNA is defined as $\text{PNA}(\mathbf{v}) = \text{cat}(\max(\mathbf{v}), \min(\mathbf{v}), \text{mean}(\mathbf{v}), \text{std}(\mathbf{v}))\boldsymbol{W}$ where $\boldsymbol{W}$ is a learnable weight.

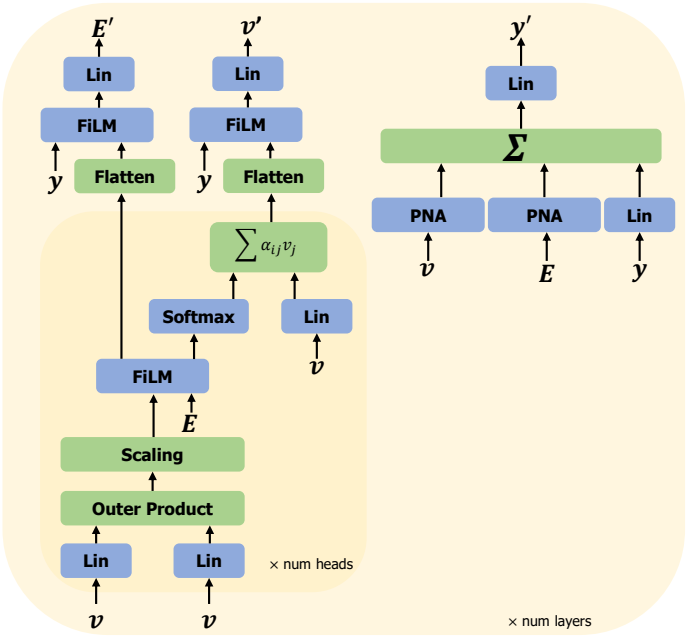


Figure 6: An overview of the graph attention module used in the graph transformer network. The output features are passed through a normalization layer and a fully connected layer at the end.

A.2 Training

Our training runs are done on either an NVIDIA RTX 3090 (24 GB of memory) or V100 (32 GB of memory). We train all of our models up to 600 epochs which can take up to 32 hours. We compute top-k accuracy metrics on a portion of the validation set every fixed number of epochs and select the checkpoint that has the highest top-1 accuracy. The models are trained using a batch size of 32. We use AdamW [26] with a learning rate of 0.0002.

A.3 Additional Features

We utilize the additional features proposed by [51] and used in Igashov et al. [18] as input to our models. We briefly state these features here for completeness.

Cycles. Message Passing Neural Networks cannot detect graph cycles, so we add them as features using formulas up to cycles of size 6. We compute node-level features (how many cycles does this node belong to) up to size 5 and graph-level features (how many cycles does this graph have) up to size 6. Fortunately, we can use formulas to compute the graph-level features $\mathbf{y}_i$ and node-level features $\mathbf{X}_i$, which can be efficiently computed on the GPU. In the following formulas, d denotes the

vector containing node degrees and $\|\cdot\|_F$ denotes the Frobenius norm:

$$
\begin{aligned}
\mathbf{X}_3 &= \text{diag}(\mathbf{A}^3)/2 \\
\mathbf{X}_4 &= \left(\text{diag}(\mathbf{A}^4) - d(d-1) - \mathbf{A}(d\mathbf{1}_n^T)\mathbf{1}_n\right)/2 \\
\mathbf{X}_5 &= \left(\text{diag}(\mathbf{A}^5) - 2\,\text{diag}(\mathbf{A}^3) \odot d - \mathbf{A}((\text{diag}(\mathbf{A}^3)\mathbf{1}_n^T)\mathbf{1}_n) + \text{diag}(\mathbf{A}^3)\right)/2 \\
\mathbf{y}_3 &= \mathbf{X}_3^T\mathbf{1}_n/3 \\
\mathbf{y}_4 &= \mathbf{X}_4^T\mathbf{1}_n/4 \\
\mathbf{y}_5 &= \mathbf{X}_5^T\mathbf{1}_n/5 \\
\mathbf{y}_6 &= \text{Tr}(\mathbf{A}^6) - 3\,\text{Tr}(\mathbf{A}^3 \odot \mathbf{A}^3) + 9\|\mathbf{A}(\mathbf{A}^2 \odot \mathbf{A}^2)\|_F \\
&\quad - 6\left\langle \text{diag}(\mathbf{A}^2), \text{diag}(\mathbf{A}^4)\right\rangle + 6\,\text{Tr}(\mathbf{A}^4) - 4\,\text{Tr}(\mathbf{A}^3) \\
&\quad + 4\,\text{Tr}(\mathbf{A}^2\mathbf{A}^2 \odot \mathbf{A}^2) + 3\|\mathbf{A}^3\|_F - 12\,\text{Tr}(\mathbf{A}^2 \odot \mathbf{A}^2) + 4\,\text{Tr}(\mathbf{A}^2).
\end{aligned}
$$

Spectral Features. We compute graph-level features: the number of connected components (which is the multiplicity of the 0 eigenvalue), and the first 5 non-zero eigenvalues of the graph Laplacian. We also compute node-level features: an estimate of the biggest connected component and the first two eigenvectors associated with the first two non-zero eigenvalues. Since molecular graphs in USPTO-50k have fewer than 100 nodes, the computation of these spectral features is not a concern.

B Additional Ablation Studies

B.1 Synthon Prediction

In this section, we provide some additional ablation studies examining the performance of our methods. In Table 5, we evaluate the performance of RETRO PRODFLOW-RS when sampling $N = 100$ reactants with $M = 2$ synthon predictions. We vary N_1, the number of reactant predictions generated for the highest ranking synthon prediction from the reaction center identification model. We verify that we need to generate more reactants for the highest-scoring synthon prediction to obtain competitive top-k accuracy.

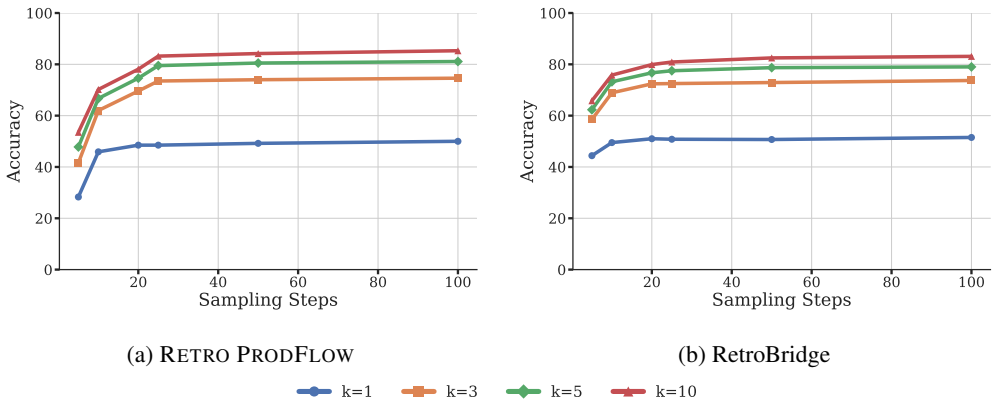
	Top-k Accuracy			
N_1	$k=1$	$k=3$	$k=5$	$k=10$
90	58.2	77.4	81.9	84.4
80	58.3	78.0	82.3	84.7
70	58.1	77.5	82.0	84.6
60	56.6	77.0	81.6	84.5
50	48.5	76.1	81.2	84.2

Table 5: Top-k accuracy (exact match) on the USPTO-50k validation dataset of RETRO SYNFLOW with $M = 2$ synthon predictions, sampling $N = 100$ reactants and varying the split. Given N_1, we have $N_2 = 100 - N_1$.

B.2 Sampling Steps

Next, we conduct a study to understand how the number of sampling steps affects the performance of flow matching compared to RetroBridge. We find that $T = 50$ sampling steps is sufficient for RETRO PRODFLOW to obtain SOTA results. Although RetroBridge achieves a higher accuracy at $T = 5$ or $T = 10$ sampling steps compared to flow matching, both methods fail to reach competitive performance. At $T = 50$ steps, RETRO PRODFLOW achieves some improvement over RetroBridge.

We also investigate the inference time required by RETRO SYNFLOW to sample $N = 100$ reactants with $T = 50$ sampling steps. We run this test on an RTX 3090 with 24 GB of memory. The average time required to sample 100 reactants for each product molecule in USPTO-50k test dataset is 5.46 ± 2.95 seconds.



(a) RETRO PRODFLOW (b) RetroBridge

Figure 7: The performance of RETRO PRODFLOW and RetroBridge as we vary the number of sampling steps.

B.3 Inference Time Comparison

In this section, we provide an ablation comparing the inference time scaling of RSF with the number of particles against other template-based and template-free models. Our model has 3.38 million parameters. We benchmark on an RTX 3090 with 24 GB of memory.

Method	**Mean time**
GLN	0.295
LocalRetro	0.022
RetroBridge	56.3
RSF (K=1)	5.4
RSF (K=2)	15.6
RSF (K=4)	30.6
RSF (K=6)	45.7
RSF (K=8)	53.9

Table 6: Average inference time (seconds) to sample 100 reactants for a given product in the USPTO-50k test set.

We note that 30 seconds for sampling a set of 100 reactants for a given product is a completely feasible time for applications of models like ours. In particular, this speed does not prevent the use of our method as a component of a multistep retrosynthesis planning pipeline. Additionally, the sampling time for RetroBridge with T=500 steps and 100 reactants is 50 seconds.

C Round-trip Visualization

This section provides a short case study analyzing the outputs of RETRO PRODFLOW and RETRO PRODFLOW-RS with $K = 2$ particles. We aim to understand how top-k accuracy can decrease when applying inference-time steering to guide generations towards outputs that optimize round-trip accuracy. We look at the top-1 accuracy results on the USPTO-50k test dataset for simplicity. Our main finding from this ablation is that it is still possible for RETRO PRODFLOW to generate reactants that are incorrect, *i.e.*, do not match the true reactants and are not feasible, *i.e.*, the forward synthesis model prediction does not match the ground-truth product. Table 7 shows how steering based on a round-trip reward affects the incorrect/correct prediction made by RETRO PRODFLOW. In total, 257 correct examples in the test dataset get converted to incorrect examples when applying reward steering. On the other hand, 251 incorrect examples are converted to correct examples when applying steering. As we increase the number of particles, *i.e.*, increase the strength of steering, this gap widens. This results in an overall decrease in exact-match accuracy as we force reactants towards

more diverse and feasible predictions. Figures 8 and 9 show the visualizations between the outputs of RETRO PRODFLOW and RETRO PRODFLOW-RS. The predicted product column refers to the prediction of the forward-synthesis model given the predicted reactants as input.

Table 7: Top-1 predicted reactants from RPF and RPF-RS quantified into four categories. The round-trip match column indicates whether the prediction made by RPF-RS is a round-trip match.

RPF	RPF-RS	Round-Trip Match	Count	Percentage
Correct	Incorrect	T	225	4.5
Correct	Incorrect	F	32	0.64
Incorrect	Correct	T	226	4.5
Incorrect	Correct	F	25	0.50
Correct	Correct	T	1848	36.9
Correct	Correct	F	388	7.7
Incorrect	Incorrect	T	1810	36.1
Incorrect	Incorrect	F	453	9.0

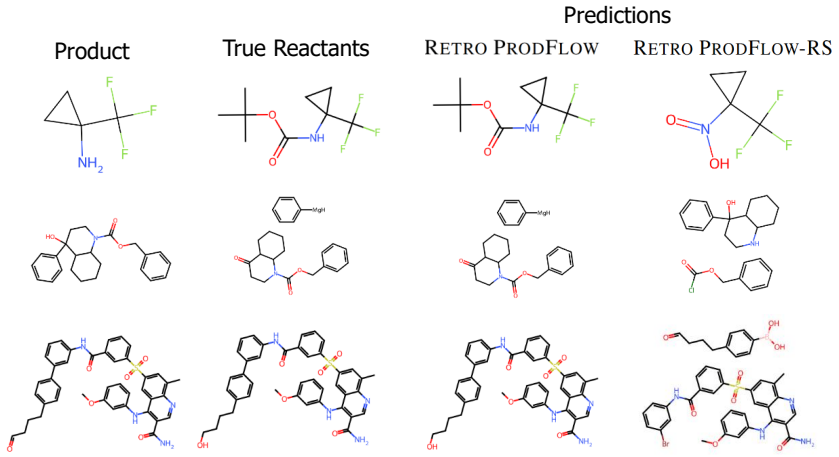


Figure 8: A visualization of reactions where RETRO PRODFLOW-RS generates an incorrect reactant prediction that is still feasible, while the RETRO PRODFLOW generates the correct reactant prediction. There are 225 examples in the test set that correspond to this case.

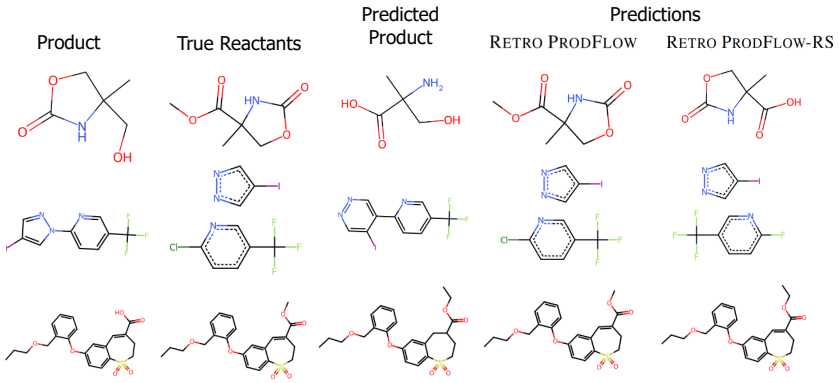


Figure 9: A visualization of reactions where RETRO PRODFLOW-RS generates an incorrect reactant prediction that is infeasible. RETRO PRODFLOW generates the correct reactant prediction. There are 32 examples in the test set that correspond to this case.

To obtain a better understanding of the errors of the forward-synthesis model, we evaluate the top-k accuracy of Molecular Transformer on the USPTO-50K dataset. Furthermore, we find that for the

non-steering outputs, 25% of the chemically valid reactants were misscored by the forward-synthesis model. For the FK-steered outputs, only 14% of the chemically valid reactants were misscored by the forward-synthesis model.

Top-1	**Top-3**	**Top-5**	**Top-10**
75.0	82.0	83.0	83.0

Table 8: Top-k accuracy of Molecular Transformer on the USPTO-50k evaluation set.

D Additional Sampling Scheme

Recently, there has been increasing interest in developing advanced adaptive sampling schemes for discrete diffusion and flow matching models [16, 32, 21, 31]. These developments aim to reduce errors in the generation process while improving inference speed. As explained in Section 2.1, we update the intermediate sample $\mathbf{x}_t$ using the following transition kernel, $\mathbf{x}^i_{t+h} \sim \text{Cat}(\mathbf{x}^i_{t+h}; \delta(\mathbf{x}^i_t) + hu^i_t(\mathbf{x}^i_{t+h}, \mathbf{x}_t))$, which is analogous to the Euler update step in continuous flow matching. Inspired by this analogy, we explore a higher-order sampling scheme [32] based on the Runge-Kutta (RK) method for solving ODEs. The update step for the method is as follows:

$$\hat{\mathbf{x}}^i_{t+h} \sim \text{Cat}(\hat{\mathbf{x}}^i_{t+h}; \delta(\mathbf{x}^i_t) + hu^i_t(\hat{\mathbf{x}}^i_{t+h}, \mathbf{x}_t)) \tag{3}$$

$$\mathbf{x}^i_{t+h} \sim \text{Cat}\left(\mathbf{x}^i_{t+h}; \delta(\mathbf{x}^i_t) + \frac{1}{2}hu^i_t(\mathbf{x}^i_{t+h}, \mathbf{x}_t) + \frac{1}{2}hu^i_{t+h}(\mathbf{x}^i_{t+h}, \hat{\mathbf{x}}^i_{t+h})\right). \tag{4}$$

This update step requires two model evaluations from p_θ instead of one. Table 9 compares the performance of RETRO PRODFLOW using this sampling scheme with 25 steps against RETRO PRODFLOW using the Euler-inspired sampling scheme with 50 steps.

Table 9: Top-k accuracy of RPF on the USPTO-50k test set sampling $N = 50$ reactants per product.

Model	1	3	5	10
RPF	49.6	73.3	79.6	83.6
RPF (RK 25 steps)	49.3	71.2	76.4	80.0
RPF (RK 50 steps)	49.3	72.5	78.6	82.3

E Predictions Visualization

We provide some additional visualizations of the generated reactants from our methods. In the following figures, an "E" represents an exact-match between the prediction reactant and an "R" indicates a round-trip match but not an exact-match. We show the top-3 reactants.

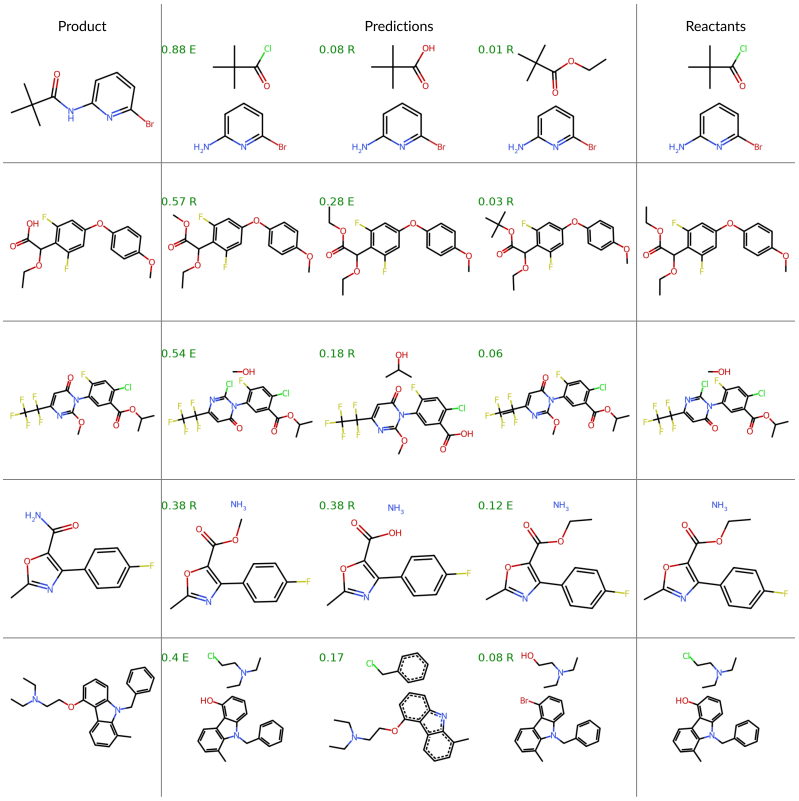


Figure 10: Visualizations of predictions made by RETRO PRODFLOW. Examples are taken from the USPTO-50k test set randomly.

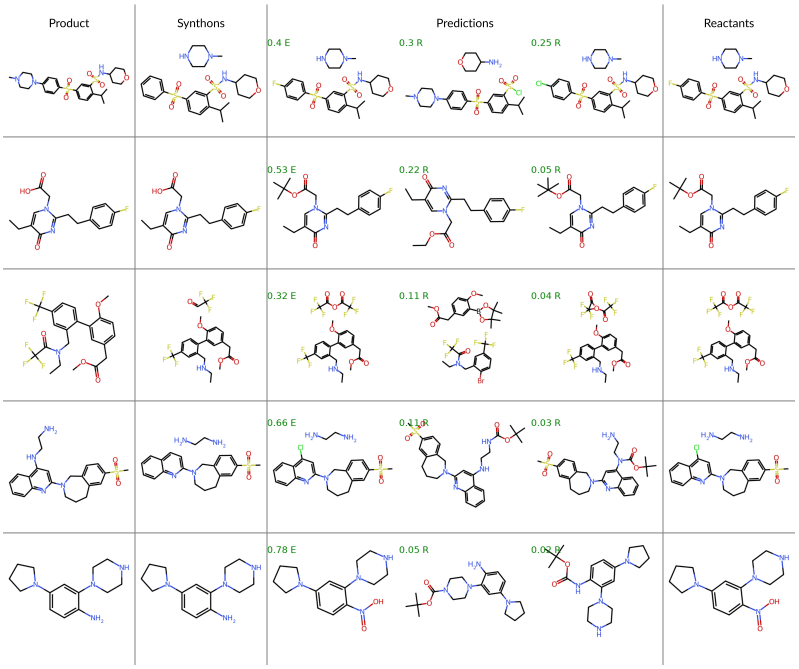


Figure 11: Visualizations of predictions made by RETRO SYNFLOW. Examples are taken from the USPTO-50k test set randomly.

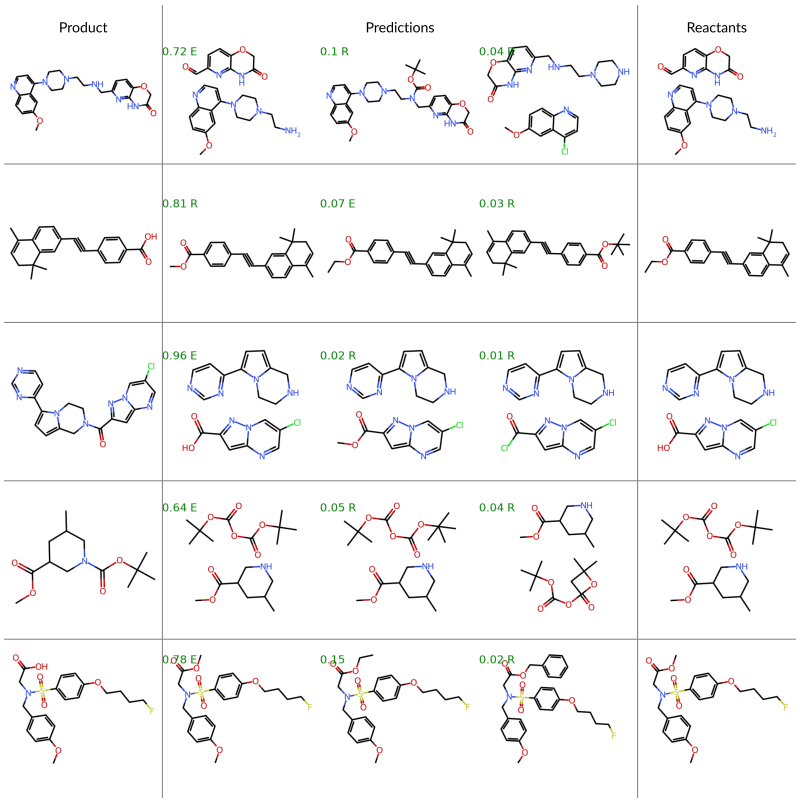


Figure 12: Visualizations of predictions made by RETRO PRODFLOW-RS. Examples are taken from the USPTO-50k test set randomly.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract are about experimental results which are shown in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, the limitations are mentioned in the conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper is about deep learning methods for retrosynthesis and does not include theory.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The methods are discussed in detail and the hyperparameters are provided in the experiments section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The dataset used is open-source and we will provide code upon paper acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Experimental details such as optimizer are discussed in the supplemental material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report error bars in our main result but not for additional ablation studies because that will be too computationally expensive.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the amount of compute resources used in the supplemental material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The work adheres to the NeurIPS code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses the impacts of our work in the introduction.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper which is about retrosynthesis does not pose such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All previous works that we use are properly credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, the code includes documentation.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Paper does not involve crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Paper does not involve crowdsourcing nor human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLMs are not involved in the core methodology.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.