

# Single-step Retrosynthesis via Reinforcement Learning from Forward Feedback

Zixun Lan<sup>1</sup>, Zhenfu Liu<sup>2</sup>, Yi Li<sup>1</sup>, Xinheng Wang<sup>1</sup>, Mingmin Chi<sup>3</sup>, Qibiao Wang<sup>2</sup>, Jie Zhu<sup>4</sup>, Xiaohui Tian<sup>4</sup>, Fei Ma<sup>1\*</sup>

**Abstract**—Single-step retrosynthesis is inherently a one-to-many inverse problem, yet most models are trained and evaluated against a single recorded reactant set. Reference-based training provides no explicit positive signal for unrecorded alternatives, while exact-match evaluation counts non-matching predictions as incorrect regardless of their chemical plausibility. We propose Retrosynthesis via Reinforcement Learning from Forward Feedback (RetroRLFF), which optimizes reactant generation using forward consistency with the target product rather than reference agreement alone. Starting from a supervised decoder-only Transformer, RetroRLFF generates complete reactant candidates and uses a frozen forward reaction model to predict products for comparison with the target. A sequence-level reward retains reference agreement to preserve recovery of recorded reactions and adds forward consistency to provide feedback for non-reference candidates, while penalizing invalid outputs. Direct Advantage-Based Policy Optimization then updates the retrosynthesis policy from this feedback without requiring human preference annotations or a preference-trained reward model. On USPTO-50K, RetroRLFF outperforms all 14 baselines in candidate-level round-trip accuracy at  $k \in \{1, 3, 5, 10\}$ . With approximately 50 million parameters, it reaches 90.1% top-1 round-trip accuracy, exceeding R-SMILES, the strongest competing baseline on this metric, by 8.5 percentage points. Relative to its supervised policy, round-trip accuracy improves from 83.1% to 90.1%, while exact-match accuracy is maintained at 57.9% versus 57.5%. To complement forward-model-based evaluation, we conduct an anonymized expert assessment on 131 randomly selected test cases, in which organic synthesis experts prefer RetroRLFF over EditRetro in 67.9% of comparisons. These results support forward feedback as an effective training signal for improving forward consistency while retaining top-1 reference recovery. Expert assessment provides complementary evidence of expert-assessed synthetic feasibility on the selected cases, although forward consistency and expert preference do not establish experimental feasibility.

**Index Terms**—Single-step retrosynthesis, reinforcement learning, forward reaction prediction, round-trip consistency, template-free retrosynthesis.

## I. INTRODUCTION

Single-step retrosynthesis prediction aims to identify possible reactant sets that can generate a target product. In computer-aided synthesis planning (CASP), these predictions define the precursor branches explored during multistep

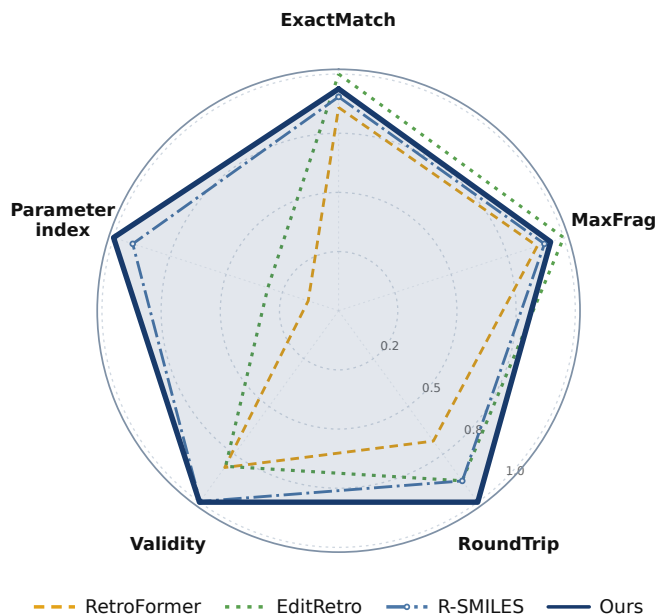


Fig. 1. Normalized comparison of RetroRLFF (Ours) with three state-of-the-art retrosynthesis models on USPTO-50K across four prediction metrics and a parameter index. The parameter index is proportional to the reciprocal of the retrosynthesis model’s parameter count; higher values indicate smaller models. Higher scores are better on all axes. RetroRLFF achieves higher round-trip consistency with fewer parameters.

search, thereby affecting search efficiency and the quality of the proposed routes [1], [2]. CASP supports applications in drug discovery, materials design, and automated chemical experiments.

Existing research has mainly developed three technical approaches to this task. Template-based methods extract local transformation rules from historical reactions and then select applicable templates through similarity retrieval, template classification, or neural symbolic reasoning [2]–[7]. Semi-template-based methods typically first identify the reaction center, decompose the target product into synthons—hypothetical precursor fragments obtained by disconnecting the product—and then complete these fragments into reactants [8]–[13]. Template-free methods do not rely on explicit reaction templates but directly learn the mapping between molecular sequences or molecular graphs. These template-free methods have expanded from SMILES-based autoregressive generation and graph editing to pre-training, diffusion models, and discrete flow matching [14]–[23].

Despite these architectural differences, most of the methods

\*corresponding author

<sup>1</sup> Xi’an Jiaotong-Liverpool University, Suzhou 215000, China

<sup>2</sup> Sichuan University of Science & Engineering, Zigong 643000, China

<sup>3</sup> Fudan University, Shanghai 200433, China

<sup>4</sup> Sinochem Digital Intelligence Technology Co., Ltd., Beijing, China

This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ.

Manuscript received April 19, 2021; revised August 16, 2021.

discussed above follow a similar supervision paradigm. Template classification loss, sequence cross-entropy, and graph-edit likelihood differ in form, but their supervision is centered on the reference reactions recorded in the dataset. Model performance is also commonly evaluated using exact-match accuracy, which measures whether the predicted reactant set matches the recorded reference. Thus, both training and evaluation primarily emphasize recovery of the recorded solutions.

However, single-step retrosynthesis is inherently a one-to-many inverse problem: a target product can often be synthesized from several different reactant sets, whereas patent datasets usually record only one or a few of these options. A prediction that does not match the recorded reference is therefore not necessarily chemically invalid. Manual evaluation and error analyses have shown that some non-matching predictions still represent plausible reaction schemes [3], [16], [24], [25]. Reference-based training provides no explicit positive signal for these unrecorded alternatives, while exact-match evaluation counts them as incorrect. Consequently, reference agreement does not fully capture the practical goal of identifying reactants that can produce the target product. This motivates a complementary training signal that can assess candidate reactants against the target product without requiring them to match the recorded reference.

To address the mismatch between training and application objectives, we propose Retrosynthesis via Reinforcement Learning from Forward Feedback (RetroRLFF). The framework uses forward reaction prediction to assess consistency between candidate reactants and the target product. Compared with inferring reactants from a product, predicting the major product from a given set of reactants generally involves less ambiguity [14], and models such as the Molecular Transformer have demonstrated strong forward prediction performance [26]. These observations motivate using forward prediction as scalable proxy feedback, although agreement between the predicted product and the target does not establish experimental feasibility. Round-trip consistency has been used to evaluate retrosynthetic candidates [1], [2], [19], [23], [24], [27]; RetroRLFF incorporates it into policy optimization for complete reactant-set generation.

Starting from a supervised decoder-only policy, RetroRLFF samples complete reactant candidates for each target product. A frozen forward reaction model predicts a product from each candidate for comparison with the target. We retain reference agreement to preserve recovery of recorded reactions, while forward consistency provides an additional learning signal for candidates that differ from the recorded reference. The sequence-level reward combines these two signals using canonicalized comparisons, with negative penalties for invalid reactants, invalid forward products, and processing failures. It requires neither human preference annotations nor a preference-trained reward model. Direct Advantage-Based Policy Optimization (DAPO) translates this feedback into token-level policy updates through critic-estimated advantages, with the supervised policy serving as a fixed reference [28].

The same frozen forward predictor supplies the consistency component of the training reward and computes round-trip accuracy during evaluation. We assess subsequent RLFF op-

timization by comparing the supervised and optimized policies while keeping their architecture and parameter count unchanged. Both use a standard decoder-only Transformer with approximately 50 million parameters and no reaction templates, graph-editing modules, or explicit reasoning components. As summarized in Fig. 1, RetroRLFF also achieves higher round-trip consistency than the three compared baselines while using a smaller retrosynthesis model. Fig. 2 summarizes the motivation, training process, and inference procedure.

We report candidate-level round-trip accuracy at  $k \in \{1, 3, 5, 10\}$ , defined as the fraction of all top- $k$  reactant candidates whose forward-predicted products match their respective targets after canonicalization. On USPTO-50K, RetroRLFF achieves the highest round-trip accuracy among the compared methods at every evaluated cutoff. Relative to the supervised policy, RLFF improves top-1 round-trip accuracy from 83.1% to 90.1%, while top-1 exact-match accuracy changes from 57.5% to 57.9%. Reward-configuration analysis further reveals tradeoffs between reference recovery and forward consistency.

Because round-trip consistency remains a model-based proxy for synthetic feasibility, we complement it with anonymized expert rankings based on synthetic feasibility. On a randomly selected subset of 131 USPTO-50K test cases, organic synthesis experts prefer RetroRLFF over EditRetro in 67.9% of comparisons. Qualitative case studies further illustrate plausible non-reference reactant alternatives. Together, these evaluations support improved forward consistency and the plausibility of selected predictions, while neither round-trip consistency nor expert preference establishes experimental success.

In summary, we outline our primary contributions as follows:

- We propose RetroRLFF, a reinforcement learning framework that incorporates forward reaction feedback into single-step retrosynthesis. It combines reference agreement with forward consistency to update the retrosynthesis policy, providing an additional learning signal for non-reference candidates without requiring human preference annotations.
- We demonstrate the effectiveness of RetroRLFF using a standard decoder-only Transformer with approximately 50M parameters. On USPTO-50K, it achieves the highest round-trip accuracy among the compared methods at all reported cutoffs; relative to its supervised policy, top-1 round-trip accuracy increases from 83.1% to 90.1%, while exact-match accuracy changes from 57.5% to 57.9%.
- We complement benchmark evaluation with anonymized expert assessment, qualitative analyses of non-reference reactant alternatives, and illustrations of recursive precursor expansion. On the randomly selected subset of 131 test cases, experts prefer RetroRLFF over EditRetro in 67.9% of comparisons and over the recorded reference in 71.8%, providing complementary evidence beyond exact reference matching.

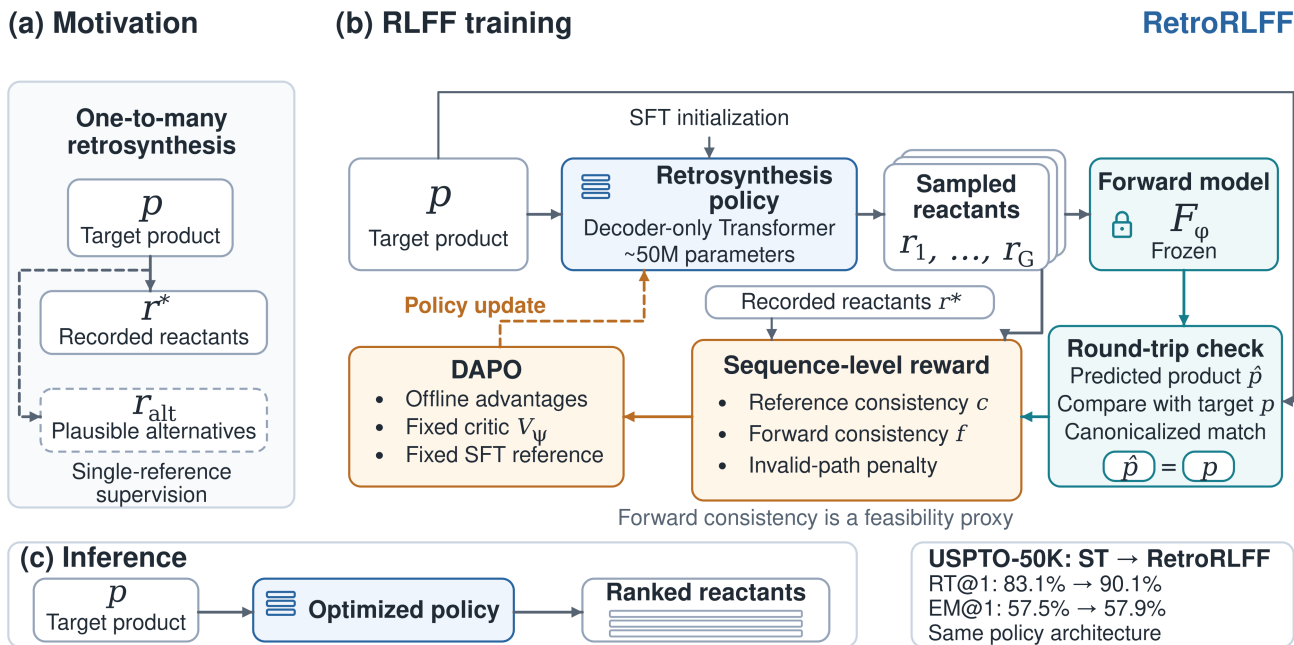


Fig. 2. Overview of RetroRLFF. (a) Single-step retrosynthesis is a one-to-many inverse problem: plausible reactant alternatives may differ from the recorded reference. (b) A supervised decoder-only policy samples complete reactant candidates, which are evaluated by a frozen forward model. Canonicalized reference agreement and forward consistency determine a sequence-level reward, with penalties for invalid evaluation paths. Direct Advantage-Based Policy Optimization (DAPO) updates the policy using offline advantages while keeping the critic and supervised reference policy fixed. (c) The optimized policy generates ranked reactant candidates. On USPTO-50K, RLFF improves top-1 round-trip accuracy from 83.1% to 90.1% while retaining exact-match accuracy (57.5% versus 57.9%) with the same policy architecture. Forward consistency is a proxy for synthetic feasibility, rather than experimental validation.

## II. RELATED WORK

### A. Template-Based and Semi-Template-Based Methods

Template-based methods represent a reaction as a local subgraph transformation. RetroSim retrieves reaction templates from similar reactions in the training set. NeuralSym formulates template selection as a neural classification problem, while GLN jointly models template applicability and the corresponding reactants [3]–[5]. LocalRetro replaces reaction-level templates with local atom and bond transformations [2]. MHNreact learns associations between products and templates with modern Hopfield networks, whereas RetroKNN uses local nearest-neighbor retrieval to improve predictions for less frequent reactions [6], [7]. Reaction templates provide explicit transformation rules and reduce the search space. However, their performance depends strongly on the quality and coverage of template extraction. Long-tailed template frequencies, costly subgraph matching, and the choice of template granularity can also limit generalization to unseen transformations.

Semi-template-based methods preserve the chemically meaningful process of bond disconnection followed by reactant completion. G2Gs and RetroXpert first identify reaction centers and split the product into synthons. They then reconstruct the reactants through graph generation or Transformer-based completion [8], [9]. GraphRetro uses stepwise graph transformations, Graph2Edits generates an autoregressive sequence of graph edits, and Retro-MTGR introduces multitask graph representation learning [10], [27], [29]. These structural

priors make intermediate decisions easier to interpret. However, an incorrect reaction center can propagate errors to the completion stage. Dependence on atom mapping and predefined edit sets also constrains the possible outputs. Moreover, each stage is usually supervised with the recorded bond disconnection or reactant set. The decomposition therefore does not remove the bias caused by single-reference supervision.

### B. Template-Free Retrosynthesis Methods

Sequence-based methods formulate retrosynthesis as a machine translation problem. The product SMILES is treated as the source sequence, and the reactant SMILES form the target sequence. Early seq2seq models used recurrent neural networks, whereas SCROP combined a Transformer with an auxiliary SMILES correction module [14], [15]. SMILES augmentation exposes the model to multiple valid representations of the same molecule and reduces its dependence on one fixed serialization [30]. R-SMILES aligns the root atoms on both sides of a reaction, thereby shortening the distance between corresponding atoms [18]. RetroFormer introduces local-global attention to capture both reaction-center information and global molecular context [19]. Chemformer first learns general chemical representations from unlabeled SMILES and is then fine-tuned for reaction prediction and retrosynthesis [20]. These methods are flexible and easy to scale. However, autoregressive decoding remains sensitive to token order, suffers from exposure bias, and may propagate early errors through the remaining sequence.

Graph-based methods operate more directly on molecular topology. MEGAN represents a reaction as a sequence of graph-edit operations [16]. Graph2SMILES combines a permutation-invariant graph encoder with a sequence decoder [17]. G2GT instead uses graph-to-graph attention and self-training to generate reactant graphs [31]. These models preserve structural information more explicitly and reduce their dependence on arbitrary SMILES order. Their main difficulties arise from the discrete action space, the ordering of graph edits, and the cost of searching over possible transformations.

Recent work has also explored non-autoregressive and iterative generative processes. EditRetro generates reactants by repeatedly applying explicit Levenshtein editing operations to the product SMILES [32]. Although it formulates this process as a Markov decision process, its reward is defined by the distance to the recorded reactant sequence. It therefore remains tied to reference supervision rather than forward reaction feedback. RetroBridge constructs a Markov bridge between product and reactant graph distributions [21]. DiffER uses categorical diffusion [22], while RetroSynFlow adopts discrete flow matching [23]. These approaches broaden the available sampling mechanisms and provide more control over candidate diversity. Nevertheless, their basic training objectives still rely mainly on paired product–reactant records. RetroSynFlow additionally uses Feynman–Kac steering with sequential Monte Carlo resampling and a learned forward-synthesis reward oracle. This signal is applied only during inference and does not update the parameters of the trained generator.

### C. Reinforcement Learning from Human and Model Feedback

Reinforcement learning makes it possible to optimize a reward defined over a complete output rather than the likelihood of individual reference tokens. Early preference-based reinforcement learning learned reward functions from pairwise human comparisons and then optimized a policy against the learned reward [33]. This idea was later applied to language generation. Stiennon et al. trained a summarization model with a reward learned from human preferences [34]. Instruct-GPT adopted a widely used three-stage pipeline consisting of supervised fine-tuning, reward-model training from ranked outputs, and policy optimization with reinforcement learning [35]. Proximal Policy Optimization (PPO) is commonly used in the final stage because its clipped objective limits the size of each policy update [36]. A Kullback–Leibler divergence penalty can further keep the optimized policy close to the supervised reference model [35]. In this way, RLHF moves the training signal from reproducing one reference output toward evaluating complete generations according to human preferences.

Human preference labels are costly and difficult to collect at scale. Reinforcement learning from AI feedback (RLAIF) addresses this limitation by replacing some human rankings with preferences produced by another model [37]. A standard RLAIF pipeline uses these preferences to train a reward model. Direct RLAIF instead obtains rewards directly from an AI evaluator during policy optimization, without fitting a

separate reward model [37]. This idea can be extended to task-specific evaluators that provide automatic feedback. However, the evaluator remains an imperfect proxy for the true objective. Strong optimization may exploit its errors, causing the proxy reward to increase while actual output quality stops improving or even decreases [38].

Model-based feedback has also been introduced into chemistry and retrosynthesis. RLSynC uses a standalone forward synthesis model to reward synthon completion within an offline–online multi-agent reinforcement learning framework [39]. RetroGFN trains a GFlowNet with a learned feasibility proxy so that it can sample diverse reactions beyond those recorded in the training set [40]. More broadly, ether0 applies reinforcement learning across a large collection of experimentally grounded chemistry tasks [41]. Round-Trip Reinforcement Learning uses the success of a forward–reverse transformation as a reward for improving the self-consistency of chemical language models [42]. These studies show that chemical models can learn from automatic evaluators rather than from exact reference matching alone. However, they differ in the generated object, the source of the reward, and the policy optimization algorithm.

RetroRLFF follows this feedback-driven view but is not RLHF in the strict sense. It does not require human preference comparisons or a reward model trained to reproduce such preferences. Given a target product  $p_i$ , the retrosynthesis policy generates a complete candidate reactant string  $\hat{r}_{i,j}$ . A forward reaction model then maps  $\hat{r}_{i,j}$  to a predicted product  $\hat{p}_{i,j} = F_\phi(\hat{r}_{i,j})$ . Agreement between  $\hat{p}_{i,j}$  and  $p_i$  provides the forward-consistency component of the reward used for DAPO. Unlike post-hoc round-trip evaluation or the inference-time steering used by RetroSynFlow, this reward directly updates the parameters of the retrosynthesis policy [23]. Unlike RLSynC, RetroRLFF optimizes complete reactant generation rather than only synthon completion. It is therefore more accurately described as reinforcement learning from forward-model feedback. The resulting reward measures forward-model-based consistency, not experimentally verified reaction feasibility.

## III. METHODOLOGY

Fig. 2 summarizes RetroRLFF. We first formulate single-step retrosynthesis and define reference-based and forward-consistency metrics. We then initialize a decoder-only policy through supervised training and optimize it using forward-feedback rewards and Direct Advantage-Based Policy Optimization. The following subsections detail the task, supervised initialization, reward construction, and policy update.

### A. Task Definition

Let  $\mathcal{D} = \{(p_i, r_i^*)\}_{i=1}^N$  denote a dataset of recorded reactions. Here,  $p_i$  is the SMILES of the target product,  $r_i^*$  is the dot-separated SMILES of its reference reactants, and  $N$  is the number of test samples.

A retrosynthesis model with parameters  $\theta$  defines a conditional policy  $\pi_\theta(r | p)$ . Given  $p_i$ , it produces an ordered list of  $K$  candidates,

$$\hat{\mathcal{R}}_i^K = (\hat{r}_{i,1}, \hat{r}_{i,2}, \dots, \hat{r}_{i,K}), \quad (1)$$

where  $\hat{r}_{i,j}$  is the  $j$ -th candidate ranked by the model score.

We use  $\mathcal{C}(\cdot)$  to denote SMILES canonicalization. For a string containing several dot-separated molecules,  $\mathcal{C}(\cdot)$  canonicalizes each molecule separately, sorts the resulting component strings, and then joins them with dots. The comparison is therefore independent of molecule order. We further define  $\mathcal{V}(s) \in \{0, 1\}$  as an indicator of whether every molecule in  $s$  can be parsed and canonicalized. Single-step retrosynthesis predicts a set of possible reactants from one product. Since several reactant sets may produce the same product,  $r_i^*$  is a recorded reference rather than the only possible chemical solution.

### B. Evaluation Metric

We report both reference-based exact matching accuracy and forward-model-based round-trip accuracy. Round-trip results can depend on candidate aggregation, forward decoding, and the treatment of invalid outputs. These differences can prevent direct comparison across studies. We therefore fix the complete evaluation protocol described below.

1) *Exact Matching Accuracy*: Each test product is assigned  $K$  ranked reactant candidates. For  $1 \leq k \leq K$ , Top- $k$  exact matching accuracy is defined as

$$\text{EM@}k = \frac{1}{N} \sum_{i=1}^N \mathbb{I}[\exists j \leq k : \mathcal{V}(\hat{r}_{i,j}) = 1 \wedge \mathcal{C}(\hat{r}_{i,j}) = \mathcal{C}(r_i^*)], \quad (2)$$

where  $\mathbb{I}[\cdot]$  is the indicator function. A test sample is correct if at least one of its first  $k$  candidates exactly matches the reference after canonicalization.

2) *Round-Trip Accuracy*: Let  $F_\phi$  be a forward reaction prediction model with fixed parameters  $\phi$ . It maps each predicted reactant string to one product string:

$$\hat{p}_{i,j} = F_\phi(\hat{r}_{i,j}). \quad (3)$$

The forward model always uses greedy decoding with `beam_size = 1` and `do_sample = False`. Thus, each reactant candidate produces exactly one deterministic forward prediction.

We define  $\text{RT@}k$  as a candidate-level metric:

$$\text{RT@}k = \frac{1}{Nk} \sum_{i=1}^N \sum_{j=1}^k \mathbb{I}[\mathcal{V}(\hat{r}_{i,j}) \wedge \mathcal{V}(\hat{p}_{i,j}) \wedge \mathcal{C}(\hat{p}_{i,j}) = \mathcal{C}(p_i)]. \quad (4)$$

Each reactant candidate is evaluated independently. It is a round-trip success if its forward prediction matches the original target product after canonicalization. The denominator is  $Nk$ , so all top- $k$  candidates from all samples receive equal weight. Invalid reactants, invalid forward products, parsing errors, and failed model calls are counted as failures. This detailed protocol is intended to reduce differences caused by inconsistent implementations of round-trip accuracy and to support fair comparisons across studies.

### C. Pretraining

Let  $\mathcal{D}_{\text{tr}} = \{(p_i, r_i^*)\}_{i=1}^M$  be the training set containing  $M$  reaction pairs. We use a standard decoder-only Transformer to learn the mapping:  $p_i \rightarrow r_i^*$ .

Both sides are represented by their SMILES strings. Let  $\mathcal{T}(\cdot)$  denote the tokenizer. The tokenized product and reference reactants are

$$x_i = \mathcal{T}(p_i), \quad y_i = \mathcal{T}(r_i^*) = (y_{i,1}, \dots, y_{i,T_i}), \quad (5)$$

where  $T_i$  is the number of reactant tokens. Conditioned on  $x_i$  and the previously generated tokens  $y_{i,<t}$ , the model predicts the next reactant token. The supervision training objective is

$$\mathcal{L}_{\text{ST}}(\theta) = -\frac{1}{M} \sum_{i=1}^M \sum_{t=1}^{T_i} \log \pi_\theta(y_{i,t} | x_i, y_{i,<t}). \quad (6)$$

The optimized parameters are denoted by  $\theta_{\text{ST}}$ .

We deliberately use a simple decoder-only architecture. It contains no reaction templates, graph modules, or additional handcrafted features. Its parameter count is kept below that of the SOTA single-step retrosynthesis models. This controlled setup reduces the effects of model scale and specialized components. It allows us to focus on how RLFF changes reference consistency and synthetic feasibility. The SMILES augmentation and reranking strategy follows common settings from previous work [18], [30].

### D. Reinforcement Learning from Forward Feedback

RLFF starts from the pretrained supervision policy. For each product, the retrosynthesis policy samples a group of complete reactant candidates. The fixed forward model then evaluates each candidate against the target product. Sampling is used for retrosynthesis rollouts, whereas  $F_\phi$  always uses deterministic greedy decoding.

1) *Reward Function*: The reward combines reference consistency and forward consistency. For a candidate reactant string  $r$ , reference consistency is

$$c_i(r) = \mathbb{I}[\mathcal{C}(r) = \mathcal{C}(r_i^*)]. \quad (7)$$

Forward consistency is defined as

$$f_i(r) = \mathbb{I}[\mathcal{C}(F_\phi(r)) = \mathcal{C}(p_i)]. \quad (8)$$

It serves as a proxy for synthetic feasibility under the forward model, but it does not prove that the reaction will succeed experimentally.

We also define the validity of the complete evaluation path:

$$v_i(r) = \mathbb{I}[\mathcal{V}(r) = 1 \wedge \mathcal{V}(F_\phi(r)) = 1]. \quad (9)$$

A failed forward call is treated as an invalid output, and any canonicalized equality involving an invalid string is false. The reward is

$$R_i(r) = \begin{cases} r_{\text{inv}}, & v_i(r) = 0, \\ \alpha c_i(r) + \beta f_i(r) + \gamma c_i(r) f_i(r) + \delta, & v_i(r) = 1, \end{cases} \quad (10)$$

where  $r_{\text{inv}} < 0$  penalizes invalid outputs and processing errors. The nonnegative coefficients  $\alpha$  and  $\beta$  weight reference

agreement and forward consistency, respectively;  $\gamma$  controls their interaction, and  $\delta$  is a constant offset for valid evaluation paths.

A valid candidate receives  $\alpha + \beta + \gamma + \delta$  when both conditions hold,  $\alpha + \delta$  for reference-only agreement,  $\beta + \delta$  for forward-only agreement, and  $\delta$  when neither condition holds.

2) *Direct Advantage-Based Policy Optimization*: The sequence-level reward defined above is used as the terminal signal for policy optimization. Following DAPO [28], we first form an offline advantage dataset from sampled rollouts. A state  $s$  contains the target product and the current reactant prefix, while an action  $a$  is the next reactant token. A separately trained critic  $V_\psi$  estimates the value of the resulting prefix from completions scored by  $R_i(r)$ . The advantage  $\hat{A}(s, a)$  is obtained by centering these values over the candidate actions sampled for the same state. This provides step-specific credit instead of assigning the same terminal reward to every token.

We set the frozen reference policy to  $\pi_{\text{ref}} = \pi_{\theta_{\text{ST}}}$  and minimize

$$\mathcal{L}_{\text{DAPO}}(\theta) = \frac{1}{2} \mathbb{E}_{(s, a, \hat{A}) \sim \mathcal{D}_{\text{adv}}} \left[ \left( \frac{\hat{A}(s, a)}{\tau} - \log \frac{\pi_\theta(a | s)}{\pi_{\text{ref}}(a | s)} \right)^2 \right]. \quad (11)$$

Here,  $\mathcal{D}_{\text{adv}}$  contains offline state-action-advantage tuples rather than pairwise preference labels, and  $\tau > 0$  is the temperature coefficient controlling deviation from the reference policy. The loss fits the log probability ratio to the scaled advantage, increasing the probability of advantageous actions and suppressing disadvantageous ones. We update  $\theta$  by gradient descent while keeping  $\pi_{\text{ref}}$  and  $V_\psi$  fixed.

#### IV. EXPERIMENTS

We evaluate RetroRLFF from seven complementary perspectives:

(1) We compare overall performance on USPTO-50K using Exact Matching Accuracy, Round-Trip Accuracy, and Maximal Fragment Accuracy at  $k \in \{1, 3, 5, 10\}$ . These metrics assess reference recovery and candidate-level forward consistency, providing complementary views of prediction quality.

(2) We compare the supervised training policy with three subsequent RLFF configurations to assess the effect of reinforcement learning and different reward coefficients. The analysis examines the balance between reference agreement and forward consistency while keeping the interaction coefficient, constant offset, and invalid-output penalty fixed.

(3) We evaluate round-trip accuracy across non-ring, ring-opening, and ring-forming reactions, comparing RetroRLFF with R-SMILES, EditRetro, G2Retro, and RetroFormer to examine performance across different structural transformations and candidate ranks.

(4) We investigate architectural sensitivity by varying the parameter count, number of Transformer layers, and number of attention heads, assessing the stability of both exact-match and round-trip performance across five configurations.

(5) We conduct an anonymized human expert evaluation on 131 randomly selected test instances. Experts rank the

RetroRLFF predictions, EditRetro predictions, and recorded reference reactants by synthetic feasibility, allowing ties. We report rank distributions and pairwise preferences.

(6) We recursively apply RetroRLFF to six representative target molecules to illustrate successive precursor predictions, candidate ranks, and branching patterns in multistep retrosynthetic exploration.

(7) We examine five qualitative cases in which RetroRLFF proposes alternatives to the recorded reactants, discussing their chemical plausibility, potential synthetic advantages, and limitations beyond exact reference matching.

##### A. Dataset

We use USPTO-50K, a standard benchmark for single-step retrosynthesis derived from reactions in United States patents [3], [5], [14]. The curated set contains 50,016 atom-mapped reactions assigned to ten broad classes: heteroatom alkylation and arylation, acylation and related processes, carbon-carbon bond formation, heterocycle formation, protections, deprotections, reductions, oxidations, functional group interconversion, and functional group addition.

Following the fixed protocol used by the Conditional Graph Logic Network (GLN) [5], we use 40,008 reactions for training, 5,001 for validation, and 5,007 for testing. These subsets account for approximately 80%, 10%, and 10% of the benchmark, respectively. For evaluation, dot-separated reactants are processed using the component-wise canonicalization  $\mathcal{C}(\cdot)$  defined in the Methodology section.

##### B. Baselines

We compare RetroRLFF with the 14 retrosynthesis baselines reported in Table I, covering template-based, semi-template-based, and template-free approaches. Template-based baselines include RetroSim [3], NeuralSym [4], GLN [5], and LocalRetro [2]. The comparison also includes RetroXpert [9], GraphRetro [10], RetroPrime [24], and G2Retro [25], together with MEGAN [16], GTA [43], RetroFormer [19], Graph2Edits [27], R-SMILES [18], and EditRetro [32], spanning reaction-center prediction, graph editing, and sequence generation. For reaction-group analysis, we compare against R-SMILES, EditRetro, G2Retro, and RetroFormer. EditRetro serves as the competing model in human expert evaluation. We additionally use our supervision training policy (ST, without RL) as the internal baseline for assessing subsequent RLFF optimization.

##### C. Evaluation Metrics

We use the notation established in the Methodology section. Top- $k$  Exact Matching Accuracy (EM@ $k$ ) measures reference consistency: a test product is correct if at least one of its first  $k$  ranked candidates is valid and equals the recorded reactant set after canonicalization. Top- $k$  Round-Trip Accuracy (RT@ $k$ ) measures forward-model-based synthetic feasibility: every one of the  $Nk$  candidates is evaluated independently, and a success requires the deterministic forward prediction to equal the original product. Their complete definitions, invalid-output rules, and normalization conventions are given in the preceding

evaluation-metric subsection and are not repeated here. The forward model  $F_\phi$  is a pretrained Molecular Transformer [26]. The same frozen checkpoint is used for both training-reward computation and round-trip evaluation.

In this paper, the definitions of the other two evaluation metrics are as follows:

1) *Maximal Fragment Accuracy*: Maximal Fragment Accuracy (MaxFrag) evaluates whether a prediction recovers the principal reactant fragment, while ignoring differences in the remaining components [30]. Let  $\mathcal{M}(s)$  select the component of a valid dot-separated reactant string  $s$  whose canonical SMILES has the greatest character length. Top- $k$  MaxFrag Accuracy is

$$\text{MaxFrag}@k = \frac{1}{N} \sum_{i=1}^N \mathbb{I} \left[ \exists j \leq k : \mathcal{V}(\hat{r}_{i,j}) = 1 \wedge \mathcal{C}(\mathcal{M}(\hat{r}_{i,j})) = \mathcal{C}(\mathcal{M}(r_i^*)) \right]. \quad (12)$$

The symbols  $N$ ,  $\hat{r}_{i,j}$ ,  $r_i^*$ ,  $\mathcal{V}$ ,  $\mathcal{C}$ , and  $\mathbb{I}$  have the same meanings as in the Methodology section. Invalid predictions cannot produce a match. A larger MaxFrag@ $k$  is better; it indicates that the main recorded reactant is recovered by at least one candidate within the first  $k$  ranks.

2) *Validity*: Validity measures whether the model generates reactant SMILES that can be processed as molecular structures. Following the definition of  $\mathcal{V}(\cdot)$  in the Methodology section, Top- $k$  Validity is defined as

$$\text{Valid}@k = \frac{1}{Nk} \sum_{i=1}^N \sum_{j=1}^k \mathbb{I}[\mathcal{V}(\hat{r}_{i,j}) = 1]. \quad (13)$$

A candidate is valid if RDKit can parse and canonicalize every molecule in its dot-separated reactant string. Each of the first  $k$  candidates is checked independently. The denominator is  $Nk$ , so every candidate has the same weight, and an invalid candidate is counted as a failure. A higher Valid@ $k$  means that the model produces fewer malformed reactant SMILES.

#### D. Implementation Details

RetroRLFF uses a standard decoder-only Transformer with six layers, eight attention heads, and approximately 50 million parameters. The architecture is retained during supervision training (ST) and subsequent RLFF optimization, so the gains over the supervised policy are obtained without increasing its parameter count. Unless otherwise specified, we set  $\alpha = 1.8$  and  $\beta = 1.2$  for reference consistency and forward consistency, respectively. The interaction coefficient  $\gamma$  and constant offset  $\delta$  are both set to  $-1$ ; invalid evaluation paths receive the direct penalty  $r_{\text{inv}} = -1$ . The frozen forward model uses deterministic greedy decoding.

ST runs on two Intel Xeon Gold 6133 CPUs at 2.50 GHz and four NVIDIA GeForce RTX 4090 GPUs, each with 24 GB of memory. RLFF runs on eight Intel Xeon Gold 6240 CPUs at 2.60 GHz and twelve NVIDIA Quadro RTX 6000 GPUs with 24 GB each; nine GPUs support policy fine-tuning and sampling, while three perform forward-model scoring. Early stopping occurs at 325,000 ST steps and 130,000 RLFF steps.

#### E. Results and Discussion

1) *Overall Performance Comparison*: Table I compares RetroRLFF with template-based, semi-template-based, and template-free methods on the USPTO-50K test set described in Section IV-A. We report  $k \in \{1, 3, 5, 10\}$  for three complementary criteria. The forward model evaluates each reactant candidate separately for RT@ $k$ , whereas EM@ $k$  and MaxFrag@ $k$  count a test product once if any of its first  $k$  predictions matches the recorded full reactant set or its maximal fragment, respectively (Section IV-C). Consequently, the decreasing RT@ $k$  values at larger  $k$  reflect the average consistency of a wider candidate pool; they are not a top- $k$  success rate over products.

RetroRLFF obtains the highest round-trip consistency at every reported cutoff: 90.1%, 78.3%, 72.0%, and 62.7% at  $k = 1, 3, 5, 10$ , respectively. The strongest competing round-trip results are 81.6%, 74.5%, 69.9%, and 59.1% from R-SMILES, so the corresponding absolute gains are 8.5, 3.8, 2.1, and 3.6 percentage points. In particular, the 90.1% top-1 result indicates that the first-ranked reactant set more frequently regenerates the target under the fixed forward model. The advantage persists when lower-ranked candidates are included, although the average round-trip rate decreases for RetroRLFF from 90.1% at  $k = 1$  to 62.7% at  $k = 10$ .

The reference-based metrics give a more qualified picture. RetroRLFF achieves 57.9% EM@1 and 61.2% MaxFrag@1, compared with EditRetro’s leading 60.8% and 65.8%, respectively. At  $k = 10$ , RetroRLFF reaches 86.4% exact match and 88.3% maximal-fragment accuracy, while LocalRetro reaches 91.0% and 92.8%. Thus, the round-trip gains do not imply a gain in recovering the single recorded reactant set or its largest fragment. The distinct patterns are consistent with the motivation for forward feedback: a prediction can agree with the target under the forward model without reproducing the recorded reference.

2) *Ablation Studies and Reward Coefficient Analysis*: Table II compares supervised training (ST, without RL) with three RLFF configurations on USPTO-50K, using top-1 Exact Matching Accuracy (EM@1) and Round-Trip Accuracy (RT@1). ST+RL denotes subsequent optimization of the supervised policy through Direct Advantage-Based Policy Optimization. All three ST+RL configurations fix  $\gamma = \delta = r_{\text{inv}} = -1$  and vary only  $\alpha$  and  $\beta$ ; invalid evaluation paths therefore receive the same penalty. This comparison examines the RL stage and coefficient settings rather than isolating individual reward terms.

With  $(\alpha, \beta) = (1.8, 1.2)$ , RLFF improves round-trip accuracy from 83.1% to 90.1%, a gain of 7.0 percentage points, while exact-match accuracy changes from 57.5% to 57.9%. This setting therefore improves forward consistency while retaining the observed level of reference recovery, matching the reported RetroRLFF results in Table I.

For valid candidates, the reward vectors for  $(c_i(r), f_i(r)) = (1, 1), (1, 0), (0, 1), (0, 0)$  are  $(1, 0.8, 0.2, -1), (1, 1, 0, -1)$ , and  $(2, 1, 1, -1)$  under  $(\alpha, \beta) = (1.8, 1.2), (2, 1), (2, 2)$ , respectively. The first setting rewards forward-consistent non-reference predictions while favoring reference agreement. The second assigns zero reward to forward-only agreement; the

TABLE I

OVERALL PERFORMANCE ON THE USPTO-50K TEST SET. ROUNDTrip REPORTS CANDIDATE-LEVEL FORWARD-MODEL CONSISTENCY AMONG THE FIRST  $k$  PREDICTIONS (RT@ $k$ ); EXACTMATCH AND MAXFRAG REPORT THE PROPORTIONS OF PRODUCTS WITH AT LEAST ONE MATCHING PREDICTION AMONG THE FIRST  $k$  (EM@ $k$  AND MaxFrag@ $k$ ). ALL ENTRIES ARE PERCENTAGES. BOLD VALUES INDICATE THE HIGHEST RESULT IN EACH COLUMN.

| Model                      | Top- $k$ Performance (%) |             |             |             |             |             |             |             |             |             |             |             |
|----------------------------|--------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                            | RoundTrip                |             |             |             | ExactMatch  |             |             |             | MaxFrag     |             |             |             |
|                            | 1                        | 3           | 5           | 10          | 1           | 3           | 5           | 10          | 1           | 3           | 5           | 10          |
| <b>Template-Based</b>      |                          |             |             |             |             |             |             |             |             |             |             |             |
| RetroSim                   | 61.1                     | 55.3        | 52.4        | 47.6        | 37.3        | 54.7        | 63.3        | 74.1        | 40.5        | 58.9        | 67.9        | 78.7        |
| NeuralSym                  | 66.2                     | 67.2        | 64.8        | 51.4        | 44.4        | 65.3        | 72.4        | 78.9        | 53.6        | 74.6        | 81.5        | 87.8        |
| GLN                        | 66.0                     | 63.2        | 61.4        | 57.4        | 52.5        | 74.6        | 80.5        | 86.9        | 57.2        | 72.4        | 78.2        | 84.0        |
| LocalRetro                 | 65.6                     | 61.9        | 59.8        | 56.0        | 53.1        | 76.1        | 84.4        | <b>91.0</b> | 57.4        | 79.4        | 86.9        | <b>92.8</b> |
| <b>Semi-Template-Based</b> |                          |             |             |             |             |             |             |             |             |             |             |             |
| RetroXpert                 | 66.5                     | 42.8        | 36.1        | 30.3        | 50.4        | 61.1        | 62.3        | 63.4        | 59.1        | 73.7        | 78.0        | 83.1        |
| GraphRetro                 | 65.1                     | 59.5        | 55.5        | 48.5        | 53.7        | 68.3        | 72.2        | 75.5        | 58.5        | 71.6        | 75.0        | 77.7        |
| RetroPrime                 | 80.4                     | 62.6        | 53.9        | 43.3        | 51.4        | 70.8        | 74.0        | 76.1        | 56.7        | 75.3        | 79.5        | 81.4        |
| G2Retro                    | 63.7                     | 58.1        | 54.1        | 46.3        | 54.1        | 74.1        | 81.2        | 86.7        | 56.4        | 75.7        | 81.4        | 86.5        |
| <b>Template-Free</b>       |                          |             |             |             |             |             |             |             |             |             |             |             |
| MEGAN                      | 65.6                     | 61.0        | 57.6        | 51.0        | 48.1        | 70.7        | 78.4        | 86.1        | 53.3        | 74.9        | 82.3        | 88.5        |
| GTA                        | 65.5                     | 54.2        | 47.2        | 36.8        | 51.1        | 67.6        | 74.8        | 81.6        | 56.7        | 75.4        | 79.9        | 81.7        |
| RetroFormer                | 64.8                     | 60.6        | 57.2        | 49.1        | 53.2        | 71.1        | 76.6        | 82.1        | 58.0        | 70.8        | 74.8        | 78.4        |
| Graph2Edits                | 64.9                     | 58.8        | 55.3        | 48.9        | 54.9        | 76.7        | 83.0        | 89.0        | 59.2        | 79.7        | 85.7        | 90.6        |
| R-SMILES                   | 81.6                     | 74.5        | 69.9        | 59.1        | 55.7        | 77.1        | 84.1        | 90.4        | 60.1        | 80.2        | 86.7        | 92.2        |
| EditRetro                  | 81.5                     | 67.7        | 58.6        | 44.4        | <b>60.8</b> | <b>80.6</b> | <b>86.0</b> | 90.3        | <b>65.8</b> | <b>84.7</b> | <b>89.0</b> | 92.3        |
| <b>Ours</b>                | <b>90.1</b>              | <b>78.3</b> | <b>72.0</b> | <b>62.7</b> | 57.9        | 74.5        | 82.8        | 86.4        | 61.2        | 78.2        | 84.7        | 88.3        |

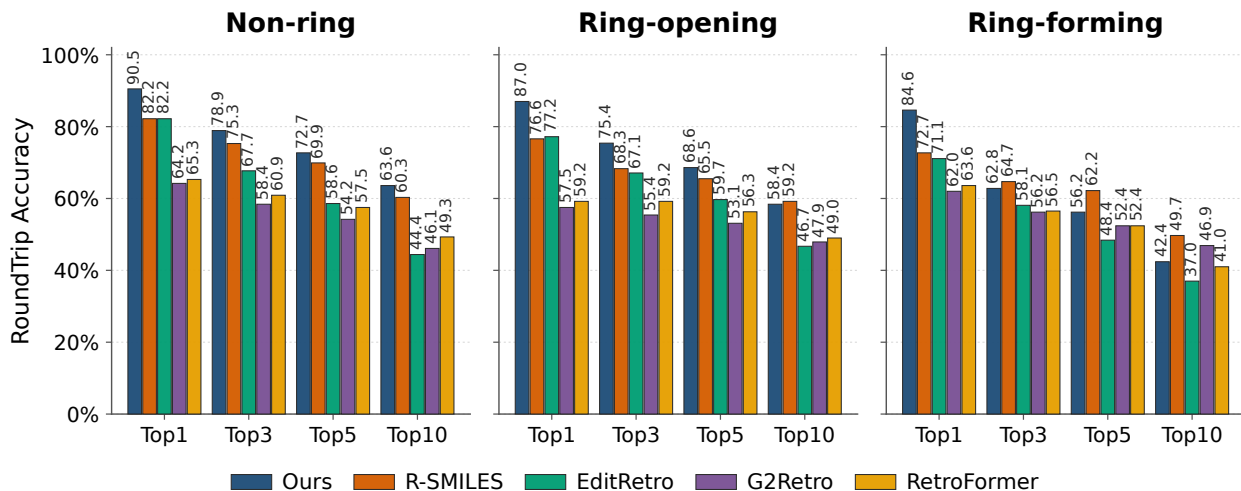


Fig. 3. Candidate-level round-trip accuracy (RT@ $k$ , %) on USPTO-50K for non-ring, ring-opening, and ring-forming reactions at  $k \in \{1, 3, 5, 10\}$ .

TABLE II

TOP-1 EXACT-MATCH AND ROUND-TRIP ACCURACY ON USPTO-50K FOR THE SUPERVISED POLICY AND THREE RLFF REWARD CONFIGURATIONS. ALL ST+RL CONFIGURATIONS USE  $\gamma = \delta = r_{\text{inv}} = -1$ . COEFFICIENT ENTRIES ARE NOT APPLICABLE TO ST.

| Configuration | $\alpha$ | $\beta$ | ExactMatch | RoundTrip |
|---------------|----------|---------|------------|-----------|
| ST (w/o RL)   | —        | —       | 57.5%      | 83.1%     |
| ST+RL         | 1.8      | 1.2     | 57.9%      | 90.1%     |
| ST+RL         | 2        | 1       | 48.7%      | 93.8%     |
| ST+RL         | 2        | 2       | 52.4%      | 88.3%     |

third rewards either condition equally and increases the reward for joint agreement.

The  $(\alpha, \beta) = (2, 1)$  setting achieves the highest round-trip accuracy, 93.8%, but reduces exact match to 48.7%.

Increasing  $\beta$  to 2 while keeping  $\alpha = 2$  raises exact match to 52.4% while lowering round-trip accuracy to 88.3%. Thus, a larger feasibility coefficient does not necessarily improve the resulting round-trip performance. These results indicate sensitivity to the full reward configuration, without establishing the underlying optimization mechanism. Among the tested settings,  $(\alpha, \beta) = (1.8, 1.2)$  achieves the highest exact-match accuracy and outperforms  $(\alpha, \beta) = (2, 2)$  on both metrics, supporting its use when preserving reference recovery is also required.

3) *Performance across Reaction Types*: Fig. 3 compares RetroRLFF with R-SMILES, EditRetro, G2Retro, and RetroFormer across three reaction groups. We use the candidate-level RT@ $k$  defined in Section IV-C, evaluated within each group.

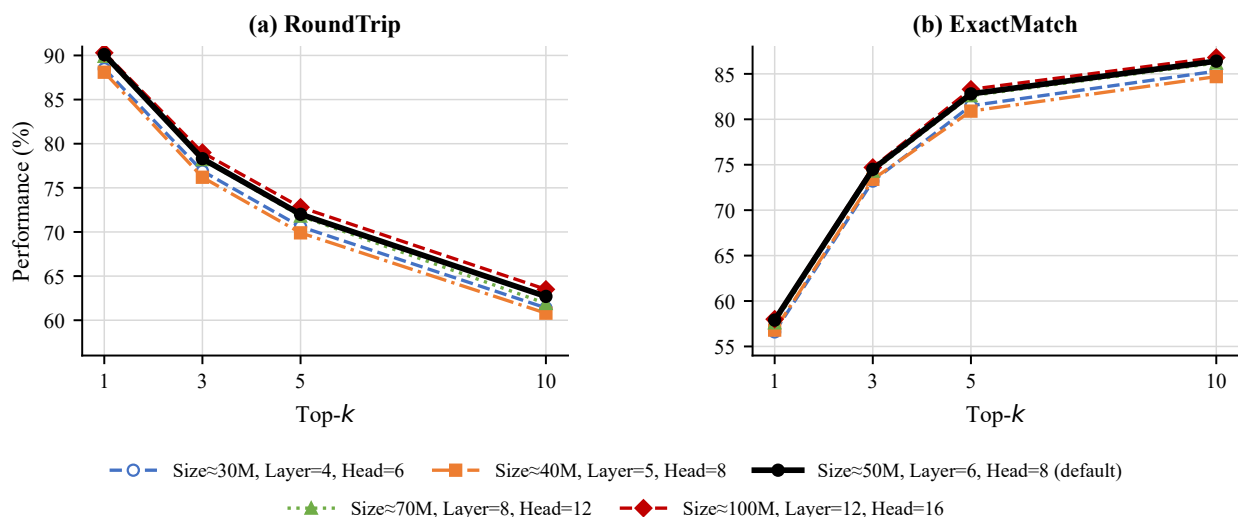


Fig. 4. Sensitivity of RetroRLFF to model size, layer count, and attention head count on USPTO-50K: (a) candidate-level round-trip accuracy (RT@ $k$ ) and (b) product-level exact matching accuracy (EM@ $k$ ), for  $k \in \{1, 3, 5, 10\}$ . The legend specifies each architecture; the default configuration has approximately 50M parameters, six layers, and eight attention heads.

RetroRLFF achieves the highest RT@1 for non-ring, ring-opening, and ring-forming reactions, reaching 90.5%, 87.0%, and 84.6%, respectively. These results exceed the strongest baseline in each group by 8.3, 9.8, and 11.9 percentage points. It also leads at  $k = 3$ , with 78.9%, 75.4%, and 64.7%. The advantage for leading predictions therefore extends across all three groups, including both types of ring transformations.

At larger cutoffs, the advantage depends on the reaction group. RetroRLFF remains best for non-ring reactions at  $k = 5$  and  $k = 10$  (72.7% and 63.6%). For ring-opening reactions, it leads at  $k = 5$  but slightly trails R-SMILES at  $k = 10$  (58.4% versus 59.2%). For ring-forming reactions, its 56.2% and 42.4% at  $k = 5$  and  $k = 10$  are below R-SMILES’s 62.2% and 49.7%. This identifies forward consistency among lower-ranked ring-forming candidates as a remaining limitation. Since RT@ $k$  averages candidate outcomes, its decline reflects reduced average consistency as the pool expands.

4) *Hyperparameter Sensitivity and Robustness:* We examine architectural sensitivity by comparing five RetroRLFF configurations in Fig. 4. Their approximate parameter counts, layer counts, and attention head counts are (30M, 4, 6), (40M, 5, 8), (50M, 6, 8), (70M, 8, 12), and (100M, 12, 16), respectively. The 50M configuration is the default. We assess both forward consistency and reference recovery using the metrics defined in Section IV-C.

The curves remain close across all evaluated cutoffs. Approximate readings from the figure place RT@1 at 88–90% and RT@10 at 61–64%, while EM@1 is about 57–58% and EM@10 about 85–87%. The 100M model performs best on both metrics, but its advantage over the default configuration is less than approximately one percentage point at each cutoff. Increasing capacity therefore provides only modest additional accuracy within the tested range. The 70M configuration also remains close to the default, and the two smaller models exhibit limited degradation. These observations support the use of the 50M configuration as a compact model with

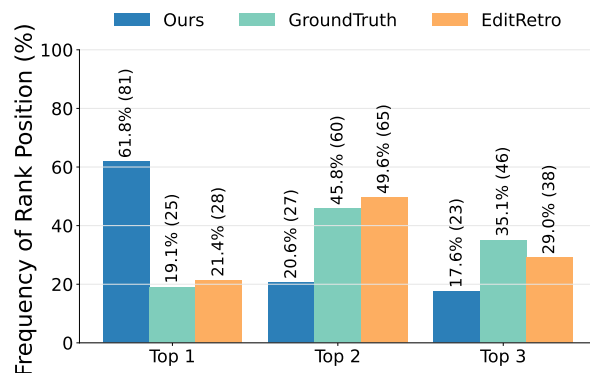


Fig. 5. Expert-assigned rank positions for RetroRLFF (Ours), recorded reference reactants (GroundTruth), and EditRetro on 131 selected test cases. Top 1, Top 2, and Top 3 denote individual rank positions, not cumulative top- $k$  accuracy. Labels report percentages and case counts; equal ranks are permitted.

performance close to that of the largest tested architecture.

All configurations exhibit decreasing RT@ $k$  and increasing EM@ $k$  as  $k$  grows, consistent with the candidate-level and product-level aggregation rules, respectively. Thus, the relationship between forward consistency and reference recovery persists across the tested architectures.

5) *Human Expert Evaluation:* To complement forward-model-based evaluation, we assess predictions on 131 instances randomly selected from the USPTO-50K test set. For each target product, organic synthesis experts compare three reactant sets: the recorded reference, the RetroRLFF prediction, and the EditRetro prediction [32]. The sets are anonymized and randomly ordered. Experts rank them by synthetic feasibility, with a better rank indicating a reactant set judged more feasible for synthesizing the target product. Equal ranks are permitted and retained as their assigned numerical positions: sets tied at rank 1 are each counted as Top 1, and sets tied at rank 2 are each counted as Top 2. These positions are reported as individual ranks rather than cumulative top-

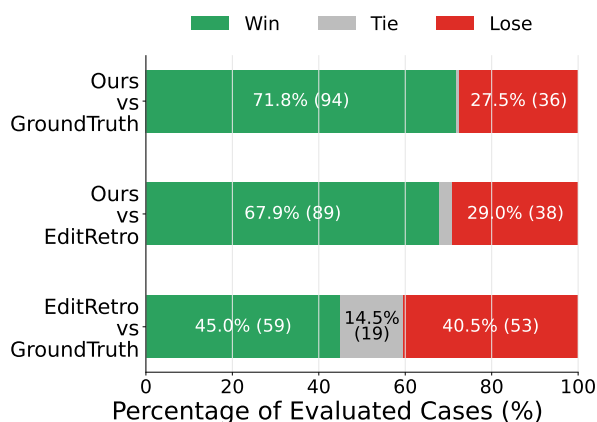


Fig. 6. Pairwise expert preferences derived from the rankings of 131 test cases. Outcomes are reported from the perspective of the first named reactant set. Green, gray, and red indicate wins, ties, and losses, respectively. Percentages use 131 as the denominator; counts appear in parentheses where labeled.

$k$  accuracy. We summarize the assigned rank positions and derive all three pairwise comparisons from these rankings. A win means that the first set is ranked ahead of the second, a loss means the reverse, and a tie means equal ranks.

Fig. 5 shows that RetroRLFF receives first place in 81 cases (61.8%), compared with 25 (19.1%) for the reference and 28 (21.4%) for EditRetro. It is assigned second and third place in 27 (20.6%) and 23 (17.6%) cases, respectively. By comparison, the reference and EditRetro are most frequently placed second, in 60 (45.8%) and 65 (49.6%) cases. Thus, expert preference on this subset favors RetroRLFF, complementing the forward-consistency results in Table I.

RetroRLFF versus the reference (Fig. 6) yields 94 wins (71.8%), one tie (0.8%), and 36 losses (27.5%). The preference for model predictions over recorded reactants supports the distinction between reference agreement and expert-assessed synthetic feasibility: reproducing the recorded solution is not sufficient to capture relative chemical preference.

RetroRLFF versus EditRetro yields 89 wins (67.9%), four ties (3.1%), and 38 losses (29.0%). Although EditRetro achieves higher exact-match accuracy in Table I, RetroRLFF is preferred more often on the reviewed cases, indicating that reference recovery and expert preference provide complementary assessments.

EditRetro versus the reference yields 59 wins (45.0%), 19 ties (14.5%), and 53 losses (40.5%). This comparatively balanced outcome contrasts with RetroRLFF’s larger preference advantage over both alternatives. These findings support the practical relevance of RetroRLFF’s predictions within the selected subset.

6) *Multistep Retrosynthesis Prediction by Our Method:* We recursively apply RetroRLFF to six target molecules, expanding intermediates through successive single-step predictions. Fig. 7 shows the proposed precursors and their candidate ranks. Transformations are discussed below in the forward synthetic direction.

Lenalidomide (Fig. 7a) involves three predictions ranked 1, 4, and 1. They suggest nitro-group reduction, lactam construction from an aminoglutarimide and a bromomethyl-substituted

nitrobenzoate, and benzylic bromination. The central disconnection requires exploring beyond the first-ranked candidate.

Nirmatrelvir (Fig. 7b) involves six predictions, all within the top two. The proposed transformations include amide dehydration to the nitrile, amide coupling, trifluoroacetylation, Boc deprotection, and ester hydrolysis. The intermediates retain stereochemical annotations on the bicyclic scaffold, illustrating recursive decomposition involving both coupling and protected precursors.

DDR1-032 (Fig. 7c) has eight predictions, all within the top five. Later expansions identify alkynyl coupling, halogenated benzimidazole intermediates, heterocycle construction, and nitro-group reduction. However, the acyl bromide depicted after the first expansion and the aldehyde-bearing precursor in the second require clarification before claiming a chemically continuous route.

Febuxostat (Fig. 7d) involves three predictions ranked 1, 2, and 1. They identify ester hydrolysis, Suzuki coupling between an arylboronic acid and a brominated thiazole ester, and borylation from an aryl bromide. The nitrile and ether substituents remain intact throughout the illustrated sequence.

DDR1-037 (Fig. 7e) illustrates convergent decomposition. The rank-2 amide disconnection produces a substituted benzoic acid and an azaindole-containing cyclic diamine. Subsequent expansions identify amide-forming precursors at rank 1 and aromatic-substitution precursors at rank 2, exposing both synthetic branches.

Cimetidine (Fig. 7f) involves five predictions ranked 8, 1, 2, 3, and 2. These suggest cyanoguanidine formation, thioether assembly, ester reduction, and imidazole construction. The rank-8 initial suggestion highlights the potential importance of retaining lower-ranked candidates during recursive exploration.

All 28 illustrated predictions occur within the top 10, including 22 within the top two. These selected cases illustrate diverse precursor relationships, rather than establish a multistep success rate or isolate the effect of RLFF.

7) *Case Study:* Fig. 8 compares five USPTO-50K test products, their recorded reactants, and RetroRLFF predictions. We discuss each column from left to right, focusing on chemically plausible alternatives and potential synthetic advantages.

For the enone-containing product (first column), the reference suggests acylative coupling of a vinylstanane with benzoyl chloride. RetroRLFF proposes Horner–Wadsworth–Emmons olefination of an aldehyde with a  $\beta$ -ketophosphonate. This alternative avoids an organotin precursor and tin-containing waste, potentially simplifying purification. Chemoselectivity toward the aldehyde over the cyclopentanone and alkene geometry require verification.

For the indazole-containing secondary amine (second column), the reference reduces an  $N$ -methyl amide, whereas the prediction suggests reductive amination of the corresponding aldehyde with methylamine. The latter offers a recognizable alternative that may permit milder reducing conditions, subject to compatibility with the heterocycle.

For the nitroaryl amino diol (third column), RetroRLFF replaces the reference’s  $N$ -acetyl precursor with a Boc-protected amine. Acid-mediated Boc removal could avoid the more

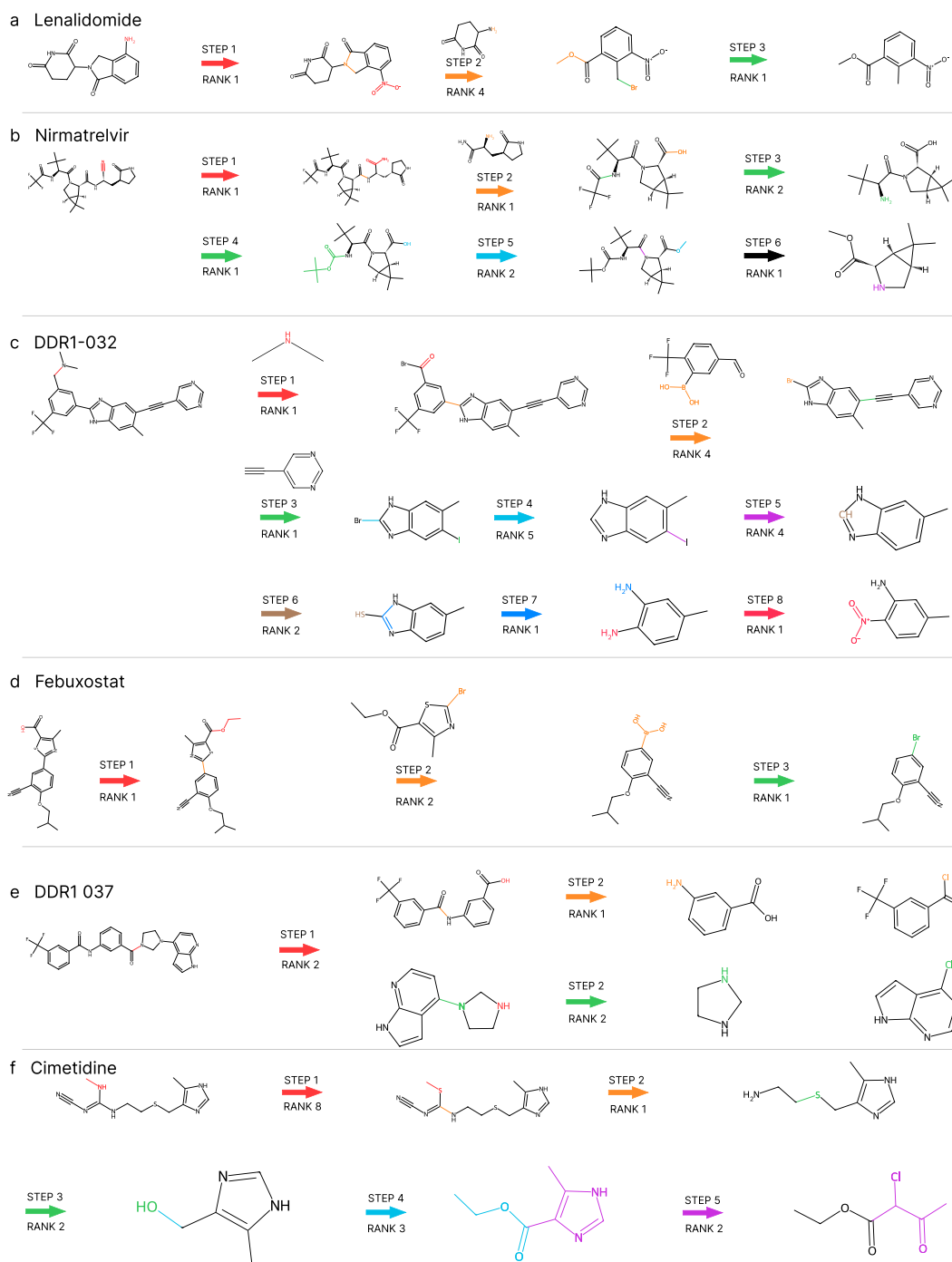


Fig. 7. Retrosynthetic suggestions for (a) lenalidomide, (b) nirmatrelvir, (c) DDR1-032, (d) febuxostat, (e) DDR1-037, and (f) cimetidine. Arrows point to proposed precursors; STEP and RANK indicate the expansion step and candidate position. Colors highlight structural changes. Panel (e) expands both precursor branches.

forcing hydrolysis potentially required for deacetylation while retaining the nitro group and both hydroxymethyl groups.

For the heterocyclic alcohol (fourth column), the reference proposes methyl-ether cleavage, whereas RetroRLFF suggests TBS-ether deprotection. Selective silicon-oxygen bond cleavage offers a plausible strategy for exposing the alcohol while

preserving the scaffold's other ether linkage, although substrate compatibility remains to be tested.

For the pyridyl-substituted lactam (fifth column), the reference uses Stille coupling with a pyridylstannane. RetroRLFF instead proposes lactam *N*-alkylation with a pyridyl-substituted allylic bromide. This avoids an organotin

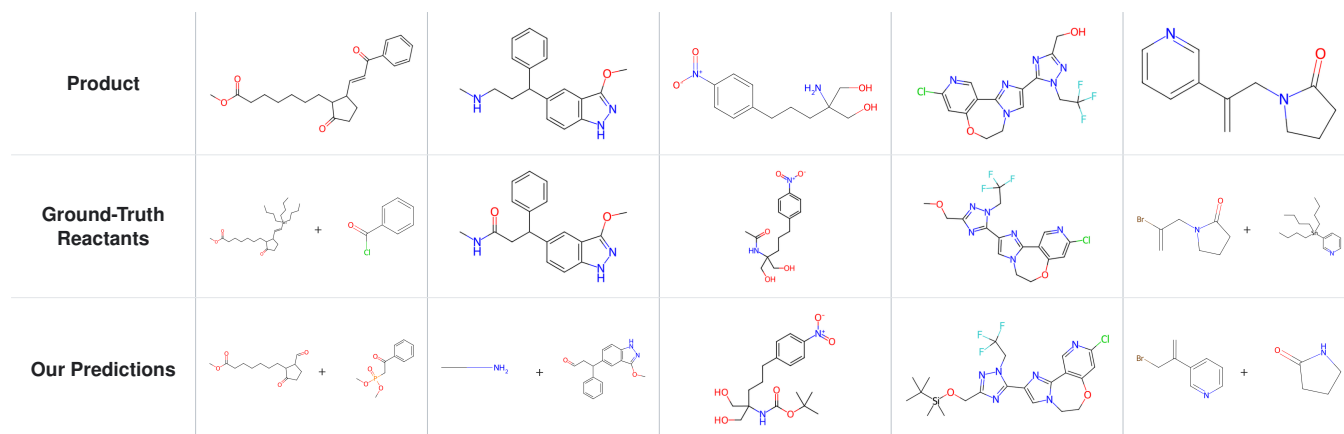


Fig. 8. Five USPTO-50K test cases. Rows show target products, recorded ground-truth reactants, and RetroRLFF predictions. Columns illustrate enone construction, secondary-amine formation, amino-diol deprotection, heterocyclic alcohol deprotection, and lactam alkylation.

coupling partner, but competing *O*-alkylation and reactions involving the pyridine nitrogen require assessment.

These cases support evaluating plausible non-reference alternatives beyond exact matching. Without reaction conditions, yields, or precursor-accessibility data, the proposed advantages remain structure-based hypotheses rather than evidence of experimental or overall route superiority.

## V. CONCLUSION

RetroRLFF incorporates forward reaction feedback into single-step retrosynthesis to address the mismatch between reference agreement and synthetic feasibility. On USPTO-50K, it improves top-1 round-trip accuracy from 83.1% to 90.1% while retaining reference recovery, with expert evaluation supporting the plausibility of its predictions on the reviewed subset.

### A. Limitations

Although the reward balances reference agreement and forward consistency, its feasibility signal remains a surrogate provided by a fixed forward model. Prediction errors and biases can separate round-trip success from experimental outcomes, and expert preference does not replace experimental validation. Moreover, the reward does not explicitly account for precursor cost and availability, catalyst requirements, reaction conditions, or yield. Consequently, forward-consistent predictions are not necessarily economical or practical to execute.

The current policy directly maps product SMILES to reactant SMILES without explicit intermediate chemical analysis or tool use. Its feedback evaluates the generated reactant set, rather than a sequence of chemical decisions. It therefore does not explicitly model how chemists examine bond disconnections, functional-group compatibility, and competing transformations, or consult literature and computational tools before selecting precursors.

### B. Future Work

We will extend the reward toward a weighted, multi-objective assessment combining feasibility with precursor accessibility, material and catalyst costs, reaction conditions, and expected yield. Independent predictors and experimental evidence will be explored to calibrate these signals and reduce dependence on a single forward evaluator. The resulting objective should reflect practical synthesis requirements while making tradeoffs between feasibility and resource use explicit.

A second direction is to extend RetroRLFF into an agent harness that supports iterative analysis and tool use within each single-step prediction. The policy could inspect alternative disconnections, retrieve reaction precedents, query precursor availability, and evaluate candidates through multiple tools, revising its decisions from the returned evidence. Such predictions would form longer decision trajectories, which reinforcement learning could optimize using the comprehensive weighted reward and credit assignment across analysis, tool selection, and final precursor generation.

## REFERENCES

- [1] P. Schwaller, R. Petraglia, V. Zullo, V. H. Nair, R. A. 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*, vol. 11, no. 12, pp. 3316–3325, 2020. [Online]. Available: <https://doi.org/10.1039/c9sc05704h>
- [2] S. Chen and Y. Jung, "Deep Retrosynthetic Reaction Prediction using Local Reactivity and Global Attention," *JACS Au*, vol. 1, no. 10, pp. 1612–1620, 2021. [Online]. Available: <https://doi.org/10.1021/jacsau.1c00246>
- [3] C. W. Coley, L. Rogers, W. H. Green, and K. F. Jensen, "Computer-Assisted Retrosynthesis Based on Molecular Similarity," *ACS Central Science*, vol. 3, no. 12, pp. 1237–1245, 2017. [Online]. Available: <https://doi.org/10.1021/acscentsci.7b00355>
- [4] M. H. S. Segler and M. P. Waller, "Neural-Symbolic Machine Learning for Retrosynthesis and Reaction Prediction," *Chemistry – A European Journal*, vol. 23, no. 25, pp. 5966–5971, 2017. [Online]. Available: <https://doi.org/10.1002/chem.201605499>
- [5] 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*, vol. 32, 2019. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2019/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2019/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf)

- [6] P. Seidl, P. Renz, N. Dyubankova, P. Neves, J. Verhoeven, J. K. Wegner, M. Segler, S. Hochreiter, and G. Klambauer, "Improving Few- and Zero-Shot Reaction Template Prediction Using Modern Hopfield Networks," *Journal of Chemical Information and Modeling*, vol. 62, no. 9, pp. 2111–2120, 2022. [Online]. Available: <https://doi.org/10.1021/acs.jcim.1c01065>
- [7] S. Xie, R. Yan, J. Guo, Y. Xia, L. Wu, and T. Qin, "Retrosynthesis Prediction with Local Template Retrieval," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, pp. 5330–5338, 2023. [Online]. Available: <https://doi.org/10.1609/aaai.v37i4.25664>
- [8] 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*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 8818–8827. [Online]. Available: <https://proceedings.mlr.press/v119/shi20d.html>
- [9] 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*, vol. 33, 2020, pp. 11248–11258. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/819f46e52c25763a55cc642422644317-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/819f46e52c25763a55cc642422644317-Paper.pdf)
- [10] 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*, vol. 34, 2021, pp. 9405–9415. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2021/file/4e2a6330465c8ffcaa696a5a16639176-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2021/file/4e2a6330465c8ffcaa696a5a16639176-Paper.pdf)
- [11] Z. Lan, B. Hong, M. Xu, Z. Zeng, Z. Liu, L. Yu, and F. Ma, "Retrosynthesis prediction via search in (hyper) graph," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 17749–17760, 2025.
- [12] Z. Lan, Z. Zeng, B. Hong, Z. Liu, and F. Ma, "Rcsearcher: Reaction center identification in retrosynthesis via deep q-learning," *Pattern Recognition*, vol. 150, p. 110318, 2024.
- [13] J. Zhu, B. Hong, Z. Lan, and F. Ma, "Single-step retrosynthesis via reaction center and leaving groups prediction," in *2023 16th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. IEEE, 2023, pp. 1–5.
- [14] 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*, vol. 3, no. 10, pp. 1103–1113, 2017. [Online]. Available: <https://doi.org/10.1021/acscentsci.7b00303>
- [15] 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*, vol. 60, no. 1, pp. 47–55, 2020. [Online]. Available: <https://doi.org/10.1021/acs.jcim.9b00949>
- [16] 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*, vol. 61, no. 7, pp. 3273–3284, 2021. [Online]. Available: <https://doi.org/10.1021/acs.jcim.1c00537>
- [17] 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*, vol. 62, no. 15, pp. 3503–3513, 2022. [Online]. Available: <https://doi.org/10.1021/acs.jcim.2c00321>
- [18] Z. Zhong, J. Song, Z. Feng, T. Liu, L. Jia, S. Yao, M. Wu, T. Hou, and M. Song, "Root-aligned SMILES: a tight representation for chemical reaction prediction," *Chemical Science*, vol. 13, no. 31, pp. 9023–9034, 2022. [Online]. Available: <https://doi.org/10.1039/d2sc02763a>
- [19] 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*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 2022, pp. 22475–22490. [Online]. Available: <https://proceedings.mlr.press/v162/wan22a.html>
- [20] R. Irwin, S. Dimitriadis, J. He, and E. J. Bjerrum, "Chemformer: a pre-trained transformer for computational chemistry," *Machine Learning: Science and Technology*, vol. 3, no. 1, 2022, article 015022. [Online]. Available: <https://doi.org/10.1088/2632-2153/ac3ffb>
- [21] I. Igashov, A. Schneuing, M. Segler, M. Bronstein, and B. Correia, "RetroBridge: Modeling Retrosynthesis with Markov Bridges," in *Proceedings of the Twelfth International Conference on Learning Representations*, 2024, pp. 39622–39640. [Online]. Available: [https://proceedings.iclr.cc/paper\\_files/paper/2024/file/acb94e709f02895fd98b5867f0b184f3-Paper-Conference.pdf](https://proceedings.iclr.cc/paper_files/paper/2024/file/acb94e709f02895fd98b5867f0b184f3-Paper-Conference.pdf)
- [22] S. Current, Z. Chen, D. Adu-Ampratwum, X. Ning, and S. Parthasarathy, "Differ: categorical diffusion ensembles for single-step chemical retrosynthesis," *Journal of Cheminformatics*, vol. 17, no. 1, 2025, article 112. [Online]. Available: <https://doi.org/10.1186/s13321-025-01056-7>
- [23] R. Yadav, Q. Yan, G. Wolf, J. Bose, and R. Liao, "RETRO SYNFLOW: Discrete Flow-Matching for Accurate and Diverse Single-Step Retrosynthesis," in *Advances in Neural Information Processing Systems*, vol. 38, 2025, pp. 55229–55258. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2025/file/4f86833d5cc98ec32e470ef1c8cb82e3-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2025/file/4f86833d5cc98ec32e470ef1c8cb82e3-Paper-Conference.pdf)
- [24] X. Wang, Y. Li, J. Qiu, G. Chen, H. Liu, B. Liao, C.-Y. Hsieh, and X. Yao, "RetroPrime: A Diverse, plausible and Transformer-based method for Single-Step retrosynthesis predictions," *Chemical Engineering Journal*, vol. 420, 2021, article 129845. [Online]. Available: <https://doi.org/10.1016/j.cej.2021.129845>
- [25] Z. Chen, O. R. Ayinde, J. R. Fuchs, H. Sun, and X. Ning, "G2Retro as a two-step graph generative models for retrosynthesis prediction," *Communications Chemistry*, vol. 6, no. 1, 2023, article 102. [Online]. Available: <https://doi.org/10.1038/s42004-023-00897-3>
- [26] P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. A. Hunter, C. Bekas, and A. A. Lee, "Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction," *ACS Central Science*, vol. 5, no. 9, pp. 1572–1583, 2019. [Online]. Available: <https://doi.org/10.1021/acscentsci.9b00576>
- [27] W. Zhong, Z. Yang, and C. Y.-C. Chen, "Retrosynthesis prediction using an end-to-end graph generative architecture for molecular graph editing," *Nature Communications*, vol. 14, no. 1, 2023, article 3009. [Online]. Available: <https://doi.org/10.1038/s41467-023-38851-5>
- [28] J. Liu, C. Wang, C. Liu, L. Zeng, R. Yan, Y. Sun, and Y. Liu, "DAPO: Improving Multi-Step Reasoning Abilities of Large Language Models with Direct Advantage-Based Policy Optimization," in *Advances in Neural Information Processing Systems*, vol. 38, 2025, pp. 71445–71477. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2025/file/6789f033ebde742552e5db84fb5d414a-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2025/file/6789f033ebde742552e5db84fb5d414a-Paper-Conference.pdf)
- [29] P.-C. Zhao, X.-X. Wei, Q. Wang, Q.-H. Wang, J.-N. Li, J. Shang, C. Lu, and J.-Y. Shi, "Single-step retrosynthesis prediction via multitask graph representation learning," *Nature Communications*, vol. 16, no. 1, 2025, article 814. [Online]. Available: <https://doi.org/10.1038/s41467-025-56062-y>
- [30] 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*, vol. 11, no. 1, 2020, article 5575. [Online]. Available: <https://doi.org/10.1038/s41467-020-19266-y>
- [31] Z. Lin, S. Yin, L. Shi, W. Zhou, and Y. J. Zhang, "G2GT: Retrosynthesis Prediction with Graph-to-Graph Attention Neural Network and Self-Training," *Journal of Chemical Information and Modeling*, vol. 63, no. 7, pp. 1894–1905, 2023. [Online]. Available: <https://doi.org/10.1021/acs.jcim.2c01302>
- [32] Y. Han, X. Xu, C.-Y. Hsieh, K. Ding, H. Xu, R. Xu, T. Hou, Q. Zhang, and H. Chen, "Retrosynthesis prediction with an iterative string editing model," *Nature Communications*, vol. 15, no. 1, 2024, article 6404. [Online]. Available: <https://doi.org/10.1038/s41467-024-50617-1>
- [33] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep Reinforcement Learning from Human Preferences," in *Advances in Neural Information Processing Systems*, vol. 30, 2017. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf)
- [34] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano, "Learning to summarize with human feedback," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 3008–3021. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf)
- [35] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 27730–27744. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf)
- [36] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017, arXiv preprint. [Online]. Available: <https://arxiv.org/abs/1707.06347>
- [37] H. Lee, S. Phatale, H. Mansoor, T. Mesnard, J. Ferret, K. R. Lu, C. Bishop, E. Hall, V. Carbune, A. Rastogi, and S. Prakash, "RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback

- with AI Feedback,” in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 2024, pp. 26 874–26 901. [Online]. Available: <https://proceedings.mlr.press/v235/lee24t.html>
- [38] L. Gao, J. Schulman, and J. Hilton, “Scaling Laws for Reward Model Overoptimization,” in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 2023, pp. 10 835–10 866. [Online]. Available: <https://proceedings.mlr.press/v202/gao23h.html>
- [39] F. N. Baker, Z. Chen, D. Adu-Ampratwum, and X. Ning, “RLSynC: Offline–Online Reinforcement Learning for Synthon Completion,” *Journal of Chemical Information and Modeling*, vol. 64, no. 17, pp. 6723–6735, 2024. [Online]. Available: <https://doi.org/10.1021/acs.jcim.4c00554>
- [40] P. Gaiński, M. Koziarski, K. Maziarz, M. Segler, J. Tabor, and M. Śmieja, “Diverse and feasible retrosynthesis using GFlowNets,” *Information Sciences*, vol. 714, 2025, article 122194. [Online]. Available: <https://doi.org/10.1016/j.ins.2025.122194>
- [41] S. Narayanan, J. Braza, R.-R. Griffiths, A. Bou, G. Wellawatte, M. Caldas Ramos, L. Mitchener, M. Pieler, S. Rodriques, and A. White, “Training a Scientific Reasoning Model for Chemistry,” in *Advances in Neural Information Processing Systems*, vol. 38, 2025, pp. 157 671–157 710. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2025/file/e754c82afb9bb0150bd6b71f5f49a9e6-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2025/file/e754c82afb9bb0150bd6b71f5f49a9e6-Paper-Conference.pdf)
- [42] L. Kong, X. Wang, Y. Chen, and M. Zhang, “Round-trip Reinforcement Learning: Self-Consistent Training for Better Chemical LLMs,” 2025, arXiv preprint. [Online]. Available: <https://arxiv.org/abs/2510.01527>
- [43] 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*, vol. 35, no. 1, pp. 531–539, 2021. [Online]. Available: <https://doi.org/10.1609/aaai.v35i1.16131>