

---

# ReactionReasoner: Towards Reasoning LLM for Chemical Reaction Prediction

---

**Hanbum Ko**  
Korea University  
hanbumko95@korea.ac.kr

**Chanhui Lee**  
Korea University  
chanhui-lee@korea.ac.kr

**Ye Rin Kim**  
Korea University  
pimang62@korea.ac.kr

**Rodrigo Hormazabal**  
LG AI Research  
rodrigo@lgresearch.ai

**Sehui Han**  
LG AI Research  
hansse.han@lgresearch.ai

**Sungbin Lim**  
Korea University  
sungbin@korea.ac.kr

**Sungwoong Kim**  
Korea University  
swkim01@korea.ac.kr

## Abstract

Chemical reaction prediction involves predicting reaction outcomes from given molecules, inferring the required starting materials from a given product, and identifying reagents that bring about the transformation. It is a complex problem that demands explicit reasoning such as functional group recognition, reaction mechanism analysis. Recent advances in general-purpose large language models (LLMs) have led to improved performance on reaction prediction tasks and can generate some reasoning traces. However, due to a lack of domain-specialized training, they consistently struggle with reaction-specific reasoning and consequently exhibit poor accuracy. In parallel, molecular LLMs have also been recently developed, yet such models typically predict only the molecules without providing reasoning and thus their performance on complex reaction prediction tasks remains limited. To address this limitation and move toward reasoning-capable LLMs for chemical reaction prediction, we present SyntheticReact, a synthetic reasoning data generation method for chemical reactions, mirroring practicing chemists’ strategies, and ReactionReasoner, an LLM-based reaction reasoning model trained on data produced by SyntheticReact. In particular, given reaction SMILES (RXN SMILES), SyntheticReact collects reaction documents via web scraping, extracts information about human chemists’ strategies, and uses an LLM to structure it into reasoning data. Using reasoning data generated by SyntheticReact, we train ReactionReasoner through supervised fine-tuning. In addition, we apply a self-bootstrapping approach: reasoning data that lead to correct answers are used for an additional supervised training, while those that fail are used to generate reflection data, capturing why the reasoning is unsuccessful. Through our experiments, we show that detailed, step-by-step reasoning, similar to how human chemists approach problems, is more valuable than reasoning that only provides general explanations. Moreover, when proper reasoning is given, ReactionReasoner significantly outperforms models that attempt predictions without reasoning.

## 1 Introduction

Recent advances in large language models (LLMs) have broadened their applications to molecular tasks, including chemical reaction prediction, establishing what are now referred to as molecular

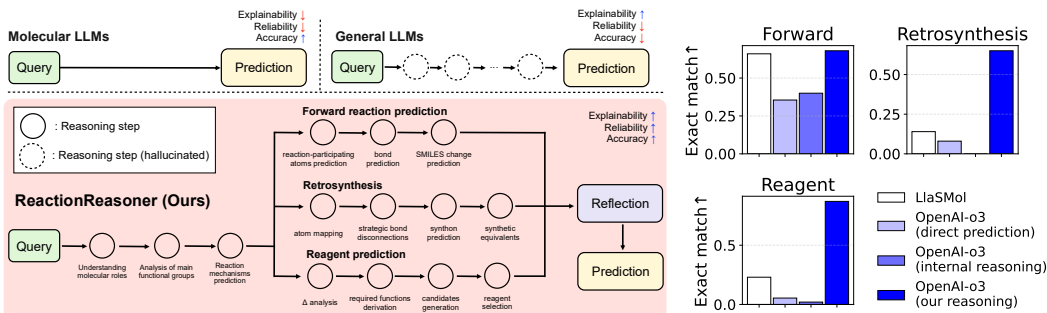


Figure 1: (Left) Comparison of reasoning processes among three models: a molecular LLM that predicts molecules without any reasoning, a general LLM that generates reasoning with a high likelihood of hallucination and does not follow the actual reasoning process of chemists, and ReactionReasoner which performs more faithful reasoning. (Right) Comparison of prediction performance when using reasoning text generated by SyntheticReact, a reasoning text generation method for training ReactionReasoner, as the reasoning to OpenAI-o3.

LLMs [1–4]. Predicting chemical reactions is generally known to involve determining the products of given reactants, tracing back the starting materials from a product, and identifying the reagents that enable the transformation. This task is inherently complex due to its reaction mechanisms and requires multi-step reasoning, whereas existing molecular LLMs provide only predictions without explicit reasoning [2, 3, 5], leading to performance limitations. Meanwhile, recent reasoning-oriented general-purpose LLMs such as OpenAI-o3 [6], Qwen3 [7], and Gemini 2.5 [8] can perform reasoning [9] before answering in many domains including chemistry. However, their reasoning paths for chemical reaction prediction diverge significantly from those of practicing human chemists and commonly suffer from hallucinations.

To address these limitations, we develop a reasoning LLM that follows an expert chemist’s multi-step reasoning process. This is achieved through a precisely defined data-generation method that ensures chemically valid reasoning without hallucinations, together with an efficient training strategy for challenging chemical reaction reasoning. Therefore, we first introduce a method for reasoning data generation, called SyntheticReact, that reduces hallucinations and mimics general reasoning steps of a human chemist for reaction prediction tasks based on molecule-specific supporting information. Specifically, given a reaction SMILES (RXN SMILES) [10, 11], SyntheticReact extracts the following supporting information: directly usable information, model-predicted information, externally collected information, and rule-derived information. By utilizing this supporting information, SyntheticReact generates large-scale reasoning data using an LLM, and to ensure the data quality, SyntheticReact verifies that the generated reasoning faithfully reflects the supporting information by employing rule-based filtering logic. Using a dataset generated by SyntheticReact, we train ReactionReasoner via supervised fine-tuning (SFT) to follow the reasoning processes of human chemists. Then, we collect self-bootstrapped reasoning data using the trained ReactionReasoner that generate the correct output SMILES as well as reasoning data that lead to incorrect outputs. Especially, using the failed reasoning data, we employ another LLM to compare the failed reasoning data with the successful reasoning data and identify why the failure occurs and how to fix it, producing reasoning data with error reflections. Finally, we use this self-bootstrapped reasoning data for another stage of SFT for ReactionReasoner.

The left part of Figure 1 summarizes the reasoning steps of ReactionReasoner compared to molecular LLMs and general-purpose LLMs. Molecular LLMs are fine-tuned to only output SMILES without reasoning steps on the complex reaction prediction tasks. While general-purpose LLMs generate reasoning steps, their reasoning often deviates from that of human chemists and is highly prone to hallucinations, resulting in limited performance on reaction prediction tasks. In the right part of Figure 1, an ablation study is shown to investigate the impact of reasoning data generated by SyntheticReact on the chemical reaction prediction tasks. The results show that, when reasoning texts (rationales) generated by SyntheticReact are provided to a general-purpose LLM (OpenAI-o3 [6]), it outperforms the prediction performance using their internal knowledge. It also shows higher performance than LlaSMol [2], a molecular LLM. This demonstrates that our proposed reasoning steps can substantially improve chemical reaction prediction performance. By training ReactionReasoner with the three-stage SFT, we show that the three-stage training gradually improves its predictive performance. However, we observe that training LLMs to learn these intermediate reasoning steps is

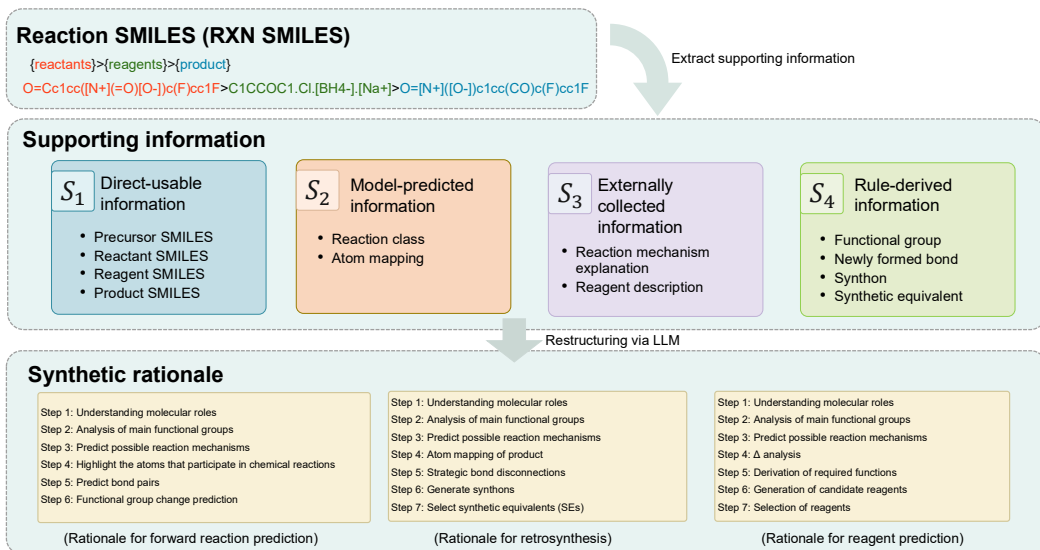


Figure 2: Overview of SyntheticReact. Supporting information is extracted from a reaction SMILES in four different categories, and then restructured using an LLM to generate task-specific rationales. Example rationale can be found in Appendix H.

more challenging than directly training them for final prediction, and their performance has therefore not yet reached a satisfactory level.

In summary, our contributions are:

- Development of SyntheticReact, a method for automatically generating large-scale and high-quality chemical reaction reasoning (rationale) data based on reliable supporting information.
- Development of ReactionReasoner, a reasoning LLM for chemical reaction tasks that produces intermediate reasoning steps reflecting human chemists.
- Empirical validation of the benefits in using generated multi-step reasoning by SyntheticReact for accurate reaction prediction as well as the improved performance obtained by ReactionReasoner, with rigorous limitation analysis.

## 2 SyntheticReact: Synthetic Chemical Reaction Rationale Generator

### 2.1 Generation Overview

SyntheticReact is an LLM-based reasoning text (rationale) generator for three chemical reaction prediction tasks—forward reaction prediction (**f**orward), retrosynthesis (**r**etro), and reagent prediction (**r**eagent)—from a given RXN SMILES. To mitigate hallucinations that arise when an LLM naively generates rationales using only SMILES of molecules, SyntheticReact utilizes four categories of supporting information, indicated  $\{S_1, S_2, S_3, S_4\}$ , into the rationale generation prompt.

### 2.2 Reasoning Steps

In Figure 1, all tasks share three initial reasoning steps ( $R_1$ – $R_3$ ). In  $R_1$ , the input molecules are identified. In  $R_2$ , functional groups and substructures of the input molecules are identified, and their chemical properties are analyzed. In  $R_3$ , a plausible reaction mechanism is predicted on the basis of the information from  $R_1$  and  $R_2$ . After the common reasoning steps, each task proceeds through its own detailed reasoning procedure. For forward, reasoning steps ( $R_4^F, R_5^F, R_6^F$ ) successively predict the set of atoms that participate in the reaction ( $R_4^F$ ) and the atom pairs themselves and bond type ( $R_5^F$ ), after which the SMILES of the molecule—expressed by simply connecting those atom pairs—is converted into the final valid product SMILES ( $R_6^F$ ). In **retro**, reasoning steps ( $R_4^R, R_5^R, R_6^R, R_7^R$ ) perform atom mapping of the product ( $R_4^R$ ), strategic bond disconnection ( $R_5^R$ ),

synthon generation ( $R_6^R$ ), and selection of synthetic equivalents ( $R_7^R$ ) for each synthon. Finally, reagent is carried out in steps ( $R_4^G, R_5^G, R_6^G, R_7^G$ ),  $\Delta$  analysis that examines structural changes before and after the reaction ( $R_4^G$ ), derives the functions required of the reagent ( $R_5^G$ ), explores the candidate reagents ( $R_6^G$ ), and determines the final reagents ( $R_7^G$ ). Details of the reasoning steps design are described in Appendix C.

## 2.3 Supporting Information

$S_1$  consists of directly usable information, namely the original SMILES of molecules (reactant, product, reagent) contained in the input RXN SMILES, and  $S_2$  is model-predicted information (reaction class, atom mapping between reactant and product) obtained from deep learning-based models [12, 13].  $S_3$  comprises externally collected information, such as chemical reaction mechanism explanations and descriptions of reagent roles gathered via web scraping, while  $S_4$  consists of rule-derived information obtained by applying SMILES arbitrary target specification (SMARTS) patterns and chemical rules, including the presence and positions of functional groups in the SMILES, formed or cleaved bond pairs, synthons, and synthetic equivalents. Each subset is ultimately inserted into the prompt so that the LLM can generate credible reasoning contents.

## 2.4 Synthetic Generation

We utilize the ORDERly [14] dataset, which is a high-quality chemical reaction prediction dataset containing RXN SMILES for each reaction instance, and the general LLM used to generate the rationale is Qwen3-8B [7]. The subsets  $S_1$ – $S_4$  appropriate to each task are selected and used in the rationale generation prompt. To maintain prompt length and formatting consistency, the model is instructed to output placeholders such as `<SYNTHONS>` and `<DISCONNECTED_BONDS>` first; these placeholders are then replaced with the actual information during post-processing. After generation, rule-based filtering is applied to check whether the supporting information is well contained in the rationale, and the passed rationales are finally used as training data. All details of the rule-based filtering logic, generation prompts, and the generated rationale examples for each task are provided in Table 3, Appendix G, and Appendix H.

# 3 ReactionReasoner: Reasoning LLM for Chemical Reaction Prediction

## 3.1 Overview

ReactionReasoner is trained using synthetic data generated by SyntheticReact. The base model is also Qwen3-8B, which can produce both reasoning and non-reasoning outputs. In Qwen3-8B, reasoning contents are enclosed by the special tokens `<think>` and `</think>`. The SyntheticReact-generated rationales are placed between these special token pairs. The loss function is cross-entropy loss over tokens, and LoRA [15] is used for training. The detailed hyperparameters for each stage are presented in Appendix E.3.

## 3.2 Stage 1: Supervised Fine-tuning

Stage 1 training consists of supervised fine-tuning (SFT) using reasoning texts generated by SyntheticReact. To enable a systematic acquisition of reasoning skills, stage 1 training is structured into multiple phases. Initially, the model is trained on the three reasoning steps that are common across all tasks in a joint manner. Subsequently, task-specific reasoning texts are trained incrementally in a stepwise manner. In the final phase, the training objective focuses on generating the correct answer SMILES. A detailed description of the training algorithm is provided in Appendix E.1.

## 3.3 Stage 2: Direct Bootstrapping

Stage 2 uses the ReactionReasoner trained in stage 1 to generate multiple reasoning paths for each training instance, where the ReactionReasoner leads to correct answer. These reasoning paths are then directly used as additional training data. To first verify whether such a self-bootstrapping strategy is feasible, we perform sampling with  $k = 30$  at a temperature of 0.8 and measure EXACT@k, which measures whether at least one correct SMILES is produced. The results are shown in Figure 3. Across all tasks, the EXACT@k values are mostly in the range of 0.8–0.9, demonstrating that, for the

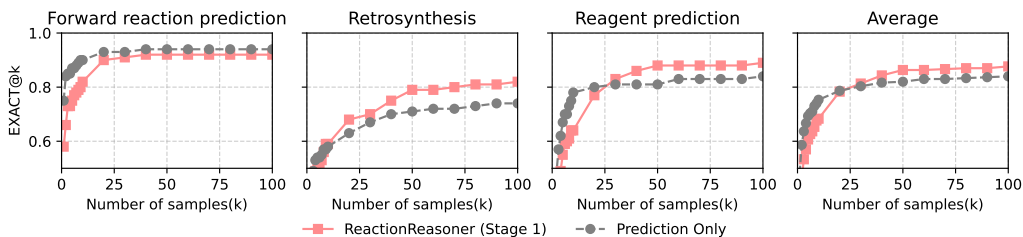


Figure 3: EXACT@k performance according to the number of samples.

majority of instances, the bootstrapping method can indeed generate more diverse reasoning paths. These are then further added to the stage 1 dataset for continued training.

### 3.4 Stage 3: Reflective Bootstrapping

Stage 3 is the stage where the model learns to further reflect on the generated reasoning. In each task-specific reasoning step, there are contents that must be answered precisely for the given task (e.g., synthon in retro), which are cases where the correct answer is uniquely defined. Therefore, unlike simple bootstrapping methods, this does not produce diverse paths. To encourage multiple reasoning paths and to make the model reach the correct answer not through a single prediction but through reflection, reflection is appended to the existing rationale. This process leverages the rationales sampled in stage 2: the LLM is given both the task-specific steps of rationales that lead to the correct answer and those that lead to an incorrect one, and it generates content that reviews and corrects the mistakes. Even when only the correct rationale is used, the model learns the ability to verify through reflection that the content is indeed correct. For reflection confirming the correct answer, the loss is computed on the rationale, reflection, and answer tokens. In the case of reflecting on an incorrect rationale, the rationale itself is excluded from the loss computation, and only the reflection and answer tokens are used for loss computation.

## 4 Experiments

### 4.1 Evaluation Dataset

One hundred instances are randomly selected from the ORDERly test dataset to serve as the validation dataset, and 1,000 instances that do not overlap with the molecular LLMs’ training data or with our validation dataset are selected as the test dataset. For prediction performance evaluation, all 1,000 test instances are used. However, due to the high evaluation cost, only the first 100 test instances are selected for rationale quality comparison, analysis, and ablation studies.

### 4.2 Baseline Models and Evaluation Metrics

The comparison models are broadly categorized into molecular LLMs, which are trained to handle molecules, and general LLMs, which are used for general purposes. The molecular LLMs include LlaSMol [2], Mol-Instructions [5], BioT5+ [3], and Mol-LLM [4], while the general LLMs encompass both reasoning and non-reasoning models, including GPT-4o, OpenAI-o3 [6], and Qwen3-8B/32B [7] models. The Qwen3 can control whether or not to use reasoning, and if it does, it is differentiated by adding ‘-think’ to the end of the name. For the evaluation of rationale quality, GPT-4o is used as the evaluator, inspired by Zheng et al. [16], and the metrics include FACT (factfulness, accuracy of information), COHR (coherence, consistency of content), and DEDU (deduction, logical inference of the answer). To measure factfulness, GPT-4o utilizes tools constructed with RDKit [17] functions and ChemToolAgent’s [18] tools, allowing the LLM evaluation to assess accuracy. GPT-4o scores range from 1 to 5, indicating a spectrum from poor to excellent. The evaluation of molecule prediction includes checking whether the prediction is exactly correct (EXACT), the text similarity between the ground truth and the predicted SMILES (BLEU, Levenshtein), the fingerprint similarity between the ground truth and predicted molecules (RDK, MACCS, MORGAN), and whether a chemically valid molecule is generated (VALIDITY).

	EXACT $\uparrow$	BLEU $\uparrow$	LEVENSHTEIN $\downarrow$	RD $\uparrow$	MACCS $\uparrow$	MORGAN $\uparrow$	VALIDITY $\uparrow$
<b>Forward reaction prediction</b>							
<i>Molecular LLMs</i>							
LlaSMol	0.605	0.877	7.455	0.843	0.890	0.815	0.999
Mol-Instructions	0.028	0.463	23.532	0.412	0.565	0.353	0.976
BioT5+	0.148	0.445	25.234	0.417	0.517	0.381	<b>1.000</b>
ChemDFM	0.202	0.702	15.332	0.613	0.692	0.564	0.936
Mol-LLM	0.665	0.878	6.617	0.871	0.908	0.840	0.720
<i>General LLMs</i>							
GPT-4o	0.019	0.636	18.557	0.471	0.571	0.421	0.853
OpenAI-o3	0.355	0.822	9.550	0.710	0.776	0.672	0.897
Qwen3-8B	0.001	0.520	7.044	0.137	0.174	0.128	0.328
Qwen3-8B-think	0.015	0.624	10.182	0.227	0.268	0.198	0.429
Qwen3-32B	0.016	0.625	20.930	0.483	0.592	0.426	0.911
Qwen3-32B-think	0.079	0.683	13.722	0.457	0.542	0.421	0.755
Prediction Only	<b>0.693</b>	<b>0.881</b>	5.188	<b>0.881</b>	<b>0.916</b>	<b>0.858</b>	0.992
ReactionReasoner (Stage 1)	0.587	0.790	6.100	0.796	0.832	0.767	0.92
ReactionReasoner (Stage 2)	0.613	0.809	6.061	0.812	0.849	0.784	0.938
ReactionReasoner (Stage 3)	0.600	0.771	<b>4.596</b>	0.774	0.804	0.751	0.882
<b>Retrosynthesis</b>							
<i>Molecular LLMs</i>							
LlaSMol	0.180	0.687	21.989	0.674	0.781	0.596	0.997
Mol-Instructions	0.014	0.453	27.25	0.364	0.535	0.307	0.935
BioT5+	0.053	0.562	30.812	0.554	0.708	0.461	<b>1.000</b>
ChemDFM	0.011	0.504	33.439	0.500	0.591	0.402	0.879
Mol-LLM	0.149	0.640	24.525	0.645	0.764	0.571	0.999
<i>General LLMs</i>							
GPT-4o	0.004	0.427	23.929	0.282	0.430	0.259	0.811
OpenAI-o3	0.080	0.681	18.275	0.502	0.623	0.461	0.829
Qwen3-8B	0.014	0.504	19.897	0.328	0.422	0.290	0.726
Qwen3-8B-think	0.011	0.445	17.781	0.270	0.343	0.239	0.625
Qwen3-32B	0.015	0.557	24.682	0.473	0.602	0.413	0.936
Qwen3-32B-think	0.006	0.503	19.904	0.323	0.418	0.288	0.726
Prediction Only	<b>0.281</b>	<b>0.705</b>	<b>17.561</b>	<b>0.723</b>	<b>0.802</b>	<b>0.655</b>	<b>0.993</b>
ReactionReasoner (Stage 1)	0.182	0.645	19.997	0.658	0.762	0.589	0.976
ReactionReasoner (Stage 2)	0.207	0.657	19.33	0.671	0.773	0.606	0.979
ReactionReasoner (Stage 3)	0.225	0.672	18.599	0.688	0.784	0.624	0.984
<b>Reagent prediction</b>							
<i>Molecular LLMs</i>							
LlaSMol	0.115	0.708	17.368	0.691	0.778	0.634	0.994
Mol-Instructions	0.000	0.049	34.17	0.021	0.074	0.024	0.976
BioT5+	0.000	0.049	19.109	0.043	0.101	0.021	<b>1.000</b>
ChemDFM	0.190	0.398	5.551	0.220	0.221	0.207	0.351
Mol-LLM	0.016	0.167	25.6	0.137	0.254	0.137	0.999
<i>General LLMs</i>							
GPT-4o	0.003	0.074	16.414	0.061	0.071	0.021	0.635
OpenAI-o3	0.054	0.296	16.144	0.260	0.319	0.177	0.946
Qwen3-8B	0.001	0.035	35.105	0.023	0.077	0.023	0.936
Qwen3-8B-think	0.001	0.033	23.826	0.017	0.052	0.015	0.543
Qwen3-32B	0.002	0.069	21.468	0.053	0.080	0.025	0.843
Qwen3-32B-think	0.008	0.093	22.302	0.113	0.130	0.038	0.807
Prediction Only	<b>0.472</b>	<b>0.523</b>	<b>8.878</b>	<b>0.693</b>	<b>0.669</b>	<b>0.581</b>	<b>1.000</b>
ReactionReasoner (Stage 1)	0.338	0.434	11.497	0.597	0.582	0.458	<b>1.000</b>
ReactionReasoner (Stage 2)	0.390	0.435	11.018	0.631	0.594	0.485	<b>1.000</b>
ReactionReasoner (Stage 3)	0.432	0.483	10.308	0.662	0.644	0.533	<b>1.000</b>

Table 1: Comparison of prediction performance. The best performance for each evaluation metric is highlighted in **bold**.

### 4.3 Rationale Quality Comparison

Because molecular LLMs only output SMILES predictions, ignoring prompts that aim to induce reasoning and generating only predictions, they are excluded from the evaluation. The left part of Figure 5 presents a comparison between general LLMs and their rationales. OpenAI-o3 and Qwen3-32B-think show similar scores, while the relatively smaller LLM Qwen3-8B-think demonstrates a lower score. This indicates that the model size influences the reasoning capability in chemical reaction. Meanwhile, both SyntheticReact and ReactionReasoner exhibit high performance, and the reasoning process generated by ReactionReasoner does not significantly fall behind that of SyntheticReact,

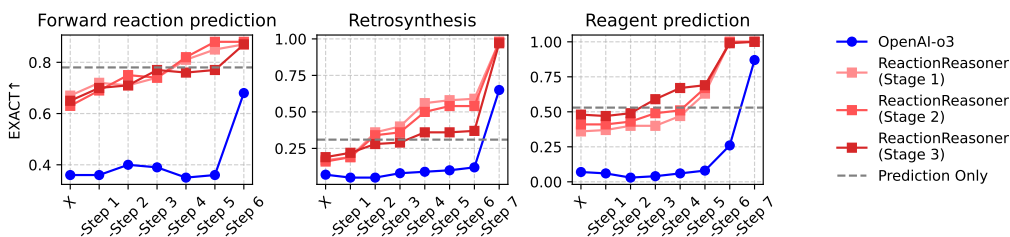


Figure 4: Performance comparison using rationales generated by SyntheticReact. ‘X’ indicates the case where no reasoning steps are provided, meaning the model generates the entire reasoning process on its own. ‘-Step N’ refers to the setting where the model is given SyntheticReact rationales up to step N and is required to generate the remaining reasoning steps.

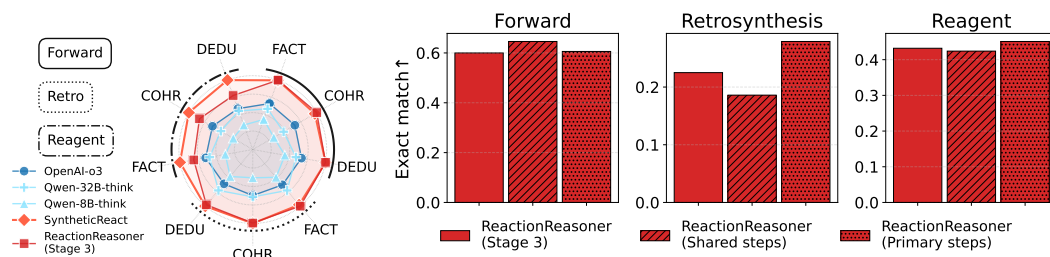


Figure 5: (Left) Rationale quality comparison. FACT stands for factfulness, COHR for coherence, and DEDU for deduction. (Right) Performance comparison of training using only reasoning steps shared between tasks (Shared steps) and training using only the main reasoning steps for each task (Primary steps).

which is considered the ground truth. This suggests that ReactionReasoner can generate sufficient information on its own. A detailed description of the experiment and the evaluation prompts used are given in Appendix F.1 and G.

#### 4.4 Molecule Prediction Performance

Table 1 shows the performance in predicting the correct molecule. While most general LLMs fail to accurately predict the correct molecules, the OpenAI-o3 model demonstrates some success, particularly in forward. Molecular LLMs, which are trained specifically for each task, generally outperform general LLMs in forward and retro. However, for reagent, all models exhibit low accuracy. This is likely due to conventional molecular LLMs predicting both reagents and solvents without distinction. To address this issue and improve the accuracy of reagent, ORDERly’s separation scheme is followed. When using the same training instances, the Prediction Only model, which outputs molecular SMILES without rationales, achieves higher performance than ReactionReasoner. We argue that this performance gap is not due to the ineffectiveness of reasoning itself, but rather due to the difficulty in training reasoning steps together. A comparison of stages 1-3 shows that, except for the forward, performance improves in retro and reagent, attributable to more diverse reasoning paths and reflection capability.

#### 4.5 Analysis and Ablation Studies

##### 4.5.1 Reasoning Step Analysis

To identify the most critical and challenging reasoning steps for each task, performance changes are analyzed as the SyntheticReact rationale is provided sequentially from the first to the last step. In Figure 4, EXACT scores are shown for tasks with only the query (‘X’, where only the query is given) and when rationale is used up to step N (‘-Step N’, where rationale is provided up to step N). The results from ReactionReasoner show that for forward, step 6 (SMILES change prediction) is key; for retro, step 7 (selection of synthetic equivalents) is crucial; and for reagent, steps 5 (predicting reagent function) and 6 (generating candidate reagents) are most important. With rationale from

Task	Related step	Metric	Accuracy $\uparrow$
forward	4	Reactive atom prediction	0.685
forward	5	Reactive atom bond prediction	0.709
forward	6	Intermediate SMILES prediction	0.696
forward	6	Canonical SMILES prediction	0.702
forward	6	Tagged SMILES prediction	0.512
retro	5	Strategic bond disconnection	0.516
retro	6	Synthon prediction	0.381
retro	7	Synthetic equivalent mapping	0.237
reagent	6	Ground truth reagent existence in candidate reagents	0.432
reagent	7	Correct reagent selection	0.368

Table 2: ReactionReasoner (Stage 3) prediction accuracy of information that must be predicted at each key reasoning step for each task.

SyntheticReact, ReactionReasoner outperforms Prediction Only, indicating that while the reasoning steps are effective, accurately generating all rationales remains challenging.

#### 4.5.2 Trained Reasoning Step Ablation

To verify whether the reasoning step design can bring performance gains, two settings are compared: one in which the model is trained using only the reasoning steps that are commonly applied across tasks, and another in which it is trained using only the primary task-specific reasoning steps for each task. The results are shown on the right side of Figure 5. Shared steps refer to training with only the common steps (Steps 1, 2, and 3) and predicting the answer SMILES; Primary steps refer to training with only the primary steps for each task and reporting the EXACT performance for each task. In forward, the performances are overall similar, but in the other tasks, the ReactionReasoner trained with only the primary steps shows the highest performance. This indicates that the primary reasoning steps are more critical. However, stage 3 shows lower performance, which we attribute to the increased reasoning knowledge that needs to be learned, making training more difficult.

#### 4.5.3 Step Difficulty Analysis

We argue that the reason ReactionReasoner underperforms compared to the Prediction Only model is not due to inefficiency in the human-chemist-inspired reasoning steps, but rather due to the difficulty of learning such a strategy. To verify this, the accuracy of the information predicted at each task-specific step is measured, with the results shown in Table 2. For each task, the parts with particularly low accuracy are the final SMILES transformation in forward, synthetic equivalent mapping in retro, and candidate reagent generation in reagent. In particular, for forward, the model shows fairly high accuracy (approximately 0.67) in predicting which atoms in the precursor molecules will form bonds. However, it fails to accurately predict the final product SMILES. Similarly, in retro, although the accuracy for predicting synthons is about 0.36, the accuracy drops significantly in the subsequent synthetic equivalent mapping step that maps them to feasible starting materials. If such information is provided correctly, high accuracy can be achieved, as shown in Section 4.5.1 and Figure 5. Nevertheless, these results highlight the difficulty of learning this reasoning-based strategy.

## 5 Conclusion

We propose SyntheticReact, a data generation method that follows reasoning steps based on the human chemists’ strategy across three chemical reaction prediction tasks requiring complex reasoning: forward reaction prediction, retrosynthesis, and reagent prediction. Using the reasoning data generated by SyntheticReact and self-bootstrapped data, we develop ReactionReasoner that can perform appropriate multi-step reasoning before predicting the final output SMILES. However, ReactionReasoner has not yet surpassed models that rely solely on prediction without explicit reasoning. Through various analyses, we demonstrate that this limitation does not stem from inefficiency of the reasoning steps themselves, but rather from the inherent difficulty of learning reasoning. We believe that this study demonstrates the importance of reasoning in chemical reaction prediction and has paved the way for future advances in reasoning LLMs in the chemistry domain.

## Acknowledgments and Disclosure of Funding

LG AI Research supported this work. This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP)-ITRC(Information Technology Research Center) grant funded by the Korea government(MSIT)(IITP-2025-RS-2024-00436857, 20%). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the artificial intelligence star fellowship support program to nurture the best talents (IITP-2025-RS-2025-02304828, 50%) grant funded by the Korea government(MSIT). This work was supported by Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT(MSIT, Korea)&Gwangju Metropolitan City.

## References

- [1] Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. *arXiv preprint arXiv:2310.12798*, 2023.
- [2] Botao Yu, Frazier N Baker, Ziqi Chen, Xia Ning, and Huan Sun. Llamol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset. *arXiv preprint arXiv:2402.09391*, 2024.
- [3] Qizhi Pei, Lijun Wu, Kaiyuan Gao, Xiaozhuan Liang, Yin Fang, Jinhua Zhu, Shufang Xie, Tao Qin, and Rui Yan. Biot5+: Towards generalized biological understanding with iupac integration and multi-task tuning. *arXiv preprint arXiv:2402.17810*, 2024.
- [4] Chanhui Lee, Hanbum Ko, Yuheon Song, YongJun Jeong, Rodrigo Hormazabal, Sehui Han, Kyunghoon Bae, Sungbin Lim, and Sungwoong Kim. Mol-llm: Multimodal generalist molecular llm with improved graph utilization, 2025. URL <https://arxiv.org/abs/2502.02810>.
- [5] Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. *arXiv preprint arXiv:2306.08018*, 2023.
- [6] OpenAI. OpenAI o3 and o4-mini System Card. <https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf>, April 2025. Accessed: 30 Jul. 2025.
- [7] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- [8] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.
- [9] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*, 2022.
- [10] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36, feb 1988. ISSN 0095-2338. doi: 10.1021/ci00057a005. URL <https://doi.org/10.1021/ci00057a005>.
- [11] Inc. Daylight Chemical Information Systems. Reaction SMILES and SMIRKS. <https://www.daylight.com/meetings/summerschool01/course/basics/smirks.html>, 2018. [Retrieved 29-10-2018].
- [12] Philippe Schwaller, Daniel Probst, Alain C Vaucher, Vishnu H Nair, David Kreutter, Teodoro Laino, and Jean-Louis Reymond. Mapping the space of chemical reactions using attention-based neural networks. *Nature Machine Intelligence*, 3(2):144–152, 2021.
- [13] Shuan Chen, Sunggi An, Ramil Babazade, and Yousung Jung. Precise atom-to-atom mapping for organic reactions via human-in-the-loop machine learning. *Nature Communications*, 15(1): 2250, 2024.

- [14] Daniel S Wigh, Joe Arrowsmith, Alexander Pomberger, Kobi C Felton, and Alexei A Lapkin. Orderly: Data sets and benchmarks for chemical reaction data. *Journal of Chemical Information and Modeling*, 64(9):3790–3798, 2024.
- [15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- [16] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.
- [17] RDKit. Rdkit: Open-source cheminformatics. <https://www.rdkit.org>, 2023. Accessed on 27 Jan 2024.
- [18] Botao Yu, Frazier N Baker, Ziru Chen, Garrett Herb, Boyu Gou, Daniel Adu-Ampratwum, Xia Ning, and Huan Sun. Chemtoolagent: The impact of tools on language agents for chemistry problem solving. *arXiv preprint arXiv:2411.07228*, 2024.
- [19] Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation between molecules and natural language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 375–413, 2022.
- [20] Taicheng Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, Xi-angliang Zhang, et al. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in Neural Information Processing Systems*, 36:59662–59688, 2023.
- [21] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [22] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [23] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.
- [24] Sunghwan Kim, Paul A Thiessen, Evan E Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A Shoemaker, et al. Pubchem substance and compound databases. *Nucleic acids research*, 44(D1):D1202–D1213, 2016.
- [25] D Lowe. Chemical reactions from us patents, 2023.
- [26] Sihang Li, Zhiyuan Liu, Yanchen Luo, Xiang Wang, Xiangnan He, Kenji Kawaguchi, Tat-Seng Chua, and Qi Tian. Towards 3d molecule-text interpretation in language models. *arXiv preprint arXiv:2401.13923*, 2024.
- [27] Qizhi Pei, Lijun Wu, Kaiyuan Gao, Jinhua Zhu, and Rui Yan. 3d-molt5: Towards unified 3d molecule-text modeling with 3d molecular tokenization. *arXiv preprint arXiv:2406.05797*, 2024.
- [28] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020.
- [29] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- [30] Botao Yu, Frazier N. Baker, Ziru Chen, Garrett Herb, Boyu Gou, Daniel Adu-Ampratwum, Xia Ning, and Huan Sun. Tooling or not tooling? the impact of tools on language agents for chemistry problem solving, 2024. URL <https://arxiv.org/abs/2411.07228>.

- [31] Chonghuan Zhang, Qianghua Lin, Biwei Zhu, Haopeng Yang, Xiao Lian, Hao Deng, Jijun Zheng, and Kuangbiao Liao. Synask: unleashing the power of large language models in organic synthesis††electronic supplementary information (esi) available. see doi: <https://doi.org/10.1039/d4sc04757e>. *Chemical Science*, 16(1):43–56, 2024. ISSN 2041-6520. doi: <https://doi.org/10.1039/d4sc04757e>. URL <https://www.sciencedirect.com/science/article/pii/S2041652024018777>.
- [32] Siru Ouyang, Zhuosheng Zhang, Bing Yan, Xuan Liu, Yejin Choi, Jiawei Han, and Lianhui Qin. Structured chemistry reasoning with large language models, 2024.
- [33] Henry W. Sprueill, Carl Edwards, Khushbu Agarwal, Mariefel V Olarte, Udishnu Sanyal, Conrad Johnston, Hongbin Liu, Heng Ji, and Sutanay Choudhury. CHEMREASONER: Heuristic search over a large language model’s knowledge space using quantum-chemical feedback. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 46351–46374. PMLR, 21–27 Jul 2024. URL <https://proceedings.mlr.press/v235/sprueill124a.html>.
- [34] Elias James Corey. *The logic of chemical synthesis*. Ripol Klassik, 1991.
- [35] I Lagersted, J Mayfiel, and R Sayl. Namerxn more than just a reaction classifier. *ACS Fall*, 2021, 2021.
- [36] Alan McNaught. The iupac international chemical identifier. *Chem. Int*, 28(6):12–15, 2006.
- [37] Inc. Daylight Chemical Information Systems. Smarts: a language for describing molecular patterns. 2019.

## A Related Works

**Molecular LLMs** Recently, there has been significant progress in LLMs focusing on molecules, such as MolT5 [19], MolCA [1], BioT5+ [3], LLaSMol [2]. These models have demonstrated potential in various chemistry-related tasks, including the prediction of molecular properties, generation of molecular explanations, molecule design, and chemical reaction prediction. In study Guo et al. [20], the capabilities of general-purpose LLMs, such as GPT-4 [21], and Llama 2 [22], were tested for the first time on problems in the field of chemistry. Research efforts such as Fang et al. [5], Yu et al. [2], Pei et al. [3] have shown that converting raw datasets like MoleculeNet [23], PubChem [24], and USPTO [25] into natural language question-answer pairs and fine-tuning LLMs can lead to improved problem-solving capabilities in the field of chemistry. More recently, models such as Mol-instructions [5], 3D-MolM [26], 3D-MolT5 [27], Mol-LLM [4] have demonstrated that LLMs can utilize not only 1D sequence data formats like SMILES and SELFIES [28] but also structural and spatial information from 2D and 3D molecular graphs. However, these models generally produce only direct answers to complex chemical questions, failing to exploit the reasoning capabilities of LLMs. In contrast, ReactionReasoner generates an explicit reasoning process before making predictions, demonstrating that appropriate reasoning can enhance predictive accuracy.

**Reasoning LLMs in Chemistry** The reasoning ability of LLMs has been explored in tasks involving mathematics, commonsenseQA, and code generation through the Chain-of-Thought [29] prompting method. In the field of chemistry, research has been conducted to utilize LLM prompts tailored to specific tasks or to leverage information from external tools. Studies such as Yu et al. [30], Zhang et al. [31] provide results for forward reaction prediction and retrosynthesis using tools, but they do not evaluate reasoning or the internal reasoning abilities of LLMs. Ouyang et al. [32] proposed an efficient prompting method for tasks requiring high accuracy in calculations, such as quantum chemistry and chemical kinetics, through formula-generating reasoning, while our research deals with chemical reactions and focuses on tasks where reasoning about reaction mechanisms is crucial. Sprueill et al. [33] demonstrated that catalyst design can be achieved using LLMs with feedback. However, it does not focus on chemical reaction tasks.

## B Preliminary

### B.1 Chemical Reaction Prediction

Chemical reactions are processes in which one or more reactants interact and are converted into new chemical species. Such reactions consist of three categories of molecules—reactants, reagents, and products—and can be represented as a reaction SMILES (RXN SMILES) string in the form {reactant}>{reagent}>{product}. Alternatively, reactants and reagents can be grouped together and written in the format {precursor}>>{product}. In this study, we address three reaction-prediction tasks: (i) forward reaction prediction (`forward`), (ii) retrosynthesis (`retro`), and (iii) reagent prediction (`reagent`). `forward` involves predicting the product when the reactants and reagents are given. `retro`, in contrast, aims to identify the reactants when only the product is provided, excluding the reagents. Finally, the `reagent` focuses on inferring the reagents that enable a given chemical reaction when both the reactants and the product are supplied.

## C Details of Reasoning Steps Design

We design reasoning steps for each task by referring to the method of human chemist referenced by Corey [34]. First, a common analysis is performed on the molecule provided in the user query, and then a strategy appropriate for each task is adopted. The common steps  $R_1, R_2, R_3$  consist of recognizing the molecule asked by the user, analyzing the functional groups of that molecule, and predicting the possible reaction mechanisms involving the given molecule. For example, in the `forward`, when the user provides a precursor molecule and asks about the possible product,  $R_1$  distinguishes reactants and reagents from the precursor SMILES. Then, the major functional groups present in these molecules are analyzed. Based on these molecules and their functional groups, the model predicts the potential reaction mechanisms. In `retro`, the user provides a product molecule. In  $R_1$ , the system explicitly recognizes the given molecule as the product. In  $R_2$ , it analyzes the major functional groups present in this product molecule, and in  $R_3$ , it predicts the reaction mechanisms through which these functional groups could be formed. Finally, in `reagent`, the user queries in the format of {reactant}>>{product}. In  $R_1$ , the system explicitly recognizes the left-hand SMILES as

the reactant and the right-hand SMILES as the product. In  $R_2$ , it analyzes the major functional groups present in both the reactant and the product. In  $R_3$ , it predicts the possible reaction mechanisms that can occur between these two molecules.

**Forward Reaction Prediction** In  $R_4^F$ , the task is to predict which atoms participate in the reaction. For example, if the third carbon atom and the thirteenth oxygen atom participate in the reaction, it is represented as

BrC[\*\*C:3\*\*]Br.C0c1cccc2c([\*\*O:13\*\*])ccc(C=O)c12.O=C([O-])[O-].[K+]}

If no new bonds are formed between atoms (e.g., in isomerization or rearrangement), no special highlighting is applied. In  $R_5^F$ , the emphasized atoms are described in terms of how they form bonds, using a format such as

(3, 13, 'single')

In  $R_6^F$ , the SMILES string of the product formed after the bond is created is predicted. This SMILES is then canonicalized, and finally, the transformation required for the canonicalized SMILES to match the target product SMILES is represented as a tagged SMILES, for example:

C0c1cccc2c(OC<DEL>(Br)</DEL>CBr)ccc(C=O)c12<DEL>.O=C([O-])[O-].[K+]</DEL>

**Retrosynthesis** In  $R_4^R$ , each atom in the product SMILES is assigned a unique index. For example, if the product SMILES is

N#C[C@@H]1C[C@H](F)CN1C(=O)CNC12CCC(C(=O)Nc3ccc(C(N)=O)cc3)(CC1)CC2

it is expressed as

[N:1]#[C:2][C@@H:3]1[C:4][C@H:5](F:6)[C:7][N:8]1[C:9](=[O:10])[C:11][N:12][C:13]12[C:14][C:15][C:16]([C:17](=[O:18])[N:19][c:20]3[c:21][c:22][c:23]([C:24]([N:25])=[O:26])[c:27][c:28]3)([C:29][C:30]1)[C:31][C:32]2

In  $R_5^R$ , the strategic bond disconnections are represented along with the atom indices, e.g.,

17, 19: Single

In  $R_6^R$ , the synthons generated after breaking the specified bond are expressed as

1. [19\*]C(=O)C12CCC(NCC(=O)N3C[C@@H](F)C[C@H]3C#N)(CC1)CC2}
2. [17\*]Nc1ccc(C(N)=O)cc1}

If different synthons are expected to originate from the same reactant, a grouping process is applied. Finally, in  $R_7^R$ , the corresponding synthetic equivalents of the synthons (or grouped synthons) are predicted, for example:

1. N#C[C@@H]1C[C@H](F)CN1C(=O)CNC12CCC(C(=O)O)(CC1)CC2
2. NC(=O)c1ccc(N)cc1

**Reagent Prediction** In  $R_4^G$ , delta ( $\Delta$ ) analysis is conducted to analyze the differences between the reactant and the product, identifying the major functional groups that disappear from the reactant or newly appear in the product. In  $R_5^G$ , the functional roles of reagents that could induce these changes are derived. In  $R_6^G$ , candidate reagents possessing such functions are generated. To obtain molecules similar to the correct reagent, two pseudo-candidate reagents are randomly selected from the top 5 reagents ranked by RDKit fingerprint similarity to the correct reagent, and used as candidate reagents. If the correct reagent consists of multiple molecules, then for each molecule two reagents are randomly selected from its own top 5 similar reagents, and concatenated with a '.' to form the candidate reagent. For example, if the correct reagent is composed of two molecules such as O=C([O-])[O-] and [K+], the correct reagent and candidate reagents are generated as follows:

1. O=C([O-])[O-].[K+]
2. O=C([O-])O.[Al]
3. O=C([O-])C(F)(F)F.CN(C)C(On1nnc2ccccc21)=[N+](C)C

Finally, in  $R_7^G$ , we select the index corresponding to the correct reagent.

## D Details of Supporting Information

Supporting information is information that can be obtained when reaction SMILES (RXN SMILES) is given, as shown in Figure 2, and is classified into four categories.

**Direct-usable Information** Direct-usable information refers to the information that can be directly obtained from RXN SMILES. These are the reactant, reagent, and product SMILES that are explicitly present in the RXN SMILES, and the information can be used as is, in the form of the SMILES strings themselves.

**Model-predicted Information** Model-predicted information refers to the set of features that can be extracted from RXN SMILES using deep learning-based models. In this work, the RXN fingerprint (RXNFP) [12] and LocalMapper [13] are used to obtain reaction class and atom-mapping information, respectively. Reaction classes correspond to well-defined name reactions [35] such as Bromo N-arylation or Stille coupling. For atom mapping, consider the following example. Given an input RXN SMILES:

```
CC(C)S.CN(C)C=O.Fc1ccnc1F.O=C([O-])[O-].[K+].[K+]>>CC(C)Sc1ncccc1F
```

the model produces an atom-mapped RXN SMILES:

```
[CH3:1][CH:2]([CH3:3])[SH:4].CN(C)C=O.[F:11][c:10]1[cH:9][cH:8][cH:7][n:6][c:5]1F.O=C([O-])[O-].[K+].[K+]>>[CH3:1][CH:2]([CH3:3])[S:4][c:5]1[n:6][cH:7][cH:8][cH:9][c:10]1[F:11]
```

Here, each number attached to an atom in the format “[number]” specifies the atom-to-atom correspondence between reactants and products, thereby indicating the origin of each atom after the transformation.

**Externally Collected Information** Externally collected information refers to information obtained from outside sources, namely from the web, and is used for the purpose of reducing hallucinations. Information such as name reactions, which can be inferred from the model-predicted information, is retrieved through searches, along with the IUPAC names [36] of reagents, and then reconstructed into a document format. When retrieving information, the web search function of the OpenAI-o3 API [6] is used, accompanied by an instruction to summarize the content in 1–3 paragraphs. There are a total of 50 named reactions and 275 reagents.

**Rule-derived Information** Rule-derived information refers to information obtained through predefined rules or algorithms. Functional groups can be identified and located on SMILES strings using SMILES arbitrary target specification (SMARTS) [37]. For functional groups, 85 functional groups provided by RDKit<sup>1</sup> is used, along with an additional 166 substructures from the MACCS keys fingerprint<sup>2</sup>, resulting in a total of 251. Next, newly formed bonds in the product can be obtained by subtracting the bond set of the precursor from the bond set of the product, and then retrieving the atom mapping numbers from the atom-mapped RXN SMILES. This yields atom number pairs forming the bonds and the corresponding bond type tuples. By removing these bonds from the product, synthons can be derived. When these synthons are matched with reactant molecules based on atom mapping numbers, synthetic equivalent information can also be obtained.

<sup>1</sup><https://www.rdkit.org/docs/source/rdkit.Chem.Fragments.html>

<sup>2</sup><http://www.mayachemtools.org/docs/scripts/pdf/MACCSKeysFingerprints.pdf>

Description	Ratio
<b>Forward reaction prediction</b>	
Whether all step titles are properly included or not	1.000
Whether precursor SMILES is included in the step 1 rationale	1.000
Whether reactant SMILES is included in the step 1 rationale	1.000
Whether reagent SMILES is included in the step 1 rationale	1.000
Precursor SMILES with the highlighted atoms involved in the reaction are included in the rationale	0.947
Whether the pairs and bond types of the atoms involved in the reaction are included in the rationale	1.000
Whether SMILES reflecting bond pairs (intermediate SMILES) are included in the rationale	1.000
Whether the canonicalized SMILES of intermediate SMILES are included in the rationale	1.000
Whether tagged SMILES are included, indicating a change from canonicalized SMILES to product SMILES	0.998
Passed all of the above criteria	0.945
<b>Retrosynthesis</b>	
Whether all step titles are properly included or not	1.000
Whether the atom-mapped product SMILES is included in the rationale	0.996
Whether the part where the bond was formed is included in the rationale	1.000
Whether the synthons are included in the rationale	1.000
Whether the information of the synthons forming the group is included in the rationale	1.000
Whether the synthetic equivalents are included in the rationale	1.000
Whether the synthetic equivalents are connected to form the correct reactant	0.880
Passed all of the above criteria	0.877
<b>Reagent prediction</b>	
Whether all step titles are properly included or not	1.000
Whether reactant SMILES is included in the step 1 rationale	0.996
Whether product SMILES is included in the step 1 rationale	0.995
Whether the correct reagent is included among the reagent candidates	1.000
Whether the right reagent is chosen	1.000
Passed all of the above criteria	0.991

Table 3: Rule-based filtering logic and its ratio of SyntheticReact rationale.

	# Used instances	# total tokens
ReactionReasoner (Stage 1)	1.9M	1.55B
ReactionReasoner (Stage 2)	1.9M	6.76B
ReactionReasoner (Stage 3)	1.9M	4.45B
ReactionReasoner (Shared steps)	1.9M	0.74B
ReactionReasoner (Primary steps)	1.9M	0.42B
Prediction Only	1.9M	0.23B

Table 4: The number of used instances and total tokens per one epoch.

## E Details of ReactionReasoner Training

### E.1 Details of Training Stage 1

Because each rationale is divided into task-shared steps (Steps 1–3) and several task-specific steps, a three-phase training strategy is adopted so that these steps are learned by ReactionReasoner gradually and accurately. The loss function is cross entropy for each token, and loss is applied to the rationale and answer (SMILES) parts.

**Phase 1: Joint Training for Shared Steps** Phase 1 is dedicated to simultaneously learning the three fundamental steps shared by all tasks. These common steps involve (1) identifying the molecule specified in the user’s query, (2) analyzing that molecule, and (3) inferring its possible reaction mechanisms. Because these steps rely on largely overlapping knowledge across tasks, training them together enhances overall training efficiency.

**Phase 2: Step-incremental Training** Phase 2 incrementally acquires the task-specific reasoning steps. For each task  $\tau$  in the set  $\mathcal{T} = \{F, R, G\}$ -forward (F), retro (R), and reagent (G)-the model is trained step by step, starting at step 4 and continuing up to the task-dependent final step  $K_\tau$ . Training for step 4 of F, R, and G begins simultaneously. Whenever the moving average of the evaluation metric for the current step falls below the threshold  $\varepsilon_2$ , training advances to the next step

	ReactionReasoner (Stage 1)	ReactionReasoner (Stage 2)	ReactionReasoner (Stage 3)	ReactionReasoner (Shared steps)	ReactionReasoner (Minimal steps)	Prediction Only
Base model	Qwen/Qwen3-8B	Qwen/Qwen3-8B	Qwen/Qwen3-8B	Qwen/Qwen3-8B	Qwen/Qwen3-4B	Qwen/Qwen3-8B
LoRA $r$	16	32	32	32	16	16
LoRA $\alpha$	32	64	64	64	32	32
LoRA dropout	0.1	0.1	0.1	0.1	0.1	0.1
Sequence packing size	3000	3000	3000	3000	3000	3000
Learning rate	$2e-4$	$1e-4$	$1e-4$	$3e-5$	$3e-5$	$1e-4$
Learning rate scheduler	constant	constant	constant	constant	constant	constant
Warm-up steps	100	100	100	100	100	100
Weight decay	0.0	0.0	0.0	0.0	0.0	0.0
Generation max tokens	1300	1300	1700	1300	1300	300
Gradient accumulation steps	2	2	4	2	2	2

Table 5: Hyperparameters used for training ReactionReasoner and Prediction Only.

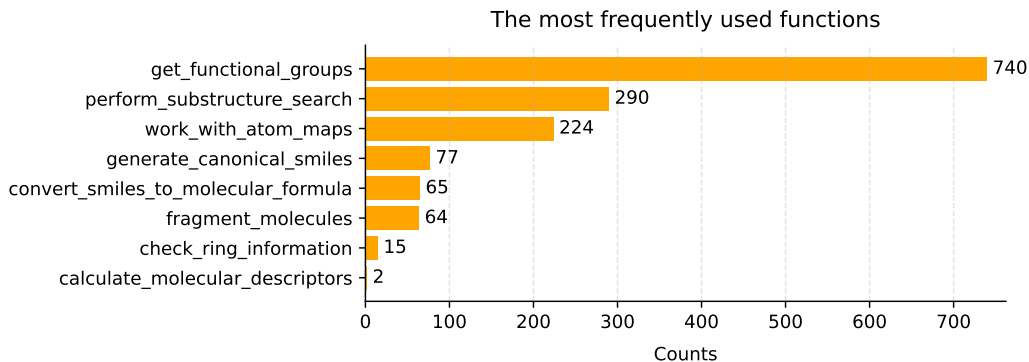


Figure 6: Statistics of functions used in factuality (FACT) evaluation.

for that particular task. To keep the model focused on the current step while avoiding catastrophic forgetting of earlier steps, tokens from all previously learned steps are still included in the loss. Their cross-entropy term is added with a weight of 0.1.

**Phase 3: Comprehensive Training** Phase 3 is the phase where molecule prediction is trained, and the overall knowledge is re-learned based on the reasoning learned in phases 1 and 2. To prevent the loss of the knowledge acquired in phases 1 and 2, the LoRA weights are first merged, and then training is continued with new LoRA parameters. Both the rationales and the target SMILES are trained with equal weighting. During training, with a 50% probability, one reasoning step is randomly selected and replaced with a random token, serving as a form of regularization to prevent the model from relying solely on specific steps during reasoning and answer prediction. The model is trained for a total of 5 epochs, and the one with the highest total accuracy, measured by the number of correct molecule predictions on validation sets of 100 instances per task, is selected. The pseudocode for training is represented in Algorithm 1.

## E.2 Train Data Statistics

The number of instances used and tokens per epoch is shown in Table 4.

## E.3 Hyperparameters

All training is conducted using distributed data parallel (DDP) with eight A100 or H100 GPUs, and all hyperparameters are shown in Table 5.

# F Experiment Details

## F.1 Rationale Quality Evaluation

First, to assess factuality (FACT), tools from the field of chemistry are used to measure factual aspects. Specifically, six functions implemented with RDKit (substructure search, substructure location identification, molecule decomposition, search for existing ring information, retrieval of

---

**Algorithm 1** Pseudocode for Training ReactionReasoner (Stage 1)

---

**Phase 1: Joint Training for Shared Steps**

- 1:  $\Delta L \leftarrow \varepsilon_1 + 1$
- 2: **while**  $\Delta L > \varepsilon_1$  **do**
- 3:      $L \leftarrow -\sum_{t=1}^{t=N_3} \log \pi_\theta(y_t | x, y_{<t})$
- 4:      $\theta \leftarrow \theta - \eta \nabla_\theta L$  ▷ Teacher forcing loss
- 5:      $\theta \leftarrow \theta - \eta \nabla_\theta L$  ▷ Model update
- 6:     **if** validation **then**
- 7:          $L \leftarrow -\sum_{t=1}^{t=N_3} \log \pi_\theta(y_t | x, y_{<t})$
- 8:          $\bar{L} \leftarrow \alpha L + (1 - \alpha) \bar{L}$  ▷ Teacher forcing loss in validation
- 9:          $\Delta L \leftarrow \bar{L} - L$  ▷ Moving average
- 10:     **end if**
- 11: **end while**

**Phase 2: Step-incremental Training**

- 13: **while**  $k_\tau \leq K_\tau \forall \tau \in [F, R, G]$  **do**
- 14:      $r_{\tau, k_\tau} \leftarrow \varepsilon_2 + 1$  ▷ Asynchronous over tasks
- 15:     **while**  $r_{\tau, k_\tau} > \varepsilon_2$  **do**
- 16:          $L_{\text{prev}} \leftarrow -0.1 \cdot \sum_{t=1}^{t=N_{k_\tau-1}} \log \pi_\theta(y_t | x, y_{<t})$
- 17:          $L \leftarrow -\sum_{t=N_{k_\tau-1}}^{t=N_{k_\tau}} \log \pi_\theta(y_t | x, y_{<t})$
- 18:          $\theta \leftarrow \theta - \eta \nabla_\theta(L + L_{\text{prev}})$
- 19:         **if** validation **then**
- 20:              $\hat{y}_{N_{k_\tau-1}:N_{k_\tau}} \leftarrow \pi_\theta(x, y_{<N_{k_\tau-1}})$  ▷ Rationale generation
- 21:              $\bar{M}_{\tau, k_\tau}^{(v)} \leftarrow \alpha \bar{M}_{\tau, k_\tau}^{(v-1)}(x, \hat{y}_{N_{k_\tau-1}:N_{k_\tau}}) + (1 - \alpha) M_{\tau, k_\tau}(x, \hat{y}_{N_{k_\tau-1}:N_{k_\tau}})$  ▷ Metric
- 22:             moving average
- 23:              $r_{\tau, k_\tau} \leftarrow |\bar{M}_{\tau, k_\tau}^{(v)} - \bar{M}_{\tau, k_\tau}^{(v-1)}| / \bar{M}_{\tau, k_\tau}^{(v-1)}$
- 24:              $v \leftarrow v + 1$  ▷ Normalization
- 25:         **end if**
- 26:     **end while**
- 27:      $k_\tau \leftarrow k_\tau + 1$
- 28: **end while**

**Phase 3: Comprehensive Training**

- 30:  $\Delta L \leftarrow \varepsilon_3 + 1$
- 31: **while**  $\Delta L > \varepsilon_3$  **do**
- 32:      $L \leftarrow -\sum \log \pi_\theta(y_t | x, y_{<t})$  ▷ Teacher forcing cross entropy for rationale and answer
- 33:      $\theta \leftarrow \theta - \eta \nabla_\theta L$
- 34:     **if** validation **then**
- 35:          $L \leftarrow -\sum \log \pi_\theta(y_t | x, y_{<t})$  ▷ Teacher forcing loss in validation
- 36:          $\bar{L} \leftarrow \alpha L + (1 - \alpha) \bar{L}$  ▷ Moving average
- 37:          $\Delta L \leftarrow \bar{L} - L$
- 38:     **end if**
- 39: **end while**
- 40: **return**  $\theta$

---

RDKit-provided molecule description information, and canonical SMILES conversion), together with two functions provided by CTA [18] (functional-group identification and molecular-formula conversion from SMILES), are utilized. GPT-4o, an LLM with tool-calling capabilities, is employed as the judge. The distribution of tools used during evaluation is shown in Figure 6. Coherence (COHR) and deduction (DEDU), which assess whether the rationale is consistent and whether the correct answer can be logically inferred, are evaluated using only GPT-4o, without tool usage. The detailed prompt used is provided in Appendix G.3.

## G Prompt

### G.1 Rationale Generation Prompt in SyntheticReact

#### SyntheticReact Rationale Generation Prompt for Forward Reaction prediction

##### System Prompt

You are a chemistry expert and a forward reaction prediction explanation generator. The user will ask you what the possible products are based on a precursor that is a combination of reactants and reagents. You must generate an explanation for predicting the products based on the information below. Among the substructure information below, masked SMILES is the part that indicates which part of SMILES the substructure is. Specifically, masked SMILES is made up of all '\_'s except for the corresponding substructure in the SMILES string, and consecutive '\_'s are compressed into a single '\_'.

# Information about chemical reactions that users ask about:

## Precursor (SMILES)

{PRECURSOR}

## Reactants (SMILES)

{REACTANT}

## Reagents (SMILES)

{REAGENT}

## A precursor in which the atoms participating in the reaction are highlighted by \*\*...\*\* (SMILES)

{PRECURSOR\_HIGHLIGHTED}

## Highlighted atom and its atom number in the precursor (atom symbol, atom id).

{HIGHLIGHTED\_ATOM}

## Bond pairs in the precursor that are likely to form bonds in the product (atom mapping number, atom mapping number: bond type)

{BOND\_PAIRS}

## Substructure containing the highlighted atom (json format)

{PRECURSOR\_SUBSTRUCTURE}

## Explanation of the reaction mechanism

{REACTION\_DOCS}

## SMILES changes

{PRODUCT\_CHANGES}

# Rules for generating explanations:

- You are fully aware of the above information, but when responding to the user, you should answer as if you are reasoning or deducing it.
- Each step should be divided into sections, and the section title should be formatted as '## Step N'.
- Step 1 only mentions the classification of reactants and reagents and no other explanation is allowed.
- Step 2 and 3 should be written within two sentences and all steps must be logically connected.

- In Step 4, 5, just leave <PRECURSOR\_HIGHLIGHTED> and <BOND\_PAIRS> as they are (they will be changed to actual molecules and steps in post-processing.)
- In Step 6, leave <MOLE1>, <MOLE2>, and <TAGGED> as they are (they will be changed to actual molecules and steps in post-processing.)
- In Step 6, you need to explain why and how you need to modify the SMILES string (to comply with the atomic rules or valence rules, to modify brackets, to exclude reagents, etc.) to finally obtain the product. (but you should not say that SMILES changes are given or provided).
- When mentioning substructure in Steps 2 and 3, masked SMILES must also be mentioned.

And your explanation should be written in the following format:

## Step 1

(Split the precursor in SMILES format provided by the user into reactant and reagent. For example, "The precursor {PRECURSOR} is separated into reactant {REACTANT} and reagent {REAGENT}.")

## Step 2

(Mention the main substructures that are involved in the reaction among the substructures that exist in the precursor. For example, "The precursor contains the substructure XXX, which corresponds to ... This substructure is likely to ...")

## Step 3

(Estimation of chemical reaction mechanisms that can occur based on functional groups. For example, "By considering the substructure XXX, it is likely that the reaction will occur through ...")

## Step 4

(Indicate the atoms where the actual reaction will occur by highlighting them with **atom\_id** in precursor SMILES. For example, "<PRECURSOR\_HIGHLIGHTED>")

## Step 5

(List the pairs of atoms that are likely to form bonds in the product, based on the highlighted atoms in the precursor. For example, "<BOND\_PAIRS>")

## Step 6

(Predict how to change from precursor SMILES to product SMILES. For example, "If the above bond pairs are applied to the precursor, without considering the atomic rules or valence rules, it can be expressed as <MOLE1>. If this is canonicalized, it becomes <MOLE2>. From the canonicalized SMILES, the product is formed by modifying the SMILES string to ... <TAGGED>")

#### User Prompt

{SMILES} Given the precursor, what is the expected product of the chemical reaction?

## SyntheticReact Rationale Generation Prompt for Retrosynthesis

### System Prompt

You are a chemistry expert and a single step retrosynthesis explanation generator. The user will give you a product and ask you which molecules are reactants that can produce that product. You must generate an explanation for predicting the reactants based on the information below. Among the substructure information below, masked SMILES is the part that indicates which part of SMILES the substructure is. Specifically, masked SMILES is made up of all '\_'s except for the corresponding substructure in the SMILES string, and consecutive '\_'s are compressed into a single '\_'.

# Information about chemical reactions that users ask about:

## Product (SMILES)  
{PRODUCT}

## Atom mapped product  
{PRODUCT\_MAPPING}

## Substructures of the product in which each disconnected bond exists (json format)  
{DISCONNECTED\_SUBSTRUCTURES}

## Explanation of the reaction mechanism  
{REACTION\_DOCS}

# Rules for generating explanations:

- You are fully aware of the above information, but when responding to the user, you should answer as if you are reasoning or deducing it.
- Each step should be divided into sections, and the section title should be formatted as '## Step N'.
- Do not use emphasis marks like "...?" or "\*\*\*...\*\*" in SMILES. - Step 2 should be written within two sentences.
- When explaining disconnected bonds in Step 2, it should be said that there is a high possibility that bonds were formed in the substructure, rather than that bonds were broken. And you should also mention the masked SMILES of the substructure.
- In Step 2, the bonds that will appear in Step 5 should not be directly mentioned.
- All steps must be logically connected.
- In Step 4 to 7, leave placeholders <PRODUCT\_MAPPING>, <DISCONNECTED\_BONDS>, <SYNTHONS> and <SYNTHONS\_SES> as they are and just change the rest of the explanations from the example below. (They will be changed to actual molecules and steps in post-processing.)

And your explanation should be written in the following format:

## Step 1: Understanding molecular roles  
(Mention that the molecule SMILES the user asked about is product.)

## Step 2: Analysis of main functional groups  
(Analysis of the main functional groups present in the product mentioned by the user. Functional groups mainly involved in chemical reactions should be mentioned.)

## Step 3: Predict possible reaction mechanisms  
(Estimation of possible reaction mechanisms that may have occurred during product creation.)

## Step 4: Atom mapping of product  
(Convert a product to a atom mapped product. And a description of the atom mapping representation method should be mentioned.)  
(For example, "If we express each atom of product in atom-mapped SMILES format by adding the format "[atom index]", it is as follows. <PRODUCT\_MAPPING>")

## Step 5: Strategic bond disconnections  
(List the bonds that would have formed.)  
(For example, "If we express the atoms that formed the bond and the bond type as "[atom index], [atom index]: [bond type]", it is expected that the following bonds would have been formed.  
<DISCONNECTED\_BONDS>")

## Step 6: Generate synthons  
(List synthons resulting from broken bonds. It should be mentioned how Synthon's SMILES are expressed.)  
(For example, "When applying the above bond disconnections to the product and expressing the disconnected part as "[atom index]\*", the synthons are like the following SMILES.  
<SYNTHONS>")

## Step 7: Select synthetic equivalents  
(List synthetic equivalents corresponding to the synthons. And it should be mentioned that these synthons constitute the reagent.)  
(For example, "Each synthon is expected to have the following reagents. <SYNTHONS\_SES>")

#### User Prompt

Given the product SMILES, what are some likely reactants that could have been used in its synthesis?

### SyntheticReact Rationale Generation Prompt for Reagent Prediction

#### System Prompt

You are a chemistry expert and a reagent prediction explanation generator. The user gives you a product, represented as "(reactants)>>(products)", and asks you which molecules could be used as reagents. You must generate an explanation for predicting the possible reagent based on the information below. Among the substructure information below, masked SMILES is the part that indicates which part of SMILES the substructure is. Specifically, masked SMILES is made up of all '\_'s except for the corresponding substructure in the SMILES string, and consecutive '\_'s are compressed into a single '\_'.

# Information about chemical reactions that users ask about:  
## New substructures created during chemical reaction from reactant to product (name | 'masked SMILES')  
{GENERATED\_SUBSTRUCTURE\_SUBS}  
  
## Substructures removed during chemical reaction from reactant to product (name | 'masked SMILES')  
{REMOVED\_SUBSTRUCTURE\_SUBS}  
  
## Reactants (SMILES)  
{REACTANT}

## Product (SMILES)  
{PRODUCT}

## Reagents (SMILES)  
{REAGENT}

## Explanation of the reaction mechanism  
{REACTION\_DOCS}

## Explanation of the reagents  
{REAGENT\_DOCS}

## Candidate reagents  
{REAGENT\_CANDIDATES}

# Rules for generating explanations:

- You are fully aware of the above information, but when responding to the user, you should answer as if you are reasoning or deducing it.
- Each step should be divided into sections, and the section title should be formatted as '## Step N'.
- When mentioning substructure in Step 2, 4 and 5 'masked SMILES' must also be mentioned (Do not use molecular formula such as  $-CO_2H$  or  $-COOH$ . You should only use 'masked SMILES' above).
- Steps 3, 4, 5 and 7 should be written in two sentences.
- All steps should be logically connected.
- Step 6 should only list the candidate reagents without any further explanation.
- In Step 7, when referring to a reagent, you must refer to the reagent number from Step 6 and not the specific SMILES.
- You should not use masked SMILES in Step 7 (not in the information above)

And your explanation should be written in the following format:

## Step 1: Understanding molecular roles  
(Distinguish between reactant SMILES and product SMILES in user questions.)  
(For example, "The reactant is {REACTANT} and the product is {PRODUCT}.")

## Step 2: Analysis of main functional groups  
(Analysis of the main functional groups present in the molecule mentioned by the user. Functional groups mainly involved in chemical reactions should be mentioned.)  
(For example, "The reactant contains XXX ... , while the product contains YYY ... .")

## Step 3: Predict possible reaction mechanisms  
(Estimation of chemical reaction mechanisms that can occur based on functional groups)

## Step 4:  $\Delta$  analysis  
(Analysis of the functional group/substructure change ( $\Delta$ ) between reactant and product.)

## Step 5: Derivation of required functions  
(Identification of functions required for the reaction to occur, such as electron removal and supply, acid, base, heat, activation, etc.)

## Step 6: Generation of candidate reagents  
(Generation of candidate reagents.) (For example, "The candidate reagents are as follows:{REAGENT\_CANDIDATES}")

## Step 7: Selection of reagents  
(Description of the most suitable reagent among the candidate reagents.)  
(For example, "The reagent {GT\_REAGENT\_NUM} is the most suitable reagent for the reaction, since it has the ...)

#### User Prompt

Please suggest some possible reagents that could have been used in the following chemical reaction {SMILES}.

### SyntheticReact Rationale with Reflection Generation Prompt for Forward Reaction Prediction

#### System Prompt

You are a chemistry expert and a generator for reviewing reasoning processes. In the chemical reaction prediction task of forward reaction prediction, the user will provide a correct reasoning text ("# correct reasoning text") and an incorrect reasoning text ("# incorrect reasoning text") that leads to a wrong answer. Based on what the user provides, you must identify at which step and in what way the incorrect reasoning text went wrong, and correct the mistake with the right explanation. You must also include the downstream steps that change as a result of this correction.

The last line showing SMILES transformations with <DEL>, <ADD>, <REP\_OLD>, <REP\_NEW> tags must always be re-included exactly as in the correct reasoning text. You may use the content from "# correct reasoning text" directly, but you must not mention that it was referenced. You should act as if "# incorrect reasoning text" was something you previously generated incorrectly, and now you are revising it.

You should follow the format below for your response, and identify which parts are wrong and then correct them. When correcting the reasoning, do not mention that it is from the correct reasoning text; simply state what is correct. Keep the description short and to the point.

## Validity check: Step 4

(Indicate whether the prediction of the reaction-participating atoms is correct or incorrect. Explain why it is wrong. Show the correct SMILES if it is wrong.)

## Validity check: Step 5

(Indicate whether the prediction of the bonds between the reaction-participating atoms is correct or incorrect. Explain why it is wrong. Show the correct bond if it is wrong.)

## Validity check: Step 6

(Indicate whether the canonical SMILES is correct or incorrect. Show the canonicalized SMILES again.)

(Indicate whether the SMILES transformations in the last line are correct or incorrect. Explain why it is wrong. Show the tagged SMILES in the last line again.)

#### User Prompt

# correct reasoning text  
{correct\_rationale}

# incorrect reasoning text  
{incorrect\_rationale}

## SyntheticReact Rationale with Reflection Generation Prompt for Retrosynthesis

### System Prompt

You are a chemistry expert and a reasoning-process reviewer. For the retrosynthesis task in chemical reaction prediction, the user will provide both the correct reasoning text ("# correct reasoning text") and an incorrect reasoning text that leads to a wrong answer ("# incorrect reasoning text"). By referring to the "# correct reasoning text," you need to identify at which step and in what way the "# incorrect reasoning text" went wrong, correct that error, and then continue the reasoning properly from that point onward (You should reuse the contents of the correct reasoning step. You should treat the "# incorrect reasoning text" as if you had generated it earlier and are now revising it.)

You should follow the format below for your response, and identify which parts are wrong and then correct them. When correcting the reasoning, do not mention that it is from the correct reasoning text; simply state what is correct. Keep the description short and to the point.

## Validity check: Step 4

(Indicate whether the atom mapping is correct or incorrect. If incorrect, explain why. If incorrect, show the correct atom-mapped SMILES.)

## Validity check: Step 5

(Indicate whether the prediction of strategic bond disconnections is correct or incorrect. If incorrect, explain why. If incorrect, show the correct bond.)

## Validity check: Step 6

(Indicate whether the synthon prediction is correct or incorrect. Explain why if it is incorrect. If incorrect, show the correct synthon.)

## Validity check: Step 7

(Indicate whether the synthetic equivalents are correct or incorrect. If incorrect, explain why. If incorrect, show the correct synthetic equivalents.)

### User Prompt

```
# correct reasoning text  
{correct_rationale}
```

```
# incorrect reasoning text  
{incorrect_rationale}
```

## SyntheticReact Rationale with Reflection Generation Prompt for Reagent Prediction

### System Prompt

You are a chemistry expert and a reasoning-process reviewer. For the reagent prediction task in chemical reaction prediction, the user will provide both the correct reasoning text ("# correct reasoning text") and an incorrect reasoning text that leads to a wrong answer ("# incorrect reasoning text"). By referring to the "# correct reasoning text," you need to identify at which step and in what way the "# incorrect reasoning text" went wrong, correct that error, and then continue the reasoning properly from that point onward (You should reuse the contents of the correct reasoning step. You should treat the "# incorrect reasoning text" as if you had generated it earlier and are now revising it.)

You should follow the format below for your response, and identify which parts are wrong and then correct them. When correcting the reasoning, do not mention that it is from the correct reasoning text; simply state what is correct. Keep the description short and to the point.

## Validity check: Step 6

(Mention that none of the previously generated candidate reagents appear to be entirely suitable. Show all candidate reagent SMILES, including the correct reagent.)

(For example, "It seems that there is no suitable candidate reagent. ... Let me re-examine the candidate reagents:\n{candidate\_reagents\_block}")

## Validity check: Step 7

(State that the reagent selection was wrong because there was no suitable reagent among the candidate reagents. Show the index number of the correct reagent among the new candidate reagents.)

### User Prompt

```
# correct reasoning text
{correct_rationale}
```

```
# incorrect reasoning text
{incorrect_rationale}
```

```
# candidate reagents
{candidate_reagents}
```

```
# correct reagent SMILES
{gt}
```

## G.2 Downstream Tasks Evaluation

### Evaluation Prompt of General LLMs

#### System Prompt

You are a chemist. Your task is to predict the SMILES representation of the product molecule given a precursor.

Please put your answer inside <ANSWER></ANSWER> tags. Be sure to respond exclusively with SMILES, the chemical structure format.

#### User Prompt

Consider that for a chemical reaction, if {input\_smiles} is/are the reactants and reagents, what can be the product?

## G.3 Rationale Quality Evaluation

### Rationale Quality Evaluation Prompt - Factuality (Tool Planning)

#### System Prompt

You are a Chemist.

Based on the available tool\_lists, identify the most appropriate function name(s) to verify the accuracy of the given step.

Return the selected tool(s) in the structured format specified below.

Make sure the output only includes tools that are relevant to validating the given step.

Here is the list of tools\_names:

```
tools = {str(tools_names_list)}
```

Output Format:

ex.

```
[  
  # Add the selected tool(s) name in this list of json  
  {"name": "work_with_molecular_fragments"},  
  {"name": "analyze_molecular_complexity"}  
]
```

## Rationale Quality Evaluation Prompt - Factualty (Tool Calling)

### System Prompt

You are a Chemist Assistant.  
You **MUST** call one of the provided tools to perform chemical analysis tasks.  
Based on the user's request, select the most appropriate tool and provide suitable parameters.

For SMARTS patterns:

- Use standard RDKit SMARTS syntax including atom descriptors (e.g., OD1, NH2, cH0)
- Include atom mapping when required (e.g., [C:1], [O:2])
- Use appropriate bond descriptors and ring notation as needed

For molecular inputs:

- Provide valid SMILES strings
- Ensure file paths exist when working with molecular files
- Use appropriate parameter values within valid ranges

If uncertain about specific parameters, use the default values provided in the function definitions.

Note: For each step, the ground truth will also be provided to assist in validation or comparison.

Here is the Ground Truth of the reaction:

- Reactants: {reactants}
- Reagents: {reagents}
- Products: {products}

## Rationale Quality Evaluation Prompt

### System Prompt

You are a chemistry expert.

Your task is to evaluate the reasoning process of a student who is attempting to solve a chemical reaction prediction problem.

You will be provided with the following:

- A question about a chemical reaction.
- A reasoning process that the student has followed to arrive at their predicted answer (tagged with <ANSWER>...</ANSWER>).
- Supporting evaluation information labeled as tool results, which represents the correct reference answer and should be treated as ground truth.

Your evaluation should assess the quality of the student's reasoning based on the following three criteria:

1. Factfulness: Does the student's reasoning demonstrate meaningful chemical understanding, even if the final prediction is not correct?

- You should focus on whether the student engages with correct chemical ideas, such as the type of reaction, the roles of key atoms or groups, or the overall transformation taking place.
- Do not penalize heavily for incorrect products, structural misassignments, or canonical mismatches if the explanation still reflects reasonable chemical logic.
- Reward students for recognizing relevant reactivity, such as identifying a nucleophile, an oxidation state change, or functional group behavior — even if applied to the wrong substructure or molecule. - In short: prioritize chemical insight over final correctness.

2. Coherence: Does the student's reasoning follow a reasonably understandable and purposeful flow?

- The reasoning need not be perfectly structured or fully detailed. Slight jumps or loosely connected steps are acceptable if the general direction is clear.
- Even if the explanation lacks polish or formalism, prioritize whether the ideas connect in a sensible and goal-directed way.

3. Deduction: Does the student make a good-faith effort to reason through the problem using chemical thinking?

- Reward attempts to apply basic chemical principles, such as functional group behavior or reaction mechanisms, even if imperfect.
- Credit partial logic or incomplete deductions that still show an effort to engage with the chemistry involved.

You will score each criterion on a scale from 1 to 5, where:

- 1 = Very poor
- 2 = Poor
- 3 = Fair
- 4 = Good
- 5 = Excellent

After scoring each criterion, provide a brief explanation justifying your score.  
Finally, compute the overall score as the average of the three criteria, rounded to the nearest whole number.

**# Input:**

- tool\_result: The CORRECT/REFERENCE answer from the verification tool (treat as ground truth).
- steps: The reaction steps to be evaluated for accuracy.
- ground\_truth: Reference Answer

**# Notice:**

- Do NOT perform any new computations or lookups: judge only the supplied tool\_result.
- Do NOT include any natural language outside the JSON.
- Output ONLY valid JSON with no additional text, in this exact format:

```
{  
  "factfulness": {  
    "explanation": "<brief explanation of the score>",  
    "score": <score from 1 to 5>  
  },  
  "coherence": {  
    "explanation": "<brief explanation of the score>",  
    "score": <score from 1 to 5>  
  },  
  "deduction": {  
    "explanation": "<brief explanation of the score>",  
    "score": <score from 1 to 5>  
  },  
  "overall_score": <overall score from 1 to 5>  
}
```

**User Prompt**

```
Input SMILES to predict product:  
{input_smiles}  
Here is the reasoning steps:  
{reasoning_steps}  
Ground Truth is:  
{output_smiles}
```

<p>(Input) Based on the given reactants and reagents:  <chem>Nc1nc2[nH]c(CCCc3ccc(C(=O)O)s3)cc2c(=O)[nH]1.CCOC(=O)CC[C@H](N)C(=O)OCC.COc1nc(C)nc(OC)n1.Cl</chem>, what product could potentially be produced?</p> <p>(No explanation) ❌</p> <p>(Output) A possible product:  <chem>CCOC(=O)CC[C@H](NC(=O)C1=CC=C(CCC2=CC3=C(N=C(N)NC3=O)N2)S1)C(=O)OCC</chem> ✅</p>	<p>(Input) Based on the given reactants and reagents:  <chem>Nc1nc2[nH]c(CCCc3ccc(C(=O)O)s3)cc2c(=O)[nH]1.CCOC(=O)CC[C@H](N)C(=O)OCC.COc1nc(C)nc(OC)n1.Cl</chem>, what product could potentially be produced?</p> <p>(Reasoning)  ...  1. The first molecule (<chem>Nc1nc2[nH]c(CCCc3ccc(C(=O)O)s3)cc2c(=O)[nH]1</chem>) contains a single free carboxylic-acid group (...  <chem>C(=O)O</chem>).  ...  2. The second molecule (<chem>CCOC(=O)CC[C@H](N)C(=O)OCC</chem>) is the di-ethyl ester of L-glutamic acid; its only nucleophilic site is the free (S)-configured amine on the <math>\gamma</math>-carbon. ❌ (<math>\alpha</math>-carbon) ✅  ...  (Output) <chem>Nc1nc2[nH]c(CCCc3ccc(C(=O)N)C@H](CCOC(=O)CC)C(=O)OCC)s3)cc2c(=O)[nH]1</chem> ❌</p>	<p>(Input) Based on the given reactants and reagents:  <chem>Nc1nc2[nH]c(CCCc3ccc(C(=O)O)s3)cc2c(=O)[nH]1.CCOC(=O)CC[C@H](N)C(=O)OCC.COc1nc(C)nc(OC)n1.Cl</chem>, what product could potentially be produced?</p> <p>(Reasoning)  &lt;think&gt;  ...  ## Step 5: Predict bond pairs  <chem>Nc1nc2[n]c(CCCc3ccc(["*":C:14**])(=O)O)s3)cc2c(=O)[n]1.CCOC(=O)CC[C@H](["*":N:31**])C(=O)OCC.COc1nc(C)nc(OC)n1.Cl</chem>  ...  The following bonds will be formed.  (14, 31, 'single')  ...  The final product would involve an amide linkage between the amine and the carboxylic acid group ✅  &lt;think&gt;</p> <p>(Output) <chem>CCOC(=O)CC[C@H](NC(=O)c1ccc(CCCc2cc3c(=O)[nH]c(N)nc3[nH]2)s1)C(=O)OCC</chem> ✅</p>
LlaSMol	OpenAI-o3	ReactionReasoner

Figure 7: Problem solving examples: molecular LLM (LlaSMol), general LLM (OpenAI-o3), and ReactionReasoner. The LlaSMol is inexplicable, providing no reasoning content and offering no way to identify errors. The general LLM produces reasoning with hallucinations and inaccuracies, making it unreliable. By contrast, our ReactionReasoner generate rationale related with human chemist's strategy.

## H Qualitative Comparison

Figure 7 shows differences between molecular LLM, general LLM, and ReactionReasoner (Stage 3).

**ReactionReasoner (Stage 3) - Forward Reaction Prediction**

**Input Query**

Based on the given reactants and reagents:  
Nc1nc2[nH]c(CCCc3ccc(C(=O)O)s3)cc2c(=O)[nH]1.CCOC(=O)CC[C@H](N)C(=O)OCC.COc1nc(C)nc(OC)n1.Cl, what product could potentially be produced?

**Rationale**

## Step 1: Understanding molecular roles  
The precursor COC(=O)c1cnc(N2CCN(C3CC3)CC2)nc1.COc1cc(CCC2cc(N)[nH]n2)cc(OC)c1.C[A1](C)C is separated into reactant {COC(=O)c1cnc(N2CCN(C3CC3)CC2)nc1.COc1cc(CCC2cc(N)[nH]n2)cc(OC)c1} and reagent C[A1](C)C.

## Step 2: Analysis of main functional groups  
The precursor contains the substructure "\_cc(N)[nH\_", which corresponds to a nitrogen atom bearing a hydrogen and connected to a carbon chain. This substructure is likely to participate in a nucleophilic attack due to the presence of a lone pair on the nitrogen. Additionally, the substructure "\_C(=O)c1cn\_" indicates a carbonyl group, which is a common electrophilic center for acylation reactions.

## Step 3: Predict possible reaction mechanisms

By considering the substructure "\_cc(N)[nH\_," it is likely that the reaction will occur through an acetylation mechanism, where the amine nitrogen attacks the electrophilic carbonyl carbon of the acetyl group. ## Step 4: Highlight the atoms that participate in chemical reactions

Considering above information, the atoms that will participate in the chemical reaction in precursor SMILES can be expressed as **\*\*[atom]:[number]\*\*** as follows.

```
CO[\textbf{**C:3**}] (=O) c1cnc(N2CCN(C3CC3)CC2)nc1.C0c1cc(CCc2cc([\textbf{**N:30**}]) [n]n2) cc(OC) c1. [C] [A1] ([C]) [C]
```

## Step 5: Predict bond pairs

Based on the above atomic mapping numbers, if the bonds to be formed by the highlighted atoms are expressed in the form of ([atom mapping number], [atom mapping number], [bond type]), the following bonds will be formed.

(3, 30, 'single')

## Step 6: SMILES change prediction

If the above bond pairs are applied to the precursor, without considering the atomic rules or valence rules, it can be expressed as

```
COC(=O) (c1cnc(N2CCN(C3CC3)CC2)nc1)Nc1cc(CCc2cc(OC)cc(OC)c2)n[nH]1.[CH3][A1]([CH3])[CH3].
```

If this is canonicalized, it becomes

```
C0c1cc(CCc2cc(NC(=O)(OC)c3cnc(N4CCN(C5CC5)CC4)nc3)[nH]n2)cc(OC)c1.[CH3][A1]([CH3])[CH3].
```

From the canonicalized SMILES, the product is formed by modifying the SMILES string to remove the reagent and adjust the bonding around the nitrogen and carbonyl group to reflect the acetylation.

The product will involve an amide linkage formed between the nitrogen and the carbonyl carbon.

If we represent the changes from canonicalized SMILES as <DEL> for deleted parts, <ADD> for added parts, and <REP\_OLD> and <REP\_NEW> for modified parts, we get the following:

```
C0c1cc(CCc2cc(NC(=O<DEL>)(OC</DEL>)c3cnc(N4CCN(C5CC5)CC4)nc3)[nH]n2)cc(OC)c1<DEL>.[CH3][A1]([CH3])[CH3]</DEL>
```

# Reflection

## Validity check: Step 4

The prediction of the reaction-participating atoms is correct. The atoms marked (C:3) and (N:30) are correctly identified as the carbonyl carbon and amine nitrogen, which are key participants in the acetylation reaction.

## Validity check: Step 5

The prediction of the bonds between the reaction-participating atoms is correct. The bond between atom 3 (carbonyl carbon) and atom 30 (amine nitrogen) is a single bond, which is consistent with the formation of an amide linkage in an acetylation reaction.

## Validity check: Step 6

The canonical SMILES is correct. The canonicalization process properly rearranges the molecule to follow IUPAC nomenclature rules.

The SMILES transformations are correct as they reflect the removal of the reagent and the formation of an amide linkage, accurately representing the acetylation product.

## Answer

<ANSWER>

```
C0c1cc(CCc2cc(NC(=O)c3cnc(N4CCN(C5CC5)CC4)nc3)[nH]n2)cc(OC)c1
```

</ANSWER>

## ReactionReasoner (Stage 3) - Retrosynthesis

### Input Query

With the given product FC(F)(F)Cn1c(N2CCCNCC2)nc2ccccc21, suggest some likely reactants that were used in its synthesis.

### Rationale

#### ## Step 1: Understanding molecular roles

The molecule provided, FC(F)(F)Cn1c(N2CCCNCC2)nc2ccccc21, is the product, and the task is to identify likely reactants that could have been used in its synthesis.

#### ## Step 2: Analysis of main functional groups

The product contains several key functional groups. The trifluoromethyl group ( $\text{CF}_3$ ) is attached to a carbon that is part of a nitrogen-containing ring system. Additionally, there is a pyrrolidine-like ring (N2CCCNCC2) connected to a benzene ring via a nitrogen atom. The presence of these groups suggests that the molecule may have been synthesized through reactions involving nucleophilic substitution, ring formation, or functional group modification.

#### ## Step 3: Predict possible reaction mechanisms

Based on the explanation provided, the synthesis likely involves the removal of a tert-butoxycarbonyl (Boc) group from an N-Boc protected amine. This process typically involves protonation of the carbonyl oxygen, followed by cleavage of the Boc group and subsequent decarboxylation to yield the free amine. The mechanism also suggests that the molecule may have been synthesized using acid-catalyzed or base-catalyzed conditions, possibly with scavengers or solvents that facilitate the elimination of volatile byproducts like isobutylene.

#### ## Step 4: Atom mapping of product

If we express each atom of the product in atom-mapped SMILES format by adding the format "[atom index]", it is as follows.

```
[F:1][C:2]([F:3])([F:4])[C:5][n:6]1[c:7]([N:8]2[C:9][C:10][C:11][N:12][C:13][C:14]2)[n:15][c:16]2[c:17][c:18][c:19][c:20][c:21]12
```

#### ## Step 5: Strategic bond disconnections

In this case, it does not appear that a product is formed by combining the reactant and reagent through a direct bond formation. The molecule likely underwent a transformation involving the removal of a protecting group rather than a simple coupling reaction.

#### ## Step 6: Generate synthons

Since there is no expected bond disconnection, synthon can be said to be equivalent to product.

1. FC(F)(F)Cn1c(N2CCCNCC2)nc2ccccc21

#### ## Step 7: Select synthetic equivalents

The above synthon most likely originated from the following molecule:

1. CC(C)(C)OC(=O)N1CCCN(c2nc3ccccc3n2CC(F)(F)F)CC1

### Answer

```
<ANSWER>  
CC(C)(C)OC(=O)N1CCCN(c2nc3ccccc3n2CC(F)(F)F)CC1  
</ANSWER>
```

## ReactionReasoner (Stage 3) - Reagent Prediction

### Input Query

Please provide possible reagents based on the following chemical reaction CCCCBr.C1CSCSC1>>C0CCC1SCCCS1.

### Rationale

#### ## Step 1: Understanding molecular roles

The reactant is CC(C)=O.CCOC[C@H]1Cc2ccccc2CN1Cc1c(CC)nm(-c2c(C)cc(Br)cc2C)c1CC and the product is CCOC[C@H]1Cc2ccccc2CN1Cc1c(CC)nm(-c2c(C)cc(C(C)(C)O)cc2C)c1CC.

#### ## Step 2: Analysis of main functional groups

The reactant contains a carbonyl group (C=O), a bromine atom (Br), and a complex structure with a tert-butyl group (C(C)(C)O). The product contains a tert-butyl group (C(C)(C)O), indicating that the bromine atom has been replaced by this group. The overall transformation suggests a substitution reaction where the bromine is replaced by a tert-butyl group.

#### ## Step 3: Predict possible reaction mechanisms

The reaction likely involves a nucleophilic substitution mechanism, where the bromine atom is replaced by a tert-butyl group. This type of reaction is often facilitated by a strong nucleophile and a good leaving group, such as bromide. The presence of a carbonyl group in the reactant may also play a role in stabilizing the transition state or activating the substrate.

#### ## Step 4: $\Delta$ analysis

The functional group/substructure change ( $\Delta$ ) between reactant and product involves the removal of bromine (\_Br\_) and the addition of a tert-butyl group (\_c(C(C)(C)O)\_). This indicates that the bromine atom has been replaced by a tert-butyl group, which is likely achieved through an SN2 or SN1 mechanism depending on the reactivity of the substrate.

#### ## Step 5: Derivation of required functions

The reaction requires a strong nucleophile to attack the electrophilic carbon bearing the bromine atom and a good leaving group (bromide). Additionally, the reaction may require a base to deprotonate or activate the substrate, and possibly an acid to protonate or stabilize intermediates. The presence of a carbonyl group may also influence the reaction pathway by stabilizing transition states or acting as an electrophilic center.

#### ## Step 6: Generation of candidate reagents

The candidate reagents are as follows:

1. [Li]CCCC
2. CCCC[N+](CCCC)(CCCC)CCCC
3. CCOC(=O)Cl

#### ## Step 7: Selection of reagents

The reagent [Li]CCCC is the most suitable reagent for the reaction, as it can act as a strong base and nucleophile, facilitating the substitution of bromine with a tert-butyl group. It also provides the necessary conditions for forming new carbon-carbon bonds through its carbanionic character.

### Answer

```
<ANSWER>
[Li]CCCC
</ANSWER>
```