ChemActor: Enhancing Automated Extraction of Chemical Synthesis Actions with LLM-Generated Data

Yu Zhang[1†], Ruijie Yu[1†], Jidong Tian[1], Feng Zhu[1], Jiapeng Liu[2], Xiaokang Yang[1], Yaohui Jin[1*], Yanyan Xu[1*],
[1] AI Institute, Shanghai Jiao Tong University, China
[2] X-Imaging Intelligent Technology (Shanghai) Co. LTD., China
{cynthiazhang, el_iauk, frank92, fzchem, xkyang, jinyh, yanyanxu}@sjtu.edu.cn
liujiapeng@x-imaging.com

Abstract

With the increasing interest in robotic synthesis in the context of organic chemistry, the automated extraction of chemical procedures from literature is critical. However, this task remains challenging due to the inherent ambiguity of chemical language and the high cost of human annotation required for developing reliable computer-aided extraction protocols. Here, we present **ChemActor**, a fully fine-tuned large language model (LLM), as a chemical executor to convert between unstructured experimental procedures and structured action sequences. We propose a sequential LLM-generated data framework to address the challenges of insufficient and low-quality annotated data. This framework integrates a data selection module that selects data based on distribution divergence, with a general-purpose LLM, to generate machine-executable actions from a single molecule input. Additionally, we introduce a novel multi-round LLMs circle review metric, which reflects the model's advanced understanding of chemical experimental procedures. Extensive experiments on reaction-to-description (R2D) and description-to-action (D2A) tasks demonstrate that ChemActor, augmented by LLM-generated data, achieves state-of-the-art performance, outperforming the baseline model by 10%. The code is available at: https://github.com/Zhanghahah/ChemActor.

1 Introduction

In the field of organic synthesis, robotic systems for chemical synthesis have grown in popularity. The availability of structured chemical data becomes more important owing to its potential to accelerate the discovery of transformative molecules with minimal human intervention (Godfrey et al., 2013; Peplow, 2014; Guo et al., 2021; Canty and Abolhasani, 2024). The execution of chemical synthesis necessitates a thorough extraction of the precise sequence of procedures, including addition, stirring, and concentration, as well as the optimal parameters for these actions, such as temperature and atmosphere (Walker et al., 2019; Fan et al., 2024), etc. However, chemical experimental procedures consist of various writing styles in natural language (Zhang et al., 2024), and extraction of experimental procedures from literature requires manual revisions (Mehr et al., 2020). Although chemists have focused on designing reliable tools through natural language processing (Guo et al., 2021), quite often these solutions rely on identifying and utilizing rules specific to each data item at the cost of a substantial human effort (Davies, 2019; Vaucher et al., 2020, 2021; Zeng et al., 2023).

Nowadays, the emergence of generative pre-trained transformer-based large language models (LLMs), typified by GPT-4, has sparked significant interest in the field of AI for chemistry (Baum et al., 2021; Boiko et al., 2023; M. Bran et al., 2024). Pre-trained on a vast corpus of chemical reactions literature, LLMs are endowed with fundamental chemical knowledge through text-to-text generation (Achiam et al., 2023). However, the sparsity of chemical data and the lack of high-quality annotations still limit the development of applying LLMs to autonomous chemical experiments (Coley et al., 2019). Due to the ambiguity and diverse writing styles in the chemical literature (Zhang et al., 2024), extraction of experimental procedures from literature by LLMs often requires manual revisions (Zhong et al., 2023). This presents a significant challenge for converting chemical descriptions into machine-readable annotated experimental actions (Ji et al., 2023; Zhang et al., 2023). Overall, *the design of an automated conversion from unstructured chemical descriptions into structured ones for LLM-based autonomous chemical experiments is a desirable and needed technology.*

Recently, the technology of LLM-generated data

[†] Equal contribution.
[*] Corresponding authors.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 24291–24314

has been very popular as it can enhance the model's own performance by generating valuable training data (Li et al., 2024). Several studies, particularly in vision understanding and nature language fields (Tremblay et al., 2018), have highlighted that generated data augmentation markedly improves the performance of foundational models (Gao et al., 2023; Trinh et al., 2024; Li et al., 2024). Considering that the standard of the recording and subsequent reporting of the synthesis of new reactions varies greatly, we investigate a naturally arisen question: *can scaling up high-quality LLM-generated data enhance the performance of the LLMs for generating experimental action sequences tasks?* To answer this, we propose a novel LLM-generated data framework into ChemActor for generating machine-executable actions starting from a molecule. The reason why LLM-generated data can be a potential solution is that (i) via joint learning with generated data and real data, LLMs learn relationships between chemical synthetic descriptions and action sequences, thereby acquiring chemical knowledge akin to the learning process of chemists; (ii) the data selection module generates more valuable data that compensates for the uneven distribution of chemical space, thereby enhancing the representation of reaction description. The contributions of this work can be summarized as follows:

1. We propose a 7B-scale fine-tuned LLM, a.k.a. ChemActor, as a chemical executor to convert unstructured human-readable descriptions to structured machine-executable actions.

2. We design a sequential LLM-generated data framework that integrates a data selection module–based on distribution divergence–with a general-purpose LLM to generate actions from a single molecule input.

3. We design a multi-round LLMs circle review metric, using interactive debate prompting to prove ChemActor's advanced semantic understanding of chemical experimental knowledge.

2 Related Work

Automated Extraction of Chemical Synthesis Actions Chemical experimental actions extraction aims at extracting structural information, which is prepared for an automated synthesis platform. According to different sources, this task includes two categories of sub-tasks: description-to-action extraction (D2A) and reaction-to-action extraction (R2A). D2A is a natural language information extraction task that converts unstructured experimental descriptions to structured synthetic steps. Vaucher et al. (Vaucher et al., 2020) first propose the D2A task with pre-defined synthesis actions and design a pre-trained Transformer-based model to solve it. Zeng et al. (Zeng et al., 2023) introduce a comprehensive schema for D2A, and also design a new dataset, CHEMTRANS. Meanwhile, they propose knowledge-enhanced methods for fine-tuned models (T5) and large language models (GPT-3.5), respectively. Results show that fine-tuned models with chemical knowledge exhibit better performance on D2A. R2A is to predict chemical experimental actions directly from chemical equations. Vaucher et al. (Vaucher et al., 2021) propose the first R2A task that converts reaction SMILES into actions. Meanwhile, they also design a series of Transformer-based models to solve R2A. Liu et al. (Liu et al., 2024) construct a new R2A dataset, OPENEXP, and propose a novel incrementally pre-trained language model, ReactXT, to achieve such a task. Compared with other general or scientific language models, ReactXT shows a significant improvement on OPENEXP. All these studies have provided effective datasets and baseline models, but D2A and R2A remain to be addressed. In this work, we introduce a novel LLM-generated augmentation framework that can effectively enhance the pre-trained models' ability for the extraction of chemical experimental actions.

LLM-Generated Data In recent years, data generation technologies have transformed data science by mimicking real-world data, addressing challenges of data scarcity (Wang et al., 2024). Several studies, particularly in mathematical and medical fields, have highlighted that realistic data augmentation markedly improves the performance of machine learning models (Trinh et al., 2024; Gao et al., 2023). Recently, LLMs have exhibited the ability to generate highly reliable data (Li et al., 2023, 2024). These methods are widely used in the field of natural language processing (Chen et al., 2023). To be specific, Whitehouse et al. (Whitehouse et al., 2023) utilize pre-trained LLM, such as GPT-4, to generate more data for low-resource machine translation. Cai et al. (Cai et al., 2023) propose a generative framework that uses LLaMA to resolve the data imbalance problem. Yuan et

al. (Yuan et al., 2023) take advantage of LLM to enhance the compatibility of clinical trial descriptions. These studies provide a foundation for LLM-generated data methods, but applying them to the scientific domain is still challenging. This is because pre-trained LLMs do not necessarily possess the corresponding domain knowledge.

3 Methods

3.1 Problem Setup

Given an unstructured reaction description text $\mathbf{d}$ from the chemical literature, our objective is to extract all action sequences $\mathbf{a}$.

3.2 The Framework of LLM-generated Data

Here, we first introduce a novel LLM-generated data framework for generating machine-executable actions starting from a single molecule. The overview is shown in Figure 1. In Figure 1A, we start with a sampled chemical reaction $\mathbf{r}$, along with a detailed description and its experimental actions. Firstly, we design a comprehensive data template to construct pair-wised instructed Q&A datasets for supervised fine-tuning based on the original LLaMA-2 foundational model. The detailed data construction is illustrated in the Appendix A.2 and Figure 4. Subsequently, we obtain a reaction-to-description model, a.k.a. **R2D**, and a description-to-action model, a.k.a. **D2A**, repspectively. **R2D** converts $\mathbf{r}$ into reaction descriptions $\mathbf{d} = \{d_1, d_2, \cdots, d_n\}$. The objective of **R2D** is shown in Equation 1, where θ is the trainable parameters, $d_{<i}$ represents $\{d_0, \cdots, d_{i-1}\}$ tokens, and d_0 is the starting token. **D2A** model focuses on converting reaction descriptions into a series of corresponding experimental actions $\mathbf{a} = \{a_1, \cdots, a_m\}$, such as adding, refluxing, etc. The objective of **D2A** are provided in Equation 2, where ψ is the trainable parameters, $a_{<i}$ represents $\{a_0, \cdots, a_{i-1}\}$, and a_0 is the starting token.

In Figure 1B, we present the LLM-generated data framework, which begins with a target molecule. First, we employ an efficient single-step retrosynthesis model from previous work (Zeng et al., 2024) to generate the necessary reactants. To ensure the validity of the generated reaction, we further utilize a forward prediction model (Schwaller et al., 2020) for self-correctness. Next, we leverage a general-purpose LLM, GPT-4o, to generate reaction descriptions using an in-context learning strategy, enhancing generation accuracy through context augmentation with relevant examples. Subsequently, the Paragraph2Actions model (Vaucher et al., 2020) is applied to predict procedural actions, and general-purpose LLM, GPT-4o, is utilized for formatting refinement. During training, we augment the existing datasets with LLM-generated data to enhance ChemActor's capability in translating reaction descriptions into procedural actions.

Algorithm 1 Data Selection Module

Input: triplet set $O = [(r_1^*, d_1^*, a_1^*), (r_2^*, d_2^*, a_2^*), ...(r_k^*, d_k^*, a_k^*)]$ of LLM-generated data, the encoder of R2D model f_θ^{enc}, the encoder of D2A model f_ψ^{enc}, distribution difference δ, the original input data points of reaction descriptions $\mathbf{d}$, the LLM-generated data points $\mathbf{d}^*$, threshold of distribution difference τ, total number of LLM-generated data n.
Output: selected LLM-generated data sets A.
Initialization: $A = \emptyset$
Initialization: $i = 0$
while $i * k < n$ **do**
  $\mathbf{e} = [e_1, e_2, \cdots, e_k] = f_\psi^{enc}(\mathbf{d})$
  $\mathbf{e}^* = [e_1^*, e_2^*, \cdots, e_k^*] = f_\psi^{enc}(\mathbf{d}^*)$
  $\delta = |KL(\mathbf{e}) - KL(\mathbf{e}^*)|$
  if $\delta \geq \tau$ **then**
    A.append(O)
    $i = i + 1$
  end if
end while
return A

3.3 Data Selection Module

We further design the data selection module to enhance the model performance, as outlined in Algorithm 1. First, we extract the embeddings of original datasets by **D2A** model mentioned in Figure 1A, denoted as $\mathbf{e} = [e_1, e_2, \cdots, e_k]$. We randomly select the triplet set k-consisting of data (reaction formulation, description, actions) from LLM-generated data sets, and extract its embeddings by **D2A**, denoted as $\mathbf{e}^* = [e_1^*, e_2^*, \cdots, e_k^*]$. Next, we project the embeddings of both real and LLM-generated data into a two-dimensional representation space using the uniform manifold approximation and projection (UMAP) algorithm. To ensure the quality of the LLM-generated data, we employ a selection criterion based on the Kullback-Leibler (KL) divergence between the distributions of LLM-

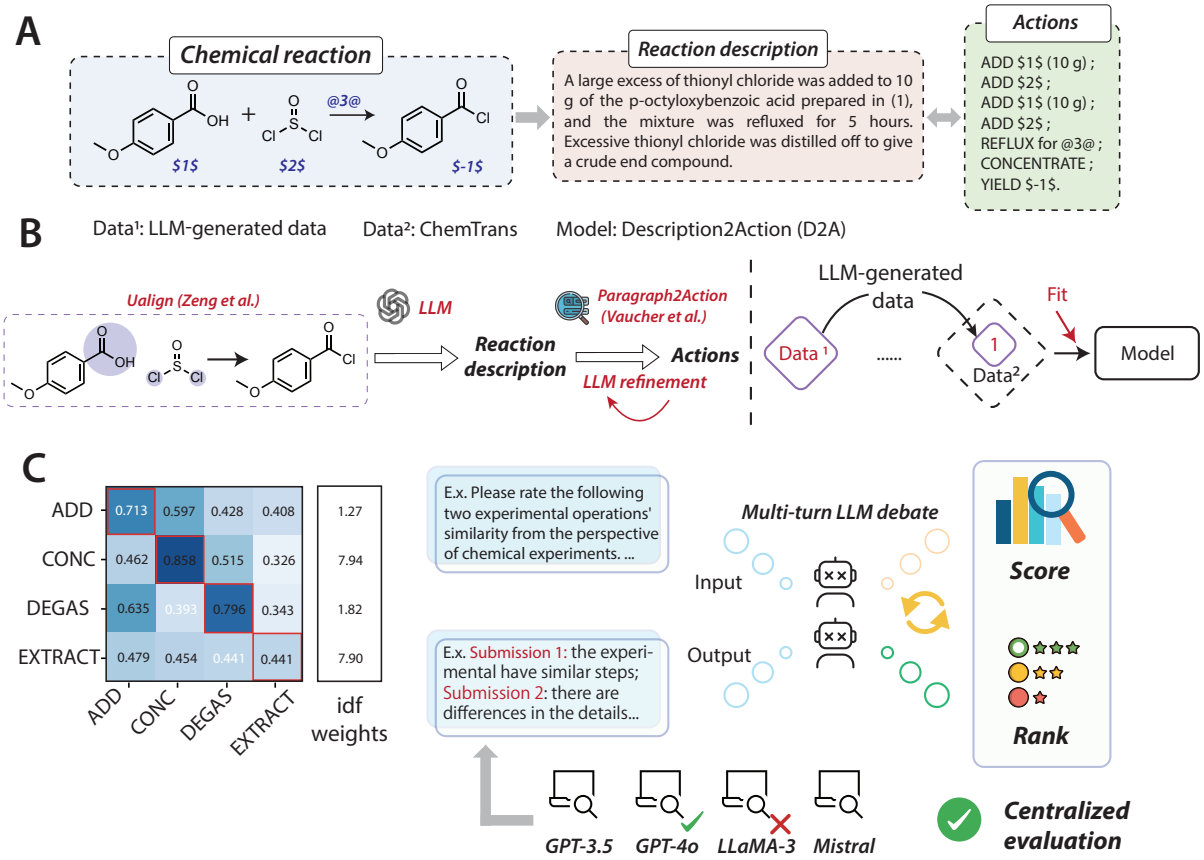

Figure 1: The overview of ChemActor. **(A)** Task definition; **(B)** Illustration of the framework of LLM-generated data; **(C)** BERTScore and multi-round LLMs circle review metrics for evalution.

generated and real data in this low-dimensional space. Specifically, we compute the KL divergence for the distributions of LLM-generated data and original data, denoted as $KL(e^*)$, and $KL(e)$, respectively. We then define a threshold τ for the difference $\delta = |KL(\mathbf{e}) - KL(\mathbf{e}^*)|$. The selection process operates as follows: if $\tau > \delta$, the corresponding triplet sets from the LLM-generated dataset are retained; otherwise, the data are resampled. This iterative procedure continues until a predefined number n of qualified LLM-generated samples is obtained.

$$\max_{\theta} \log P_\theta(\mathbf{d}|\mathbf{r}) = \sum_{i=1}^{n} \log P_\theta(d_i|\mathbf{r}, d_{<i}) \quad (1)$$

$$\max_{\psi} \log P_\psi(\mathbf{a}|\mathbf{d}) = \sum_{i=1}^{m} \log P_\psi(a_i|\mathbf{d}, a_{<i}) \quad (2)$$

3.4 Semantic Evaluation for Actions

To better evaluate ChemActor, we introduce two additional metrics: BERTScore (Zhang et al., 2019) and LLMs circle review, as shown in Figure 1C. The detailed introduction of the metric of BERTScore can be found in the Appendix B.3. Further, LLM-based peer review presents a promising approach for evaluating generated outputs (Chu et al., 2024; Zhao et al., 2024). Inspired by (Chern et al., 2024), we design an LLMs circle review metric based on a multi-round debating strategy to better evaluate ChemActor. Specifically, we implement an interactive evaluation framework employing four LLMs (GPT-3.5-turbo, GPT-4, LLaMA-3, and Mistral) to independently assess prediction-annotation semantic similarity. Through iterative prompting, these LLMs transparently share scores and rationales, then engage in structured debate. The debate continues until a predetermined number of rounds is reached, after which the final average score is adopted. The prompts for multi-round debating are shown in Figure 9.

4 Experiment

4.1 Data

We evaluate the effectiveness of our method using two datasets, CHEMTRANS and OPENEXP. CHEMTRANS (Zeng et al., 2023) consists of 3,950 description–action pairs with a detailed schema of instructions. OPENEXP proposed by (Liu et al., 2024) are derived from USPTO-Applications (Lowe, 1976) and the Open Reaction Database (Kearnes et al., 2021). Further, Appendix A.1 provides the data statistics and distribution of these datasets.

4.2 Experimental Setup

Firstly, we organize training data as instruction prompt datasets for supervised fine-tuning. For the training strategy, we propose an alternating data mixing paradigm. This approach employs an iterative training scheme where the model is first exposed to a batch of real data, followed by a batch of LLM-generated data. Fully fine-tuning LLaMA-2-7B for 8 epochs is completed in approximately 12 hours using 8 $\times$ NVIDIA A800 GPUs. We set τ as 0.7 in data selection module, the batch size to 12, and the default learning rate to 9.65e-6. The further detailed training setting can be seen in Appendix B.2.

4.3 Performance Comparison

We evaluate baseline and ChemActor methods on the two benchmark datasets, CHEMTRANS and OPENEXP, respectively. Descriptions of metrics applied for evaluation and introduction of all baseline methods are introduced in Appendix B.3.

Evaluation on CHEMTRANS. We compare ChemActor with various existing methods, as presented in Table 1. Additionally, we evaluate the performance of general-purpose model GPT-3.5 and GPT-3.5-chat API for comparison, which can be seen in the Appendix Figure 12. Following previous work (Zeng et al., 2023), we assess performance using the Sequence Matching (SM) and ExactMatch (EM) metrics.

In Table 1, **ChemActor, w/generated** refers to ChemActor trained with an additional 50,000 LLM-generated data points, while **ChemActor, w/o generated** represents ChemActor trained without LLM-generated data. We evaluate the performance of our ChemActor for both D2A and A2D tasks. From the results, we conclude that (i) the GPT-3.5 series, even with few-shot instructions, struggles with translating between human-readable synthetic descriptions and machine-executable actions, leading to low similarity metrics; (ii) ChemActor wo/generated outperforms both T5-ChemTrans and Paragraph2Actions(Vaucher et al., 2021), achieving 68.37% BLEU-4 and 31.9% EM scores for the D2A task, representing improvements of 41.18% and 31%, respectively. Furthermore, by incorporating LLM-generated data into the training process, ChemActor, w/generated achieves a 9% improvement in BLEU-2 and a 4.5% increase in EM compared to ChemActor, w/o generated for generating action sequences. The improvement can be attributed to the incorporation of LLM-generated data, which enriches the model's understanding of the relationship between reaction descriptions and corresponding experimental actions, thereby enhancing its ability to generate accurate action sequences. Furthermore, we conduct a type-matching experiment to evaluate the model's ability to predict action types and their corresponding necessary components, as shown in Appendix C.2.

Evaluation on OPENEXP. For OPENEXP, we compare our ChemActor with the state-of-the-art scientific LLMs and comparative baseline methods. Notably, all these scientific LLMs are pre-trained on molecules using 1D or 2D SMILES representations rather than reaction descriptions. They take the reaction equation as input and generate experimental action sequences. In contrast, the baselines proposed by (Vaucher et al., 2021) and (Zeng et al., 2023) use reaction descriptions as input and generate structural experimental actions.

Following (Vaucher et al., 2021; Liu et al., 2024), we employ the additional metrics for performance evaluation: Validity and Levenshtein score (LEV). The performance of our ChemActor on OPENEXP datasets is shown in Table 2. The results reveal that ChemActor, wo/generated consistently outperforms baseline methods across all metrics. Specifically, it surpasses ReactXT by 34.2% for BLEU-2 and 68.5% for 75%LEV, demonstrating ChemActor's effectiveness for text-based reaction understanding. Furthermore, baseline models that use reaction equations as input consistently underperform compared to those that leverage reaction descriptions. This is because reaction descriptions contain detailed contextual and procedural information about experimental conditions, which are essential for accurate predictions. In contrast, reaction equations primarily represent reactants and products, lacking explicit details about reaction conditions, solvents, catalysts, and procedural steps

Task	Description-to-Action (D2A)					Action-to-Description (A2D)			
Model	**SM-A**	**SM-O**	**BLEU-2**	**BLEU-4**	**EM**	**Distinct-4**	**ROUGE-4**	**BLEU-2**	**BLEU-4**
Paragraph2Actions	21.57	57.91	44.88	27.19	0	8.308	0.517	5.210	0.365
Paragraph2Actions+	22.43	58.45	44.97	27.97	0	18.36	1.225	8.168	0.933
GPT-3.5	0.441	4.471	7.520	0.931	0	67.99	5.261	10.83	2.920
GPT-3.5, 3-shot	37.53	66.96	59.69	44.91	4.94	56.51	13.39	20.41	8.816
GPT-3.5, 3-shot*	45.11	70.45	62.84	50.16	6.71	59.26	15.06	23.19	10.69
GPT-3.5-chat	2.708	35.49	14.17	2.72	0	**74.62**	3.016	6.423	1.982
GPT-3.5-chat, 3-shot	25.75	57.99	49.25	31.92	0.72	70.70	8.619	15.73	5.486
GPT-3.5-chat, 3-shot*	34.88	62.28	55.57	40.45	3.25	69.59	10.33	17.96	6.913
T5-ChemTrans, base	65.59	83.78	59.24	43.46	18.31	56.67	20.06	27.61	15.01
T5-ChemTrans, large	67.12	85.41	75.89	67.33	22.36	57.17	21.82	29.54	16.55
ChemActor, wo/generated	68.29	85.07	75.63	68.37	31.90	56.71	19.18	29.87	18.43
ChemActor, w/generated	**76.88**	**89.01**	**84.74**	**76.93**	**36.40**	64.19	**25.79**	**31.76**	**34.27**

Table 1: Experimental Results of ChemActor on CHEMTRANS dataset. The best performance is in **bold.**

Method	Validity	BLEU-2	BLEU-4	100%LEV	90%LEV	75%LEV	50%LEV	ROUGE-1	ROUGE-2	ROUGE-L
With molecule pre-training, reaction equation → experimental actions										
TextChemT5$_{220M}$	99.3	54.1	40.6	0.4	4.6	13.7	61.2	61.5	40.3	56.4
MolT5-Large$_{780M}$	99.6	54.5	41.0	0.6	6.6	16.6	63.7	62.5	40.9	57.2
Galactica$_{1.3B}$	99.9	53.5	39.5	0.4	5.7	13.4	60.5	60.9	38.6	55.2
MolCA, Galac$_{1.3B}$	99.9	54.9	41.5	1.0	9.2	18.9	65.3	62.5	40.4	57.0
ReactXT, Galac$_{1.3B}$	**100.0**	57.4	44.0	1.0	**9.5**	22.6	70.2	64.4	42.7	58.9
ChemActor*, wo/g	**100.0**	**86.7**	**85.4**	**2.0**	7.0	**78.5**	**99.0**	**86.7**	**85.3**	**86.1**
Without molecule pre-training, reaction description → experimental actions										
Paragraph2Actions	63.2	34.5	19.1	0.0	0.0	0.0	13.6	46.6	18.1	36.4
Paragraph2Actions+	76.0	45.0	30.7	0.6	6.5	13.0	38.4	55.7	29.2	47.0
T5-ChemTrans, base	99.5	92.2	90.4	38.7	**69.6**	**91.6**	99.0	**95.1**	**93.6**	**94.6**
ChemActor, wo/g	**99.8**	**92.9**	**91.0**	**40.1**	69.4	91.4	**99.2**	92.5	**93.6**	94.5

Table 2: Experimental Results of ChemActor on OPENEXP dataset. The best performance is in **bold.**

that significantly influence outcomes.

4.4 Multi-Round Circle Review

As previously discussed, existing metrics are insufficient for evaluating the semantic correctness of predicted outputs. Thus, we propose BERTScore and LLMs circle review for further evaluation. The details of the proposed metric are illustrated in Appendix Section B.3. We also perform a human evaluation experiment to substantiate the effectiveness of our proposed metric. Specifically, we invite six Ph.D. and M.S. students from the department of chemistry to assess the prediction results on the CHEMTRANS test set. Each prediction is required to be scored on a scale from 0 to 10. The entire test set is divided into six groups, and each student is responsible for evaluating the cases in their assigned group. Next, we collect all scores, average them, and scale them to a range of 0 to 1 to obtain the final evaluations.

In Table 3, evaluation results reveal the insights: 1) within the LLM circle review metric, all general-purpose LLMs consistently achieve the same ranking across all compared methods: ChemActor>T5-ChemTrans>Mistral>T5-Base. The superior performance of T5-ChemTrans over Mistral can be attributed to its pre-training on literature data encompassing experimental actions, which significantly enhances its understanding of chemical knowledge. In contrast, T5-Base and Mistral-7B are fine-tuned using LoRA from general foundation models, making it challenging for them to perform well with the limited CHEMTRANS training datasets. 2) Human evaluation scores tend to be conservative, as they are all lower than the scores from the LLM circle review. Importantly, the ranking trend of the human evaluations is consistent with the results from the LLM circle review. We hypothesize that the reason Mistral performs better than T5-ChemTrans in this context is that Mistral's generated results

Model	GPT-3.5	GPT-4o	LLaMA-3	Mistral	Human review
Mistral	0.74	0.76	0.78	0.77	0.61
T5-Base	0.68	0.70	0.75	0.70	0.50
T5-ChemTrans	0.76	0.78	**0.81**	**0.78**	0.58
ChemActor (wo/g)	**0.77**	**0.79**	**0.81**	**0.78**	**0.66**

Table 3: Results for multi-round LLMs circle reviews and human evaluations on CHEMTRANS dataset. The best performance is in **bold.**

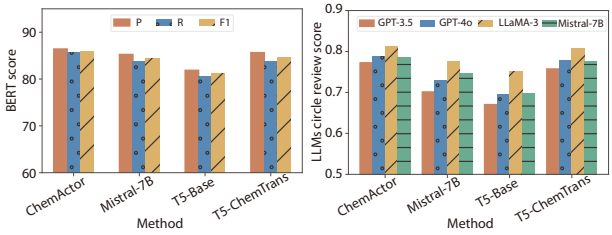

Figure 2: Results for multi-round LLMs circle reviews.

are more closely aligned with human preference since it is pre-trained on a massive, unlabeled text corpus.

In addition, we conduct multi-round LLM debate experiments and calculate the scores of methods to verify the effectiveness of our proposed method. Detailed prompt examples can be seen in Appendix Figure 9. In Figure 2, it shows that ChemActor outperforms other baselines on both BERTScore and LLMs circle review scores. Both ChemActor and T5-ChemTrans achieve BERTScore F1 scores exceeding 0.85, with ChemActor obtaining the highest score, highlighting its superior ability to capture the semantics of chemical procedures. Although circle review scores generally align with the BERTScore, there are variations between models. Specifically, GPT-4o and LLaMA-3 rate ChemActor the highest, whereas GPT-3.5 and Mistral-7B favor T5-ChemTrans.

4.5 Ablation Study

LLMs show increasingly advanced emergent capabilities and are being incorporated across various domains. Understanding the relationship between the LLM-generated data and its abilities holds significant importance. Thus, we design several ablation studies to verify the effectiveness of our proposed framework.

4.5.1 Data Selection Module

The data selection module employs a threshold τ to filter LLM-generated data based on distribution divergence. Here we discuss the effect of hyperepameter τ for model performance. The results in Table 4 demonstrate that ChemActor's performance is significantly influenced by both the divergence threshold τ and the dimension reduction method. Increasing τ from 0.5 to 0.8 consistently improves all evaluation metrics. However, this improvement comes at a computational cost, as processing time increases dramatically from 141s to 1031s. The comparison between the UMAP and the t-SNE reveals that UMAP outperforms t-SNE across all metrics. It is indicated that as τ increases, the spatial distribution of sampled points becomes more uniform, which helps mitigate the issue of data skewness for training. Further, the results demonstrate that $\tau = 0.7$ with UMAP projection represents an optimal balance between model performance and computational efficiency.

4.5.2 Training Size Proportions

Before investigating the effect of LLM-generated data on performance, we examine whether increasing the training data can improve the performance of ChemActor. To this end, we evaluate ChemActor with varying proportions of training data on the CHEMTRANS and OPENEXP datasets. We also introduce T5-Base (Raffel et al., 2020), LLaMA-3, and Mistral-7B (Jiang et al., 2023) for comparison. Note that the T5-Base refers that we use the T5 model proposed by (Raffel et al., 2020) for fine-tuning. The entire five scenarios for evaluation are illustrated in Appendix C.1. We list four training set proportions, 30%, 50%, 70%, and 100% in Table 5 and Table 6. For the CHEMTRANS dataset, it is notable that Mistral-7B consistently outperforms T5-Base and LLaMA-3-7B, achieving a Modified BLEU score of 73.08 at 100% data. However, ChemActor, wo/g demonstrates a significant performance boost, surpassing all other methods with a score of 80.17 at 100% data. For the OPENEXP datasets, we notice that the performance of the LLaMA-3-7B and Mistral models is significantly lower than that of the T5-Base. We hypothesize that this phenomenon can be attributed to two aspects: 1) from the data aspect, the diver-

Model	τ	Projection	SM-A	SM-O	BLEU-2	BLEU-4	EM	Cost Time
ChemActor, w/g	0.5	UMAP	67.13	83.31	74.03	66.87	30.94	141s
	0.6	UMAP	68.29	84.97	76.17	68.74	32.41	385s
	0.7	UMAP	68.92	85.51	76.01	68.71	34.14	427s
	0.8	UMAP	68.93	85.53	76.07	69.12	34.33	1031s
ChemActor, w/g	0.7	t-SNE	66.86	83.09	73.79	66.81	29.89	517s

Table 4: Performance evaluation of ChemActor with varying amounts of τ on the CHEMTRANS dataset. Cost time refers to the time spent searching for LLM-generated data points that fulfill the requirements.

sity of actions in OPENEXP is greater compared to the CHEMTRANS dataset. When sampling 30% of OPENEXP for training, there is a risk that certain actions may not be adequately sampled. This sampling issue poses a challenge for LLaMA-3 and Mistral, as these general-purpose LLMs struggle to perform optimally on skewed data distributions. Therefore, we may need more comprehensive data for supervised fine-tuning to eliminate the domain gap between general language tasks and domain-specific chemical knowledge. 2) From the metric aspect, we conduct a human evaluation experiment to evaluate the prediction results of all methods. The results demonstrate that Mistral achieves a score of 0.61, while T5-Base scored 0.50. These scores underscore Mistral's superior ability to align with human evaluators' expectations compared to T5-Base. Therefore, to some extent, Mistral doesn't exactly perform less well than T5-base.

4.5.3 Performance of LLM-Generated Data

With the ascent of LLMs, natural language processing has seen improvements, including LLM-based data augmentation. However, an oversight arises from the random generation of augmented data by LLMs, suggesting that not all data may have equal training value, potentially impeding generative performance. We examine that ***"how much will LLM-generated data benefit performance?"*** To answer this, we incorporate varying proportions of LLM-generated data into the training process and evaluate the model's performance on action-generation tasks.

Table 7 presents the performance of ChemActor with varying amounts of LLM-generated data on the CHEMTRANS dataset. Method criterion refers to strategies for data augmentation. To assess the impact of our proposed data selection module, as illustrated in Figure 1B, we also randomly sample data from the OPENEXP dataset to construct training sets for training ChemActor.

The results reveal that ChemActor, when trained with an additional 50,000 LLM-generated samples achieves strong performance, attaining a BLEU-4 score of 76.93 and an EM score of 36.4, reflecting a 14.1% improvement in EM. In contrast, incorporating the same volume of randomly sampled data from OPENEXP leads to only a 2.7% EM improvement. Additionally, when 300 randomly selected samples are added to the training set, the EM score decreases from 31.9 to 30.19, suggesting that indiscriminate data augmentation may introduce noise rather than improve model performance. In summary, our proposed data selection module is designed to identify reactions that differ from those in the original dataset, thereby expanding the representation of chemical reactions. This broader representation enhances the model's ability to perform the D2A task more effectively.

5 Conclusion

In this paper, we present a fine-tuned LLM, a.k.a. ChemActor for automated extraction of chemical synthesis actions. Trained on Q&A instruction datasets with both real and LLM-generated data, ChemActor efficiently converts unstructured reaction descriptions into experimental actions. We introduce a data selection module that generates valuable data to address the uneven distribution of chemical space. Experimental results show that ChemActor achieves competitive results with state-of-the-art models, and our LLM-generated data proves more effective than other augmentation methods.

6 Limitations

Although ChemActor has been trained on extensive chemical reaction data, and we introduce a data selection module to tackle data sparsity issues, predicting actions with very limited observations remains a challenging task. In such cases, ChemActor might generate suboptimal answers.

Metrics	90% LEV				75% LEV				Modified BLEU				Levenshtein similarity			
Training proportions	30%	50%	70%	100%	30%	50%	70%	100%	30%	50%	70%	100%	30%	50%	70%	100%
T5-Base	0.00	0.13	0.38	1.01	0.89	3.42	6.58	13.16	52.37	59.25	64.38	68.94	46.19	51.02	54.33	58.75
LLaMA-3-7B	0.38	0.51	0.51	0.38	4.94	6.71	5.32	7.97	47.89	51.12	49.22	52.92	46.65	48.51	46.86	49.45
Mistral-7B	**3.8**	4.56	4.56	5.44	**23.67**	26.33	28.61	30.51	70.05	70.13	71.51	73.08	62.44	63.83	64.37	65.52
ChemActor, wo/g	2.78	**5.7**	**7.72**	**10.0**	22.53	**29.62**	**33.8**	**39.75**	**72.83**	**77.03**	**79.32**	**80.17**	**63.75**	**67.76**	**69.5**	**70.87**

Table 5: Performance evaluation with varying different training proportions on the CHEMTRANS dataset.

Metrics	90% LEV				75% LEV				Modified BLEU				Levenshtein similarity			
Training proportions	30%	50%	70%	100%	30%	50%	70%	100%	30%	50%	70%	100%	30%	50%	70%	100%
T5-Base	43.22	52.14	56.29	63.02	79.95	84.85	86.77	89.85	82.94	85.39	86.29	88.37	85.13	87.51	88.58	90.41
LLaMA-3-7B	13.83	13.25	13.95	16.08	51.66	50.72	51.95	55.01	57.37	55.87	56.92	60.64	72.53	71.97	72.54	73.98
Mistral-7B	38.56	41.7	43.42	43.98	75.81	77.62	78.7	78.88	78.67	79.97	80.47	80.31	83.16	84.05	84.58	84.63
ChemActor, wo/g	**64.2**	**65.59**	**68.31**	**69.4**	**89.72**	**89.98**	**91.13**	**91.4**	**89.57**	**89.87**	**91.05**	**92.9**	**90.47**	**90.32**	**91.55**	**91.63**

Table 6: Performance evaluation of ChemActor with varying different training proportions on the OPENEXP dataset.

Model	Method criterion	SM-A	SM-O	BLEU-2	BLEU-4	EM
T5-Base	3k wo/generated	41.55	66.26	45.19	30.07	2.15
	add 300 LLM-generated samples	44.38	67.87	47.34	31.91	3.29
	add 600 LLM-generated samples	54.36	75.88	52.78	37.17	8.99
	add 2,100 LLM-generated samples	57.71	77.9	55.08	39.24	10.76
ChemActor	randomly add 300 samples from OPENEXP	67.10±0.03	83.29±0.05	74.17±0.05	66.32±0.06	30.19±0.08
	randomly add 600 samples from OPENEXP	67.11±0.08	83.31±0.05	74.03±0.08	66.31±0.05	30.28±0.05
	randomly add 2,100 samples from OPENEXP	69.81±0.41	84.97±0.5	77.36±0.5	68.89±0.68	31.04±0.32
	randomly add 50,000 samples from OPENEXP	71.08±0.8	85.31±0.8	78.48±0.5	71.93±0.8	32.77±0.8
ChemActor	3k wo/generated	68.29	85.07	75.63	68.37	31.9
	add 300 LLM-generated samples	68.29±0.07	84.97±0.05	76.17±0.03	68.74±0.05	32.41±0.1
	add 600 LLM-generated samples	68.93±0.03	85.53±0.05	76.07±0.03	68.72±0.03	34.33±0.1
	add 2,100 LLM-generated samples	70.88±0.04	86.52±0.06	79.66±0.05	69.67±0.08	35.90±0.2
	add 50,000 LLM-generated samples	**76.88±0.1**	**89.01±0.06**	**84.74±0.1**	**76.93 ±0.2**	**36.4±0.4**

Table 7: Performance evaluation of ChemActor with varying amounts of LLM-generated data on the CHEMTRANS dataset. The best performance is in **bold**.

Furthermore, when applying ChemActor to a chemical automated platform, it's crucial to address potential safety risks from executing LLM-powered recommendations. To cope with this, developers must design comprehensive prompt guidelines that prioritize safety. If deemed dangerous, execution should be halted immediately, ensuring ChemActor's application in automated platforms is secure and effective.

7 Acknowledgments

We thank SJTU AI for Science platform for the computing support. This work was supported by the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), the Shanghai Municipal Science and Technology Explorer Project (24TS1403300), the National Natural Science Foundation of China (62102258), and the Fundamental Research Funds for the Central Universities.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Zachary J Baum, Xiang Yu, Philippe Y Ayala, Yanan Zhao, Steven P Watkins, and Qiongqiong Zhou. 2021. Artificial Intelligence in Chemistry: Current Trends and Future Directions. *Journal of Chemical Information and Modeling*, 61(7):3197–3212.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous Chemical Research with Large Language Models. *Nature*, 624(7992):570–578.

Xunxin Cai, Meng Xiao, Zhiyuan Ning, and Yuanchun Zhou. 2023. Resolving the imbalance issue in hierarchical disciplinary topic inference via llm-based data augmentation. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1424–1429. IEEE.

Richard B Canty and Milad Abolhasani. 2024. Reproducibility in automated chemistry laboratories using computer science abstractions. *Nature Synthesis*, 3(11):1327–1339.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2023. An empirical survey of data augmentation for limited data learning in nlp. *Transactions of the Association for Computational Linguistics*, 11:191–211.

Steffi Chern, Ethan Chern, Graham Neubig, and Pengfei Liu. 2024. Can large language models be trusted for evaluation? scalable meta-evaluation of llms as evaluators via agent debate. *arXiv preprint arXiv:2401.16788*.

Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, and Matteo Manica. 2023. Unifying molecular and textual representations via multi-task language modelling. In *International Conference on Machine Learning*, pages 6140–6157. PMLR.

Zhumin Chu, Qingyao Ai, Yiteng Tu, Haitao Li, and Yiqun Liu. 2024. Pre: A peer review based large language model evaluator. *arXiv preprint arXiv:2401.15641*.

Connor W Coley, Dale A Thomas III, Justin AM Lummiss, Jonathan N Jaworski, Christopher P Breen, Victor Schultz, Travis Hart, Joshua S Fishman, Luke Rogers, Hanyu Gao, et al. 2019. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science*, (6453):eaax1566.

Ian W Davies. 2019. The digitization of organic synthesis. *Nature*, 570(7760):175–181.

Linnea Evanson, Yair Lakretz, and Jean-Rémi King. 2023. Language acquisition: Do children and language models follow similar learning stages? *arXiv preprint arXiv:2306.03586*.

Vincent Fan, Yujie Qian, Alex Wang, Amber Wang, Connor W Coley, and Regina Barzilay. 2024. Openchemie: An information extraction toolkit for chemistry literature. *Journal of Chemical Information and Modeling*, 64(14):5521–5534.

Cong Gao, Benjamin D Killeen, Yicheng Hu, Robert B Grupp, Russell H Taylor, Mehran Armand, and Mathias Unberath. 2023. Synthetic data accelerates the development of generalizable learning-based algorithms for X-ray image analysis. *Nature Machine Intelligence*, 5(3):294–308.

Alexander G Godfrey, Thierry Masquelin, and Horst Hemmerle. 2013. A remote-controlled adaptive medchem lab: An innovative approach to enable drug discovery in the 21st century. *Drug Discovery Today*, 18(17-18):795–802.

Jiang Guo, A Santiago Ibanez-Lopez, Hanyu Gao, Victor Quach, Connor W Coley, Klavs F Jensen, and Regina Barzilay. 2021. Automated chemical reaction extraction from scientific literature. *Journal of chemical information and modeling*, 62(9):2035–2045.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Steven M Kearnes, Michael R Maser, Michael Wleklinski, Anton Kast, Abigail G Doyle, Spencer D Dreher, Joel M Hawkins, Klavs F Jensen, and Connor W Coley. 2021. The open reaction database. *Journal of the American Chemical Society*, 143(45):18820–18826.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, et al. 2024. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. *arXiv preprint arXiv:2402.13064*.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10443–10461, Singapore. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Zhiyuan Liu, Yaorui Shi, An Zhang, Sihang Li, Enzhi Zhang, Xiang Wang, Kenji Kawaguchi, and Tat-Seng Chua. 2024. Reactxt: Understanding molecular" reaction-ship" via reaction-contextualized molecule-text pretraining. *arXiv preprint arXiv:2405.14225*.

Daniel Lowe. 1976. Chemical reactions from us patents (1976-sep2016), 2017. *DOI*, 10:m9.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2024. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, pages 1–11.

S Hessam M Mehr, Matthew Craven, Artem I Leonov, Graham Keenan, and Leroy Cronin. 2020. A universal system for digitization and automatic execution of the chemical synthesis literature. *Science*, 370(6512):101–108.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Mark Peplow. 2014. the pobo-chemist. *Nature*, 512(7512):20.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Laria Reynolds and Kyle McDonell. 2021. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Philippe Schwaller, Riccardo Petraglia, Valerio Zullo, Vishnu H Nair, Rico Andreas Haeuselmann, Riccardo Pisoni, Costas Bekas, Anna Iuliano, and Teodoro Laino. 2020. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chemical science*, 11(12):3316–3325.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*.

Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. 2018. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 969–977.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482.

Alain C Vaucher, Philippe Schwaller, Joppe Geluykens, Vishnu H Nair, Anna Iuliano, and Teodoro Laino. 2021. Inferring experimental procedures from text-based representations of chemical reactions. *Nature communications*, 12(1):2573.

Alain C Vaucher, Federico Zipoli, Joppe Geluykens, Vishnu H Nair, Philippe Schwaller, and Teodoro Laino. 2020. Automated extraction of chemical synthesis actions from experimental procedures. *Nature communications*, 11(1):3601.

Eric Walker, Joshua Kammeraad, Jonathan Goetz, Michael T Robo, Ambuj Tewari, and Paul M Zimmerman. 2019. Learning to predict reaction conditions: Relationships between solvent, molecular structure, and catalyst. *Journal of chemical information and modeling*, 59(9):3645–3654.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Zhenhua Wang, Guang Xu, and Ming Ren. 2024. Llm-generated natural language meets scaling laws: New explorations and data augmentation methods. *arXiv preprint arXiv:2407.00322*.

Chenxi Whitehouse, Monojit Choudhury, and Alham Fikri Aji. 2023. LLM-powered data augmentation for enhanced cross-lingual performance. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 671–686, Singapore. Association for Computational Linguistics.

Jiayi Yuan, Ruixiang Tang, Xiaoqian Jiang, and Xia Hu. 2023. Large language models for healthcare data augmentation: An example on patient-trial matching. In *AMIA Annual Symposium Proceedings*, volume 2023, page 1324. American Medical Informatics Association.

Kaipeng Zeng, Bo Yang, Xin Zhao, Yu Zhang, Fan Nie, Xiaokang Yang, Yaohui Jin, and Yanyan Xu. 2024. Ualign: pushing the limit of template-free retrosynthesis prediction with unsupervised smiles alignment. *Journal of Cheminformatics*, 16(1):80.

Zheni Zeng, Yi-Chen Nie, Ning Ding, Qian-Jun Ding, Wei-Ting Ye, Cheng Yang, Maosong Sun, E Weinan, Rong Zhu, and Zhiyuan Liu. 2023. Transcription between human-readable synthetic descriptions and machine-executable instructions: An application of the latest pre-training technology. *Chemical Science*, 14(35):9360–9373.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wei Zhang, Qinggong Wang, Xiangtai Kong, Jiacheng Xiong, Shengkun Ni, Duanhua Cao, Buying Niu, Mingan Chen, Yameng Li, Runze Zhang, et al. 2024. Fine-tuning large language models for chemical text mining. *Chemical Science*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Ruochen Zhao, Wenxuan Zhang, Yew Ken Chia, Deli Zhao, and Lidong Bing. 2024. Auto arena of llms: Automating llm evaluations with agent peer-battles and committee discussions. *arXiv preprint arXiv:2405.20267*.

Ming Zhong, Siru Ouyang, Minhao Jiang, Vivian Hu, Yizhu Jiao, Xuan Wang, and Jiawei Han. 2023. ReactIE: Enhancing chemical reaction extraction with weak supervision. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12120–12130, Toronto, Canada. Association for Computational Linguistics.

A Dataset Details

A.1 Data Collection

We utilize two datasets to evaluate the effectiveness of our method across description-to-action (D2A) and action-to-description (A2D) tasks.

- **OPENEXP (Liu et al., 2024).** OPENEXP is an open-source dataset that is derived from the raw data from USPTO-Applications (Lowe, 1976) and ORD (Kearnes et al., 2021). It includes chemical reactions and the corresponding unstructured descriptions of experimental procedures. To obtain structured action sequences from reaction descriptions, authors conduct data preprocessing to filter low-quality data, leverage the action space defined by (Vaucher et al., 2020), and run the D2A model released by (Christofidellis et al., 2023). The processed database features 274,439 pairs of chemical reactions and the corresponding descriptions and step-by-step actions for experimental procedures. It is randomly split into train/valid/test sets by the 8:1:1 ratio.

  We explore the inner distribution characteristics of the OPENEXP dataset, which provides valuable insights into the data quality. Appendix Figure 5A-C displays the distribution of the number of actions, the number of description tokens, and the action taxonomy from the training and test set in the OPENEXP dataset, respectively. It is indicated that the distribution of the number of actions is similar. Additionally, we can see that each data query comprises at least four actions and an average of ten actions (Appendix Figure 5A). Further, from Appendix Figure 5B, we can see that both of the distribution of the number of description tokens keeps a long-tail feature, and is characterized by an average token number of around 300. Appendix Figure 5C gives an idea of the frequency of action types in the dataset, which indicates that 'ADD' and 'STIR' account for the majority of the action space, while very few reactions involve 'MICROWAVE' and 'SONICATE'.

  Further, we format the raw data from the OPENEXP dataset to make it suitable to the D2A task. As the dataset is initially designed for the reaction-to-action (R2A) task, the molecules and the duration involved in the action are translated into tokens, such as '1' and '@5@'. To ensure that the action sentence is demonstrated in natural language, we revert the tokens to their original forms. However, most of the chemicals related to the 'YIELD' action can not be mapped to their corresponding IUPAC names, which may cause confusion during the training period. As a result, we remove the 'YIELD' action and the corresponding components for all experiments on the OPENEXP dataset.

- **CHEMTRANS (Zeng et al., 2023).** CHEMTRANS is an open-source human-annotated dataset sourced from the raw data from the supplementary information of Organic Syntheses. The database defines a concise and complete instruction schema for chemical synthetic actions, which contains 16 types of actions. It consists of 3,950 description-action pairs, on average with 154.6 tokens per input description and 9.2 actions per output action sequence. It is randomly split into training (2,765), validation (395), and test (790) sets. Additionally, by randomly sampling and substituting the operation sequences and corresponding arguments, T5-ChemTrans (Zeng et al., 2023) conducts data augmentation and expands the size of the training set to 29,326. To make a fair comparison to the T5-ChemTrans method, we employ the augmented dataset named as CHEMTRANS-30k for our experimentation.

  To obtain more knowledge about the distribution characteristics of the training and test set from the CHEMTRANS dataset, we do an

investigation on the distribution of the number of actions, the number of description tokens and the frequency of action types, as illustrated on Appendix Figure 5D-F. It is implied that most of the reactions contains nine actions (Appendix Figure 5D). Moreover, when the number of actions is fewer than five or more than fifteen, which is indicted in the two-side bars of the figure, the test set contains more instances with these action counts compared to the training set. As illustrated in Appendix Figure 5E, each reaction description consists of an average of 300 tokens, and very few descriptions contains more than 800 tokens. Thus, we set the maximum output text token length as 800 for the A2D task. Appendix Figure 5F shows that 'ADD' and 'SETTEMP' make up the majority of the action space. Besides, we can see that the distribution of the frequency of the action types from the training set is not very consistent to that from the test set. Actions such as 'DISTILL' and 'TRANSFER' occur much more frequently in the test set than in the training set, which poses a challenge on exactly deducing action types during the evaluation period.

Apart from the two open-source description-action datasets mentioned above, there are some close-source action datasets. Vaucher et al. (Vaucher et al., 2020) curate a database from the raw data from the commercial dataset Pistachio, and a number of studies employ it to perform the D2A task (Vaucher et al., 2020; Zhang et al., 2024; Zhong et al., 2023). To make comparisons to the prior works, we do dataset statistics which is presented in Appendix Table 8. Moreover, to obtain more insights into the pre-defined action space, Appendix Figure 3 lists the action types annotated in the CHEMTRANS and the OPENEXP dataset. The action types written in black appear in both datasets, while the action types written in blue only appear in one dataset. It is worth mentioning that though some action types are expressed in different forms, but they are similar in meaning. For instance, the 'SETTEMP' action annotated in the CHEMTRANS dataset and the 'SETTEMPERATURE' action annotated in the CHEMTRANS dataset share the same meaning. Additionally, the 'DRY' action from the CHEMTRANS dataset is separated into the 'DRYSOLUTION' and 'DRYSOLID' actions from the OPENEXP dataset.

A.2 Construction of Pair-wised Instruction Datasets

Instruction prompt datasets refer to formatting structured or unstructured data as natural language instructions, enabling LLMs to respond appropriately (Reynolds and McDonell, 2021). Recent advancements indicate that constructing high-quality prompt datasets facilitates effective reasoning of LLMs (Wang et al., 2023). Towards the task of generating structural experimental actions from literature, we design a tailored instruction prompts system for better instruction tuning (Appendix Figure 4). Specifically, for both CHEMTRANS and OPENEXP datasets, we first design instruction prompts for the definition of tasks, a.k.a. ***Instruction*** in Appendix Figure 4. Next, we collect pair-wised reaction description-action sequences to construct high-quality Q&A datasets. Question templates such as *'Please generate a sequence of structured actions according to the given description of experimental procedures'* are generated by GPT-4 autonomously using prompt engineering. '***Instruction***' and '***Source***' in Appendix Figure 4 are integrated together as the input for instruction tuning. It is important to note that, we generate approximately 2,000 templates using GPT-4 to construct datasets, thereby ensuring the diversity and completeness of training sets. After constructing pair-wised Q&A datasets, we further analyze the token distribution for two training sets, seen in Appendix Figure 5B. This indicates that the maximum text token length is 800, so we set it to 800 during training.

B Experimental Details

B.1 Baseline methods

We briefly introduce the baselines:

- **Paragraph2Actions (Vaucher et al., 2020)**. Paragraph2Actions is the first to propose the D2A task with pre-defined synthesis actions. Both Paragraph2Actions (Vaucher et al., 2020) and Paragraph2Actions+ (Vaucher et al., 2021) utilize a simple transformer-based model without any pre-training tasks to predict action sequences. Specifically, Paragraph2Actions+ incorporates additional knowledge-based enhancements to improve performance. Authors curate the data from the commercial Pistachio dataset, conduct data preprocessing, and collect description-action

Dataset	Total	Train	Valid	Test	Open source
Pistachio (Vaucher et al., 2020)	693k	555k	69k	69k	No
OpenExp (Liu et al., 2024)	274k	220k	27k	27k	Yes
ChemTrans (Zeng et al., 2023)	3,950	2,765	395	790	Yes
ChemTrans-30k (Zeng et al., 2023)	30,511	29,326	395	790	Yes

Table 8: The statistics and comparison of different datasets.

Pre-defined action types in two datasets

ChemTrans		OpenExp	
ADD	SETTEMP	ADD	SETTEMPERATURE
YIELD	DRY	YIELD	DRYSOLUTION
WASH	COLUMN	WASH	DRYSOLID
EXTRACT	DISTILL	EXTRACT	STIR
FILTER	EVAPORATE	FILTER	CONCENTRATE
QUENCH	TRANSFER	QUENCH	MAKESOLUTION
RECRYSTALLIZE		RECRYSTALLIZE	COLLECTLAYER
PARTITION		PARTITION	WAIT
TRITURATE		TRITURATE	PHASESEPARATION
REFLUX		REFLUX	DEGAS
			MICROWAVE
			SONICATE
			PH

Figure 3: Comparison of the action types on the ChemTrans and OpenExp datasets.

pairs.

- **T5 (Raffel et al., 2020)**. T5 is a method based on an encoder-decoder architecture, which has achieved competitive performance on a broad range of natural language processing tasks. For the T5-Base model, it consists of 12 encoder layers and 12 decoder layers, and is scaled up to 220M parameters.

- **T5-ChemTrans (Zeng et al., 2023)**. T5-ChemTrans utilizes a T5-based foundational model to facilitate transcription between human-readable reaction descriptions and machine-executable instructions. It is pre-trained on four knowledge enhancement tasks and fine-tuned on the augmented ChemTrans dataset.

- **GPT (Achiam et al., 2023)** GPT is a transformer-based large language model, which marks a massive leap in the process of language models. During its development, OpenAI enveils a series of GPT-based versions, including GPT-3.5-turbo and GPT-4o. In this work, we introduce GPT-3.5 to conduct D2A task. GPT-3.5 (text-davinci-003 completion mode) and GPT-3.5-chat (gpt-3.5-turbo chat completion mode) are employed for comparison. The original version displays the zero-shot performance. Further, GPT-3.5, 3-shot is given 3 randomly picked training instances, and GPT-3.5, 3-shot* is given 3 training instances with the highest similarity with the current testing instance. Additionally, we introduce GPT-4o for LLMs circle review. Compared to GPT-3.5-turbo, GPT-4o features a larger parameter scale and context window length, which improves its natural language processing capabilities.

- **ReactXT (Liu et al., 2024)**. ReactXT is developed based on the MolCA model. It incorporates three types of input contexts to incrementally pre-train LMs, which makes it suitable for experimental procedure prediction, retrosynthesis and molecule captioning tasks.

- **Mistral (Jiang et al., 2023)**. Mistral-7B-Instruct-v0.2 is a large language model with about 7.3B parameters. It trains transformers with grouped-query attention (GQA) and sliding window attention (SWA).

- **LLaMA (Touvron et al., 2023).** LLaMA

(a) ***A example of pair-wised instruction datasets on ChemTrans***

Instruction	**Chemist:** Considering a chemical reaction, unstructured descriptions of experimental procedures in chemical reactions can often offer valuable insights into the details of the experiment that standardized formats might overlook. Please generate a sequence of structured actions according to the given description of experimental procedures
Source	(B) Reduction of Cinnamic Acid.—After the apparatus is assembled, 2 l. of 7–8 per cent sodium sulfate solution is placed in the battery jar and the porous cup is filled to the same level with more of this solution. The stirrer is started, and 200 g. (1.35 moles) of a good grade of cinnamic acid is suspended in the catholyte. A solution of 35 g. (0.88 mole) of sodium hydroxide in 150 cc.

Annotated actions	***Predicted actions***
[add] reagent: (name: sodium sulfate & type: pure & volume: 2 l. & concentration: 7–8 per cent &) & [add] reagent: (name: cinnamic acid & type: pure & mole: 1.35 moles & mass: 200 g. & note: a good grade &) & reagent: (name: cinnamic acid & type: pure & mole: 1.35 moles & mass: 200 g. &)	[add] reagent: (name: sodium sulfate & type: pure & volume: 2 l. & concentration: 7–8 per cent &) & reagent: (name: cinnamic acid & mole: 1.35 moles & mass: 200 g. & note: good grade &) & reagent: (name: A solution of 35 g. 0.88 mole of sodium hydroxide in 150 cc. of water & type: mixture & speed: a rate as to avoid forming lumps of sodium cinnamate. &) [add] reagent: (name: sodium cinnamate & type: pure & note: suspended &)

(b) ***A example of pair-wised instruction datasets on OpenExp***

Instruction	**Chemist:** You are an experienced chemcial assistant. In a reaction, a sequence of structured actions gives synthetic knowledge that can be documented into the instruction library efficiently. Could you generate a sequence of structured actions?
Source	Crude 7-(3,4-dichlorophenyl)-8-hydroxy-5H-pyrido[2,3-b]pyrazin-6-one (0.13 g) was heated in acetic anhydride (10 mL) for 3 hours and then allowed to cool and stored at ambient temperature for 16 hours. The reaction mixture was poured onto ice and water, stirred vigorously and extracted with ethyl acetate. The organic extract was washed with water and concentrated. The residue was purified by column chromatography on silica gel (eluent: ethyl acetate/hexane 1:1) to give Compound B29 of Table B as a yellow solid (0.077 g)

Annotated actions	***Predicted actions***
ADD 7-(3,4-dichlorophenyl)-8-hydroxy-5H-pyido[2,3-b]pyrazin-6-one (0.13 g) ; ADD acetic anhydride (10 mL) ; STIR for 3 hours ; ADD water at ambient temperature over 16 hours ; STIR ; EXTRACT with ethyl acetate ; WASH with water ; CONCENTRATE ;	ADD 7-(3,4-dichlorophenyl)-8-hydroxy-5H-pyido[2,3-b]pyrazin-6-one (0.13 g) ; ADD acetic anhydride (10 mL) ; STIR for 3 hours ; ADD water at ambient temperature over 16 hours ; STIR ; EXTRACT with ethyl acetate ; WASH with water ; CONCENTRATE ;

Figure 4: Examples of constructed Q&A datasets on CHEMTRANS and OPENEXP datasets.

is an open-source fine-tuned large language model, which has launched three versions so far. Compared to LLaMA-2, LLaMA-3 has made improvements on tasks such as chemistry-related questioning, code generation, and mathematics.

B.2 Hyperparameters

We offer a comprehensive summary of the training settings and hyperparameter values among ChemActor and baseline methods.

- **ChemActor.** During the training process, we utilize the single-step retrosynthesis model proposed by (Zeng et al., 2024) to predict reactants given a product molecule. LLaMA-2 (Touvron et al., 2023) is employed as our scientific foundational model. Fully fine-tuning LLaMA-2-7B for 8 epochs is completed in approximately 12 hours using 8 × NVIDIA A800 GPUs for OPENEXP datasets. We set the batch size as 12, the default learning rate as 9.65e-6.

- **T5-ChemTrans.** To perform the D2A and A2D tasks on the CHEMTRANS dataset, we utilize the T5-ChemTrans (Base) model checkpoints provided by (Zeng et al., 2023) and conduct generative inferences on the test set. Besides, for D2A tasks evaluated on the OPENEXP dataset, we employ the source codes released by (Zeng et al., 2023) and fine-tune the T5-ChemTrans (Base) model on 220k description-action pairs from the OPENEXP training set. During the fine-tuning period, we set the batch size to be 16, the learning rate to be 1e-4.

- **T5-Base.** We fine-tune the T5-Base model for performance improvement, which leverages

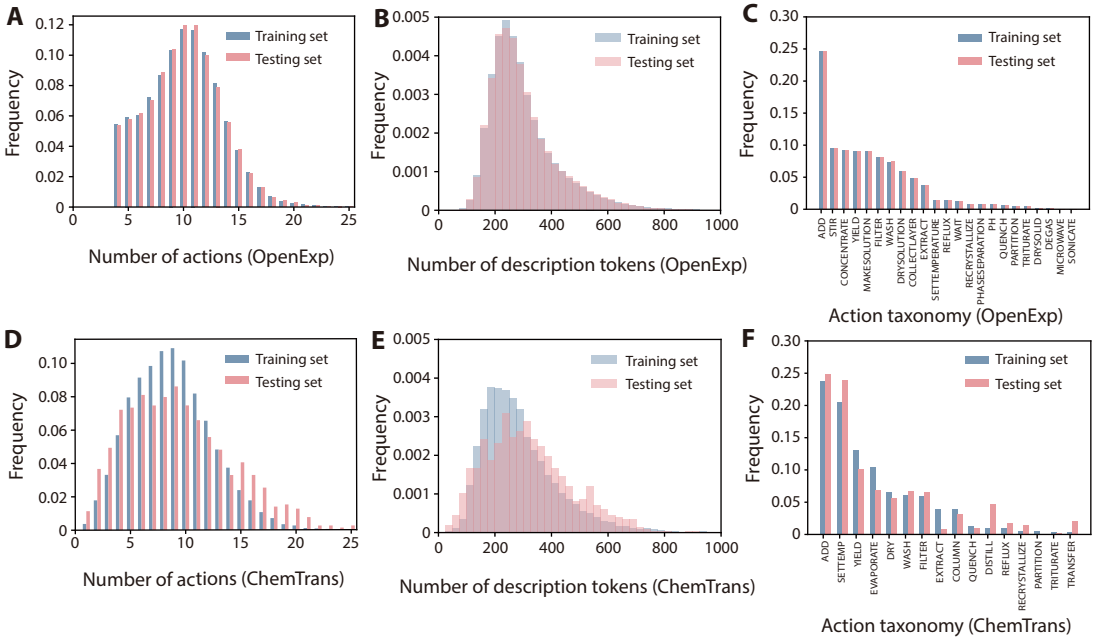

Figure 5: Data distribution of the OPENEXP and CHEMTRANS datasets.

the codes released by (Zhang et al., 2024). During the training period, we set the batch size as eight, the maximum number of epochs as five and the learning rate as 1e-5.

- **Mistral-7B-Instruct-v0.2.** We follow (Zhang et al., 2024) to do full parameter fine-tuning for Mistral-7B-Instruct-v0.2. Our experimentation is capped for no more than three epochs, with the learning rate initialized as 1e-6 and the batch size as two.

- **LLaMA-3-7B-Instruct.** We also do full parameter fine-tuning for LLaMA-3-7B-Instruct. Our training settings are similar to those on the Mistral-7B-Instruct-v0.2 model.

B.3 Metrics

Common Metric. We assess performance using the Sequence Matching (SM) and ExactMatch (EM) metrics: SM-O evaluates the accuracy of operation prediction, while SM-A additionally considers the accuracy of argument prediction. For ExactMatch (EM), we report the proportion of items that are predicted perfectly. The detailed calculation of metrics is discussed in (Zeng et al., 2023).

Subsequently, following (Vaucher et al., 2021; Liu et al., 2024), we employ the additional metrics for performance evaluation in Table 2, validity, which checks the syntactical correctness of the action sequence; the normalized Levenshtein similarity (LEV) (Levenshtein et al., 1966). Specifically, 90%LEV refers to the proportion of predictions that achieve a normalized Levenshtein score greater than 0.9. The BLEU score (Papineni et al., 2002) and the ROUGE score (Lin, 2004). BLEU focuses on precision, while ROUGE emphasizes recall in the generated text.

BERTScore. Traditional natural language generation metrics assess the similarity between two strings by counting the minimum number of single-character edits (insertions, deletions, or substitutions) needed to transform one string into the other. These metrics cannot effectively capture the semantics of synthesis actions. To effectively evaluate ChemActor, we introduce two additional metrics, including BERTScore (Zhang et al., 2019) and multi-round LLMs circle review (Chu et al., 2024), shown in Figure 1C-D. Considering BERTScore, it adopts encoder-only LLMs to evaluate the word embedding similarity of the prediction and the ground truth. BERTScore can adopt different LMs as the encoder, but the $deberta - xlarge - mnli$ model provides scores that are most closely aligned with human evaluations. Therefore, we select $deberta - xlarge - mnli$ to evaluate BERTScore, which can be computed by Appendix Equation 3, where $\mathbf{E}_p = \{\mathbf{e}_p^1, \cdots, \mathbf{e}_p^k\}$ and $\mathbf{E}_t = \{\mathbf{e}_t^1, \cdots, \mathbf{e}_t^k\}$ are embeddings of the predictions and the ground truths, derived from the $deberta - xlarge - mnli$ model.

$$\begin{cases} R = \frac{1}{|\mathbf{E}_t|} \sum_{i=1}^{k} \max_{\mathbf{e}_p^j \in \mathbf{E}_p} \mathbf{e}_t^{i\top} e_p^j \\ P = \frac{1}{|\mathbf{E}_p|} \sum_{j=1}^{k} \max_{\mathbf{e}_t^i \in \mathbf{E}_t} \mathbf{e}_t^{i\top} e_p^j \\ F1 = 2 \times \frac{P \times R}{P + R} \end{cases} \quad (3)$$

Multi-Round LLMs Circle Review. Besides BERTScore, we ultilize the multi-round LLMs circle review to measure the semantic rationality of generated actions. We follow the debating strategy proposed by (Chern et al., 2024). To be detailed, at round 0, we introduce a number of LLMs to evaluate the semantic similarity between the predicted and annotated actions. To standardize the evaluation results, we set a restriction that the LLM responses should output a score ranging from 0 to 1 and a sentence of explanation about why the score is marked. In this way, the evaluations from multiple LLMs are obtained at the initial stage. After that, we carry out a debate between LLMs. Specifically, at round N, we make the output scores and explanations generated by LLMs at round N-1 transparent to each other. Then, we apply a prompt-based method to create a debate among LLMs, which is illustrated in Appendix Figure 9. Given the peer responses, each LLM will reconsider its previous response to the annotated and predicted actions, and provide a revision of its evaluation score and rationale. From the figure, we can see the score by GPT-3.5 shifts from 1.0 to 0.8 after the debate. After revising the responses on the whole dataset, the debate at round N is completed and the revised results will be spread to LLMs at round N+1. The debate will last for a fixed number of rounds, and the final average score is taken as the LLM score.

C Performance Evaluation

C.1 Training Size Proportions

To verify that the size of the training set can indeed improve the model performance, we adjust the proportions of training data on the CHEMTRANS and OPENEXP datasets for the D2A task. Specifically, we randomly sort the training data and select the top data by a ratio of 0.1, 0.3, 0.5, 0.7, and 1.0, which forms a series of incremental data. In addition to our proposed ChemActor, we introduce T5-Base, Mistral-7B-Instruct-v0.2 and LLaMA-3, which are competitive D2A models and are discussed in the 'Experimental Details' section. We adopt 95% LEV, 75% LEV, modified BLEU, and the average Levenshtein similarity to assess the model performance, which is introduced in the 'Metrics' section. To be more intuitive, we visualize the performance of the Levenshtein similarity and the modified BLEU on the CHEMTRANS and OPENEXP datasets, which are illustrated in Appendix Figure 7. From the figures, we can see that ChemActor performs consistently better than other methods on both evaluation metrics on the CHEMTRANS and OPENEXP datasets, which highlights the effectiveness of our method.

The visualization reveals that the distribution of our proposed LLM-generated data has fewer overlaps with the distribution of real data. However, the distribution of augmented data shows more overlaps. By increasing the number of LLM-generated data, the distribution of our generated data gradually encompasses the entire chemical space. This indicates that the proposed data selection module is effective in generating diverse and useful data, which mitigates data sparsity issues during training and subsequently improves the performance of action sequence prediction.

C.2 Type Matching

To evaluate the performance between our ChemActor and baseline methods, we employ the evaluation metrics mentioned in previous works, which have been introduced in the Section B.3. However, among these metrics, the metrics derived from natural language tasks, such as BLEU and ROUGE, focus on the exact matching of sentences. Moreover, the metrics designed for the D2A task, such as SM-O, only care about the quality of action type classification and overlook the action components following the action type, which are indeed more important for chemical experimental procedures. Thus, in addition to these metrics, we further test type-matching experiments to evaluate the model's capability of perfectly predicting both the action types and the corresponding necessary components in the meantime.

To be specific, we first calculate the recall metrics of ChemActor in action type recognition, as presented in the Table 9. From these results, we can observe that ChemActor exhibits varying recognition capabilities across different action types. For most action types, our model achieves a recall of over 75%, demonstrating its excellent performance. However, for the action types 'quench' and 'reflex', the recall is only around 60%. To investigate

Action type	Recall (%)	Action type	Recall (%)
add	85.82	extract	80.89
settemp	89.35	quench	61.63
yield	79.15	distill	75.00
evaporate	76.48	reflux	55.71
dry	74.44	recrystallize	64.75
wash	70.12	triturate	71.43
filter	77.32	transfer	77.25
column	65.45		

Table 9: Results for action type predictions on CHEMTRANS dataset.

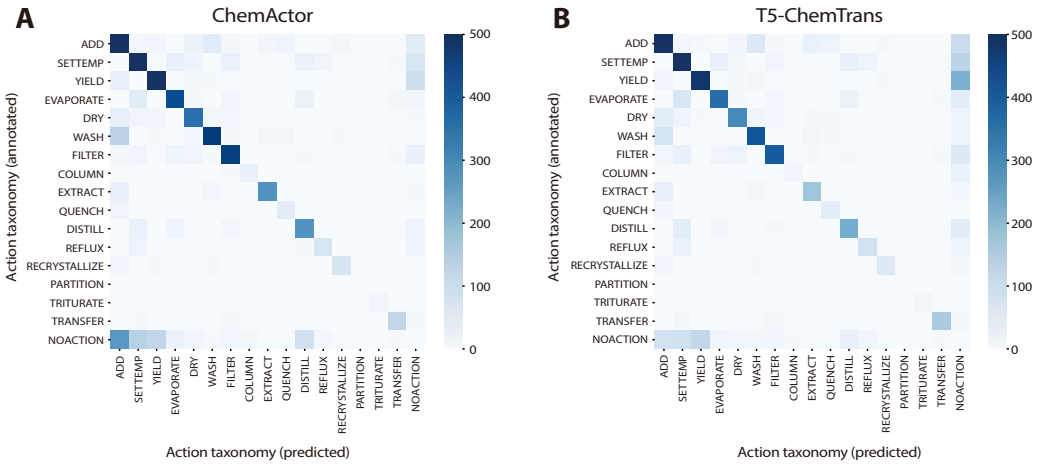

Figure 6: Heatmap of type matching results of ChemActor and T5-ChemTrans.

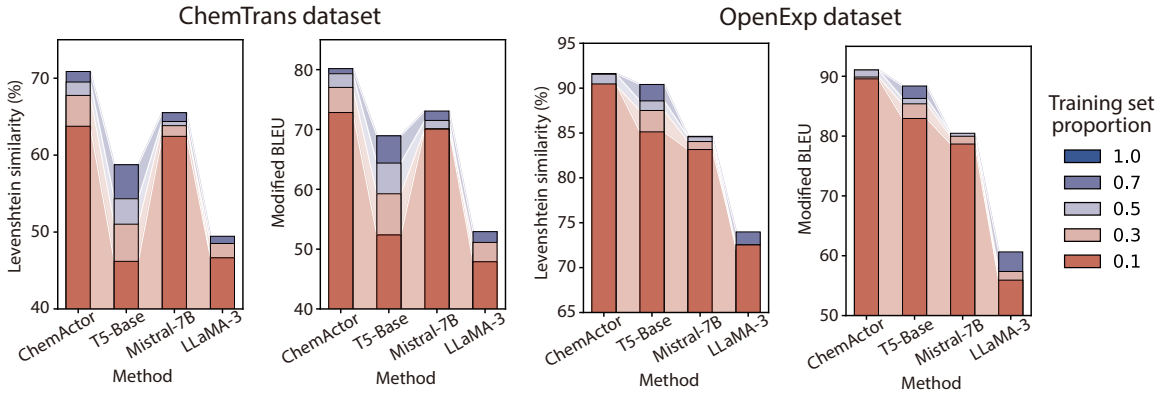

Figure 7: The performance varying different training proportions on CHEMTRANS and OPENEXP datasets.

the reasons behind the model's suboptimal performance on these two action types, we conduct an in-depth analysis of the action type distribution within the dataset. As shown in Appendix Figure 5, the action types 'quench' and 'reflex' appear with very low frequency in the training set, approximately 0.02. In that case, LLMs may struggle to learn effectively from extremely low-frequency samples, leading to the reduced recognition performance for these two action types.

Further, we try to assess the model's ability to accurately predict both the types of actions and their corresponding essential components simultaneously. We separate the action sentence into a sequence of action phrases, and every action phrase contains one action type and a paragraph of action components. For example, the action phrase *'[quench] reagent: (name: ice water & type: pure & volume: 200 mL &)'* comprises an action type *'quench'* and the paired action components *'reagent:*

(name: ice water & type: pure & volume: 200 mL &)'. After extracting the action types and the action components from the predicted and annotated actions, we traverse the data to do action type matching. It is important to mention that instead of simply matching the predicted action types to the annotated action types, we compute the difflib similarity between every predicted-annotated action component pair. For each action phrase in the annotated action, we pick out the predicted action phrase containing the most similar action component to form a matching pair. To avoid mismatching, the matching is valid only when the similarity is higher than a certain threshold (we set it as 0.4), or its paired action phrase will be changed into 'NOACTION'. For the action phrases in the predicted action, we follow similar steps. Finally, we count all the action type pairs and get the Type Matching results. In this way, the rule-based algorithm can do matches between the predicted and annotated action types that are most related to each other, thereby making the metric effectively evaluate the quality of generated actions.

We draw heatmaps of the type matching results, which are illustrated in Appendix Figure 6. The evaluation results are generated by our ChemActor and a competitive method, T5-ChemTrans, which has been introduced in the 'Experimental Details' section. In the figure, the labels of the predicted and annotated action types are on the x-axis and y-axis, correspondingly. The darker the color of the square is, the more pairs that satisfy the corresponding predicted and annotated action type. In this way, the darkness of the color of diagonal squares can effectively represent the model performance in making perfect action predictions. For the sake of clarity, the color scale is capped at 500, though many action-type pairs, particularly those on the diagonal, exceed this value.

From Appendix Figure 6, we can see that both ChemActor and T5-ChemTrans have a good performance on predicting action types, as the color of off-diagonal squares is much more shallow than that of diagonal squares. More importantly, the color of diagonal squares from ChemActor is greatly darker than that from T5-ChemTrans, which highlights the effectiveness of our method. Additionally, our proposed ChemActor presents an outstanding capability of learning from a few samples. For instance, the action type 'DISTILL' occurs with a frequency of about 0.01 in the training set as depicted in Appendix Figure 5F, which highlights the challenge of representation learning. From Appendix Figure 5, we can see that the color of the 'DISTILL-DISTILL' square in the left figure is much darker than that in the right figure, and the color of 'DISTILL-X' and 'X-DISTILL' square is much more shallow. It is indicated that ChemActor is less likely to make mistakes on the action 'DISTILL' than T5-ChemTrans, which shows the great advantage of ChemActor on few-shot learning.

C.3 Training stratiges

Various factors observed in the data may influence the speed of learning. We investigate the impact of LLM-generated data on model learning speed, focusing on data curriculum methods. Different skills require tailored training data, and the choice of data curriculum affects skill acquisition rates. Here, we define each action type in the CHEMTRANS datasets as a skill and measure the number of training steps needed to acquire these skills. We introduce two types of data curriculum methods, **uniform mixing**, where real and generated data are combined and fed to the model simultaneously, and **alternate mixing**, where real and generated data are fed in sequence, alternating during training. Instead of using next-word prediction loss, inspired by (Evanson et al., 2023), we measure acquisition time, the number of steps needed for the model to reach 90% of its final accuracy. Appendix Figure 8 discusses the relationship between data curriculum methods and model skills. All skills are sorted by the acquisition time.

From the results, we can see that different skills have different speeds of learning and emerge at different times. The smaller the percentage of actions in the training set, such as 'DISTILL', and 'REFLUX', the longer acquisition time it takes for the model to learn them. More importantly, the alternative mixing strategy enhances the speed of skill acquisition as demonstrated by the leftward movement of the data points in Appendix Figure 8(right). This adjustment allows us to observe these points earlier in the training process.

C.4 Multi-Round LLMs Circle Review

To evaluate our methods on the D2A task, we have applied a number of natural language metrics and proposed the Type Matching metric, which have been discussed in the previous sections. However, upon closely examining the predictions on the CHEMTRANS test set, we find that these metrics fail to comprehensively evaluate the accuracy

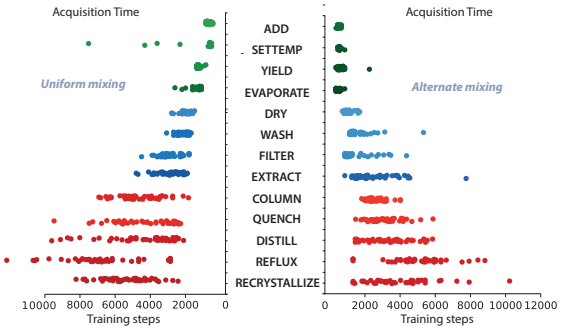

Figure 8: The performance of different data curriculum methods. Acquisition time: The number of steps required to reach 90% of final accuracy.

and quality of the generated actions, as illustrated in Appendix Figure 10 and Appendix Figure 11. To be specific, in Appendix Figure 10, the action generated by our ChemActor paraphrases the action phrase *'[wash] reagent: (name: two 60-mL portions of a 1:1 mixture of saturated aqueous sodium bicarbonate and water & type: mixture & volume: 60-mL & concentration: 1:1 & batch:each: portions &)'* as *' [wash] reagent: (name: sodium bicarbonate & type: mixture & volume: 60-mL & concentration: saturated & note: aqueous & batch:each: portions &) & reagent: (name: water & type: pure &)'*. It can be inferred that the predicted actions are semantically similar to the annotated actions. However, the generated actions are not in the same data format or in the same order as the human annotations, which leads to a low difflib similarity of 13.18. In that point, the previous metrics which only focus on exact matching fail to handle the diversity of action outputs, and underestimate the effectiveness of our model. Thus, it is crucial to apply an evaluation metric capable of effectively measuring semantic rationality.

We apply the multi-round LLMs circle review to evaluate the semantics of synthesis actions, and the details of the metric are introduced in the Appendix Section B.3. During our experimentation, to evaluate the effectiveness of our method on the D2A task, we introduce Mistral-7B-Instruct-v0.2, T5-Base, and T5-ChemTrans as baseline methods, whose experimental details are discussed in the Appendix Section B. We curate a bad case dataset and conduct the LLM-based multi-round circle review evaluation. Specifically, we traverse the actions predicted by our ChemActor and other methods and compute the difflib similarity between the ground truth and the prediction. Then, we analyze the data to filter cases with a difflib similarity greater than 0.4, remaining 279 bad cases. Additionally, we adopt GPT-3.5-turbo, GPT-4o, LLaMA-3, and Mistral-7B-Instruct-v0.2, which show great competitiveness in natural language processing tasks. Further, we set the number of debate round as two and conduct the multi-round LLMs circle review evaluation.

The evaluation results are illustrated in Figure 2. Our method outperforms other baselines on the multi-round circle review score based on all LLMs. To be more detailed, we give two examples of multi-round LLMs circle review evaluation on bad cases, which are illustrated in Appendix Figure 10 and Appendix Figure 11. We can see that the results generated by our method are semantically similar to the ground truth, and achieves a relatively low difflib similarity score but the highest LLM review score. It implies that the difflib similarity metric simply measures the sentence similarity token by token, which may overlook the diversity of action predictions and underestimate the predictions generated by our method. However, the multi-round LLMs circle review is able to evaluate the semantic rationality and provide a more credible evaluation result.

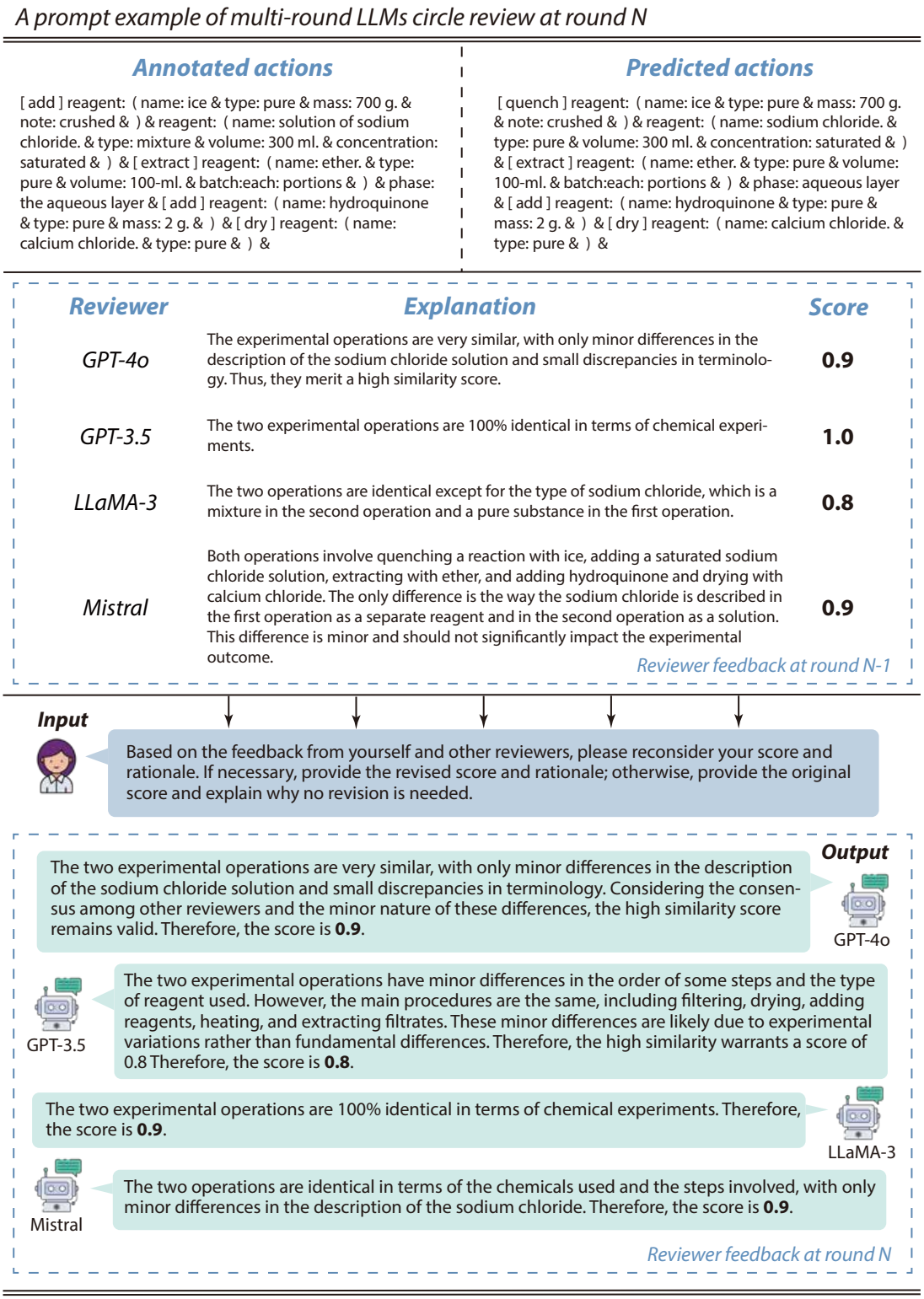

Figure 9: The pipeline of the multi-round LLMs circle review.

An example of the performance of LLMs circle review on the ChemTrans dataset at round 2

Label	[quench] reagent: (name: ice water & type: pure & volume: 200 mL &) & [extract] reagent: (name: methylene chloride. & type: pure & volume: 30-mL & batch:each: portions &) & phase: aqueous layer & [wash] reagent: (name: brine. & type: pure &) & reagent: (name: two 60-mL portions of a 1 : 1 mixture of saturated aqueous sodium bicarbonate and water & type: mixture & volume: 60-mL & concentration: 1 : 1 & batch:each: portions &) & reagent: (name: sodium bicarbonate & type: pure & concentration: saturated & note: aqueous &) & reagent: (name: water & type: pure &) & [dry] reagent: (name: sodium sulfate & type: pure &) & [evaporate] N/A: removed &		
	Predicted answers	***EM***	***LLM review (GPT-4o)***
ChemActor	[quench] reagent: (name: water & type: pure & volume: 200 mL & temperature: ice &) & [settemp] N/A: stirring & [extract] reagent: (name: methylene chloride. & type: pure & volume: 30-mL & batch:each: portions &) & phase: aqueous layer & [wash] reagent: (name: sodium bicarbonate & type: mixture & volume: 60-mL & concentration: saturated & note: aqueous & batch:each: portions &) & reagent: (name: water & type: pure &) & reagent: (name: brine. & type: pure &) & [dry] reagent: (name: sodium sulfate & type: pure &) & [evaporate] N/A: removed &	**13.18**	**0.9**
Mistral	[quench] reagent: (name: water & type: pure & volume: 200 mL & temperature: ice &) & [extract] reagent: (name: methylene chloride. & type: pure & volume: 30-mL & batch:each: portions &) & phase: aqueous layer & [wash] reagent: (name: sodium bicarbonate & type: mixture & volume: 60-mL & concentration: saturated & note: aqueous &) & reagent: (name: water. & type: pure & volume: 60-mL & concentration: saturated & note: aqueous &) & phase: The combined methylene chloride extracts & [dry] reagent: (name: sodium sulfate & type: pure &) & [evaporate] N/A: removed &	**64.75**	**0.8**
T5-Base	[add] reagent: (name: ice water & type: pure & volume: 200 mL & note: ice &) & [settemp] N/A: stirring & [extract] reagent: (name: methylene chloride. & type: pure & volume: 30-mL & batch:each: portions &) & [wash] reagent: (name: sodium bicarbonate & type: pure & volume: 60-mL & concentration: saturated & note: aqueous &) & reagent: (name: water & type: pure & volume: 60-mL & batch:each: portions &) & reagent: (name: brine. & type: pure &) & [dry] reagent: (name: sodium sulfate & type: pure &) & [evaporate] N/A: removed &	**12.88**	**0.6**
T5-ChemTrans	[quench] reagent: (name: water & type: pure & volume: 200 mL & temperature: ice &) & [extract] reagent: (name: methylene chloride. & type: pure & volume: 30-mL & batch:each: portions &) & [wash] reagent: (name: sodium bicarbonate & type: mixture & volume: 60-mL & concentration: 1 & note: saturated & batch:each: portions &) & reagent: (name: water & type: pure &) & reagent: (name: brine. & type: pure &) & [dry] reagent: (name: sodium sulfate & type: pure &) & [evaporate] N/A: removed &	**57.12**	**0.8**

Figure 10: An example of the performance of multi-round LLMs circle review (bad case 1).

An example of the performance of LLMs circle review on the ChemTrans dataset at round 2

Label	[add] reagent: (name: ice & type: pure & mass: 700 g. & note: crushed &) & reagent: (name: solution of sodium chloride. & type: mixture & volume: 300 ml. & concentration: saturated &) & [extract] reagent: (name: ether. & type: pure & volume: 100-ml. & batch:each: portions &) & phase: the aqueous layer & [add] reagent: (name: hydroquinone & type: pure & mass: 2 g. &) & [dry] reagent: (name: calcium chloride. & type: pure &) &		
	Predicted answers	***Difflib similarity***	***LLM review (GPT-4o)***
ChemActor	[quench] reagent: (name: ice & type: pure & mass: 700 g. & note: crushed &) & reagent: (name: sodium chloride. & type: pure & volume: 300 ml. & concentration: saturated &) & [extract] reagent: (name: ether. & type: pure & volume: 100-ml. & batch:each: portions &) & phase: aqueous layer & [add] reagent: (name: hydroquinone & type: pure & mass: 2 g. &) & [dry] reagent: (name: calcium chloride. & type: pure &) &	13.04	0.9
Mistral	[add] reagent: (name: ice & type: pure & mass: 700 g. & note: crushed &) & reagent: (name: sodium chloride. & type: pure & volume: 300 ml. & concentration: saturated &) & [extract] reagent: (name: ether. & type: pure & volume: 100-ml. & batch:each: portions &) & phase: aqueous layer & [add] reagent: (name: hydroquinone & type: pure & mass: 2 g. &) & [dry] reagent: (name: calcium chloride. & type: pure &) &	92.56	0.8
T5-Base	[settemp] N/A: stirring & [extract] reagent: (name: ether. & type: pure & volume: 100-ml. & batch:each: portions &) & [add] reagent: (name: hydroquinone & type: pure & mass: 2 g. &) & [dry] reagent: (name: calcium chloride. & type: pure &) & [dry] reagent: (name: calcium chloride. & type: pure &) &	60.23	0.6
T5-ChemTrans	[quench] reagent: (name: 700 g. of crushed ice & type: mixture & mass: 700 g. & note: crushed &) & reagent: (name: sodium chloride. & type: mixture & volume: 300 ml. & concentration: saturated &) & [extract] reagent: (name: ether. & type: pure & volume: 100-ml. & batch:each: portions &) & [add] reagent: (name: hydroquinone & type: pure & mass: 2 g. &) & [dry] reagent: (name: calcium chloride & type: pure &) &	8.99	0.8

Figure 11: An example of the performance of multi-round LLMs circle review (bad case 2).

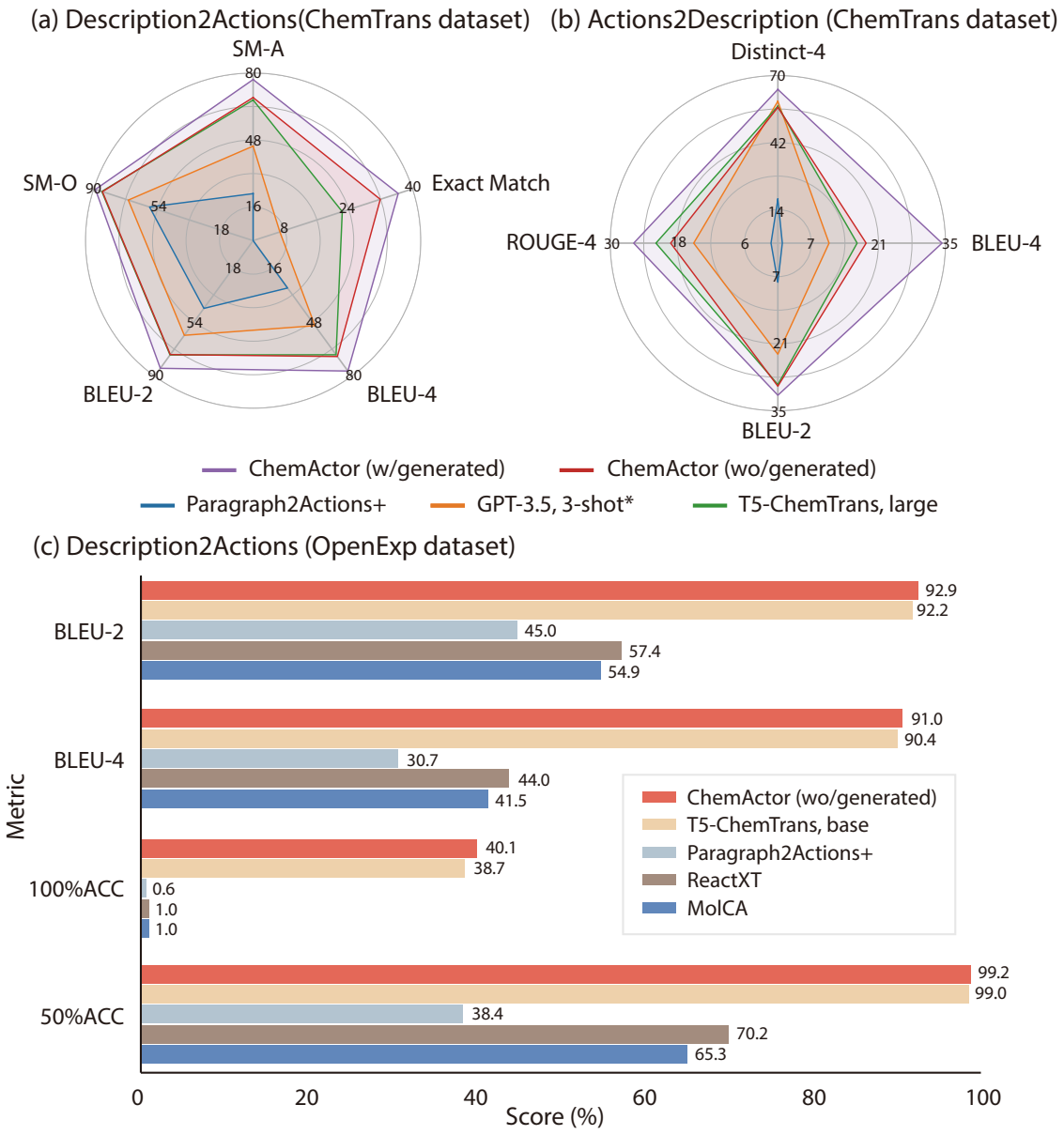

Figure 12: Experimental results of ChemActor on ChemTrans and OpenExp datasets.