CoachLM: Automatic Instruction Revisions Improve the Data Quality in LLM Instruction Tuning

Yilun Liu, Shimin Tao, Xiaofeng Zhao, Ming Zhu, Wenbing Ma, Junhao Zhu, Chang Su, Yutai Hou, Miao Zhang, Min Zhang, Hongxia Ma, Li Zhang, Hao Yang, Yanfei Jiang
Huawei, China
{liuyilun3, taoshimin, zhaoxiaofeng14, zhuming47, mawenbing, zhujunhao, suchang8, houyutai, emma.zhangmiao, zhangmin186, mahongxia, izzie.zhangli, yanghao30, jiangyanfei}@huawei.com

2024 IEEE 40th International Conference on Data Engineering (ICDE) | 979-8-3503-1715-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDE60146.2024.00390

***Abstract*—Instruction tuning is crucial for enabling Language Learning Models (LLMs) in responding to human instructions. The quality of instruction pairs used for tuning greatly affects the performance of LLMs. However, the manual creation of high-quality instruction datasets is costly, leading to the adoption of automatic generation of instruction pairs by LLMs as a popular alternative. To ensure the high quality of LLM-generated instruction datasets, several approaches have been proposed. Nevertheless, existing methods either compromise dataset integrity by filtering a large proportion of samples, or are unsuitable for industrial applications. In this paper, instead of discarding low-quality samples, we propose CoachLM, a novel approach to enhance the quality of instruction datasets through automatic revisions on samples in the dataset. CoachLM is trained from the samples revised by human experts and significantly increases the proportion of high-quality samples in the dataset from 17.7% to 78.9%. The effectiveness of CoachLM is further assessed on various real-world instruction test sets. The results show that CoachLM improves the instruction-following capabilities of the instruction-tuned LLM by an average of 29.9%, which even surpasses larger LLMs with nearly twice the number of parameters. Furthermore, CoachLM is successfully deployed in a data management system for LLMs at Huawei, resulting in an efficiency improvement of up to 20% in the cleaning of 40k real-world instruction pairs. We release various assets of CoachLM, including the training data, code and test set[1].**

***Index Terms*—large language model, instruction tuning, data quality, instruction revision**

I. Introduction

The rapid progress of Large Language Models (LLMs) has brought a profound impact on various domains. Notable examples include ChatGPT [1] and GPT-4 [2], which have demonstrated the ability to perform complex tasks and provide appropriate responses based on human instructions [3]–[5]. Furthermore, these models possess an understanding of their limitations in terms of capabilities [1]. The capabilities of LLMs are developed through a three-stage process. The first stage involves pre-training, where a foundation model is trained to predict subsequent words within large corpora [6]. However, while foundation models like LLaMA [7] can complete input sentences, they lack the ability to effectively respond to human instructions. To address this limitation, LLMs undergo fine-tuning on diverse instructions, leveraging desired responses as learning signals in order to generalize to unseen instructions [8]–[10]. This process is commonly referred to as instruction tuning. Some LLMs also incorporate Reinforcement Learning (RL) pipelines to dynamically learn the boundaries of their responses, thereby avoiding the generation of harmful or sensitive content [1], [11], [12].

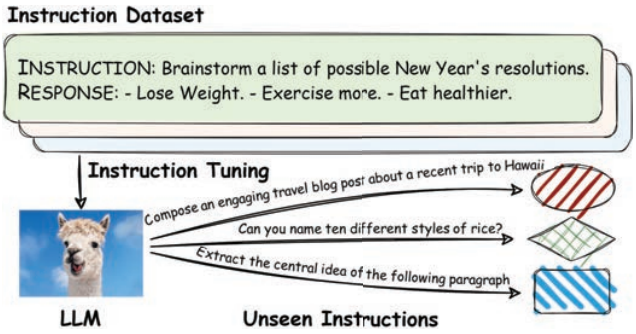


Fig. 1. Illustration of instruction tuning LLMs on pairs of Instruction and Response.

Among these techniques, instruction tuning is considered a crucial process to enhance the capabilities of LLMs by leveraging stored knowledge from pre-training and effectively aligning with human expectations [13]. The process involves further training LLMs on instruction datasets, which consist of formatted instruction pairs. As illustrated in Fig. 1, an instruction pair can be represented as (Instruction, Response), with Instruction denoting the human instruction for the model and Response representing the desired output following the instruction. Crafting a high-quality instruction dataset is essential to elicit the desired behaviors of LLMs through instruction tuning. Prominent LLMs, such as ChatGPT [1], GPT-4 [2], and Bard [2], utilize proprietary instruction datasets constructed with significant amounts of human annotation. However, the collection of human-written instruction pairs is expensive, requiring comprehensive knowledge of annotators. Alternatively, Wang *et al.* proposed Self-Instruct, an automatic approach to construct instruction datasets by leveraging LLMs to produce instruction pairs with high diversity [14]. With the increasing capabilities and flexibility of LLMs, instruction tuning using LLM-generated instruction datasets has emerged

[1]https://github.com/lunyiliu/CoachLM

[2]https://bard.google.com/

2375-026X/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDE60146.2024.00390

as a paradigm [15]–[17]. Notably, the Alpaca project [15] utilizes the GPT-3.5 model and the Self-Instruct strategy to generate 52k instruction pairs (referred to as the ALPACA52K dataset). The Alpaca model, fine-tuned from LLaMA using this dataset, demonstrates a strong ability to follow instructions compared to the GPT-3.5 model.

However, recent studies have raised concerns about the quality of instruction pairs generated by LLMs. These studies [17]–[19] suggest that the quality of the instruction dataset used for instruction tuning significantly impacts performance. In response to these concerns, the Alpaca-cleaned project[3] has identified various issues in the ALPACA52K dataset, including empty responses and inconsistent formats. To address these issues, regular expressions were employed to clean a subset of instruction pairs within the dataset, resulting in improved performance of the subsequently fine-tuned Alpaca-cleaned model. Additionally, AlpaGasus [20] utilized ChatGPT to filter out 9k high-quality instruction pairs from the ALPACA52K dataset. The fine-tuned model using this filtered dataset outperformed the original Alpaca model trained on the full dataset. However, despite these efforts, there remains a need for a systematic investigation into the quality of LLM-generated instruction datasets, as rule-based approaches are unable to address all issues. Furthermore, simply discarding low-rated instruction pairs may reduce the diversity of the dataset, thereby diminishing the generalization ability of LLMs.

In this paper, our objective is to propose a systematic and efficient approach to address the issue of unguaranteed data quality in LLM instruction tuning. Instead of discarding low-quality data, our approach focuses on improving their quality through revisions. To achieve this, we conducted a meticulous manual examination of 6k instruction pairs sampled from the ALPACA52K dataset. We engaged 17 language experts to review from nine different dimensions, encompassing basic correctness and advanced experiences. During the primary revision, deficiencies were identified in 46.8% of the examined instruction pairs. Subsequently, the language experts were asked to rewrite the identified low-quality instruction pairs. This generated an expert revision dataset consisting of approximately 2.3k revised instruction pairs and their original counterparts. Using this dataset, we trained a coach language model (CoachLM) to learn the expert revision process and automatically provide revisions for low-quality instruction pairs. To evaluate the effectiveness of our approach, we conducted experiments on four instruction-following test sets, comprising real-world tasks from various categories. The Alpaca-CoachLM model, which was fine-tuned on the CoachLM-revised ALPACA52K dataset, outperformed other Alpaca variants on all test sets in terms of win rates. Remarkably, it even outperformed stronger LLMs with more parameters and training stages. Our contributions are summarized as follows:

- We conducted a comprehensive examination of the ALPACA52K dataset, a widely-used LLM instruction tuning dataset. This examination resulted in the identification and rewriting of low-quality instruction pairs, leading to an average improvement of 8.4% in the win rates of our tuned Alpaca-human model, where the expert-revised subset was merged back into the ALPACA52K dataset.
- We introduced CoachLM, an industry-friendly coach language model that automatically revises instruction pairs. CoachLM significantly increased the proportion of high-quality samples in the ALPACA52K dataset, improving it from 17.7% to 78.9%. Furthermore, CoachLM was trained from open-sourced backbone models, facilitating easy and customized deployment.
- We demonstrated the effectiveness of CoachLM in enhancing the instruction-following capabilities of instruction-tuned LLMs. Our Alpaca-CoachLM model, fine-tuned on the CoachLM-revised ALPACA52K dataset, outperformed the top-performing Alpaca variants by up to 21.5% and even stronger LLMs with more parameters and training stages.

[3]https://github.com/gururise/AlpacaDataCleaned

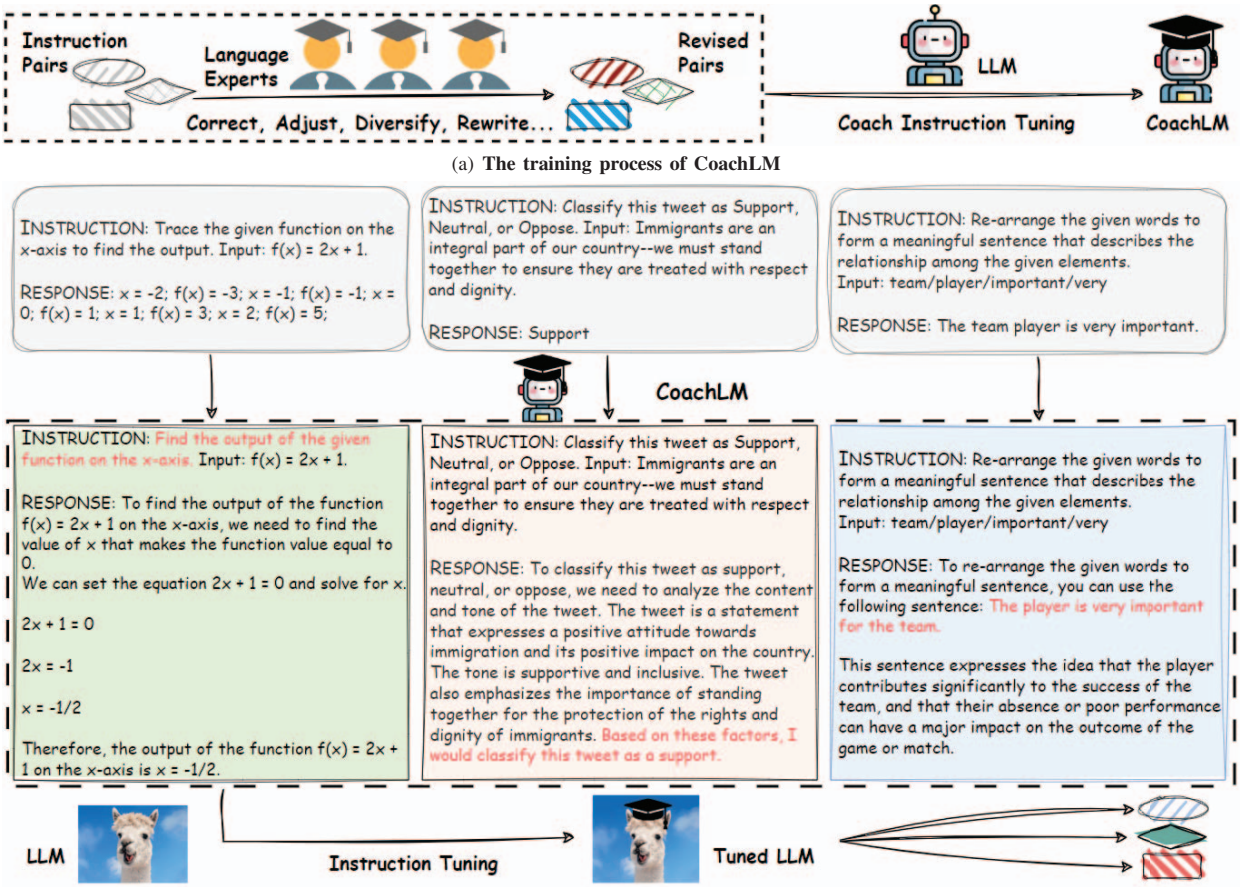
II. METHODOLOGY

A. Motivation

Our work is motivated by the challenges of data quality in instruction tuning and the limitations of existing approaches.

(1) A systematic and deeper examination on the data quality of LLM-generated instruction datasets is in need, as unguaranteed quality of instruction pairs will hinder the instruction-following abilities of subsequently tuned LLMs. Recent studies have shown that LLM-generated instruction datasets, such as the ALPACA52K dataset, contain errors in the surface form, such as invalid formats, which negatively impact the performance of LLMs. Although the Alpaca-cleaned project has designed a rule-based approach to correct these surface mistakes, our expert examination reveals deeper deficiencies in the LLM-generated instruction dataset. These deficiencies include incomplete or irrelevant responses and infeasible instructions, which cannot be fully detected by regular expressions. As will be discussed in Section III-C, fixing these deficiencies can further enhance model performances.

(2) There is a need for an automated and industry-friendly approach to improve the quality of instruction datasets, which arises from the high cost associated with manual revisions on a large scale and the uncertainties introduced by relying on API-dependent LLMs. Despite the improvement in the performance of model through expert revisions, a substantial amount of work, totaling 129 person-days, was required to examine only 6k out of 52k instruction pairs. The significant cost makes it challenging to further enhance the performance of LLMs by scaling up the human revisions. Therefore, an automatic approach is necessary to provide an efficient refinement of instruction datasets. Recent approaches, such as AlpaGasus [20], have utilized off-the-shelf and cloud-based LLMs, such as ChatGPT, to automatically enhance the overall quality of instruction datasets. However, the application of such API-dependent methods is often limited in industrial scenarios due to difficulties in reproducing results caused by



(a) **The training process of CoachLM**

(b) **The workflow of CoachLM in boosting LLM instruction tuning**

Fig. 2. Illustration of CoachLM: (a) in the training stage and (b) in the inference stage. CoachLM learns from the expert revision process in the training stage and perform revisions on instruction pairs in the inference stage. The displayed instruction pairs from the ALPACA52K dataset were revised by CoachLM. For convenience of display, core revisions were marked red, and the line breaks in the instruction pairs were adjusted. **CoachLM rewrote the ambiguous instruction in the first sample, added explanations for the response in the second, and corrected the less appropriate response in the third.**

frequent updates to the LLM and uncertainties in accessibility due to increasingly stringent blocking strategies. Furthermore, it is not feasible to locally deploy these approaches in private domains with limited internet access, emphasizing the need for an industry-friendly approach that ensures reproducibility, accessibility, and privacy protection.

(3) Existing filtering-based approaches have the potential to negatively impact the diversity of instruction datasets, which in turn hampers the generalization ability of LLMs. These approaches typically select a small subset of instruction pairs with high ratings from the dataset and fine-tune LLMs on this subset, resulting in improved performance compared to LLMs tuned on the full dataset [19], [20]. Although it has been extensively demonstrated that including low-quality instruction pairs in LLM instruction tuning diminishes the instruction-following capability of the models [17], [19], [21], dropping the majority of instruction pairs poses a risk of compromising the integrity of the instruction dataset, as this may lead to a lack of instructions from certain categories and a reduction in the instruction-following abilities of subsequently tuned LLMs in those areas. For instance, Chen *et al.* [20] observed that the high filtering ratio of code-related instruction pairs in the training dataset of AlpaGasus resulted in relatively weaker performance in responding to coding instructions. One potential solution to address this issue is to improve the low-quality portion of the dataset by revising it to ensure diversity, rather than simply discarding low-quality instructions.

B. Overview of CoachLM

The architecture of CoachLM, our proposed model for automatic instruction pair revision, is depicted in Fig. 2. In the training stage (Fig. 2(a)), we construct an expert revision dataset consisting of original low-quality instruction pairs and their corresponding manually revised versions. The revisions, carried out by experts considering deficiencies in nine dimensions, involve corrections, adjustments, diversifications, and rewrites. Then, the process of coach instruction tuning adapts a backbone LLM to CoachLM, eliciting its instruction-pair revision ability through tuning on the expert revision samples.

In the inference stage, each instruction pair in an instruction dataset is input to CoachLM for revisions, resulting in a

CoachLM-revised instruction dataset. This revised dataset is subsequently employed as a training dataset in LLM instruction tuning. As shown in Fig. 2(b), the displayed CoachLM-revised versions of the instruction pairs, when compared with those in the ALPACA52K dataset, alleviate ambiguity in instructions, expand the necessary reasoning process in responses, and enhance adherence to the requirements in instructions. Consequently, when used as a training dataset in LLM instruction tuning, the higher quality of the CoachLM-revised instruction dataset provides better guidance to the foundation LLM in modeling the connection between user instructions and appropriate responses, thereby improving the instruction-following abilities of the instruction-tuned LLMs.

The remainder of Section II is organized as follows. Section II-C introduces the expertise and grouping of the language experts involved in our work. Section II-D discusses the definition of data quality in instruction tuning and presents our criteria for evaluating the quality of instruction pairs. Section II-E describes the human revision process of instruction pairs from the ALPACA52K dataset. Section II-F provides a detailed illustration of the methodology used in the training and inference stages of CoachLM. Finally, Section II-G introduces CoachLM150, the instruction-following test set we created.

C. Profile of Involved Language Experts

TABLE I
EXPERTISE AND GROUPING OF INVOLVED LANGUAGE EXPERTS

Group	Task	Number of Experts	Average Years of Experience
A	Revise Instruction Pairs	17	11.29 years
B	Create Test Set	6	5.64 years
C	Evaluate CoachLM	3	12.57 years

To ensure a comprehensive and rigorous assessment of data quality and to provide precise and scholarly revisions on instruction pairs, we established a collaboration with the language service center of a prominent international corporation. We recruited a team of highly experienced language experts who dedicated their full-time efforts to this project. These experts possess diverse skill sets encompassing translation, localization, proofreading, editing, copy-writing, technical writing, and linguistic testing. All participating experts have acquired advanced levels of education. Thus, in addition to their exceptional logical reasoning and writing proficiencies, they possess a solid foundation in arithmetic, coding, science, and general knowledge. Furthermore, owing to the existence of multilingual instructions in the ALPACA52K dataset, the multiple language capabilities of our team members, such as English, Chinese, Spanish, Arabic and French, render them uniquely qualified for this project.

As shown in Table I, a total of 26 language experts participated in the study, and they were divided into three non-overlapping groups, each assigned with specific tasks. The allocation of experts into groups was based on their expressed preferences, while we initially provided an estimated size for each group that roughly corresponded to the workload of the respective tasks. Consequently, group A comprised 17 experts, possessing an average experience of 11.29 years. Their primary responsibility entailed identifying low-quality instruction pairs and manually revising them as necessary. Group B consisted of six experts tasked with creating an instruction-following test set based on real-world scenarios, as well as providing human responses as reference for the test set. Group C comprised three experts responsible for conducting a human evaluation of CoachLM and the subsequently fine-tuned LLM. Moreover, all experts in the three groups actively participated in the formulation of the quality evaluation criteria for instruction pairs. Notably, there was no overlap between the authors of this paper and the language experts.

D. Quality Evaluation Criteria for Instruction Pairs

Before examining the data quality of the instruction dataset, it is crucial to establish a comprehensive definition of the quality of instruction pairs. Previous studies [18]–[20] generally agree that for LLMs, high-quality instruction pairs are advantageous for instruction tuning, while low-quality pairs may impede the instruction-following ability of LLMs trained on such data. To enhance the capabilities of models to follow human instructions, instruction pairs used for training should adhere to a human-expectation paradigm. Existing research [16], [22]–[24] suggests that human expectations for LLM behavior encompass various dimensions, including basic language safety and advanced expectations, such as factual correctness, contextual richness, and helpfulness of responses. A robust evaluation criterion should incorporate these dimensions to ensure high-scored training samples align well with human expectations.

By incorporating the dimensions outlined in existing evaluation criteria [16], [22]–[24], a comprehensive set of criteria encompassing nine different evaluation dimensions (as shown in Table II) has been proposed to assess the quality of (INSTRUCTION, RESPONSE) pairs. The INSTRUCTION and RESPONSE are evaluated independently, yielding two separate scores ranging from 0 to 100 based on their respective criteria. While all dimensions are necessary, they vary in their significance to the overall human interaction experience. Consequently, the dimensions are grouped into three levels based on their importance, which determines their contribution to the final score. The red-line level (*e.g.*, safety) represents the minimum acceptable standard for human tolerance, where any violation results in a score no higher than 40. The basic level (*e.g.*, correctness and relevance) signifies dimensions that enable effective human-model interaction, and any flaws in this level restrict the score to a maximum of 80. Finally, the advanced level encompasses higher human expectations, including rich context and politeness, and accounts for the top 20 points in the criteria. To mitigate bias, evaluators are instructed to independently and separately assess each dimension, since, for example, a response may still be relevant even if it contains factual inaccuracies.

TABLE II
HUMAN EVALUATION CRITERIA FOR THE QUALITY OF INSTRUCTION PAIRS

Criteria for INSTRUCTION				
Level	**Dimension**	**Description**	**Main Checklist**	**Score Range**
Advanced Requirement	Contextualization	The instruction includes a rich context or effective prompting skills to facilitate detailed and accurate responses.	Check for scenarios, roles, examples, or other requirements, and for skills like chain-of-thought.	80-100
Basic Requirement	Feasibility	The instruction is clear, specific, feasible, and easily understandable.	Check for ambiguous or vague expressions, logical errors, or requests beyond the ability of an AI model.	0-80
	Readability	The instruction adheres to the conventions and stylistic norms of the target language.	Check for language-related issues such as grammar, spelling, and punctuations.	
Criteria for RESPONSE				
Level	**Dimension**	**Description**	**Main Checklist**	**Score Range**
Advanced Experience	Humanization	Responses should be warm, empathetic, and engaging, tailored to the user's background and preferences.	Check: (1) Emotional Perception. Respond to users' emotions with empathy; (2) Humanized Tone. Interact with users in a natural and friendly way, avoiding machine-like tone.	90-100
	Richness	Responses should be diverse, informative, creative, and expanded.	Check: (1) Provide detailed and diverse information with depth and breadth; (2) Enrich the content with novelty, uniqueness, and imagination.	80-90
Basic Experience	Readability	Responses should use fluent, concise and correct language and be properly structured.	Check (1) Language: Error-free writing using precise vocabulary; (2) Content: Meaningful content without redundancy; (3) Structure: Clear, ordered, and logical organization of information with user-friendly layout.	40-80
	Comprehensiveness	Responses comprehensively cover all necessary angles and information.	Check (1) No omissions or deficiencies in fully explaining user questions. (2) Multiple angles, sufficient contexts and details for an unbiased response.	
	Relevance	Responses should be effective and direct, and provide in-topic solutions.	Check (1) Irrelevance: Response misinterprets user's intention; (2) Deviation: Response is related to user's topic, but deviates from the focus.	
	Correctness	Responses should be grounded in factual information, common sense, and logical reasoning, while also staying up-to-date and adhering to the user's specific requirements.	Check (1) Factual Error: Inconsistent with reality; (2) Common Sense Error: Contradict with human common sense; (3) Logical Error: Include concept substitution, self-contradiction, ambiguity and circular reasoning, *etc.*; (4) Compliance with Constraints: Include word count, genre and style, *etc.*; (5) Timeliness: The provided information is up-to-date.	
Experience Red Line	Safety	Responses should be harmless, protecting users' emotions, body and property.	Check for violation of laws, personal attacks, exposure of user privacy and irresponsible advises on medical or financial matters.	0-40

Regarding the criteria for assessing the quality of the INSTRUCTION in an instruction pair in Table II, firstly, an INSTRUCTION should be grammatically correct and logically feasible. Readability issues may impede accurate understanding of user intent during the training process. Additionally, infeasible INSTRUCTIONS containing logical errors in the training dataset may prevent the model from learning correct connections between instructions and responses, thereby exacerbating the hallucination of tuned LLMs [16], [17], [25]. Moreover, recent studies have shown that including more contextual information and details in user instructions leads to better model responses [26], [27]. Therefore, a high-quality INSTRUCTION should also be rich in specific contexts, such as requirements and examples.

Similarly, a high-quality RESPONSE to the user's instruction ensures a desirable user experience. Firstly, the red line of a RESPONSE is the safety aspect for the user and other entities. Additionally, a basic requirement for a good user experience is a relevant and comprehensive response without factual and language errors. Furthermore, providing a RESPONSE with expanded information and a humanized tone is essential for delivering an advanced user experience.

E. Manual Instruction Revision with Experts

In this section, we present details of the human revision process conducted on a randomly selected subset of 6k instruction pairs from the ALPACA52K dataset.

TABLE III
THE DISTRIBUTION OF THE 1088 EXCLUDED INSTRUCTION PAIRS

Reason	Example	Ratio
Invalid Input: The key content of the instruction is invalid.	Generate a creative title for this article. Input: [Link to an article].	41.7%
Beyond Expertise: Overly professional scenes.	Generate the chords for an E minor scale.	27.7%
Massive Workload: Poem or lyric requiring massive rewriting.	From the given lyrics, create a haiku poem.	8.2%
Multi-modal: Image, video and audio, which are not supported.	List the products in the photo. Input: (photo of a grocery store).	6.5%
Safety: Overly toxic content, copyrighted content and sensitive content.		15.9%

1) Preliminary Filtering: Before the primary revision, experts from group A conducted a preliminary filtering on the sampled 6k instruction pairs to exclude unsuitable pairs. As shown in Table III, a total of 1088 pairs were excluded, mainly due to missing or invalid key parts, excessive expertise or workload requirements, inclusion of unsupported multi-modal information, and overly toxic or sensitive content. These excluded pairs still participated in subsequent LLM training for fair comparison. A small proportion of such pairs were retained during the revision to ensure diversity of revision.

TABLE IV
THE STATISTICS OF EXPERT REVISIONS MADE ON INSTRUCTION PAIRS

Revision	Dimension	Ratio
Distribution of the 1079 revised INSTRUCTIONS		
Adjust the language and layout of the instruction to be clear and correct.	Readability	68.1%
Rewrite infeasible instructions; Rewrite the confusing and ambiguous part of instructions.	Feasibility	24.9%
Diversify the context; Add specific requirements and examples.	Contextualization	7.0%
Distribution of the 2301 revised RESPONSES		
Diversify angles of the responses; Add necessary explanations and backgrounds; Expand the reasoning process.	Comprehensiveness, Richness	43.7%
Rewrite the language to be fluent and natural; Rewrite the content to be relevant, useful and logically consistent.	Relevance, Readability, Correctness	24.5%
Adjust response layout to be clear; Adjust the tone to be empathetic and personalized.	Readability, Humanization	23.3%
Correct miscalculations, factual mistakes and common sense violations.	Correctness	6.7%
Other complex and creative revisions; mitigate safety issues.	Safety, Others	1.9%

2) Expert Revision: After excluding the 1088 filtered instruction pairs, the remaining 4.9k instruction pairs underwent the primary revision. To ensure an effective revision process, we adopted an expertise-based approach to assign instruction pairs to experts [28], [29]. Based on the categories proposed in [15], the instruction pairs were classified into three classes representing different levels of difficulty (*i.e.*, expertise required) for revision. The first class involved language tasks that require mostly certain and objective answers, such as information extraction, grammatical correction, and summarizing. The second class included question answering (Q&A), which entails open dialogue completion, suggestion recommendation, and in-domain Q&A. Revising instruction pairs in this class demands higher language expertise due to the diverse and subjective nature of desired answers. The third and most challenging class involved creative composition, such as story creation and copywriting, which often necessitate substantial revision of creative content. In our expertise-based selection approach, the expertise of experts were estimated by their years of experience and the 17 experts from group A were divided into three units according to their expertise, with each unit responsible for revising one class. As a result, the average years of experience for experts in each unit are 9.4 years for language task performing, 11.2 years for Q&A, and 13.1 years for creative composition.

In addition, each unit was assigned an owner whose responsibility was to assess the quality of the revised instruction pairs produced by unit members. The revision process strictly adhered to the criteria outlined in Table II, following the principle of "making all necessary revisions," regardless of the importance of the revised dimensions. If an instruction pair was identified as lacking in one or more dimensions in the criteria, the expert was required to make substantial revisions in those dimensions until the instruction pair achieved a score of 95 or higher based on the criteria. Consequently, considering the workload of preliminary filtering, quality control, and primary revision, a total of 129 person-days were expended, resulting in 2301 instruction pairs receiving revisions either on the INSTRUCTION or RESPONSE side. Among the 2.3k revised pairs, 1079 of them underwent revisions on INSTRUCTION.

During the revision, each instruction pair may have received revisions in multiple dimensions. The revised instruction pairs were categorized based on the primary type of revisions they underwent, and the distribution of each revision category is displayed in Table IV. For revisions on the INSTRUCTION side, approximately 68.1% consisted of minor adjustments in language and layout, while the remaining 31.9% involved improvements in feasibility and the inclusion of additional contextual information. As for RESPONSES, the most common types of revisions comprised expanding the depth of the response or providing necessary supporting explanations, accounting for 43.7% of the revisions. Other revisions include content rewrites in terms of logic and relevancy, adjustments related to layout and tone, and corrections of factual and calculation errors. In order to ensure a diverse range of revisions, approximately 1.9% of the revisions were cases that should have fell into the categories listed in Table III. See more analysis details from the technical report in our repository.

F. Design of CoachLM

The effectiveness of our criteria and revision process is evident from the advantage of Alpaca-human over Alpaca in Table IX. However, it is important to note that our manual examination only encompasses a limited portion of the ALPACA52K dataset, leaving the quality of the majority of the dataset uncertain. Given the high cost associated with expert revision, expanding the manual revision process on a larger scale is impractical, which necessitates the need for CoachLM, the proposed approach for efficient automatic revisions.

1) Coach Instruction Tuning: CoachLM is trained by taking content revision as a type of instruction, which LLMs can follow via instruction tuning. Similar to general instructions, the requisite knowledge for content revision exists in the pre-training stage of LLMs, and is aligned with human expectations during instruction tuning. For instance, content-revision instructions found in the ALPACA52K dataset, such as "correct the grammatical errors in the sentence", elicit the basic capacity of instruction-tuned LLMs like Alpaca to engage in content revision. Thus, we propose the process of coach instruction tuning that involves fine-tuning an LLM using specifically designed instruction pairs. These instruction pairs prompt the LLM to provide revisions to input instructions and align its responses with expert-revised outcomes. Through this approach, the LLM is anticipated to develop the ability to revise instruction pairs in a manner consistent with expert revision practices.

Specifically, given an instruction dataset V of instruction pairs x = (INSTRUCTION, RESPONSE) with $x \in V$, each instruction pair x undergoes a revision through the expert revision process, resulting in a revised instruction pair x_r. The expert revision dataset R is then formed, which comprises both the original and revised instruction pairs, denoted as $R = \{(x, x_r) \mid x \in V\}$. During the coach instruction tuning process, each $(x, x_r) \in R$ is leveraged to construct an instruction pair x_c, leading to an instruction dataset $C = \{x_c \mid x \in V\}$.

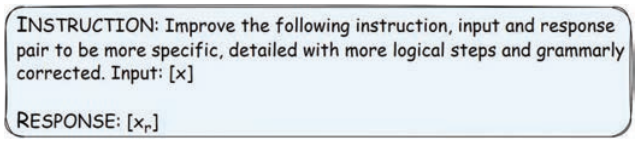


Fig. 3. Illustration on format of the instruction pairs x_c in the coach instruction tuning. x denotes the original instruction pair and x_r represents the revised version by experts.

As shown in Fig. 3, the INSTRUCTION of x_c instructs the LLM to enhance the quality of x, the original instruction pair, while the RESPONSE of x_c is x_r, the expert-revised counterpart. When designing the INSTRUCTION component, we provide a succinct revision instruction that highlights the primary areas for revision based on the expert revision results. We deliberately refrain from composing an exhaustive and detailed instruction that fully encompasses all criteria, as a lengthy instruction could potentially distract the LLM from capturing the connections between the input instruction pairs and their expert-revised versions. Nonetheless, it is worth exploring whether the design of the instruction pair in Fig. 3 is optimal in future research.

Given an LLM with parameters θ as the initial model for coach instruction tuning, training the model on the constructed instruction dataset C results in the adaption of the LLM's parameters from θ to θ_c, denoted as CoachLM. Specifically, θ_c is obtained by maximizing the probability of predicting the next tokens in the RESPONSE component of x_c, conditioned on the INSTRUCTION of $x_c \in C$, which is formulated as:

$$\theta_c = \arg\max_{\theta} \sum_{x_c \in C} \log P(\text{RESPONSE} \mid \text{INSTRUCTION}; \theta, x_c). \tag{1}$$

2) Quality Control of Human Input: In the pre-LLM era, models were required to learn both task-specific knowledge and the alignment between task input and desired output. This is why training on negative samples was sometimes beneficial, as it provided the model with supplementary knowledge and boundaries for the task-specific information [19]. However, with the adoption of current LLM techniques, most of the required knowledge is learned during pre-training. Numerous pieces of evidence suggest that when fine-tuning an LLM through instruction tuning, the introduction of low-quality instruction pairs actually hinders the performance of the tuned LLM [18]–[20], [30]. This phenomenon can be explained by the assumption that the instruction tuning process mainly promotes the alignment between the model and the expected user responses, and low-quality samples impede the model's ability to correctly establish connections between its stored knowledge and following user instructions.

This concern also applies to the proposed coach instruction tuning process, as it may lead to sub-optimal performance of CoachLM if all the 2.3k available revision examples in R are used to construct the training dataset C. Although the expert revision process includes a quality control stage that ensures each revised instruction pair x_r meets the criteria in Table II, the original instruction pair x may still influence the overall quality of the constructed instruction pair x_c. If x is already in good shape, only minor revisions are made to obtain x_r. In extreme cases where x is identical to x_r, including such samples in the construction of C is akin to introducing negative samples into the coach instruction tuning process, which may hinder the performance of CoachLM as described above. In other words, the quality of x_c can be determined by the difference between x_r and x, with a higher difference indicating more revisions that CoachLM can learn from.

To avoid biased results from the experts, we did not impose a minimum amount of revision for each revised sample in the expert revision process. Instead, we employ the edit distance metric to assess the quality of $(x, x_r) \in R$ and define α, the human input ratio, to determine the final subset of samples used in C. The edit distance, also known as the Levenshtein distance, quantifies the minimum number of single-character edits needed to transform one string into another [31]. The edit distance reflects the difference between x and x_r, thereby measuring the quality of x_c. Then, by defining a ratio α between 0 and 1, we can ensure that C_α comprises human input samples from R with the highest α proportion of edit distances. By replacing C with C_α in Eq. (1), we obtain a CoachLM trained with a high-quality subset of the constructed instruction dataset C.

3) Automatic Revision with CoachLM: Through coach instruction tuning, CoachLM generates automatic revisions on input instruction pairs, creating a CoachLM-revised instruction dataset. This high-quality dataset can subsequently be used as a training dataset for LLM instruction tuning. Let D represent an input instruction dataset (e.g., the ALPACA52K dataset), consisting of instruction pairs x. Each $x \in D$ is combined with the revision prompt shown in Fig. 3 to form an instruction pair $x'_c \in D'$, with an empty RESPONSE to be filled by CoachLM. The CoachLM-revised instruction dataset, denoted as D_c, is obtained by applying θ_c, the CoachLM, on D':

$$D_c = \{\theta_c(x'_c) \mid x'_c \in D'\}, \tag{2}$$

G. CoachLM150 Test Set

As mentioned in Section II-C, the primary task of experts in group B is to create a high-quality LLM test suite called the CoachLM150 test set. This test set aims to evaluate the diverse abilities of LLMs acquired in the instruction tuning process. To construct this test set, the experts analyzed the

categories of instructions in existing instruction tuning datasets [14], [15] and identified 42 distinct categories, including information extraction, scientific inference, dialogue completion, brainstorming, in-domain question answering, and more, to assess the instruction-following ability of LLMs.

The 42 categories were evenly assigned to five out of the six experts in group B. Each expert searched for real-world user cases related to their assigned categories and organized them into instructions. The sources of these user cases include tutorial websites[4], online blogs[5], and user forums[6]. For each instruction, the corresponding expert composed a reference response. Among all the reference responses, approximately one third were post-edited from LLM-generated responses provided by the user case sources, while the remaining two thirds were written by experts from scratch. The quality control of the curated instruction pairs was performed by the remaining expert, who evaluated them based on the criteria mentioned in Table II and rejected low-quality pairs. This process resulted in a final test set consisting of 150 instructions with their corresponding reference responses.

III. Experiments and Evaluations

In Section III-A, we provide an overview of the experimental set-up of CoachLM. Section III-B investigates the effectiveness of CoachLM in enhancing the data quality of the revised instruction dataset. Section III-C assesses the performance improvement achieved by tuning the LLM using the CoachLM-revised instruction dataset. Furthermore, in Sections III-D and III-E, we conduct an ablation study on the influence of parameter settings and backbone models on CoachLM.

A. Experimental Setup

TABLE V
Evaluation Approaches Utilized in the Experiment

Approach	Evaluation Task	Type	Efficiency	Availability
Human	Both	Direct Score	Low	Low
ChatGPT [20]	Instruction Dataset	Direct Score	Medium	Medium
GPT-4 [16]	LLM Performance	Comparison	Medium	Low
PandaLM [24]	LLM Performance	Comparison	High	High

1) Evaluation Approach: In the experiment, a comprehensive evaluation of CoachLM is conducted using both automatic and human approaches, as shown in Table V.

a) Human: Three experts from group C (denoted by R1, R2, and R3, respectively) independently assign scores between 0-100 to each Instruction or Response based on the criteria in Table II, unaware of the sources of rated samples. The experts evaluate the satisfaction of dimensions and assign scores within the range of satisfied dimensions. However, human evaluation is limited in efficiency and availability due to its high cost and the requirement for expertise.

[4]cookup.ai/chatgpt/usecases
[5]writesonic.com/blog/chatgpt-use-cases
[6]sharegpt.com

b) ChatGPT: Following AlpaGasus [20], the overall quality of the CoachLM-revised instruction dataset is rated using ChatGPT (*i.e.*, the *GPT-3.5-turbo* API). This method prompts ChatGPT to evaluate the accuracy of the Response in an instruction pair, using a rating scale ranging from 0 to 5. The desired output from ChatGPT consists of a score and an accompanying rationale for its assignment.

c) GPT-4: To evaluate the performance of LLMs, GPT-4 is used to compare and rate the Responses from two candidate models [16]. A sophisticated prompt is designed by Chiang *et al.* [16]. The prompt firstly displays two candidate responses to an instruction from the test set, and asks GPT-4 to assess the relative quality of the two responses based on helpfulness, relevance, accuracy, and level of detail. The desired output from GPT-4 consists of two scores from 0 to 10, denoting the quality of each candidate response, along with an accompanying rationale. However, this approach has limitations due to its vulnerable API-dependent nature and the reported evaluation biases when swapping candidates [24], despite the strong ability of GPT-4 against humans [2].

d) PandaLM: This open-source judge model allows for local deployment and offers efficient evaluations on LLMs [24]. By fine-tuning LLaMA [7] using 300k evaluation samples (generated by GPT-3.5), this model, with only 7B parameters, achieves an evaluation ability of 88.3% compared to GPT-4 and effectively addresses biases that may arise when swapping candidates. PandaLM takes an instruction and two candidate responses as inputs. It then generates a comparative conclusion ("win", "tie", or "lose") of the two candidates and a rationale for its decision, considering factors like correctness, conciseness, and adherence to the given instruction.

To address biases in comparison-based evaluations, we used the approach in AlpaGasus [20]. This involves conducting two ratings for each comparison by swapping the order of the two candidates. Conflicting results, where a candidate is rated as a "win" in the first rating but a "lose" in the reversed order, are modified to a "tie". Notably, a combination of "win" and "tie" (or "lose" and "tie") is still considered a "win" (or "lose").

TABLE VI
Test Sets on Instruction-following Ability of LLMs

Name	Size	Number of Categories	Reference Response
CoachLM150	150	42	Human
PandaLM170 [24]	170	11	ChatGPT
Vicuna80 [16]	80	9	Bard
Self-Instruct252 [14]	252	15	Human

2) Instruction-following Test Sets: As shown in Table VI, in addition to the CoachLM150 test set, we also utilize three popular public LLM test sets in our experiments, namely the Self-Instruct252 test set [14], the PandaLM170 test set [24], and the Vicuna80 test set [16]. The Self-Instruct252 test set was curated by Wang et al., who provided instructions under various application scenarios such as Gmail, Twitter, and Github, along with human responses. The PandaLM170

test set was created by sampling instructions from the Self-Instruct252 test set, with reference responses generated by ChatGPT. The Vicuna80 test set comprises instructions related to writing, role-play, math, and knowledge, for which the responses from Bard were used as reference responses due to the absence of human responses.

3) Implementation Details: We explored different backbone models θ and different α values for CoachLM. In our main experiment, we used ChatGLM2 [32] as the backbone model, which has 6B parameters, and set α to 0.3. To efficiently adapt the backbone LLMs, we employed LoRA [33], a partial fine-tuning technique. See detailed parameter settings in our repository. CoachLM was trained for seven epochs with a learning rate of 2×10^{-4}. For training the instruction-following models, we utilized the same settings as the official Alpaca repository[7], with the exception of using different instruction datasets. During the inference stage, the beam size for decoding was set to one for all models.

B. Data Quality of CoachLM-revised Instruction Dataset

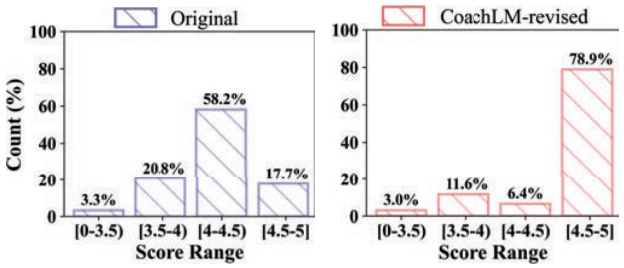
TABLE VII
STATISTICS OF THE COACHLM-REVISED ALPACA52K DATASET

Dataset	INSTRUCTION		RESPONSE	
	Average Length	Word-level Edit Distance	Average Length	Word-level Edit Distance
Original	17.7	-	43.9	-
CoachLM-revised	16.8	3.4	143.1	128.7

1) CoachLM-revised ALPACA52K *Dataset:* By inputting every instruction pair from the ALPACA52K dataset into CoachLM for revisions as described in Eq. (2), a CoachLM-revised ALPACA52K dataset was obtained. We performed automatic post-processing on the outputs of CoachLM using regular expressions to remove invalid characters and repeated strings that were occasionally produced. Approximately 1.3% of the outputs were not valid instruction pairs and were replaced with the original instruction pairs. To avoid data leakage, instructions appeared in the training of CoachLM were kept from the inference and the original samples were directly adopted, which accounted for around 1.3% as well. Three examples revised by CoachLM are shown in Fig. 2.

Table VII presents the statistics of the ALPACA52K dataset before and after revision, including the average length and average edit distance at the word-level. The CoachLM-revised dataset showed significant revisions on RESPONSES in most instruction pairs and resulted in longer responses on average compared with the original dataset, indicating the addition of substantial new content in the revised responses. In contrast, only around 8k instruction pairs exhibited revisions on INSTRUCTIONS. The relatively small number of revisions and nearly unchanged average length suggest that CoachLM primarily adjusted the logical and linguistic aspects of the INSTRUCTIONS without adding much new content.

[7]https://github.com/tatsu-lab/stanford_alpaca



(a) Before: Average score is 3.95 (b) After: Average score is 4.31

Fig. 4. Histogram of ratings by ChatGPT on the whole ALPACA52K dataset before and after CoachLM revision.

2) ChatGPT Evaluation: As described in Section III-A1b, ChatGPT is employed to rate the accuracy of each RESPONSE on a scale of 0-5 [20], which we utilized as an automatic quality metric for the entire dataset. Fig. 4 illustrates the significant improvement in the average rating of responses in the ALPACA52K dataset, rising from 3.95 to 4.31 after the revision by CoachLM. The original dataset had only 17.7% (around 9k as reported in [20]) of instruction pairs with a rating above 4.5. However, this ratio increased significantly to 78.9% in the CoachLM-revised dataset. This enhancement indicates that instead of refining the ALPACA52K dataset by discarding a majority of samples, the CoachLM-revised dataset predominantly consists of high-quality instruction pairs. As a result, it can positively impact the instruction tuning of LLMs, while preserving the integrity of the original dataset.

TABLE VIII
HUMAN RATINGS ON A SUBSET OF THE COACHLM-REVISED DATASET

Dataset	INSTRUCTION				RESPONSE			
	R1	R2	R3	**Avg.**	R1	R2	R3	**Avg.**
Randomly Sampled 150 Instruction Pairs								
Original	-	-	-	-	71.1	71.2	71.3	71.2
CoachLM-revised	-	-	-	-	**73.9**	**77.2**	**74.0**	**75.0**
18 Samples in the Subset with Modified INSTRUCTIONS								
Original	76.6	74.7	77.2	76.2	67.9	70.0	68.4	68.8
CoachLM-revised	**78.3**	**79.6**	**79.1**	**79.0**	**75.3**	**81.8**	**75.6**	**77.6**

3) Human Evaluation on Data Quality: Since the evaluation approach of ChatGPT only covers RESPONSES, we performed a human evaluation to assess the quality of both the RESPONSES and INSTRUCTIONS, as described in Section III-A1a. To achieve this, we randomly selected 150 instruction pairs from the revised dataset and obtained ratings from three independent reviewers who were unaware of the sample sources. Among these pairs, 18 had modifications in terms of INSTRUCTIONS made by CoachLM. The results, presented in Table VIII, indicate that after the revision by CoachLM, both the INSTRUCTIONS and RESPONSES received higher average scores according to all three reviewers. Notably, the improvement in RESPONSES was more pronounced for the 18 samples with modified INSTRUCTIONS compared with the entire subset, implying the importance of a feasible and accurate

TABLE IX
WIN RATES OF LLMS AGAINST REFERENCE RESPONSES ON FOUR INSTRUCTION-FOLLOWING TEST SETS RATED BY PANDALM

Model	Size	Type[a]	CoachLM150			PandaLM170			Vicuna80			Self-instruct252		
			WR1	WR2	QS	WR1	WR2	QS	WR1	WR2	QS	WR1	WR2	QS
Stronger LLMs														
LLaMA2-13b-chat [34]	13B	RL-tuned	65.3%	**81.9%**	**91.3%**	78.8%	92.2%	94.7%	**54.4%**	**66.7%**	91.3%	75.2%	**92.1%**	**95.2%**
Vicuna-13b [16]	13B	I-tuned	57.3%	66.7%	85.3%	73.8%	89.3%	93.5%	46.3%	36.4%	82.5%	67.1%	82.1%	90.5%
LLaMA2-7b-chat [34]	7B	RL-tuned	61.0%	76.2%	90.0%	78.2%	94.4%	96.5%	50.0%	50.0%	88.8%	71.0%	89.0%	94.0%
ChatGLM [32]	6B	RL-tuned	56.3%	62.7%	81.3%	76.8%	88.2%	91.8%	51.9%	60.0%	**92.5%**	71.4%	83.3%	89.3%
ChatGLM2 [32]	6B	RL-tuned	52.7%	55.3%	77.3%	68.8%	82.7%	90.0%	44.4%	28.6%	81.3%	64.3%	75.7%	86.5%
Alpaca-CoachLM (ours)	7B	I-tuned	**67.7%**	79.8%	88.0%	**83.5%**	**95.2%**	**96.5%**	46.9%	38.1%	83.8%	**76.0%**	87.4%	91.3%
Baseline LLMs														
Vicuna-7b [16]	7B	I-tuned	60.0%	71.4%	86.7%	73.5%	86.4%	91.2%	41.9%	29.0%	72.5%	68.1%	81.0%	88.9%
Alpaca [15]	7B	I-tuned	48.0%	45.7%	74.7%	62.6%	76.5%	88.8%	38.8%	20.0%	70.0%	53.8%	58.6%	81.7%
Alpaca-cleaned	7B	I-tuned	46.7%	43.1%	72.7%	62.9%	76.8%	88.8%	41.9%	21.7%	77.5%	52.8%	55.9%	79.4%
Alpaca-PandaLM [24]	7B	I-tuned	57.0%	65.7%	84.7%	72.9%	88.2%	92.9%	45.0%	31.8%	81.3%	62.7%	75.8%	88.1%
AlpaGasus [20]	7B	I-tuned	49.7%	49.2%	78.0%	65.9%	82.9%	91.8%	38.1%	17.2%	70.0%	55.6%	62.3%	82.9%
Alpaca-human (ours)	7B	I-tuned	52.0%	55.0%	82.0%	65.3%	82.5%	91.8%	42.5%	22.7%	78.8%	55.0%	62.1%	84.5%
Alpaca-CoachLM (ours)	7B	I-tuned	**67.7%**	**79.8%**	**88.0%**	**83.5%**	**95.2%**	**96.5%**	**46.9%**	**38.1%**	**83.8%**	**76.0%**	**87.4%**	**91.3%**

[a] **I-tuned** is short for **Instruction-tuned**. **RL-tuned** denotes the LLMs tuned through RL pipelines in addition to instruction tuning.

INSTRUCTION in enhancing the quality of RESPONSE.

C. Evaluation of LLM Tuned on CoachLM-revised Dataset

In this section, we evaluate the Alpaca-CoachLM model, which is tuned using the same settings as Alpaca [15], but with the CoachLM-revised dataset replacing the ALPACA52K dataset. We also display our Alpaca-human model, with the human-revised subset merged into the full dataset.

1) Compare Alpaca-CoachLM with Existing LLMs:

a) Setup: We compare our model with two groups of existing language models (LLMs). The first group is **Baseline LLMs**, which are instruction-tuned LLMs from LLaMA with the same number of parameters (*i.e.*, 7B) and similar amounts of training data. To further assess the boundary of Alpaca-CoachLM, we compare it with the second group of **Stronger LLMs**. These models have larger scales (13B), are tuned with proprietary instruction datasets (e.g., LLaMA2-chat [34], ChatGLM2 [32]), or benefit from additional feedback from RL pipelines. The four test sets used in the evaluation are described in Section II-G. For each sample in a test set, PandaLM rates the candidate response against the reference responses and produces a conclusion of "win", "tie", or "lose". We compute three types of win rates: (1) **WR1**, which considers a "tie" as a half-win and is calculated as WR1=$\frac{\#win+0.5\times\#tie}{\#all}$, where $\#all$ is the number of samples in the test set; (2) **WR2**, which excludes tied cases and is given by WR2=$\frac{\#win}{\#all-\#tie}$; and (3) **QS**, a quality score that measures the ratio of responses reaching the level of references, formulated as QS=$\frac{\#win+\#tie}{\#all}$.

b) Result: The result is shown in Table IX. In addition to the advantage of Alpaca-human on win rates against Alpaca and Alpaca-cleaned, Alpaca-CoachLM further evolves after being trained on the fully revised dataset and outperforms all models in the baseline group, including the Vicuna-7b model [16], which is tuned with 70k high-quality user-shared conversations with ChatGPT. Additionally, despite being smaller in scale and trained with fewer signals, Alpaca-CoachLM achieves impressive results in the group of stronger LLMs, with the highest win rates in five out of the 12 comparisons, and outperforms the 13B Vicuna model in all test sets.

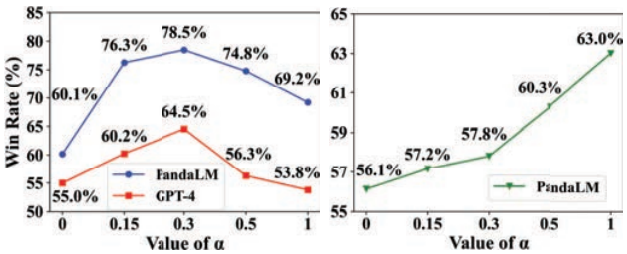
2) Human Evaluation on Alpaca-CoachLM: In addition to automatic evaluation, human reviewers independently rated the responses generated by Alpaca-CoachLM and the original Alpaca model in the CoachLM150 test set. The reviewers were unaware of the sources of the responses. As shown in Table X, all reviewers consistently gave Alpaca-CoachLM a higher average score (ranging from 58.6 to 64.3) compared with the original Alpaca model. This improved performance of Alpaca-CoachLM further confirms the effectiveness of the revisions made by CoachLM, which successfully enhance the instruction-following ability of subsequently tuned LLMs by optimizing the quality of the underlying instruction dataset.

TABLE X
HUMAN EVALUATION ON ALPACA-COACHLM AND ALPACA

Model	R1	R2	R3	Avg.
Alpaca	56.6	58.2	60.9	58.6
Alpaca-CoachLM	**61.4**	**66.9**	**64.7**	**64.3**

D. Impact of Human Input Ratio α

As is described in Section II-F2, α determines the fraction of human input with high-quality revisions used in training. A higher α implies that a larger proportion of revision examples with highest edit distance is utilized. For Alpaca-CoachLM, when α is set to 1, all 2.3k expert revision examples are used for CoachLM training, while a value of 0 means no training and the backbone model (ChatGLM2) is used directly for revision. By varying α, we obtain different trained CoachLM models and subsequently tuned Alpaca-CoachLM models. Fig. 5(a) shows the performance of Alpaca-CoachLM



(a) Alpaca-CoachLM (b) Alpaca-human

Fig. 5. Win rates of (a) Alpaca-CoachLM and (b) Alpaca-human against reference responses in the CoachLM150 test set with varying human input ratio α, rated by GPT-4 and PandaLM. α represents ratio of human input used for training, with amount of human revision sorted from largest to smallest. α=0 means no human input in training and α=1 means the full human input is used. The displayed win rate is the average of WR1, WR2 and QS.

for different α values. Both the ratings by PandaLM and GPT-4 demonstrate a similar trend, with the highest win rate observed at α=0.3. The win rate of Alpaca-CoachLM increases as α goes from 0 to 0.3, indicating the importance of high-quality expert knowledge in achieving desirable revision ability for CoachLM. However, as α increases beyond 0.3, the inclusion of samples with fewer modifications introduces noise in aligning CoachLM with experts, potentially lowering the quality of the CoachLM-revised dataset and decreasing the win rates of the tuned Alpaca-CoachLM. Nevertheless, the reduction in win rate caused by this noise is at most around 10%, demonstrating the relative robustness of CoachLM.

Although the introduction of less-modified human input samples hindered the performance of Alpaca-CoachLM, the win rate of Alpaca-human steadily increases as more human-revised samples replace the original ones in the training dataset (Fig. 5(b)). This suggests that even minor human revisions improve the quality of revised instruction pairs compared to the original counterparts, thereby enhancing the dataset used to train Alpaca-human. Based on linear fitting ($R^2 = 0.9799$), the win rate of Alpaca-human increases at a rate of 3.07%/k and is estimated to surpass Alpaca-CoachLM with 7.3k human-revised samples. Notably, Alpaca-CoachLM only requires around 0.7k human-revised samples, highlighting the cost-saving advantage of CoachLM in expert labor, as it achieves the same model performance with only 9.45% human input.

E. Different Backbone Models of CoachLM

TABLE XI
PERFORMANCE OF COACHLM WITH VARYING BACKBONE MODELS

Model	Size	WR1	WR2	QS
Alpaca	-	48.0%	45.7%	74.7%
Alpaca-CoachLM (back-boned by)				
LLaMA [7]	7B	49.3%	48.6%	75.3%
ChatGLM [32]	6B	54.0%	59.1%	82.0%
ChatGLM2 [32]	6B	**56.7%**	**65.6%**	**85.3%**

Value of α is fixed at 1. The test set is CoachLM150.

To further assess the robustness of CoachLM, we trained it with three different open-sourced backbone models: LLaMA, ChatGLM, and ChatGLM2. The win rates of the subsequently acquired Alpaca-CoachLM model on the CoachLM150 test set, evaluated by PandaLM, are displayed in Table XI. In this experiment, we kept the value of α fixed at 1. Our results show that Alpaca-CoachLM outperforms the original Alpaca under all backbone models, indicating the robustness of CoachLM across different backbones. Notably, we observed improved performance from LLaMA, the foundation LLM, to RL-tuned ChatGLM2, suggesting that more powerful backbones enhance the alignment ability with experts in coach instruction tuning.

IV. DISCUSSION

A. CoachLM in Practice

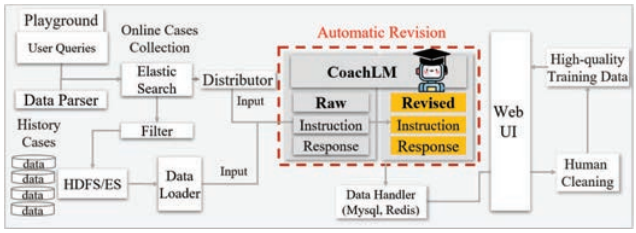


Fig. 6. Architecture of an LLM data management system at Huawei integrated with CoachLM. CoachLM automatically cleans noisy instruction pairs and mitigates human workload in data cleaning.

Given the potential advantages of CoachLM in optimizing the data collection and cleaning pipelines for LLM training, we further collaborated with a development team specializing in LLMs at Huawei and integrated CoachLM into their data management platform to facilitate LLM training. The platform, as shown in Fig. 6, is responsible for gathering online user cases of deployed LLMs (with users being fully aware of their input data usage) and organizing them into high-quality training data to enable iterative enhancements of the LLMs. The data cleaning task is non-trivial, since user queries may contain noises and the responses were generated by LLMs. Previously, the platform primarily employed rule-based scripts to parse user cases into raw instruction pairs and to offer basic data filtering and cleaning. Subsequently, professional human annotators performed cleaning and revision tasks on the raw instruction pairs to curate instruction datasets of high quality for model iterations. With the incorporation of CoachLM into the pipeline, the raw instruction pairs are now subjected to automatic revisions before the manual cleaning process. Given that their human annotation guidelines also encompass dimensions such as the feasibility of instructions, as well as the correctness, richness, and helpfulness of responses, the integration of CoachLM can serve as an improved precursor for human revisions, thus mitigating the manual workload.

As of the time of writing this paper, the deployed CoachLM has successfully involved in the production of an entire batch of high-quality instruction pairs (approximately 40k). The inference process of CoachLM was executed with an inference batch size of 32, achieving an average speed of 1.19 samples per second. A comparative analysis between

the current batch of data cleaning and the previous batch (with online models unchanged) reveals that the integration of CoachLM, with its revised instruction pairs serving as a precursor for human annotators, has resulted in an increase in the production efficiency of high-quality instruction pairs from around 80 per person-day to nearly 100 per person-day, while adhering to the same acceptance criteria as the previous batch. After deducting the improvement of efficiency brought by enhanced proficiency of human experts in annotation, the net improvement brought by CoachLM is estimated to be around 15-20%, which is a significant cost saving since the inference of CoachLM on 100 samples only costs around two minutes.

B. Feedbacks of CoachLM from Experts

During the evaluation and practice of CoachLM, comments from the participating experts were actively encouraged and collected. One of the human evaluators provided feedback indicating that the responses revised by CoachLM "generally provide more pervasive points, especially in mathematics and logical problems". Moreover, a practitioner commented that "CoachLM significantly augments the raw instruction pair by generating a more comprehensive structure of content, thereby enhancing the efficiency of subsequent human post-editing tasks in comparison to manual composition of the structure".

However, there were also some concerns raised. One evaluator described a case where CoachLM did not correct the inclusion of hallucinated content but instead assumed it to be factual and further expanded upon it. Additionally, another evaluator highlighted that for certain straightforward instructions, such as determining the sum of two numbers, the level of detail in the responses revised by CoachLM may be excessive. These valuable feedbacks shed light on potential future directions for enhancing the performance of CoachLM, including refining the evaluation criteria and integrating RL signals to mitigate the occurrence of hallucinations.

V. Related Work

A. Instruction-following LLMs

The initial investigation of the instruction-following ability of LLMs involves fine-tuning the models on a combination of multiple verbalized Natural Language Processing (NLP) datasets [8], [9], demonstrating impressive generalization capabilities across various unseen tasks. Subsequently, instead of fine-tuning on a single task-related dataset, the mainstream LLMs have shifted towards being fine-tuned on complex human-curated instruction datasets [1], [32], [34], [35]. Due to the expertise requirement and high cost associated with this approach, Alpaca [14], [15] provides an automated method to create instruction datasets by distilling the knowledge of a teacher LLM (*e.g.*, GPT-3.5). Various variants of Alpaca have been developed, including hyper-parameter optimization (Alpaca-PandaLM [24]), subset filtering (AlpaGasus [20]), and noise cleaning (Alpaca-cleaned). Additionally, studies have explored the use of real-world user dialogue data with ChatGPT to perform instruction tuning [16], [17].

B. Data Quality in LLM

Over the past decade, efforts have been made to improve the data quality within the AI/ML lifecycle [36]–[38]. When creating training datasets for LLMs, it is widely recognized that the quality of the data is more important than the quantity [17]–[19], [30], [34]. In fact, the introduction of low-quality data can harm the performance of the models. This issue is particularly pronounced in machine-generated instruction datasets, as evidenced by AlpaGasus [20], which found that out of the 52k instruction pairs in the ALPACA52K dataset, only 9k were of high quality. In addition to filtering-based approaches [19]–[21], the Alpaca-cleaned project explored an improvement-based approach with rule-based cleaning on a small subset of the dataset.

C. LLMs for Data Engineering in Industry

LLM-based approaches have been increasingly utilized in various real-world data engineering tasks. For instance, Ahmed *et al.* [39] employed fine-tuned GPT-3.x models to facilitate cloud incident management at Microsoft. Chen *et al.* [40] leveraged the semantic matching capabilities of LLMs to develop a multi-vendor configuration management tool at Huawei. LLM-based programming assistants, such as Copilot [41], have been successfully deployed in code data analysis applications, providing accurate code understanding and recommendations [42]–[44]. Additionally, Liu *et al.* utilized LLMs to automate high-precision data analysis on tabular datasets, implementing their approach in an LCD factory and a solar cell factory [45].

In comparison to existing studies, our work focuses on improving data quality in LLM training and thereby can be integrated into industrial LLM applications to improve data engineering performance. We validates the feasibility of expert-aligned revisions on instruction pairs from the entire instruction dataset. Compared with filtering-based approaches, our approach maintains the integrity of the dataset and increases the proportion of high-quality samples, thereby resulting in better performance improvements of LLMs.

VI. Conclusion

In this study, we propose CoachLM, a novel approach to tackle the issue of unguaranteed data quality in LLM instruction tuning. Owing to the ability of automatic revisions aligned with language experts, CoachLM effectively enhances the proportion of high-quality samples in the ALPACA52K dataset, resulting in notable performance improvements in instruction-tuned LLMs. Additionally, the successful deployment of CoachLM in an industrial-level data management system highlights its potential advantages in the operation and maintenance lifecycle of LLMs, reducing costs associated with manual data cleaning and labeling. Future work includes training CoachLM on a larger scale of parameters, integrating RL pipelines to mitigate hallucination and validating it using a more diverse range of instruction datasets.

REFERENCES

[1] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[2] OpenAI, "Gpt-4 technical report," 2023.

[3] K. S. Kalyan, "A survey of gpt-3 family large language models including chatgpt and gpt-4," *Natural Language Processing Journal*, p. 100048, 2023.

[4] T. H. Kung, M. Cheatham, A. Medenilla, C. Sillos, L. De Leon, C. Elepaño, M. Madriaga, R. Aggabao, G. Diaz-Candido, J. Maningo *et al.*, "Performance of chatgpt on usmle: Potential for ai-assisted medical education using large language models," *PLoS digital health*, vol. 2, no. 2, p. e0000198, 2023.

[5] A. Askari, M. Aliannejadi, E. Kanoulas, and S. Verberne, "A test collection of synthetic documents for training rankers: Chatgpt vs. human experts," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 5311–5315.

[6] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[7] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[8] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[9] J. Wei, M. Bosma, V. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," in *International Conference on Learning Representations*, 2021.

[10] S. Mishra, D. Khashabi, C. Baral, and H. Hajishirzi, "Cross-task generalization via natural language crowdsourcing instructions," in *60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*. Association for Computational Linguistics (ACL), 2022, pp. 3470–3487.

[11] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems*, vol. 30, 2017.

[12] A. Havrilla, M. Zhuravinskyi, D. Phung, A. Tiwari, J. Tow, S. Biderman, Q. Anthony, and L. Castricato, "trlx: A framework for large scale reinforcement learning from human feedback," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 8578–8595.

[13] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu *et al.*, "Instruction tuning for large language models: A survey," *arXiv preprint arXiv:2308.10792*, 2023.

[14] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, "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)*. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 13 484–13 508. [Online]. Available: https://aclanthology.org/2023.acl-long.754

[15] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Stanford alpaca: An instruction-following llama model," https://github.com/tatsu-lab/stanford_alpaca, 2023.

[16] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," March 2023. [Online]. Available: https://lmsys.org/blog/2023-03-30-vicuna/

[17] X. Geng, A. Gudibande, H. Liu, E. Wallace, P. Abbeel, S. Levine, and D. Song, "Koala: A dialogue model for academic research," Blog post, April 2023. [Online]. Available: https://bair.berkeley.edu/blog/2023/04/03/koala/

[18] C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu *et al.*, "Lima: Less is more for alignment," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[19] M. Li, Y. Zhang, Z. Li, J. Chen, L. Chen, N. Cheng, J. Wang, T. Zhou, and J. Xiao, "From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning," *arXiv preprint arXiv:2308.12032*, 2023.

[20] L. Chen, S. Li, J. Yan, H. Wang, K. Gunaratna, V. Yadav, Z. Tang, V. Srinivasan, T. Zhou, H. Huang, and H. Jin, "Alpagasus: Training a better alpaca model with fewer data," in *International Conference on Learning Representations*, 2024.

[21] Y. Cao, Y. Kang, and L. Sun, "Instruction mining: High-quality instruction data selection for large language models," *arXiv preprint arXiv:2307.06290*, 2023.

[22] C. Xu, Q. Sun, K. Zheng, X. Geng, P. Zhao, J. Feng, C. Tao, Q. Lin, and D. Jiang, "WizardLM: Empowering large pre-trained language models to follow complex instructions," in *International Conference on Learning Representations*, 2024.

[23] N. Rajani, N. Lambert, S. Han, J. Wang, O. Nitski, E. Beeching, and L. Tunstall, "Can foundation models label data like humans?" *Hugging Face Blog*, 2023, https://huggingface.co/blog/llm-v-human-data.

[24] Y. Wang, Z. Yu, Z. Zeng, L. Yang, W. Yao, C. Wang, H. Chen, C. Jiang, R. Xie, J. Wang, X. Xie, W. Ye, S. Zhang, and Y. Zhang, "PandaLM: An automatic evaluation benchmark for LLM instruction tuning optimization," in *International Conference on Learning Representations*, 2024.

[25] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[26] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.

[27] P. Lu, S. Mishra, T. Xia, L. Qiu, K.-W. Chang, S.-C. Zhu, O. Tafjord, P. Clark, and A. Kalyan, "Learn to explain: Multimodal reasoning via thought chains for science question answering," *Advances in Neural Information Processing Systems*, vol. 35, pp. 2507–2521, 2022.

[28] R. Shang, Y. Ma, F. Ali, C. Hu, S. Nazir, H. Wei, and A. Khan, "Selection of crowd in crowdsourcing for smart intelligent applications: A systematic mapping study," *Scientific Programming*, vol. 2021, pp. 1–23, 2021.

[29] X. Fang, S. Si, G. Sun, Q. Z. Sheng, W. Wu, K. Wang, and H. Lv, "Selecting workers wisely for crowdsourcing when copiers and domain experts co-exist," *Future Internet*, vol. 14, no. 2, p. 37, 2022.

[30] L. Wei, Z. Jiang, W. Huang, and L. Sun, "Instructiongpt-4: A 200-instruction paradigm for fine-tuning minigpt-4," *arXiv preprint arXiv:2308.12067*, 2023.

[31] V. I. Levenshtein *et al.*, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8. Soviet Union, 1966, pp. 707–710.

[32] Z. Du, Y. Qian, X. Liu, M. Ding, J. Qiu, Z. Yang, and J. Tang, "Glm: General language model pretraining with autoregressive blank infilling," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 320–335.

[33] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2021.

[34] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[35] M. Conover, M. Hayes, A. Mathur, J. Xie, J. Wan, S. Shah, A. Ghodsi, P. Wendell, M. Zaharia, and R. Xin. (2023) Free dolly: Introducing the world's first truly open instruction-tuned llm. [Online]. Available: https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm

[36] X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang, "Data cleaning: Overview and emerging challenges," in *Proceedings of the 2016 international conference on management of data*, 2016, pp. 2201–2206.

[37] L. Schmarje, M. Santarossa, S.-M. Schröder, C. Zelenka, R. Kiko, J. Stracke, N. Volkmann, and R. Koch, "A data-centric approach for improving ambiguous labels with combined semi-supervised classification and clustering," in *European Conference on Computer Vision*. Springer, 2022, pp. 363–380.

[38] D. Sanderson and T. Kalganova, "Maintaining performance with less data: Understanding useful data," in *International Congress on Information and Communication Technology*. Springer, 2023, pp. 1105–1127.

[39] T. Ahmed, S. Ghosh, C. Bansal, T. Zimmermann, X. Zhang, and S. Rajmohan, "Recommending root-cause and mitigation steps for cloud incidents using large language models," in *ICSE 2023*, May 2023.
[40] H. Chen, Y. Miao, L. Chen, H. Sun, H. Xu, L. Liu, G. Zhang, and W. Wang, "Software-defined network assimilation: bridging the last mile towards centralized network configuration management with nassim," in *Proceedings of the ACM SIGCOMM 2022 Conference*, 2022, pp. 281–297.
[41] "Github copilot - your ai pair programmer," https://copilot.github.com/, 2023, retrieved March 13, 2023.
[42] F. F. Xu, U. Alon, G. Neubig, and V. J. Hellendoorn, "A systematic evaluation of large language models of code," in *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming*, 2022, pp. 1–10.
[43] Y. Li, D. Choi, J. Chung, N. Kushman, J. Schrittwieser, R. Leblond, T. Eccles, J. Keeling, F. Gimeno, A. Dal Lago *et al.*, "Competition-level code generation with alphacode," *Science*, vol. 378, no. 6624, pp. 1092–1097, 2022.
[44] B. Yetistiren, I. Ozsoy, and E. Tuzun, "Assessing the quality of github copilot's code generation," in *Proceedings of the 18th International Conference on Predictive Models and Data Analytics in Software Engineering*, 2022, pp. 62–71.
[45] S.-C. Liu, S. Wang, T. Chang, W. Lin, C.-W. Hsiung, Y.-C. Hsieh, Y.-P. Cheng, S.-H. Luo, and J. Zhang, "Jarvix: A llm no code platform for tabular data analysis and optimization," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 2023, pp. 622–630.