

DeepThink: Aligning Language Models with Domain-Specific User Intents

Anonymous ACL submission

Abstract

Supervised fine-tuning with synthesized instructions has been a common practice for adapting LLMs to domain-specific QA tasks. However, the synthesized instructions deviate from real user questions and expected answers. This study proposes a novel framework called DeepThink to generate high-quality instructions. DeepThink first generates a few seed questions to mimic actual user questions, simulates conversations to uncover the hidden user needs, and refines the answer by conversational contexts and the retrieved documents for more comprehensive answers. Experiments demonstrate that DeepThink achieves an average performance improvement of 7.92% compared to a GPT-4-turbo+RAG-based assistant on the real user test set in the advertising domain across dimensions such as relevance, completeness, clarity, accuracy, and actionability.

1 Introduction

Large language models (LLMs) such as GPT-4 (OpenAI et al., 2023) have achieved remarkable advancements in question-answering (QA) tasks. Commercial and open-source LLMs are primarily trained on general-domain data and perform less effectively in vertical domains such as healthcare, finance and advertising (Goyal et al., 2024; Xu et al., 2024b; Wu et al., 2023). Supervised Fine-Tuning (Sanh et al., 2022; Wang et al., 2024, 2022) (SFT) has been widely adopted to optimize the LLM’s parameters on a curated set of instructions or task examples to enhance LLMs’ ability to answer domain-specific questions.

Due to the high cost of collecting instruction data, recent SFT methods generate synthetic data. They typically start with a few *seed instructions*, either constructed manually (Ouyang et al., 2022) or generated by LLMs from available documents (Wang et al., 2024). The seed questions are then expanded (Wang et al., 2022) or evolved (Xu

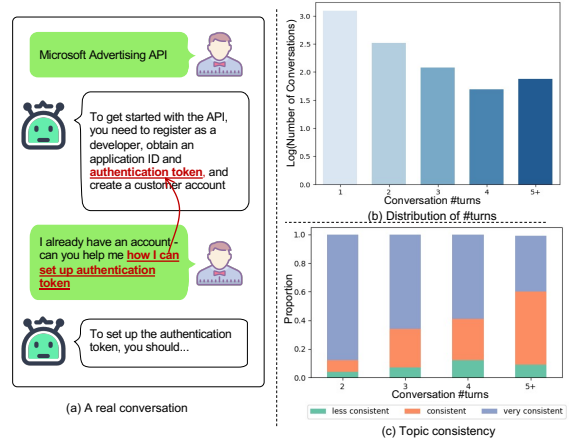


Figure 1: Three phenomena on real-world advertising platforms

et al., 2023) to provide greater complexity and diversity. However, the synthesized instructions deviate from user questions and expected answers.

We now characterize the user demands of domain-specific QA by analyzing an advertising platform. As illustrated in Figure 1(a), users usually start with a brief question and ask for more details mentioned in the assistant’s responses. Consequently, multi-turn dialogues constitute a substantial portion (Figure 1(b)), and most conversations, no matter how many turns, focus on one topic theme (Figure 1(c)).¹

The above observations reveal a critical challenge of QA in vertical domains: Questions are vague and incomplete and do not reflect the user’s hidden interests. Because specialized knowledge is required in vertical domains, users do not always possess such expertise, and they are incapable of

¹The platform currently uses GPT-4-turbo with Retrieval Augmented Generation (RAG) as an intelligent assistant. We prompt GPT-4-turbo to analyze whether the dialogue is closely centered around a single topic theme and assign a topic consistency score. The scores range from 1 to 5, where a score of 3 or below indicates a lower level of thematic consistency, and a score of 5 represents very high thematic consistency.

asking concise and accurate questions. LLMs fine-tuned by conventional instructions (e.g., synthetic questions that mismatch actual user behavior patterns and answers that fall short of user expectations) tend to provide broad responses, which will increase the cost of consultation and harm user experience. The question naturally arises. *Could we use high-quality synthetic instructions to fine-tune the LLM to capture the user’s hidden interests and give a precise and relevant answer?*

Figure 1(a) also demonstrates that authentic user interests are gradually exposed through conversations and satisfying answers are obtained by continuously expanding on a question and delving into technical details. Inspired by this insight, we propose a novel supervised fine-tuning method named DeepThink. DeepThink first generates a few seed questions guided by actual user questions. Then, to simulate human conversations, DeepThink designs a dual-role (i.e., the inquirer and the assistant) framework to generate dialogues on the seed questions. An evaluator assesses each answer based on the dialogue context and provides revision suggestions for a refiner to enhance the quality of the answers. Finally, the questions in each turn of the simulated dialogues and the corresponding refined answers are employed for supervised fine-tuning.

We evaluate the performance of DeepThink on an online advertising platform. As shown in Figure 2, DeepThink surpasses commercial LLM, i.e., GPT-4-turbo+RAG and achieves improvements of 3.43%, 12.09%, 6.69%, 3.74%, and 13.69% regarding relevance, completeness, clarity, accuracy, and actionability, respectively.

In summary, the main contribution of DeepThink is threefold. (1) DeepThink presents a novel instruction synthesis method by simulating real-world user queries and follow-up conversations for supervised fine-tuning. (2) DeepThink refines the synthesized answers based on the conversation contexts to ensure that the LLMs can generate more comprehensive answers and address user interests in vertical domains. (3) A large-scale evaluation on an advertising platform of real user questions verifies the effectiveness of DeepThink.

2 Related Works

2.1 Instruction Data Synthesis

To address the issue of limited training samples in specific domains, various works have proposed using additional data, such as manual annota-

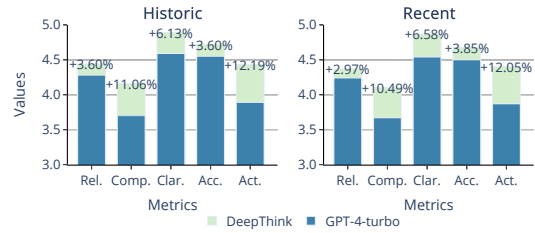


Figure 2: Performance comparison of DeepThink and GPT-4-turbo across five evaluation dimensions over different time spans ("Historic," and "Recent."). DeepThink performs better than GPT-4-turbo in relevance, completeness, clarity, accuracy, and actionability.

tion (Zhao et al., 2024; Zheng et al., 2023) and automatic generation by LLMs (Mekala et al., 2022; Wang et al., 2021, 2022; Xu et al., 2023). However, manual annotation is expensive (Honovich et al., 2023), and iterative generation by LLMs frequently introduces the risk of hallucinations.

Our work falls into the category of automatic generation by LLMs. However, our work differs from previous approaches in two main aspects. (1) We synthesize instructions by simulating conversations closer to real-world scenarios. (2) We adopt several techniques to improve the quality of synthesized instruction. We integrate Retrieval-Augmented Generation (RAG) to mitigate hallucination in conversation-based synthesis. We apply a Conversation-based Data Refiner for filtering, ensuring topic consistency and data authenticity.

2.2 Retrieval-Augmented Generation

Retrieval augmentation has become a standard solution to address hallucinations in LLMs by introducing external knowledge to compensate for factual shortcomings (Asai et al., 2023; Ma et al., 2023; Izacard et al., 2021; Ram et al., 2023). Early Retrieval Augmentation efforts focus primarily on the retriever itself, where both the neural retriever and generator are typically trainable Pretrained Language Models (PrLMs), such as BERT (Devlin et al., 2019) or BART (Lewis et al., 2019). In contrast, modern Retrieval Augmentation applied to LLMs emphasizes determining when and how to retrieve relevant information (Fatehkia et al., 2024; Asai et al., 2023; Xu et al., 2024a). For example, Self-RAG enables on-demand retrieval and generates more accurate, fact-based text through fine-grained self-reflection (Asai et al., 2023).

Our approach uses RAG throughout the data syn-

thesis, SFT, and inference stages. This not only improves the authenticity of the synthesized data but also helps the LLM learn how to effectively utilize the retrieved knowledge during the SFT stage. In contrast, previous research only used RAG during the inference stage, relying heavily on the LLM’s ability to discern the retrieved knowledge. This can lead to insufficient utilization of relevant knowledge, especially when dealing with domain knowledge that was not included in the pretraining process.

3 Approach

As illustrated in Figure 3, DeepThink consists of four key stages: (1) Seed Question and Answer Synthesis, (2) Conversation-based Data Synthesis, (3) Conversation-based Data Refinement, and (4) Retrieval-augmenting-SFT.

3.1 Seed Question and Answer Synthesis

We leverage GPT-4-turbo to synthesize instructions. Existing studies such as SELF-QA (Zhang and Yang, 2023) utilize LLMs to extract questions from documents, enabling the automatic generation of seed questions. Unfortunately, the questions generated through existing approaches exhibit significant linguistic style discrepancies compared with those found in genuine user-LLM interactions. To resolve this issue, we randomly sample a few authentic user questions and prompt GPT-4-turbo to generate queries that mirror the linguistic style and structure of these samples. This method ensures that the generated questions reflect the realistic language and format of the actual user queries. Details of the specific prompts used are in Appendix G.

3.2 Conversation-based Data Synthesis

We implement a dual-role conversation framework based on GPT-4-turbo, where one role is designated as the *Inquirer* and the other as the *Assistant*. DeepThink guides the two roles to emulate authentic and high-quality conversations. Specifically, in guiding the *Inquirer*, DeepThink instructs it to mimic the style of actual user inquiries by incorporating real user questions into *Inquirer*’s prompt. This stylistic imitation distinguishes from earlier methods (Wang et al., 2022) without instructing the LLM using actual user queries, resulting in dialogues that more closely reflect real-world conversational dynamics.

Previous methods rely on the inherent knowledge of LLMs to generate answers and often lead

to hallucinations, especially in vertical domains where LLMs lack direct training data (Abdullin et al., 2024; Liu et al., 2024). In generating the *Assistant*’s responses, DeepThink incorporates a retrieval-augmented generation framework. By retrieving domain-relevant documents to ground responses, this approach mitigates the risk of hallucination and enhances the response’s accuracy.

Furthermore, to maintain engagement and progressively deepen the dialogue, DeepThink instructs the *Assistant* to suggest follow-up questions based on topics that may interest the user. The *Inquirer* then has the option to (1) choose from these suggestions, (2) generate a new question, or (3) respond with "No more questions" to end the conversation. This structured interaction ensures that the conversation flow remains natural. The interaction is enforced to end when exceeding a predefined maximal number of turns because long dialogues are likely to drift from the original topic. The prompt is shown in Figure 19, and cases are shown in Appendix F.1.

3.3 Conversation-based Data Refinement

The answers generated from the above procedures face several critical issues: (1) they merely provide superficial responses to user queries without capturing the underlying intent behind the questions, and (2) they fail to address ambiguous or unclear user queries, resulting in answers that do not align with the user’s expectations. To mitigate these challenges, leveraging question-and-answer pairs from other turns in the conversation to supplement the current response presents a natural solution. However, this process is inherently complex, as the content from other turns may not always align perfectly with the current question, and irrelevant information should not be incorporated into the refinement. To address this, we propose an iterative answer refinement strategy based on the synthesized conversation. In each iteration of the refinement process, the refiner is prompted to refine the answer based on the conversational context, followed by an assessment phase where the refined content is evaluated and constructive feedback is generated. This feedback is then utilized as input for the subsequent iteration, guiding the refiner to improve the response further.

Initialization. Refinement focuses on enriching the current answer by incorporating relevant information from the conversation’s other turns. Specifically, DeepThink feeds synthesized questions, cor-

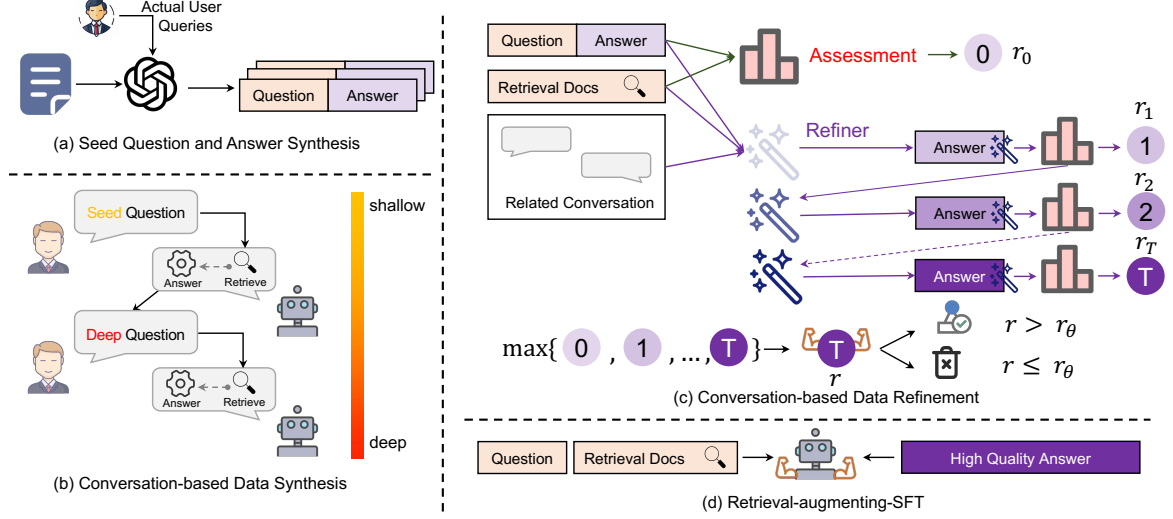


Figure 3: The framework of DeepThink

responding answers, and the related conversation context into GPT-4-turbo (Refiner). By designing specific prompts that guide the Refiner to mimic the linguistic style of real user inquiries, we ensure that the generated answers are both comprehensive and stylistically consistent with authentic user interactions. Additionally, to minimize irrelevant interference and prevent potential hallucinations, we retrieve documents closely aligned with the current question and include them in the input.

Feedback-based Refinement. As previous studies (Zheng et al., 2024; Mao et al., 2023) have demonstrated GPT-4-turbo’s capability to emulate human evaluation preferences, we employ it as an effective assessor. GPT-4-turbo evaluates responses across five dimensions: relevance, completeness, clarity, accuracy, and actionability, providing an overall score and detailed feedback. This feedback is subsequently utilized as input for the refiner to further refine the response in the next iteration. This multi-faceted assessment allows for targeted refinements, ensuring that each aspect of the response aligns with user expectations and the conversational context. The iterative process continues for a maximum of rounds T .

Instruction Update and Filtering. We calculate the overall score r_0 of the original answer a_0 and put the original answer in the selection pool $\mathcal{P}$. We also obtain the assessment score r_t for the refined answer a_t in each iteration, where $t \in [1, T]$. We put these answers in $\mathcal{P}$. We select the best answer with the highest score in the pool that exceeds a predefined quality threshold, i.e.,

$$r = r_{\arg \max_{0 \leq t \leq T} r_t, r_t > r_\theta}.$$

3.4 Retrieval-Augmented Supervised Fine-Tuning

To effectively capture and utilize domain-specific knowledge, our proposed DeepThink integrates Retrieval-Augmented Generation (RAG) within the Supervised Fine-Tuning (SFT) framework. Unlike traditional SFT approaches that rely solely on (question, answer) pairs generated from the model’s inherent knowledge, DeepThink leverages relevant external documents to enhance answer accuracy and context awareness.

Motivation for RAG Integration. In vertical domains such as advertising, healthcare, and finance, user queries often require precise and contextually rich responses that depend on up-to-date and domain-specific information. Traditional SFT methods fall short in these scenarios as they do not utilize external knowledge sources, limiting the model’s ability to generate accurate and relevant answers based on the provided context.

RAG-Augmenting-SFT. Our approach involves incorporating retrieved documents into the supervised fine-tuning process. For each training instance, given a question x_i , we first retrieve the most relevant documents d_i from a curated knowledge base. The model then generates an answer y_i conditioned on both the question and the retrieved documents. This results in a (question, document, answer) triplet that forms the basis of our fine-tuning data.

Formally, the training loss is defined as:

$$\mathcal{L}(\Phi) = - \sum_{(x_i, d_i, y_i)} \log p(y_i | x_i, d_i, \Phi), \quad (1)$$

where Φ represents the LLM parameters, y_i is the answer generated by the LLM, and $p(\cdot)$ is the likelihood of the answer given the question and documents.

4 Experiments

4.1 Experimental Setups

Dataset. We consider the Advertising Domain in this paper. We have obtained relevant documentation from an advertising platform, consisting of a total of 7.5k text segments covering Setup and Basics, Manage Ads, Measuring Results, and Billing and Payments. For testing, we have collected real user questions from our advertising platform, spanning the period from March 27 to June 6. The dataset underwent a cleaning process, including deduplication and filtering out questions unrelated to advertising, resulting in a refined collection of 7,801 user questions. Subsequently, we order these questions chronologically based on the time they were posted and divide them into two subsets: (1) *Historic*: consists of the earliest 90% of data (i.e., 6,617 questions). This subset allows us to gauge the model’s response quality over a broad range of topics, thus offering a comprehensive view of the model’s overall capabilities. (2) *Recent*: consists of the most recent 10% of data (i.e., 1,184 questions), representing the latest user needs on the platform. It is used to assess the LLM’s response quality for recent user activities. This subset helps detect any potential degradation in the model’s performance over time, ensuring its continued reliability and alignment with user expectations.

Imitation seed data. Following prior work (Zhang and Yang, 2023), we prompt GPT-4-turbo to generate at least one high-quality instruction for each text segment from the domain data. To approximate the distribution of real user inquiries, we randomly select 15 authentic user questions from our advertising platform to prompt GPT-4-turbo to generate seed questions. Our conversation-based Data synthesis also employs the same set of 15 real user questions throughout the process. We obtain approximately 5k seed data. Detailed prompts are provided in Figure 17.

Baseline. (1) *Proprietary and open-source LLMs* including GPT-4-turbo, GPT-3.5-turbo, Mis-

Table 1: Statistics of Instruction Data Generated by Different Methods

	# Examples	# Domain	Instruction Length	Output Length
Seed data	5k	advertising	10±3	87±20
Self Instruct	23k	advertising	15±13	56±27
Evol Instruct	15k	advertising	18±8	74±24
Magpie	300k	general	12±6	377±76
DeepThink	12k	advertising	18±6	90±20

tral 7B, and Llama3 8B. (2) *Data synthesis+SFT* methods which synthesize instructions and utilize Mistral 7B as the base model for SFT, including Self Instruct (Zhang and Yang, 2023), Evol Instruct (Xu et al., 2023), and Magpie (Xu et al., 2024c). Note that these data-synthesis strategies do not incorporate Retrieval Augmented Generation (RAG) in data synthesis. It is impossible to include retrieved contents as part of the question during the SFT phase. To ensure a fair comparison, we also implement a *variant of the proposed model* DeepThink-S, which does not use retrieved content in the question. The statistics of generated instructions are shown in Table 1. (3) *RAG-augmenting-SFT* baselines, which utilize Mistral 7B as the base model for retrieval augmented SFT, including RAFT (Zhang et al., 2024) and DSF (Zhang et al., 2024). More details about baselines are discussed in Appendix B

Evaluation. Following previous works (Zhu et al., 2023; Zheng et al., 2024), we leverage GPT-4-turbo to evaluate the quality of model-generated responses. Specifically, we input the question, the most relevant documents, and the model’s response into GPT-4-turbo, prompting it to score the model’s answer based on relevance, completeness, clarity, accuracy, and actionability. We further evaluate DeepThink using DeepSeek-R1 and Llama-3.1-405B, with results provided in the Appendix C.

4.2 Comparative Study

Comparison of Response Quality. We first compare the performance of DeepThink with various baseline models regarding multi-facet evaluation. The results are shown in Table 2. We have the following observations.

(1) DeepThink achieves superior performance in the advertising domain. Compared with proprietary LLMs, DeepThink achieved improvements of 3.43%, 12.09%, 6.69%, 3.74%, and 13.69% over the best-performing GPT-4-turbo in relevance, completeness, clarity, accuracy, and actionability metrics, respectively. It indicates proprietary

Table 2: Performance of different methods

Type	Model	Historic					Recent					Avg Ovr.
		Rel.	Comp.	Clar.	Acc.	Act.	Rel.	Comp.	Clar.	Acc.	Act.	
LLMs	GPT-4-turbo	4.28	3.70	4.59	4.55	3.89	4.24	3.67	4.54	4.50	3.87	4.18
	GPT-3.5-turbo	3.80	3.29	4.05	4.03	3.47	4.22	3.66	4.54	4.49	3.85	3.94
	Mistral 7B	4.23	3.51	4.70	4.60	3.90	3.97	3.24	4.44	4.37	3.60	4.05
	Llama3 8B	4.06	3.37	4.51	4.35	3.66	3.97	3.41	4.42	4.29	3.67	3.97
Data Synthesis+SFT	Self Instruct	4.29	3.77	4.79	4.51	4.09	4.25	3.74	4.75	4.46	4.06	4.27
	Evol Instruct	4.28	3.83	4.78	4.52	4.06	4.23	3.73	4.73	4.44	4.01	4.26
	Magpie	4.02	3.73	4.52	4.21	4.05	3.97	3.65	4.45	4.16	3.96	4.07
	DeepThink-S	<u>4.31</u>	<u>3.99</u>	<u>4.83</u>	<u>4.55</u>	<u>4.33</u>	<u>4.27</u>	<u>3.95</u>	<u>4.80</u>	<u>4.52</u>	<u>4.29</u>	<u>4.38</u>
RAG-augmenting-SFT	RAFT	4.27	3.66	4.66	<u>4.60</u>	3.95	4.22	3.62	4.63	<u>4.54</u>	3.93	4.21
	DSF	4.19	3.64	4.69	4.43	3.90	4.09	3.52	4.58	4.32	3.80	4.12
Data Synthesis +RAG-augmenting-SFT	DeepThink	4.44	4.16	4.89	4.72	4.43	4.37	4.10	4.86	4.68	4.40	4.50

LLMs only focus on general domain knowledge and do not perform well in the vertical domain, i.e., the advertising domain. Supervised Fine-Tuning (SFT) is necessary for the advertising domain and DeepThink proposes an efficient data synthesis strategy for SFT. Compared with other data synthesis strategies, DeepThink achieved improvements of 3.15%, 9.98%, 2.21%, 4.77%, and 8.30% over the best-performing Self Instruct in relevance, completeness, clarity, accuracy, and actionability, respectively. Besides, DeepThink-S also achieved average improvements of 2.68% over Self Instruct. This means that our model benefits from conversation-based synthetic data, which enables it to gain insights into users' hidden interests and provide higher-quality responses.

(2) Our model demonstrates significant improvements in Completeness and Actionability, outperforming other baselines by at least 9.13% and 8.34%, respectively. These gains are likely due to our use of conversation data, which enhances response quality in two key ways. First, conversation data is highly focused on specific topics, enabling the model to provide more comprehensive and detailed answers, thus improving Completeness. Second, because conversations often explore practical "how-to" details, the model generates more actionable responses, boosting Actionability.

(3) In our model, RAG-augmenting-SFT has resulted in significant performance improvements. Specifically, DeepThink demonstrates an overall improvement of 2.73% compared with DeepThink-S, which does not use RAG. This indicates that RAG helps the model generate higher-quality responses.

Comparison of Human Preference. We calculate the WinRate of each model in comparison

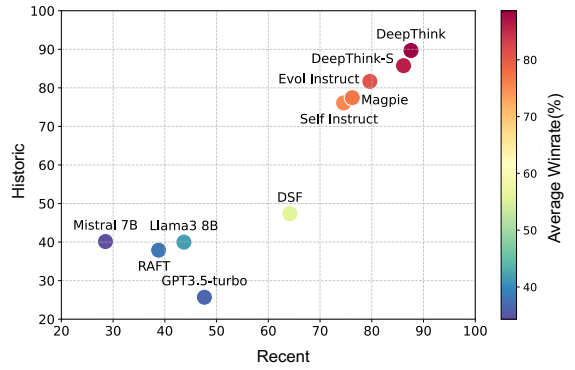


Figure 4: Human Preference Evaluation (WinRate models vs. GPT-4-turbo %)

with GPT-4-turbo (the LLM used by the online advertising assistant platform). To reflect the degree of human preference, we use the judgments from GPT-4-turbo. Detailed prompts are provided in Figure 22. We report the WinRate on the historical and recent subset in Figure 4. We also color the baselines by the average win rate. We have the following observations.

(1) Users exhibit a stronger preference for responses generated by DeepThink. DeepThink achieves the highest WinRates on both the Historic and Recent datasets, with scores of 89.69% and 87.58%, respectively. This indicates that, compared to the original advertising assistant, users prefer the responses from DeepThink.

(2) Models fine-tuned by synthesized instructions generally achieve better performance. For example, Self Instruct, Evol Instruct, Magpie, DeepThink-S, and DeepThink all achieve a WinRate of at least 70% on both datasets.

(3) DeepThink demonstrates superior performance compared with other instruction synthesis methods for SFT. DeepThink achieves a 9.90% im-

Table 3: Performance of each component in DeepThink on Recent dataset

	Rel.	Comp.	Clar.	Acc.	Act.
DeepThink	4.37	4.10	4.86	4.68	4.40
w/o CDS	4.19	3.89	4.73	4.49	4.21
w/o CDR	4.21	3.76	4.72	4.45	4.07
w/o CDS, CDR	4.14	3.45	4.65	4.38	3.85

provement over the best-performing baseline, Evol-Instruct. This improvement highlights the effectiveness of our proposed strategy, which leverages conversation data to uncover users’ deeper interests and employs an iterative refiner to optimize answers continuously. Our approach not only generates higher-quality answers but also better captures the underlying concerns and interests behind user queries.

4.3 Impact of Conversation-base Data Synthesis and Refinement

We conduct extensive experiments to show the effectiveness of different components in DeepThink. We conduct a series of ablation studies that involve: (1) removing Conversation-based Data Synthesis (w/o CDS), (2) removing Conversation-based Data Refinement (w/o CDR), and (3) the simultaneous removal of CDS and CDR in the recent dataset. The results are presented in Table 3. From these experiments, we draw the following conclusions.

Every component in our model contributes significantly to its performance. When CDS is removed, the model exhibits notable declines in response quality: Relevance drops by 4.12%, Completeness by 5.12%, Clarity by 2.67%, Accuracy by 4.06%, and Actionability by 4.32%. Similarly, removing CDR results in reductions of 3.72% in Relevance, 8.34% in Completeness, 2.83% in Clarity, 4.88% in Accuracy, and 7.52% in Actionability. Furthermore, when both CDS and CDR are removed simultaneously, the model’s performance degrades even more significantly, with Relevance decreasing by 5.32%, Completeness by 15.90%, Clarity by 4.27%, Accuracy by 6.37%, and Actionability by 12.52%. These results clearly demonstrate the importance and effectiveness of each component in our model.

(2) The removal of CDR has the most significant impact on the model’s performance. This demonstrates the critical role of CDR in leveraging conversational context to enhance response quality. Specifically, CDR refines responses by utiliz-

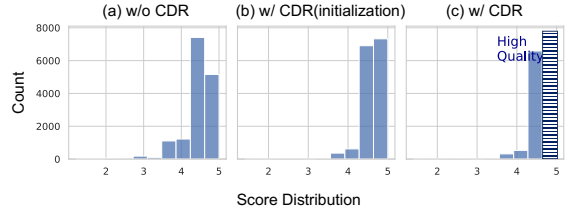


Figure 5: Score distribution of the instructions

Table 4: Performance of Imitation-based and Synthesis-only seed data

	Sim.	Rel.	Comp.	Clar.	Acc.	Act.
Synthesis-only	0.76	4.14	3.42	4.63	4.37	3.81
Imitation-based	0.79	4.15	3.46	4.64	4.37	3.84

ing assessment feedback to filter out irrelevant or meaningless information from dialogues, thereby significantly improving the overall quality of the generated answers.

(3) CDS has a particularly strong impact on improving the relevance of the model’s responses. When CDS is removed, the relevance of the model’s answers shows the most significant decline, dropping by 4.17%. This is likely because CDS generates a broader range of high-quality instructions by simulating real-world user conversations. By closely mimicking how users naturally communicate, CDS ensures that the generated instructions are more aligned with actual user queries, thereby enhancing the relevance of the model’s responses.

Furthermore, we conduct ablation studies on the refiner in Conversation-based Data Refinement. Specifically, we implement the three variants: (1) the original answers obtained by synthesized conversation without refinement (w/o CDR), (2) answers initially refined by only the conversation contexts (w/ CDR initialization), and (3) answers iteratively refined by assessment feedback (w/ CDR). We report the distribution of assessment scores in Figure 5, and we make the following observations.

(1) The initial effect of the refiner is significant. The average score has been increased from 4.63 to 4.75 after refiner initialization, representing an improvement of 2.59%. This indicates that refinement leveraging conversational context enhances response quality.

(2) Feedback-guided refiner further increases the ratio of high-quality answers, i.e., with a score of five. This demonstrates that the assessment feedback effectively guides the refiner to fine-tune responses, making them better aligned with user preferences.

4.4 Necessary of Imitation

We analyze the differences between instructions generated by GPT-4-turbo using two distinct approaches: imitation-based, which replicates the style of real user questions, and synthesis-only, i.e., *SELF-QA*, which generates instructions without such imitation. Utilizing the all-mpnet-base-v2 model, we obtain embeddings for each instruction and for real user questions from the Recent evaluation dataset. In addition to the five evaluation dimensions, we also calculate the similarity(Sim.) between the centroid of the imitation-based instruction embeddings and the centroid of real user question embeddings, as well as between the centroid of synthesis-only instruction embeddings and the centroid of real user question embeddings.

As shown in Table 4, imitation-based instructions exhibit higher similarity (Sim.=0.79) to actual user questions compared with synthesis-only instructions (Sim.=0.76). Additionally, models fine-tuned on imitation-based data demonstrate improved performance across various metrics, including Relevance, Completeness, Clarity, Accuracy, and Actionability, compared with those trained with synthesis-only data. Specifically, imitation-based methods achieve improvements of 0.24% in Relevance, 1.17% in Completeness, and 0.22% in Clarity. These results indicate that imitation-based instruction data more closely align with real user queries, leading to enhanced model performance.

4.5 Performance of RAG-augmenting-SFT

To evaluate the impact of Retrieval-Augmented Generation (RAG) on Supervised Fine-Tuning, we first compare the loss of DeepThink-S (i.e., uses only the original questions without any retrieved documents) with DeepThink (i.e., uses retrieved documents). As shown in Figure 6, the training loss for DeepThink-S is significantly higher than that for DeepThink, with an average increase of 37.28%. This discrepancy can be attributed to the distinct reliance on knowledge sources during the Supervised Fine-Tuning phase. Specifically, solely relying on the knowledge poses a non-trivial challenge for LLMs. In contrast, integrating retrieved documents within the instructional contexts allows learning objectives to align effectively with more accurate responses, reinforcing LLM’s ability during the SFT phase.

We have shown that DeepThink achieves a notable improvement regarding all performance met-

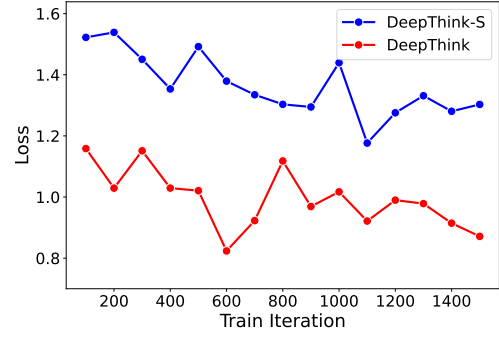


Figure 6: Training loss trend of DeepThink with and without RAG-augmenting-SFT on Recent

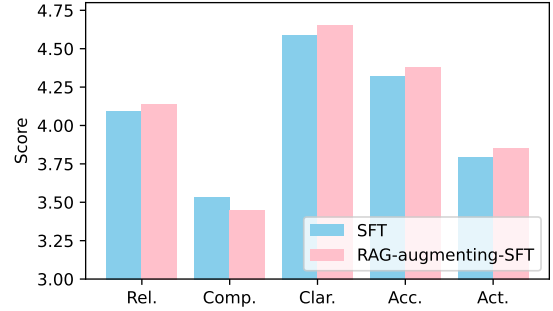


Figure 7: Performance of SFT and RAG-augmenting-SFT on Recent

rics compared with DeepThink-S in Table 2. We further remove the conversation component and implement (1) SFT that uses only seed instructions for fine-tuning and (2) RAG-augmenting-SFT that uses seed instructions along with retrieved documents. As shown in Figure 7, incorporating relevant documents as part of the input on lower-quality instructions also helps the model better understand contextual relationships and enhances QA capabilities.

5 Conclusion

In this paper, we propose DeepThink, a novel framework designed to improve the performance of large language models (LLMs) in domain-specific question-answering tasks. By integrating three key components: data synthesis based on conversations, data refinement based on conversations, and supervised fine-tuning (SFT) enhanced with retrieval, DeepThink addresses the critical challenge of adapting LLM to understand and meet hidden user needs in vertical domains. Our experiments demonstrate that DeepThink outperforms GPT-4-turbo+RAG by 7.92% across the evaluation metrics.

6 Limitations

This study has several limitations. First, the experimental validation was exclusively conducted within the advertising domain, which may constrain the generalizability of our methodology to other vertical domains (e.g., e-commerce, education, or healthcare). Future research should extend the evaluation framework by conducting cross-domain experiments to verify the robustness of our approach. Second, the assessment protocol relied primarily on GPT-4-turbo, DeepSeek-R1, and Llama-3.1-405B for automated evaluation, potentially introducing model-specific biases. Future work can explore (1) implementing human-in-the-loop evaluation with advertising professionals to assess practical utility, and (2) incorporating real-world A/B testing with actual advertisers to measure performance metrics in production environments.

References

- Yelaman Abdullin, Diego Molla-Aliod, Bahadorreza Ofoghi, John Yearwood, and Qingyang Li. 2024. Synthetic dialogue dataset generation using llm agents. *arXiv preprint arXiv:2401.17461*.
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. *Self-rag: Learning to retrieve, generate, and critique through self-reflection*. *ArXiv*, abs/2310.11511.
- XTuner Contributors. 2023. Xtuner: A toolkit for efficiently fine-tuning llm. <https://github.com/InternLM/xtuner>.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *Bert: Pre-training of deep bidirectional transformers for language understanding*. In *North American Chapter of the Association for Computational Linguistics*.
- Masoomali Fatehkia, Ji Kim Lucas, and Sanjay Chawla. 2024. T-rag: lessons from the llm trenches. *arXiv preprint arXiv:2402.07483*.
- Sagar Goyal, Eti Rastogi, Sree Prasanna Rajagopal, Dong Yuan, Fen Zhao, Jai Chintagunta, Gautam Naik, and Jeff Ward. 2024. Healai: A healthcare llm for effective medical documentation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 1167–1168.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. *Unnatural instructions: Tuning language models with (almost) no human labor*. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. *Unsupervised dense information retrieval with contrastive learning*. *Trans. Mach. Learn. Res.*, 2022.
- 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*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. *Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*. In *Annual Meeting of the Association for Computational Linguistics*.
- Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Chatqa: Building gpt-4 level conversational qa models. *arXiv preprint arXiv:2401.10225*.
- Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. *arXiv preprint arXiv:2305.14283*.
- Rui Mao, Guanyi Chen, Xulang Zhang, Frank Guerin, and Erik Cambria. 2023. Gpteval: A survey on assessments of chatgpt and gpt-4. *arXiv preprint arXiv:2308.12488*.
- Dheeraj Mekala, Tu Vu, Timo Schick, and Jingbo Shang. 2022. *Leveraging qa datasets to improve generative data augmentation*. In *Conference on Empirical Methods in Natural Language Processing*.
- OpenAI et al. 2023. *GPT-4 Technical Report*.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al.

2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. [In-context retrieval-augmented language models](#). *Transactions of the Association for Computational Linguistics*, 11:1316–1331.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. [Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Jian Wang, Chak Tou Leong, Jiashuo Wang, Dongding Lin, Wenjie Li, and Xiao-Yong Wei. 2024. Instruct once, chat consistently in multiple rounds: An efficient tuning framework for dialogue. *arXiv preprint arXiv:2402.06967*.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. [Self-instruct: Aligning language models with self-generated instructions](#). In *Annual Meeting of the Association for Computational Linguistics*.
- Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. 2021. [Towards zero-label language learning](#). *ArXiv*, abs/2109.09193.
- Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhajan Kam-badur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.
- Austin Xu, Will Monroe, and K. Bicknell. 2024a. [Large language model augmented exercise retrieval for personalized language learning](#). *Proceedings of the 14th Learning Analytics and Knowledge Conference*.
- Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. [Wizardlm: Empowering large language models to follow complex instructions](#). *ArXiv*, abs/2304.12244.
- Xuhai Xu, Bingsheng Yao, Yanzhe Dong, Saadia Gabriel, Hong Yu, James Hendler, Marzyeh Ghassemi, Anind K Dey, and Dakuo Wang. 2024b. Mental-llm: Leveraging large language models for mental health prediction via online text data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(1):1–32.
- Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024c. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *arXiv preprint arXiv:2406.08464*.
- Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*.
- Xuanyu Zhang and Qing Yang. 2023. Self-qa: Unsupervised knowledge guided language model alignment. *arXiv preprint arXiv:2305.11952*.
- Wenting Zhao, Xiang Ren, John Frederick Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. [Wildchat: 1m chatgpt interaction logs in the wild](#). *ArXiv*, abs/2405.01470.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric Xing, et al. 2023. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *arXiv preprint arXiv:2309.11998*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.
- Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*.

A Implementation

In data synthesis, we sample 15 seed queries, and the maximum number of conversation turns is three. We iterate for three rounds to refine the answers.

We utilize the Mistral 7B Instruct model (Jiang et al., 2023) as the base model for fine-tuning. In the training phase, following prior works (Taori et al., 2023; Wang et al., 2022; Xu et al., 2023), we apply supervision on the output tokens’ loss. The fine tuning is performed using the Xtuner framework (Contributors, 2023) with a learning rate $lr = 2e - 5$, a warm-up ratio of 0.03, and a batch size of 1. We employ the LORA training method with hyper-parameters rank r set to 64, α set to 16, and dropout rate p set to 0.05. During the generation phase, text generation is performed using vLLM (Kwon et al., 2023) with a temperature coefficient $T = 0.7$. In the context of Retrieval-Augmented Generation (RAG), we utilize the LangChain framework to process domain-specific data. We employ CharacterTextSplitter to segment the data into text chunks with a chunk size of 512 and an overlap of 32. These chunks are then embedded using the pre-trained all-mpnet-base-v2 model², and the embeddings are stored in a Chroma database. During retrieval, we calculate the similarity between the question and the stored chunks, selecting the top 3 most similar chunks as the retrieval results.

B Baselines

We compare DeepThink with the following instruction synthesis baselines.

- Self Instruct (Zhang and Yang, 2023): a method which leverages a small set of seed data and a pretrained language model to synthesize a large amount of instructional data for fine-tuning.
- Evol Instruct (Xu et al., 2023): a method that starts with a basic set of instructions and employs a large language model to iteratively rewrite them, progressively enhancing their complexity. This approach generates a wide array of instructional data with varying levels of complexity.
- Magpie (Xu et al., 2024c): a self-synthesis method that leverages the autoregressive feature of aligned LLMs like Llama-3-Instruct to

²<https://huggingface.co/sentence-transformers/all-mpnet-base-v2>

auto-generate 4 million high-quality instructions, with 300K selected for fine-tuning.

We also compare DeepThink with the following baselines that use RAG to augment SFT.

- RAFT (Zhang et al., 2024): a training method that enhances large language models (LLMs) for open-book question answering by utilizing Chain-of-Thought (CoT) during the Supervised Fine-Tuning (SFT) phase. It incorporates both relevant and irrelevant documents in the context, training the model to ignore the irrelevant ones and focus on citing useful information in its output.
- DSF: performing standard supervised finetuning, without documents in context. We follow the same setting as mentioned in RAFT.

Remarks. The goal of RAFT is to train the model to distinguish which documents are relevant to the question so that the model can answer based on these documents. On the other hand, DeepThink aims to help the model identify knowledge in the documents that is not only relevant to the question but also aligns with the user’s intent (since the answers after CDR incorporate conversational information, uncovering the deep user intent in the question). We aim for this process to be implicit, avoiding the reliance on explicit CoT, which can sometimes be inaccurate. User intentions are complex and diverse, and inappropriate or stereotypical CoT reasoning may hinder the model’s ability to fully capture the user’s true intent (Turpin et al., 2023). Besides, unlike RAFT, we did not deliberately introduce irrelevant documents in instructions that could confuse the model.

C Comparison of Response Quality Evaluated by Different LLMs

To further validate the effectiveness of DeepThink, we conducted an additional evaluation using a slow-thinking reasoning model. Specifically, we employed DeepSeek-R1 (Guo et al., 2025)³ and Llama-3.1-405B to assess the performance of representative baseline methods and DeepThink on the Recent dataset, following the same evaluation prompt template described in Section 4.2.

³We locally deployed the open-source DeepSeek-R1-Distill-Qwen-32B model and DeepSeek-R1-Distill-Llama-70B model

As demonstrated in Table 5, the experimental results reveal that when evaluated through the slow-thinking reasoning framework of DeepSeek-R1, DeepThink achieves consistent conclusions with those obtained from GPT-4-turbo. This alignment persists across multiple evaluation dimensions, suggesting that our method maintains robust performance even under more deliberate and systematic reasoning paradigms and different LLM-based evaluators.

Table 5: Performance of each component in DeepThink on Recent dataset evaluated by Different LLM Evaluators.

Evaluator	Setting	Rel.	Comp.	Clar.	Acc.	Act.
Llama-3.1-405B	GPT-4-turbo	4.50	3.79	4.60	4.52	3.94
	Self Instruct	4.67	4.01	4.78	4.62	4.17
	Evol Instruct	4.63	3.99	4.77	4.63	4.17
	RAFT	4.60	3.81	4.63	4.59	4.05
	DeepThink	4.78	4.25	4.79	4.73	4.57
DeepSeek-R1-Distill-Qwen-32B	GPT-4-turbo	4.43	3.83	4.55	4.56	4.14
	Self Instruct	4.53	3.92	4.72	4.57	4.27
	Evol Instruct	4.48	3.86	4.70	4.52	4.27
	RAFT	4.60	3.92	4.70	4.72	4.31
	DeepThink	4.75	4.26	4.82	4.77	4.65
DeepSeek-R1-Distill-Llama-70B	GPT-4-turbo	4.40	3.71	4.67	4.61	4.07
	Self Instruct	4.42	3.79	4.84	4.55	4.23
	Evol Instruct	4.36	3.72	4.83	4.49	4.22
	RAFT	4.48	3.74	4.78	4.68	4.22
	DeepThink	4.60	4.06	4.86	4.70	4.56

D Comparison of Synthetic Data Quality

We sample 1000 questions each from three synthetic datasets: DeepThink, Self Instruct, and Evol Instruct, as well as from the Recent evaluation dataset. Using the all-mpnet-base-v2 model, we obtain embeddings for these questions. We then use t-SNE to assess the distribution similarity between DeepThink, Self Instruct, Evol Instruct, and Recent datasets. Furthermore, we calculate the centroid embedding for each dataset and assessed the similarity of the centroid embeddings between DeepThink, Self Instruct, Evol Instruct, and Recent. This approach allows us to evaluate which synthetic data generation method yields data that is more comparable to real user questions.

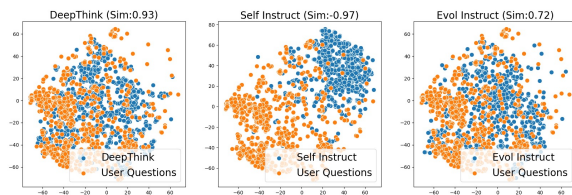


Figure 8: Similarity between different synthetic data methods and real user questions



Figure 9: Top 10 Most Common Root Verbs (Inner) and Their Top 3 Direct Noun Objects (Outer) in DeepThink



Figure 10: Top 10 Most Common Root Verbs (Inner) and Their Top 3 Direct Noun Objects (Outer) in Evol Instruct



Figure 11: Top 10 Most Common Root Verbs (Inner) and Their Top 3 Direct Noun Objects (Outer) in Self Instruct

As depicted in Figure 8, a comparison between DeepThink and other methods such as Self Instruct and Evol Instruct reveals that the instruction data generated by DeepThink exhibits significantly higher relevance to actual user questions. The centroid similarity of DeepThink is 0.93, while Self Instruct got -0.97 and Evol Instruct got 0.72. This marked relevance demonstrates that DeepThink, through its data construction approach that simulates conversational formats and style as found on real advertising platforms, generates instructions that not only better reflect user expression but also satisfy the actual demands users may present in specific scenarios.

Furthermore, we follow the previous work (Xu et al., 2024c) and show the visualization of root verbs and their direct noun objects. Figure 9, 10 and 11 visualize the top common root verbs and their direct noun objects of DeepThink, Evol Instruct and Self Instruct dataset, respectively. A notable finding is that in DeepThink, the verb "provide" holds a significantly larger proportion compared to other synthesis approaches. Additionally, expressions such as "-guidance" and "-example" are types of questions that users are more inclined to ask in the advertising domain. This result further validates that DeepThink can generate more questions that users would actually ask in this field.

E Full Parameters vs. LoRA Finetuning

We conduct two types of fine-tuning, full-parameter fine-tuning and LoRA fine-tuning, on Self Instruct, Evol Instruct, and our proposed model, DeepThink. Specifically, we employ QLoRA (Dettmers et al., 2024), a quantization-based efficient finetuning improvement of LoRA. These are subsequently evaluated on the Recent dataset. For full-parameter fine-tuning, we employ the Mistral 7B base model, while for LoRA fine-tuning, the Mistral 7B Instruct is selected as the foundation model. Our evaluation focus primarily on the relevance of the model’s responses, as this metric is a crucial indicator of the model’s accuracy and utility in understanding and generating answers. Relevance of the responses is critical because it directly influences the model’s capability to solve problems, authenticity, and user satisfaction. As depicted in Figure, we observe that full-parameter fine-tuning significantly underperformed compared to LoRA fine-tuning. One possible reason for this discrepancy is the divergence between the synthetic training data and the

distribution of real user questions, which hampers the model’s ability to generalize to authentic user data in full-parameter tuning. Additionally, our DeepThink displays superior performance in full-parameter fine-tuning compared to Self Instruct and Evol Instruct, which partially demonstrates the effectiveness of our imitation-based instruction synthesis method, as it yields instructions that more closely align with the distribution of real user questions.

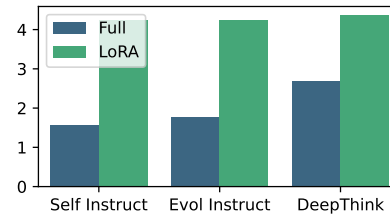


Figure 12: Performance between Full parameters and QLoRA finetuning

F Case Study

F.1 Synthesized Conversations and Refining Responses

We present a conversation synthesized by DeepThink, as shown in Figure 13. In this conversation, the user initially asks a rather vague question, leading to a less helpful and broad response. In the next round, the user clarifies the question, receiving a more precise answer. Finally, the user follows up on a specific detail in the response, obtaining a deeper and more detailed answer. This type of conversation closely mirrors the scenarios that people encounter when using language models, as described in Section 1, and DeepThink successfully synthesizes such conversations. Furthermore, Figure 14 shows that DeepThink improves the original answer to the question using Conversation-based Data refinement.

F.2 Online platform vs. DeepThink

We demonstrate the answers for the online advertising platform (GPT-4-turbo+RAG) and DeepThink regarding the same real user questions. They are shown in Figure 15 and 16. The cases show answers generated by DeepThink better than the online advertising platform in completeness and actionability.

G Prompts

#turn 1

Q: Where can I find more information about experiment splits?

A: To find more information about experiment splits in Advertising, you should access the 'Help' section directly through your Advertising account. There, you can search for 'experiment split' to find detailed articles and guides. Additionally, you can visit the Advertising support website, where resources and FAQs about various features, including experiment splits, are available. If you prefer, you can also reach out to customer support for personalized assistance.

#turn 2



This answer is so bad, perhaps I need to ask more clearly.

Q: Can you explain how the experiment split percentages are determined and can they be adjusted after the experiment has started?

A: The experiment split percentages are determined at the start of your experiment setup. This involves allocating a certain percentage of the original campaign's budget and traffic to the experiment, which you decide based on your testing objectives. Once the experiment has started running, you cannot adjust these percentages. However, **if you wish to alter the budget for the experiment, you can do so by changing the budget of the original campaign**, which will proportionally affect the experiment's budget.

#turn 3



This answer is good, but I want to ask in more detail.

Q: What specific steps should I follow to **change the budget of the original campaign** in order to adjust the experiment's budget, and how will this reflect in the experiment's performance metrics?

A: To find more information about experiment splits in Advertising, you should access the 'Help' section directly through your Advertising account. There, you can search for 'experiment split' to find detailed articles and guides. Additionally, you can visit the Advertising support website, where resources and FAQs about various features, including experiment splits, are available. If you prefer, you can also reach out to customer support for personalized assistance.



Wow, the answer is excellent. I have solved my problem.

Figure 13: The case of conversation between the user and the assistant synthesized by DeepThink.

Q: Where can I find more information about experiment splits?

A: To find more information about experiment splits in Advertising, you should access the 'Help' section directly through your Advertising account. There, you can search for 'experiment split' to find detailed articles and guides. Additionally, you can visit the Advertising support website, where resources and FAQs about various features, including experiment splits, are available. If you prefer, you can also reach out to customer support for personalized assistance.



Conversation-based Data Refinement

Q: Where can I find more information about experiment splits?

A: To find information about experiment splits, access the 'Help' section of your Advertising account and search for 'experiment split.' This split is the budget and traffic percentage allocated to your experiment from the original campaign, decided during setup. Once the experiment begins, the split cannot be altered. However, you can change the experiment's budget by adjusting the original campaign's budget. For this, go to 'Campaigns,' select the campaign, click 'Settings,' and modify the 'Budget' section. The change reflects in the experiment based on the predetermined split percentage. This information is detailed in the Help section.

Figure 14: The case of the response after Conversation-based Data Refinement.

Question: how do I check what websites my display campaign ads are showing on?

Online Platform (GPT-4-turbo+RAG): To check the websites where your display campaign ads are showing, run the Website URL (publisher) report available on the Reports page. This report will show you sites based on ad distribution rules. Additionally, you can use the Exclusions feature to exclude specific sites that you don't want your ads to run on. To learn more, see the ad distribution settings in your campaign.

DeepThink: To check the websites where your display campaign ads are showing, you can use the Website URL (publisher) report. Here's how to do it:

1. From the Reports page, select Website URL (publisher) report.
2. Choose the date range for the report.
3. Select the ad group and campaign you want to check.
4. Run the report.

This report will show you which websites your ads are being displayed on. If you find that certain websites are not providing the desired results, you can exclude them. To exclude a website:

1. Go to the Ad distribution settings in the campaign or ad group you're working on.
2. Click on Exclusions.
3. Enter the website URL you want to exclude.
4. Click Save .

DeepThink

Figure 15: Online platform answer vs. DeepThink(Case 1)

Question: change currency

Online Platform (GPT-4-turbo+RAG): You cannot change the currency for your account once it's set. If you want to be billed in a different currency, you will need to create another account.

DeepThink: To change the currency for your Advertising account, you cannot modify the currency once it's set. If you wish to be billed in a different currency, you will need to create a new account with the desired currency. To do this, sign out of your current account and then sign up for a new account with the desired currency during the sign-up process. Remember, payment methods vary by currency and country/region, so it's important to check the available payment options for your new account before proceeding.

DeepThink

Figure 16: Online platform answer vs. DeepThink(Case 2)

Input Description:

Document: <An Advertising document>

RealQ: <Some real world user questions about Advertising>

Output Description:

```
[  
  {"Question": <The question>, "Answer": <The answer>},  
  ...  
  {"Question": <The question>, "Answer": <The answer>},  
]
```

Notes:

1. Generate at least 1 high-quality question-answer pair based on the given document.
2. The answer must be based on the document. Do not answer based on common sense or personal knowledge.
3. Closely mimic the style and tone of the questions in RealQ, but do not limit the content to only what is asked in RealQ.
4. Ensure questions are diverse and cover different aspects of the Advertising document.
5. Output the final question-answer pairs in a JSON list format.

Input:

Document: {document}

RealQ: {realq}

Output:

Figure 17: The prompt of extracting Seed QAs from documents

- Task:
You are an advertising user. You need to ask the assistant about advertising.
- Input Description:
Dialogue: <The conversation related to Advertising>
RecQ: <The recommended questions generated by the assistant>
- Output Description:
{{"Question": <Ask more detailed questions about the dialogue>}}
- Notes:
 1. You need to ask more detailed questions based on the assistant's responses in the dialogue to deepen the conversation further.
 2. The questions you pose can be probing, delving deep into the concepts presented in the answers.
 3. The question should be based on the themes: Setup and Basic Information, Managing Ads, Measuring Results, Billing and Payments or other advertisement fields.
 4. You can ask the questions based on RecQ, but you can also ask other questions based on the dialogue.
 5. The question you ask should closely mimic the style and tone of the questions in RealQ, but do not limit the content to only what is asked in RealQ.
 6. If you do not have any more questions, please output "<No more question>".
 7. The output format must be JSON format.

Input:

RealQ: {realq}
Dialogue: {dialogue}
RecQ: {RecQ}

Output:

Figure 18: The prompt of *Inquirer* in Conversation-based Data Synthesis

- Task:
You are an AI assistant tasked with answering questions about Advertising. You will be provided with relevant document, and you need to generate questions based on that document.

- Input Description:
Document: <The document related to the question>
Question: <The question asked by the user>
Dialogue: <The last conversation between you and the user>

- Output Description:
{{"Answer": <Step-by-step detailed answer to the question, with clear logic, high relevant, completed, actionable and clarity in 100 words.>, "RecQ": <Recommend some questions based your answer and the dialogue.>}}

- Notes:

1. Provide clear, detailed and informative answers to the question ask by the user utilizing the given document in 100 words. Make sure your answer is different and has more informative with the dialogue. If your response includes other concepts, please address them as well.
2. Ensure that your questions and answers are coherent, well-structured, and easy to understand.
3. The recommended questions should be high-quality, in-depth, and highly relevant to your response, document, and dialogue.
4. The recommended questions should closely mimic the style and tone of the questions in RealQ, but do not limit the content to only what is asked in RealQ.

Input:
Document: {document}
RealQ: {realq}
Question: {question}
Dialogue: {dialogue}

Output:

Figure 19: The prompt of *Assistant* in Conversation-based Data Synthesis

- Task:
 You are a Conversation-based Refiner AI tasked with improving answers to questions. Your role is to refine previous answers by incorporating insights from relevant conversations, document information, and feedback. Your goal is to enhance the answer's accuracy, relevance, and clarity.

- Input Description:
 Question: <The original question asked by the user>
 Document: <The relevant document information>
 PreviousAnswer: <The answer that needs refinement>
 Conversation: <The conversation history related to the question>
 Feedback: <Any feedback provided on the previous answer>

- Output Description:
 {{"Answer": <Step-by-step detailed answer to the question, with clear logic, high relevant, completed, actionable and clarity in 100 words.>}}

- Instructions:
 1. Review the question, previous answer, relevant conversation, Document, and Feedback (if provided).
 2. Refine the previous answer following these guidelines:
 - Correct any inaccuracies in the previous answer
 - Remove any content not supported by the Document
 - Ensure all key information is supported by the Document
 - Incorporate relevant insights from the conversation and Feedback
 - Improve clarity, accuracy, and relevance to the question

Input:
 Question: "{question}"
 Document: "{Document}"
 PreviousAnswer: "{previousAnswer}"
 Conversation: "{conversation}"
 Feedback: "{Feedback}"

Output:

Figure 20: The prompt of Conversation-based Data Refinement

Task

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below.

You should first generate your answer based on the question.

You should choose the assistant that follows the user's instructions and answers the user's question better.

Begin your evaluation by comparing the two responses and provide a short explanation.

Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision.

Do not allow the length of the responses to influence your evaluation.

Do not favor certain names of the assistants.

Be as objective as possible.

Output format:

{{"verdict": "[[A]] if assistant A is better, [[B]] if assistant B is better, and [[C]] for a tie", "explanation": "The explanation goes here."}}

Input:

[User Question] {question}

[The Start of Assistant A's Answer] {answer_a} [The End of Assistant A's Answer]

[The Start of Assistant B's Answer] {answer_b} [The End of Assistant B's Answer]

Output:

Figure 21: The prompt of the evaluation prompt based on the relevant document

Task

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below.

You should first generate your answer based on the question.

You should choose the assistant that follows the user's instructions and answers the user's question better.

Begin your evaluation by comparing the two responses and provide a short explanation.

Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision.

Do not allow the length of the responses to influence your evaluation.

Do not favor certain names of the assistants.

Be as objective as possible.

Output format:

{{"verdict": "[[A]] if assistant A is better, [[B]] if assistant B is better, and [[C]] for a tie", "explanation": "The explanation goes here."}}

Input:

[User Question] {question}

[The Start of Assistant A's Answer] {answer_a} [The End of Assistant A's Answer]

[The Start of Assistant B's Answer] {answer_b} [The End of Assistant B's Answer]

Output:

Figure 22: The prompt of the winrate evaluation prompt