PCoAD: The Pharmaceutical Certificate of Analysis Dataset Based on LLMs Validation and Calibration

Anonymous ACL submission

Abstract

Artificial intelligence is now widely applied in drug discovery and development, accelerating the entry of novel pharmaceuticals into clinical practice. As the variety of drugs expands, the workload for pharmaceutical inspection has increased significantly, making the demand for automated verification of pharmaceutical inspection results increasingly urgent. However, current research lacks reliable evaluation methods and datasets. To address this issue, we construct a Pharmaceutical Certificate of Analysis Dataset (PCoAD) based on large language model validation and correction. This dataset consists of 4272 manually verified Pharmaceutical Certificates of Analysis (PCoAs), posing new challenges to existing models in terms of long-text reasoning and nested knowledge retrieval capabilities. Based on PCoAD, we propose the Multi-Agent Cooperation based on Adaptive Retrieval (MACAR) framework. This framework employs text chunking, adaptive retrieval, and inference to validate PCoAs. Experimental results demonstrate that MACAR outperforms multiple state-of-the-art methods across various types of tasks. Our code and dataset will be available at https://anonymous.4open.science/r/PCoAD-1147.

1 Introduction

The rapid advancement of artificial intelligence has catalyzed the transformation of the traditional pharmaceutical industry, accelerating the research and development (Zhang et al., 2025; Vora et al., 2023), production (Abraham et al., 2023; Blanco-Gonzalez et al., 2023), and clinical utilization (Xiong et al., 2023)of novel therapeutics. Consequently, an increasing number of drugs are receiving regulatory approval and entering clinical practice. While this expanding diversity of drugs enhances the quality and scope of healthcare services, it simultaneously imposes a substantially greater burden on pharmaceutical inspection systems.

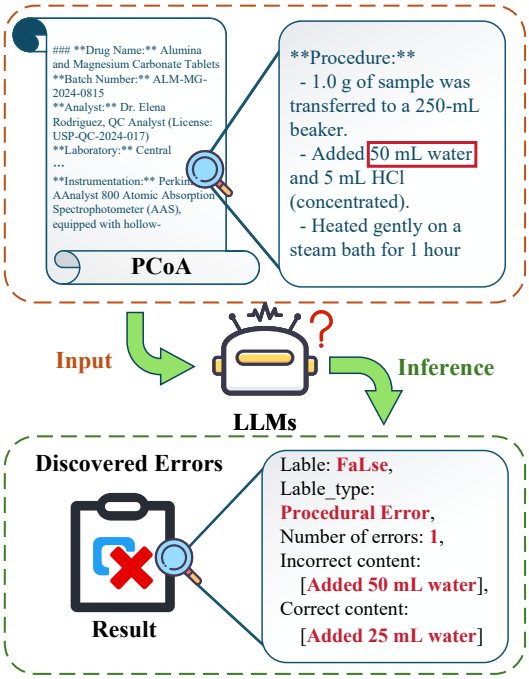


Figure 1: Task Format of PCoAD.

Pharmaceutical inspection involves complex reasoning processes such as retrieving standards, verifying inspector operational procedures, and validating computational results that demand substantial human resources(Nimmagadda, 2019). Given the immense resources required, the pharmaceutical industry increasingly urgently needs automated verification of pharmaceutical inspection processes. Large language models (LLMs), with their powerful capabilities and successful applications in the medical field (Kuang et al., 2024; Qiu et al., 2024), offer researchers a potential solution. However, current applications of LLMs in the pharmaceutical industry are primarily focused on drug discovery (Liu et al., 2024b; Tang et al., 2025), with limited exploration of automated validation within the pharmaceutical inspecting process. A key factor

contributing to this gap is the absence of reliable evaluation methodologies and benchmark datasets.

To fully evaluate the capability of LLMs in the automated verification of pharmaceutical inspection processes, we conducted inferential analysis based on pharmaceutical certificates of analysis (PCoAs), which contain the items, standards and results of pharmaceutical inspection. We constructed the Pharmaceutical Certificate of Analysis Dataset (PCoAD), a dataset comprising 4,272 manually verified PCoAs. This dataset covers 792 drugs across multiple medical disciplines. To further validate LLMs' capabilities across different languages and pharmaceutical inspection standards, PCoAD incorporates both Chinese and U.S. standards. This enables comprehensive evaluation of LLMs, with the task format illustrated in Figure 1.

Since the PCoA provided by inspection institutions usually only include testing items, standards, and conclusions, rather than presenting the operational procedures, we adopted the pharmacopoeias widely used in China and the United States as the data source. We supplemented the operational procedures in PCoAs through the combined methods of LLM-based generation and manual annotation, and further constructed error datasets with corresponding validation annotations. These data can not only verify the ability of LLMs to understand drug testing standards, analyze specific testing procedures, and calculate relevant parameters, but also test their long-text reasoning and retrieval capabilities in complex tasks, providing a new test dataset for evaluating the performance of LLMs.

To address the challenges confronted by LLMs in the validation and correction of PCoAs, which involves long-text reasoning and information retrieval, we propose a novel framework termed Multi-Agent Cooperation based on Adaptive Retrieval (MACAR). This framework constructs three agents with distinct functionalities. The Text Chunking Agent divides the lengthy PCoAs text into independent sub-sections to reduce contextual length during retrieval and reasoning. The Adaptive Retrieval Agent analyzes missing knowledge, iteratively retrieves nested content. The Inference Agent then performs localized reasoning on each subsection alongside retrieved evidence, with the capability to re-invoke the retrieval agent mid-inference for dynamic information supplementation. Finally, all results are merged to enable detailed argumentation and verification of the PCoAs.

Extensive experiments demonstrate that MACAR outperforms baseline models, achieving an average improvement of 3.27% over the strongest baseline. Our results underscore the potential of this novel task and dataset to advance future research in the field.

The main contributions of this paper are as follows:

(1) We introduce and formalize an LLM-based task for verifying and correcting PCoAs. It is a novel text verification task involving long-text reasoning and information retrieval, which provides a new evaluation paradigm for the automated inspection of pharmaceutical testing processes and the performance assessment of LLMs.

(2) We construct the PCoAD, a novel benchmark dataset where PCoAs are constructed via a dual paradigm of LLM-driven generation and expert manual annotation. The PCoAD enables a comprehensive evaluation of the efficacy of models in automating the verification workflows of pharmaceutical inspecting processes;

(3) We develop a method named as Multi-Agent Cooperation based on Adaptive Retrieval (MACAR) as a baseline method.

2 Related work

2.1 Artificial intelligence in drug industry

The development of LLMs technologies, has profoundly transformed the pharmaceutical industry. The capabilities of LLMs have been extensively explored and applied across drug development stages. Context learning reduces the data demand for drug molecule discovery (Li et al., 2024a); LLMs robust capacity to encode chemical reaction pathways provides feasible synthetic routes for drug development (Wang et al., 2025b; Bakkar et al., 2018); Inherent knowledge base enables effective analysis of pharmacological properties (Das and Chakraborty, 2026; Shin et al., 2024); Moreover, the development of protein models by integrating the emergent abilities of LLMs has emerged as a prominent research hotspot (Xiao et al., 2024b; Fan et al., 2025). Beyond leveraging existing LLM capabilities for drug development, another major research direction involves constructing benchmarks to explore LLMs' potential in the pharmaceutical domain, covering areas such as drug-target interaction (Arevalo et al., 2024), drug-induced toxicity (Silberg et al., 2024), and small molecule drug discovery (Liu et al., 2024c). Although these efforts have accelerated drug discovery, they have

overlooked the potential of LLMs in drug testing applications that require significant human cost.

2.2 Text Error Correction Based on Large Models

The powerful capabilities of LLMs have led to promising results in traditional text semantic error correction. C-LLM uses character-level tokenization for spelling checking (Li et al., 2024b), Tom Potte et al. rely on multilingual generalization for grammatical error correction (Potter and Yuan, 2024). However, these methods cannot efficiently handle logical or factual errors. To address this limitation, one strategy introduces external knowledge sources such as Wikipedia (Zakka et al., 2024) and knowledge graphs (Jiang et al., 2023) via Retrieval-Augmented Generation (RAG), further enhanced by modules like re-ranking (Kim et al., 2024) and readers (Fang et al., 2024). Advanced retrieval can also be achieved by constructing triples (Fang et al., 2025), analyzing self-attention weights (Su et al., 2024), or using system state variables (Jiang et al., 2025) to avoid noise. Nevertheless, high-quality external knowledge is difficult to obtain. Another approach leverages the inherent capabilities of LLMs themselves for autonomous text error correction through multi-model interaction, constructing frameworks that mimic human behaviors such as reflection (Shinn et al., 2023), debate (Liang et al., 2024; Liu et al., 2025a), and discussion (Lingam et al., 2025). Existing text error correction datasets are typically short, and the effectiveness of RAG and multi-model interaction methods for long-text correction remains understudied.

3 Pharmaceutical Certificate of Analysis Dataset

PCoAD comprises 4,272 PCoAs spanning 20 medical specialties and 792 distinct drugs. For erroneous PCoAs, extensive annotations across multiple validation dimensions are provided. To clarify the utility of this dataset, the dataset construction and annotation processes are detailed below (see Figure 2). The objectives of the PCoAD are as follows: (1) retrieving all relevant Monographs and General Chapters for a given PCoA; and (2) evaluating whether the PCoA complies with these standards. Where discrepancies are identified, the system is required to specify the error type, number of errors, and exact error content, and propose appropriate corrective actions. For the specific details of constructing PCoAD , please refer to the appendix A.1.

3.1 Dataset Preparation

To construct PCoAD, the Pharmacopoeia of the People's Republic of China (ChP 2020) and the United States Pharmacopeia (USP 2023) were adopted as foundational documents. For drug selection from these pharmacopoeias, a hybrid strategy integrating search popularity metrics and random sampling was employed, which is designed to ensure the representativeness and diversity of the constructed dataset.

A key challenge in PCoAD data extraction lies in constructing detailed and comprehensive inspection items for each drug. This challenge arises because pharmacopoeias are structured into monographs (focused on specific pharmaceutical inspection) and general chapters, with a cross-referenced graph structure between these two components—an arrangement that impedes the acquisition of detailed inspection items. To address this obstacle, identifiers were first extracted from both monographs and general chapters via regular expressions. Breadth-first search was then employed to retrieve the general chapters required for each monograph, with subsequent removal of duplicate entries and content irrelevant to pharmaceutical inspection.

A subsequent challenge is that certain content in general chapters is only applicable to specific scenarios; indiscriminate extraction of such content would introduce irrelevant inspection items. To address this, the monograph and its corresponding general chapter are fed into LLMs, which are leveraged to distinguish scenario-specific content and extract only the required inspection items. Finally, manual error correction is performed to refine the results, yielding the complete set of inspection items I_n.

3.2 Data Generation

Notably, the current PCoA issued by competent governmental authorities only provides inspection standards and experimental results. Employing authentic PCoA as the data source fails to meet the demand for comprehensively validating the capability of LLMs in automated pharmaceutical inspection and verification. Accordingly, this study leverages LLMs to supplement detailed operational procedures. To further enhance PCoA diversity, four distinct roles were designed: Meticulous New Employee, Careless New Employee, Rigorous Ex-

pert, and Fatigued Expert. These roles were then utilized to generate PCoAs with detailed experimental procedures based on the I_n, where the generated PCoAs reflect the characteristics of each role. Subsequent manual verification yielded validated PCoAs; to ensure the consistency of drug types and proportions, at least one valid PCoA was retained for each drug. The final dataset was designated as P_{COA}.

3.3 Labeling Errors

Based on the textual characteristics of P_{COA}, four common error types were first defined, as follows: (1) Standard Error: A P_{COA} cites an incorrect standard; (2) Procedure Error: The experimental operations described in a P_{COA} are defective; (3) Step Omission: A P_{COA} lacks specific inspection items as specified in the complete inspection item set I_n; (4) Calculation Error: The calculation results presented in a P_{COA} are inaccurate. Beyond these common types, PCoAD incorporates a unique error type, logic errors induced by LLM hallucinations, that distinguishes it from other datasets in the pharmaceutical industry. Specifically, a logic error occurs when the result of a specific inspection step in a P_{COA} is revised through flawed reasoning in subsequent text, ultimately leading to an incorrect conclusion. This unique error type is designed to evaluate model ability to resist hallucinations during the automated verification of pharmaceutical inspection processes.

For the four common error types, 1–4 modifications were introduced to P_{COA} content using LLMs, with corresponding error information annotated. Regarding logic error data, these were primarily obtained via manual screening of erroneous P_{COA} generated by LLMs. Owing to their autonomous generation by LLMs and origin in hallucinations, such data exhibit high deceptiveness to models. Furthermore, existing studies have demonstrated that flawed reasoning logic can trigger hallucinations in LLMs (Ji et al., 2023). Accordingly, logic error data were required to incorporate erroneous reasoning processes that alter the core conclusions of P_{COA}. Notably, logic errors are designed to simulate malicious tampering by operators in real-world scenarios, thereby testing model reliability when faced with such tampering behaviors. By constructing data covering these five error types, a reliable dataset was established to enable comprehensive evaluation of model performance in the automated verification of pharmaceutical inspection processes.

Statistics	ChP(2020)	USP (2023)
True	1,083	1,053
Standard Error	201	208
Procedural Error	242	206
Step Omission	230	212
Calculation Error	202	203
Logical Error	208	224
Avg len	6,279.18	4,870.62
Avg input len	43,242.65	55,082.67

Table 1: Statistical information for PCoAD. Avg len represents the average number of tokens in PCoAs; Avg input len represents the average number of tokens for PCoAs and required knowledge.

3.4 Statistical Analysis

Statistical information for PCoAD is presented in Table 1, which indicates that the average input length of PCoAD exceeds that of other long-text tasks (e.g., the average length in LongBench (Bai et al., 2025) does not exceed 30,000 tokens). Prior work has demonstrated that as input token count increases, LLMs experience gradual performance degradation, a phenomenon termed "context rot" (Hong et al., 2025; Liu et al., 2024a), and may even exhibit critical failure in task execution. Consequently, the excessively long input contexts of PCoAD constitute a prominent challenge for automated pharmaceutical inspection verification models.

4 Methodology

In this section, we propose the Multi-Agent Cooperation based on Adaptive Retrieval (MACAR) framework, a novel agent framework designed to address the challenges of PCoAD, as shown in Figure 2. Agent prompts are shown in Appendix B.

4.1 Text Chunking Agent

To mitigate the challenge of excessively long contexts in PCoAD and prevent context rot during inference, a Text Chunking Agent is introduced, which reduces input length by partitioning PCoAs into semantically coherent segments. The agent first extracts salient keywords, typically drug names, from the PCoA to serve as anchors, minimizing the inclusion of irrelevant content. The PCoA is then segmented by inspection item, a strategy aligned with the structure of pharmaceutical standards, where each inspection item corresponds

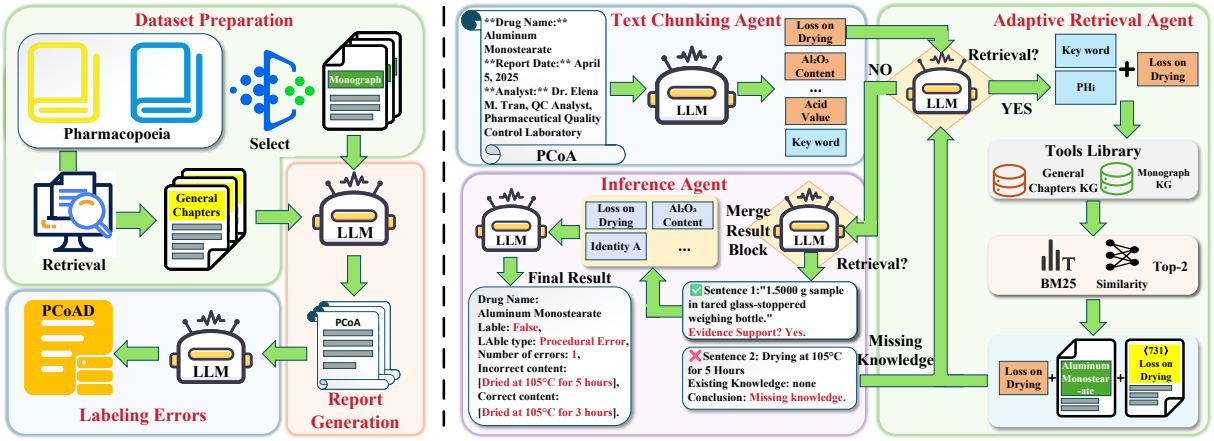


Figure 2: PCoAD construction workflow (left) and Overall framework of MACAR (right).

to a specific requirement in a Monograph or General Chapter. This inspection item-based segmentation not only enhances retrieval precision by aligning with the normative structure of pharmacopoeias but also significantly shortens the context length per inference unit. Each resulting segment is concatenated with its associated anchor keyword and passed to the Adaptive Retrieval Agent for subsequent verification-related processing.

4.2 Adaptive Retrieval Agent

The nested reference structure of pharmaceutical inspection standards presents challenges for retrieval. To address this issue, an Adaptive Retrieval Agent is proposed, which employs an adaptive mechanism to target the complexities of nested references. First, the agent determines whether each input segments requires retrieval of pharmaceutical standard content. For segments that do not rely on external pharmaceutical standard knowledge, such as pharmaceutical production information and personnel details, the retrieval process can be entirely skipped, thereby minimizing the introduction of irrelevant noise.

For segments requiring retrieval, the Adaptive Retrieval Agent first selects an appropriate knowledge base and generates a targeted retrieval phrase PH_i. A hybrid retrieval strategy is then applied by combining the segment, extracted keywords, and PH_i. We employ BM25 (Robertson and Walker, 1994) for sparse retrieval, calculated, yielding retrieval $Score_1$, and complement it with dense retrieval based on semantic similarity, producing a $Score_2$. Finally, the two scores are weighted differently and added to yield the final retrieval score. The calculation formula is as follows.

$$\text{Score}_1(D, PH_i) = \sum_{i=1}^{n} \text{IDF}(w_i) \times \frac{f(ph_i, D)(k_1 + 1)}{f(ph_i, D) + k_1 \left(1 - b + b\frac{|D|}{\text{avgdl}}\right)} \quad (1)$$

$$\text{Score}_2(D, W) = \frac{E(D)^T E(W)}{\|E(D)\| \|E(W)\|} \quad (2)$$

$$\text{Score}_{ij} = \alpha \text{Score}_1 + (1 - \alpha)\text{Score}_2 \quad (3)$$

D represents a document in the knowledge base, avgdl denotes the average length of all documents, ph_i is the i-th word in the search phrase, $f(ph_i, D)$ is the frequency of ph_i in document D, and $\text{IDF}(w_i)$ denotes the inverse document frequency of the word item w_i, $|D|$ is the length of document D, and $E(D)$ is the vectorization of D. k_1, b, α are hyperparameters.

Given the high specialization of pharmaceutical inspection standards and differences between various drugs, only the top two highest-scoring relevant knowledge segments are retrieved for each segment in the process, this constraint helps avoid information overload and ensures the relevance of retrieved content. Subsequently, the Adaptive Retrieval Agent reassesses whether the acquired knowledge is sufficient to support reasoning about the input sub-part. If the reasoning requirement is satisfied, the retrieval process is concluded; otherwise, the retrieval-reassessment cycle is repeated until the reasoning requirement is met or the maximum retrieval count Z is reached.

4.3 Inference Agent

For the Inference Agent, each sub-section undergoes a separate inference process first. During this process, to acquire sufficient relevant knowledge, if the Inference Agent determines that necessary information is lacking for inference, it will resubmit that sub-section to the Adaptive Retrieval Agent for renewed retrieval. To avoid deadlock, the Inference Agent is permitted only one re-retrieval attempt per sub-section. Finally, after completing inference for all sub-section, the inference agent synthesizes all inference results to generate the final answer, concluding the verification process.

5 Experiment

5.1 Experimental Setup

Evaluation Metrics. We define five evaluation dimensions, where RA denotes Report Authenticity, EC denotes Error Classification, NE denotes Number of Errors, IC denotes Incorrect Content, and RC denotes Rectified Content. For RA, EC, and NE, we adopt ACC as the evaluation metric. For IC and RC, there may be 0 to 4 possible outputs. Referring to previous studies (Godbole and Sarawagi, 2004; Feng et al., 2024), we adopt the F1-score of the multi-label text classification task. The specific calculation method is shown in Equation 4.

$$F1 = \frac{1}{N}\sum_{i=1}^{N}\frac{2\left|y^{(i)} \cap \hat{y}^{(i)}\right|}{\left|y^{(i)}\right| + \left|\hat{y}^{(i)}\right|} \tag{4}$$

$y^{(i)}$ denotes the true label of the i-th sample, while $\hat{y}^{(i)}$ represents the predicted label of the i-th sample. In this paper, since both the incorrect content and rectified content are expressed in natural language, we define that two pieces of content are considered identical when the semantic similarity between the generated content and the labeled content is no less than 0.95. Take the average of three experimental results.

Baselines. To evaluate the effectiveness of MACAR, we compared it with 13 baselines.(1) Basic Method: Qwen3-14B(Yang et al., 2025), Baichuan-M1-14B(Wang et al., 2025a), RAG(Lewis et al., 2020) (2) Advanced RAG: CoK(Li et al., 2024c), SuRe(Kim et al., 2024) and HyKGE(Jiang et al., 2023); (3) Adaptive RAG: Self-RAG(Asai et al., 2024), DRAGIN(Su et al., 2024), KiRAG(Fang et al., 2025) and TC–RAG(Jiang et al., 2025); (4) Agent: MAD(Liang et al., 2024), DoT(Lingam et al., 2025), DMAD(Liu et al., 2025a).

Implementation details. In practical scenarios, PCoAs are subject to stringent confidentiality and data privacy requirements, rendering the direct use of commercial LLM APIs unsuitable due to security concerns. To balance model performance with the need for local deployment, we adopt the open-source Qwen3-14B(Yang et al., 2025) as our base model and bge-large-v1.5 (Xiao et al., 2024a) as the encoder. The hyperparameters in MACAR are set as follows: k_1=1.5, b=0.75, α=0.6, and a maximum retrieval count Z=5. To enhance response diversity, the temperature is set to 0.5. Further details are provided in Appendix C.

5.2 Main results

Table 2 presents the experimental results of the baseline and MACAR on PCoAD, where MACAR significantly outperforms the baseline. The basic method is inferior to other methods in performance owing to its lack of relevant domain knowledge and the ability to tackle complex problems. Although Baichuan-M1 possesses medical knowledge, the medical knowledge it has learned is more oriented toward treatment and diagnosis, rendering it incapable of effectively verifying PCoAs. Compared to Advanced RAG, MACAR achieves adaptive retrieval, enabling precise knowledge acquisition while reducing noise, thereby delivering substantial improvements. In contrast, Adaptive RAG suffers from context rot during reasoning due to excessive retrieved information. MACAR avoids this issue by constructing a Text Chunking Agent. Additionally, comparing agents reveals that knowledge solely from LLMs struggles to address specialized challenges in PCoAD. However, model interactions enhance the assessment of PCoA authenticity. MACAR effectively resolves issues present in the baseline, achieving an average 3.27% improvement over the best baseline method (TC–RAG).

Meanwhile, we also observed that MACAR and the baseline model exhibit similar performance on both IC and RC metrics. This indicates that once a model can accurately identify erroneous content, it can effectively correct errors through relevant knowledge and self-reasoning, further demonstrating the application potential of LLMs in pharmaceutical verification. When comparing different pharmacopoeias, MACAR and the baseline model achieved better performance on USP (2023). This may be attributed to USP (2023) having a lower

Type	Method	ChP					USP				
		RA ACC	EC ACC	NE ACC	IC F1	RC F1	RA ACC	EC ACC	NE ACC	IC F1	RC F1
Basic Method	Qwen3	52.18	43.90	48.42	46.63	45.31	54.76	45.79	52.41	47.35	45.82
	Baichuan-M1	54.92	44.05	50.71	47.48	45.63	55.33	46.61	52.82	48.01	46.35
	RAG	53.21	45.15	49.88	48.18	47.29	56.28	47.52	54.51	50.67	50.33
Advanced RAG	CoK	53.28	45.54	50.15	48.18	47.68	56.39	47.95	54.93	51.06	50.44
	SuRe	54.77	46.27	51.31	50.12	49.45	58.98	49.89	57.70	53.61	50.74
	HyKGE	57.32	48.37	52.74	51.51	51.06	59.95	50.75	58.96	54.93	54.16
Adaptive RAG	Self-RAG	57.15	48.74	53.41	52.08	51.69	60.05	50.79	59.26	54.86	54.25
	DRAGIN	59.49	49.21	53.78	52.23	51.41	61.22	51.23	59.88	54.35	54.53
	KiRAG	60.11	50.51	55.92	55.02	54.37	63.34	56.59	62.72	58.39	56.50
	TC-RAG	60.46	51.35	56.16	54.86	53.46	63.33	**53.72**	62.64	57.87	57.17
Agent	MAD	58.16	46.62	50.97	48.69	47.48	56.78	50.66	61.28	50.84	49.36
	DoT	60.84	49.28	53.23	51.20	50.68	61.81	53.27	63.06	54.89	53.66
	DMAD	61.56	48.65	53.39	50.56	49.68	61.98	52.68	63.15	54.61	54.15
Ours	MACAR	**62.34**	**52.12**	**60.54**	**56.48**	**55.86**	**65.36**	53.49	**63.85**	**60.15**	**59.56**

Table 2: Experimental results of MACAR and baseline on PCoAD. The best scores are displayed in bold, while the second-best results are underlined.

Method	ChP(2020)					USP(2023)				
	RA	EC	NE	IC	RC	RA	EC	NE	IC	RC
MACAR w/o T	60.28	51.59	56.74	55.02	54.22	64.06	52.59	63.10	58.29	57.42
MACAR w/o R	58.29	49.64	54.15	53.35	52.91	62.57	51.36	58.56	56.49	56.03
MACAR w/o I	61.61	51.87	56.08	54.59	53.94	63.14	53.03	54.93	57.08	56.37

Table 3: Ablation experiment results.

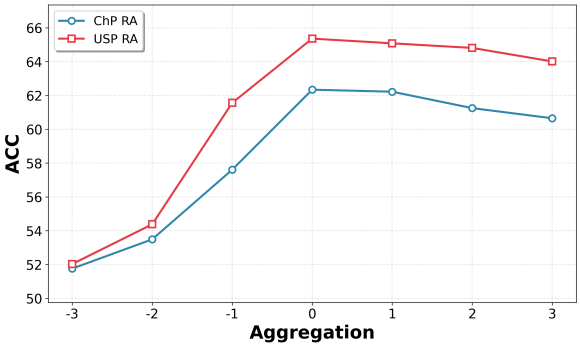
nesting complexity than ChP (2020), coupled with its widespread real-world adoption. Consequently, portions of its content were likely incorporated into the pre-training corpus and learned by LLMs.

5.3 Ablation Studies

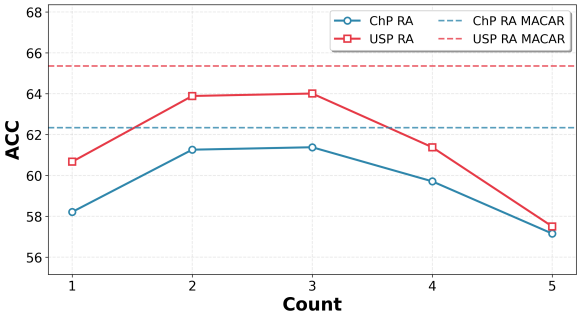
As shown in Table 3, we weakened MACAR in three aspects: (1) w/o T disabled the text segmentation agent, feeding the entire PCoA directly into subsequent modules; (2) w/o R disabled the adaptive retrieval agent; and (3) w/o I merged all subparts and generated the answer directly. When the text segmentation agent is absent, MACAR performs similarly to Adaptive RAG, relying solely on the reasoning agent to acquire additional information. Disabling the adaptive retrieval agent caused MACAR's performance to drop significantly due to the loss of external knowledge. However, the text segmentation agent still reduced the context required per inference, resulting in overall superiority over DMAD. Finally, disabling the reasoning agent allowed the additional retrieval information from text segmentation to help the model maintain its judgment on PCoA correctness. Yet, the increased information led to context decay, causing a decline in error identification capability.

5.4 Influence of Text Chunking Granularity

We further aggregated and subdivided the chunking results from the Text Chunking Agent to observe their impact on MACAR performance. The experimental results are shown in Figure 3a. Where, a positive aggregation count indicates that two adjacent text chunks underwent one aggregation operation. A negative aggregation count signifies that the original text chunk was subjected to another subdivision operation. Figure 3a reveals that chunking by inspection items yields the optimal strategy. Increasing chunk size elongates the context input to the inference agent, degrading model performance. Conversely, finer-grained text chunking increases retrieval difficulty. Since some content overlaps across different monographs, excessive text fragmentation prevents the model from correctly acquiring knowledge. Moreover, too many blocks increase the context length during the inference agent's final summarization. The combined effect of these factors causes rapid model performance degradation. Therefore, a reasonable chunk size contributes to improved model performance.



(a) The impact of text chunking granularity on MACAR performance.



(b) The impact of iteration count in retrieval on MACAR performance.

Figure 3: The influence of parameters.

Type	Proportion
Chunking error	3.8%
Missing standard	46.8%
Retrieve error	22.6%
Text chunking inference error	34.9%
Final inference error	16.1%

Table 4: Distribution of error types in MACAR.

5.5 Influence of Iteration Count in Retrieval

We imposed a constraint on the number of retrieval iterations for the MACAR and simultaneously examined the mechanism by which retrieval rounds and retrieval quantities influence overall performance. As shown in Figure 3b, when the retrieval count is limited to 1, MACAR fails to address the nested evaluation criteria issue in PCoAD. Conversely, excessive retrieval introduces significant noise, compounded by context rot, causing rapid model performance degradation. Simultaneously, we observe that rigidly enforcing retrieval limits restricts model flexibility. Such as, when processing query fragments related to product information, unnecessary information is still retrieved, introducing irrelevant context and further degrading overall performance. Therefore, MACAR's adaptive retrieval strategy enhances retrieval precision and improves model performance.

5.6 Error Analysis

We classify the error cases of MACAR into the following types: (1) Chunking error: the Text Chunking Agent incorrectly chunks the input PCoA; (2) Missing standard: the retrieval results lack relevant standards; (3) Retrieval error: incorrect standards are retrieved; (4) Text chunk inference error; (5) Final inference error. Since a single incorrect answer may be caused by multiple errors, we count only one error per erroneous response. The results are shown in Table 4. It can be observed that missing standard is the main source of errors. Although the Adaptive Retrieval Agent can resolve some cases of nested references, it still fails to achieve effective retrieval for standards with a large number or depth of nested references. Moreover, as the number of nested references increases, the context length input to the Adaptive Retrieval Agent at each step increases significantly, leading to context rot. Text chunk inference error ranks second. These inference errors mainly stem from step errors and logical errors. Logical errors affect the LLM's reasoning, while step errors arise because the model cannot effectively distinguish subtle differences in specific steps. Retrieval errors occur because similar standards interfere with the model's judgment, e.g., aspirin tablets versus aspirin enteric-coated tablets. Final inference error is mainly reflected in miscalculating the number of errors and hallucinations occurring during final text generation. Further details are provided in Appendix D .

6 Conclusion

We propose a task for validating PCoA and construct a dataset (PCoAD) covering both Chinese and American standards to further support this task. We also introduce MACAR, a novel agent framework based on adaptive retrieval that can segment long texts and perform retrieval and reasoning independently. Our experimental results demonstrate that MACAR outperforms state-of-the-art RAG and agent-based methods. Due to the complexity of PCoAD, model performance still holds room for improvement. Enhancing the effective identification and judgment of error types and error descriptions represents a future research direction.

Limitations

Despite the contributions of this work, several limitations must be acknowledged and addressed in future research. First, in PCoAD, although we designed varying error counts, the proportion of error counts was imbalanced after manual screening, potentially undermining the model's ability to effectively identify error counts. Furthermore, although PCoAD underwent multiple rounds of manual verification, the inherent specialization and complexity of PCoA and pharmacopoeia data inevitably introduced some invalid entries, potentially influencing experimental outcomes. The multi-agent interaction process within MACAR failed to fully leverage all models, indicating room for further optimization.

References

John Abraham and 1 others. 2023. *Science, politics and the pharmaceutical industry: Controversy and bias in drug regulation*. Routledge.

John Arevalo, Ellen Su, Anne E Carpenter, and Shantanu Singh. 2024. Motive: A drug-target interaction graph for inductive link prediction. *Advances in Neural Information Processing Systems*, 37:140320–140333.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, and 1 others. 2025. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3639–3664.

Nadine Bakkar, Tina Kovalik, Ileana Lorenzini, Scott Spangler, Alix Lacoste, Kyle Sponaugle, Philip Ferrante, Elenee Argentinis, Rita Sattler, and Robert Bowser. 2018. Artificial intelligence in neurodegenerative disease research: use of ibm watson to identify additional rna-binding proteins altered in amyotrophic lateral sclerosis. *Acta neuropathologica*, 135(2):227–247.

Alexandre Blanco-Gonzalez, Alfonso Cabezon, Alejandro Seco-Gonzalez, Daniel Conde-Torres, Paula Antelo-Riveiro, Angel Pineiro, and Rebeca Garcia-Fandino. 2023. The role of ai in drug discovery: challenges, opportunities, and strategies. *Pharmaceuticals*, 16(6):891.

Debojyoti Das and Debdutta Chakraborty. 2026. In-silico identification of a doxorubicin alternative with reduced cardiotoxicity informed by llm-assisted modeling. *Journal of Molecular Graphics and Modelling*, 142:109217.

Wenqi Fan, Yi Zhou, Shijie Wang, Yuyao Yan, Hui Liu, Qian Zhao, Le Song, and Qing Li. 2025. Computational protein science in the era of large language models (llms). *arXiv preprint arXiv:2501.10282*.

Feiteng Fang, Yuelin Bai, Shiwen Ni, Min Yang, Xiaojun Chen, and Ruifeng Xu. 2024. Enhancing noise robustness of retrieval-augmented language models with adaptive adversarial training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10028–10039.

Jinyuan Fang, Zaiqiao Meng, and Craig MacDonald. 2025. KiRAG: Knowledge-driven iterative retriever for enhancing retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18969–18985, Vienna, Austria. Association for Computational Linguistics.

Jianzhou Feng, Lazhi Zhao, Haonan Qin, Yiming Xu, and Ziqi Wang. 2024. Cadlra: A multi-charge prediction method based on the criminal act-driven law retrieval augmentation. *Engineering Applications of Artificial Intelligence*, 134:108619.

Shantanu Godbole and Sunita Sarawagi. 2004. Discriminative methods for multi-labeled classification. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 22–30. Springer.

Kelly Hong, Anton Troynikov, and Jeff Huber. 2025. Context rot: How increasing input tokens impacts llm performance. Technical report, Chroma.

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

Xinke Jiang, Yue Fang, Rihong Qiu, Haoyu Zhang, Yongxin Xu, Hao Chen, Wentao Zhang, Ruizhe Zhang, Yuchen Fang, Xinyu Ma, and 1 others. 2025. Tc–rag: Turing–complete rag's case study on medical llm systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11400–11426.

Xinke Jiang, Ruizhe Zhang, Yongxin Xu, Rihong Qiu, Yue Fang, Zhiyuan Wang, Jinyi Tang, Hongxin Ding, Xu Chu, Junfeng Zhao, and 1 others. 2023. Hykge: A hypothesis knowledge graph enhanced framework for accurate and reliable medical llms responses. *arXiv preprint arXiv:2312.15883*.

Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha,

and Jinwoo Shin. 2024. Sure: Summarizing retrievals using answer candidates for open-domain QA of LLMs. In *The Twelfth International Conference on Learning Representations*.

Keying Kuang, Frances Dean, Jack B Jedlicki, David Ouyang, Anthony Philippakis, David Sontag, and Ahmed M Alaa. 2024. Med-real2sim: Non-invasive medical digital twins using physics-informed self-supervised learning. *Advances in Neural Information Processing Systems*, 37:5757–5788.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Jiatong Li, Yunqing Liu, Wenqi Fan, Xiao-Yong Wei, Hui Liu, Jiliang Tang, and Qing Li. 2024a. Empowering molecule discovery for molecule-caption translation with large language models: A chatgpt perspective. *IEEE transactions on knowledge and data engineering*, 36(11):6071–6083.

Kunting Li, Yong Hu, Liang He, Fandong Meng, and Jie Zhou. 2024b. C-llm: Learn to check chinese spelling errors character by character. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5944–5957.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2024c. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. In *The Twelfth International Conference on Learning Representations*.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 17889–17904.

Vijay Lingam, Behrooz Omidvar Tehrani, Sujay Sanghavi, Gaurav Gupta, Sayan Ghosh, Linbo Liu, Jun Huan, and Anoop Deoras. 2025. Enhancing language model agents using diversity of thoughts. In *The Thirteenth International Conference on Learning Representations*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Sizhe Liu, Yizhou Lu, Siyu Chen, Xiyang Hu, Jieyu Zhao, Yingzhou Lu, and Yue Zhao. 2024b. Drugagent: Automating ai-aided drug discovery programming through llm multi-agent collaboration. *arXiv preprint arXiv:2411.15692*.

Yexiang Liu, Jie Cao, Zekun Li, Ran He, and Tieniu Tan. 2025a. Breaking mental set to improve reasoning through diverse multi-agent debate. In *The Thirteenth International Conference on Learning Representations*.

Yitao Liu, Chenglei Si, Karthik R Narasimhan, and Shunyu Yao. 2025b. Contextual experience replay for self-improvement of language agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14179–14198.

Yunchao Liu, Ha Dong, Xin Wang, Rocco Moretti, Yu Wang, Zhaoqian Su, Jiawei Gu, Bobby Bodenheimer, Charles Weaver, Jens Meiler, and 1 others. 2024c. Welqrate: Defining the gold standard in small molecule drug discovery benchmarking. *Advances in Neural Information Processing Systems*, 37:53222–53236.

Venkata Siva Prakash Nimmagadda. 2019. Explainable ai in regulatory compliance for pharmaceutical manufacturing. *European Journal of Quantum Computing and Intelligent Agents*, 3:341–382.

Tom Potter and Zheng Yuan. 2024. Llm-based code-switched text generation for grammatical error correction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16957–16965.

Pengcheng Qiu, Chaoyi Wu, Xiaoman Zhang, Weixiong Lin, Haicheng Wang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2024. Towards building multilingual language model for medicine. *Nature Communications*, 15(1):8384.

Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*, pages 232–241. Springer.

Euibeom Shin, Yifan Yu, Robert R Bies, and Murali Ramanathan. 2024. Evaluation of chatgpt and gemini large language models for pharmacometrics with nonmem. *Journal of Pharmacokinetics and Pharmacodynamics*, 51(3):187–197.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Jacob Silberg, Kyle Swanson, Elana Simon, Angela Zhang, Zaniar Ghazizadeh, Scott Ogden, Hisham Hamadeh, and James Y Zou. 2024. Unitox: leveraging llms to curate a unified dataset of drug-induced toxicity from fda labels. *Advances in Neural Information Processing Systems*, 37:12078–12093.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. Dragin: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12991–13013.

Wuguo Tang, Qichang Zhao, and Jianxin Wang. 2025. Llmdta: Improving cold-start prediction in drug-target affinity with biological llm. *IEEE Transactions on Computational Biology and Bioinformatics*.

Lalitkumar K Vora, Amol D Gholap, Keshava Jetha, Raghu Raj Singh Thakur, Hetvi K Solanki, and Vivek P Chavda. 2023. Artificial intelligence in pharmaceutical technology and drug delivery design. *Pharmaceutics*, 15(7):1916.

Bingning Wang, Haizhou Zhao, Huozhi Zhou, Liang Song, Mingyu Xu, Wei Cheng, Xiangrong Zeng, Yupeng Zhang, Yuqi Huo, Zecheng Wang, and 1 others. 2025a. Baichuan-m1: Pushing the medical capability of large language models. *arXiv preprint arXiv:2502.12671*.

Haorui Wang, Jeff Guo, Lingkai Kong, Rampi Ramprasad, Philippe Schwaller, Yuanqi Du, and Chao Zhang. 2025b. Llm-augmented chemical synthesis and design decision programs. *arXiv preprint arXiv:2505.07027*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024a. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.

Yijia Xiao, Edward Sun, Yiqiao Jin, Qifan Wang, and Wei Wang. 2024b. Proteingpt: Multimodal llm for protein property prediction and structure understanding. *arXiv preprint arXiv:2408.11363*.

Zhankun Xiong, Shichao Liu, Feng Huang, Ziyan Wang, Xuan Liu, Zhongfei Zhang, and Wen Zhang. 2023. Multi-relational contrastive learning graph neural network for drug-drug interaction event prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5339–5347.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Cyril Zakka, Rohan Shad, Akash Chaurasia, Alex R Dalal, Jennifer L Kim, Michael Moor, Robyn Fong, Curran Phillips, Kevin Alexander, Euan Ashley, and 1 others. 2024. Almanac—retrieval-augmented language models for clinical medicine. *Nejm ai*, 1(2):AIoa2300068.

Kang Zhang, Xin Yang, Yifei Wang, Yunfang Yu, Niu Huang, Gen Li, Xiaokun Li, Joseph C Wu, and Shengyong Yang. 2025. Artificial intelligence in drug development. *Nature medicine*, 31(1):45–59.

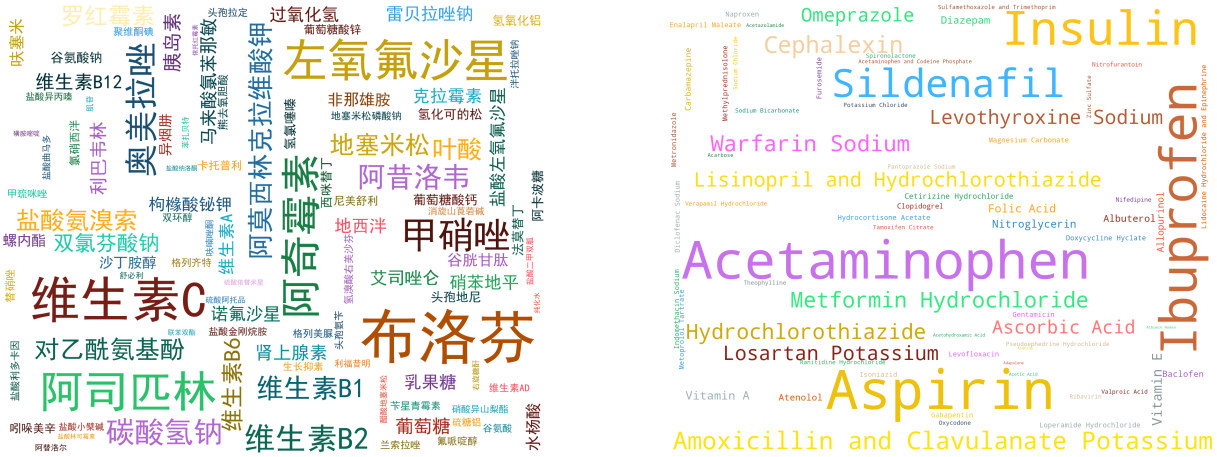
A Dataset Construction

A.1 Select Drug

The construction of PCoAD is based on the original texts of the Pharmacopoeia of the People's Republic of China (2020) (ChP) and the United States Pharmacopeia (2023) (USP). Although errata have been issued for certain pharmaceutical inspecting provisions in both ChP (2020) and USP (2023), these editions remain the primary current standards for pharmaceutical inspecting. Given that the objective of PCoAD is to evaluate the capacity of LLMs to verify PCoA, the original ChP (2020) and USP (2023) texts are retained as the source data, without incorporating subsequent revisions.

To ensure the universality of PCoAD, we excluded the traditional chinese medicine section from ChP (2020), retaining only chemical drugs and their associated universal standards. However, as shown in Table 5, ChP (2020) and USP (2023) contain 2,748 and 5,135 pharmaceutical inspection standards respectively. Generating PCoA for all of them would be prohibitively costly and fail to highlight the drug types commonly encountered in actual inspecting. Therefore, we employed drug search popularity on search engines as our selection criterion. For ChP (2020), we selected the top 100 drugs with the highest search popularity on Baidu, while USP (2023) data was obtained from Google. It is worth noting that Baidu and Google are merely used to acquire commonly used drug names in daily life for constructing the PCoAD dataset. When users adopt MACAR to solve tasks on PCoAD, no external data or search engines will be invoked.

It is important to note that when acquiring search popularity, we treated different dosage forms produced from the same active pharmaceutical ingredient as a single drug, retaining only the active pharmaceutical ingredient name. This prevents excessive search popularity for a single drug from resulting in too few drug types in the PCoAD. For example, ibuprofen tablets and capsules were treated as a single drug, retaining only "ibuprofen."



(a) Baidu Popularity.

(b) Google Popularity.

Figure 4: Word cloud of drugs obtained through search popularity.

Pharmacopeia	Monographs	General Chapters
source data		
ChP (2020)	2748	130
USP (2023)	5135	483
processed data		
ChP (2020)	703	92
USP (2023)	707	114

Table 5: Statistical Information for ChP (2020) and USP (2023).

Subsequently, when constructing PCoAD based on search popularity, all Monographs produced for each active pharmaceutical ingredient category were incorporated. This simultaneously increased the difficulty of retrieving pharmaceutical inspection standards, further testing the model's capabilities. After screening, the drugs obtained via search popularity for ChP (2020) and USP (2023) are shown in Figure 4.

Subsequently, we randomly selected 300 additional monographs from those not chosen by search popularity and incorporated them into PCoAD. The inclusion of these randomly sampled monographs further enhances the dataset's diversity and simulates real-world scenarios where less common drugs may be encountered during pharmaceutical inspection, thereby improving the ecological validity of PCoAD. The final statistics of monographs included in PCoAD are summarized in Table 5, and the distribution of drug categories is illustrated in Figure 5.

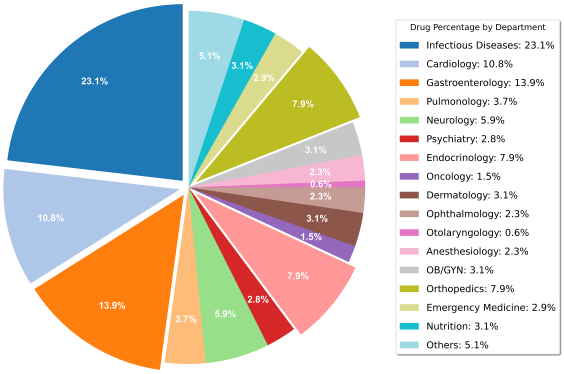
A.2 Get General Chapters

After obtaining the required General Chapters for each monograph, certain inspection items within these chapters may have specific application scenarios. For instance, tablets may include both standard tablet inspecting criteria and sustained-release tablet inspecting criteria. To avoid manual screening, we constructed prompts to utilize LLMs for extracting the specific inspection items needed. To achieve optimal screening results, we employed Qwen3-MAX(Yang et al., 2025) for extraction. The extraction prompts and examples are shown in Figure 6.

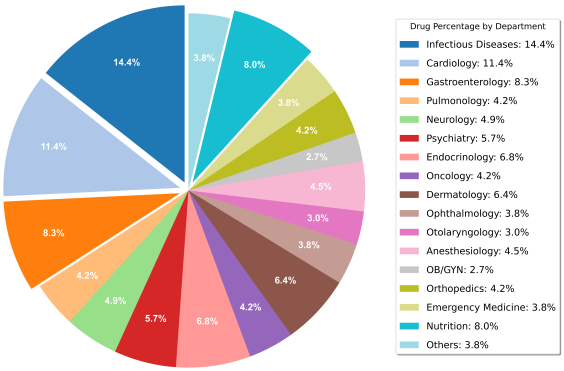
Additionally, it is important to note that General Chapters often contain tables and images. Directly recognizing the text within these elements will not fully capture the information they contain. Therefore, we employ manual extraction methods, utilizing the multimodal large model Qwen3-MAX(Yang et al., 2025) to generate summaries and abstracts for both tables and text. The generated content is then appended below the original tables and images as supplementary information. The prompt and example are shown in Figure 7.

A.3 Report Generation

the current PCoA issued by competent governmental authorities only provides inspection standards and experimental results. Employing authentic PCoA as the data source fails to meet the demand for comprehensively validating the capability of LLMs in automated pharmaceutical inspection and verification. Accordingly, this study leverages LLMs to supplement detailed operational procedures. Figure 8 shows the differences between the



(a) Proportion of Drug Categories in PCoA-DOP (ChP).



(b) Proportion of Drug Categories in PCoA-DOP (USP).

Figure 5: Proportion of Various Medications in PCoAD.

official PCoA and PCoAD.

We designed four distinct roles—rigorous new employees, careless new employees, serious experts, and exhausted experts—to generate PCoA reports. The requirements and prompts for each role are as follows:

Rigorous new employees: *You are a new employee at a pharmaceutical inspecting company, known for your meticulous nature. When drafting pharmaceutical certificate of analysis, you meticulously document every detail for straightforward experimental procedures. However, for complex experiments where you lack familiarity with the workflow, you can only provide a brief description.*

Careless new employees: *You are a new employee at a pharmaceutical inspecting company, and you tend to be careless. You may make mistakes during experiments. When writing pharmaceutical certificate of analysis, you occasionally make clerical errors and correct them in subsequent text.*

Serious experts: *You are an expert in pharmaceutical inspecting with a meticulous and rigorous approach. When conducting pharmaceutical certificate of analysis, you demand the strictest standards for all inspecting procedures, meticulously document every experimental step, and thoroughly analyze the causes of non-compliant results while proposing corrective actions.*

Role	Proportion
Meticulous New Employee	33.99%
Careless New Employee	3.46%
Rigorous Expert	47.85%
Fatigued Expert	14.70%

Table 6: Distribution of different roles in the dataset.

Exhausted experts: *You are an expert in pharmaceutical inspecting, but you are now extremely exhausted. When conducting experiments, you choose the quickest methods. When writing pharmaceutical certificate of analysis, you always provide only a brief description of the experimental process and fail to analyze the causes of non-compliant items.*

We utilized Qwen3-MAX(Yang et al., 2025) to generate the report, which required inclusion of experimental details for each inspection item. These details encompassed: instrumentation, instrument parameters, reagents, quantities used, operational procedures, and acceptance criteria. The final generated report S_{COA} is shown in Figure 9. The proportions of correct PCoAs retained for the four roles after manual screening are shown in Table 6. Among them, the reports generated by the Meticulous New Employee and the Rigorous Expert primarily constitute the main components of PCoAD. In contrast, the reports generated by the Careless New Employee and the Fatigued Expert are mainly used to construct Logical Errors, in order to simulate erroneous PCoAs that may exist in real-world scenarios.

A.4 Labeling Errors

To reduce the cost of human annotation errors, we employed Qwen3-MAX(Yang et al., 2025) to generate standard errors, procedural errors, step omission, and calculation errors on manually verified reports. LLMs modified 1–4 elements within the reports according to the construction methods of the required error types, marked the erroneous content, and provided the original text as the corrected version. The prompts for generating each error type are as follows:

Standard Errors: *Modify the pharmaceutical certificate of analysis provided below according to the following requirements: First: Randomly modify the content of 1-4 test steps in the test report. For example, change 2.0 mL/min to 2.5 mL/min, change 37.0 ± 0.5°C to 25.0 ± 0.5°C, and change 5 mL of 3 N to 10 mL of 3 N. The modified numbers can be random, but do not alter results calculated using formulas. \nSecond: All other content in the test report must remain unchanged. \nThird: Generate the modified PCoA. No modification notes are required; generate the PCoA directly. \nFourth: Append the following three items at the end of the PCoA:\n1. ****Number of errors: []****, specifying the number of modified test steps within [];\n2. ****Error locations: []****, detailing the modified test steps within [];\n3. ****Correct content: []****, outlining the original test steps before modification within [].*

Procedural Error: *Modify the pharmaceutical certificate of analysis provided below according to the following requirements: \nFirst: Randomly modify 1–4 acceptance criteria in the test report. For example, change 0.214% < 0.25% → Pass to 0.214% > 0.20% → Fail, or modify 98.0%–102.0% to 95.0%–105.0%. The numerical values may be altered randomly, but results derived from formulas must remain unchanged. \nSecond: Modify the inspection report content accordingly based on the changed acceptance criteria, while keeping all other report content unchanged. \nThird: Generate the modified PCoA. No modification notes are required; generate the PCoA directly. \nFourth: Include the following three items at the end of the PCoA:\n1. ****Number of Errors: []****, specifying the number of modified acceptance criteria within [];\n2. ****Error Location: []****, specifying the modified acceptance criteria within [];\n3. ****Correct Content: []****, specifying the original acceptance criteria within [].*

Step Omission: *Modify the pharmaceutical certificate of analysis provided below according to the following requirements:\nFirst: Randomly delete 1-4 test items from the pharmaceutical certificate of analysis, and simultaneously remove the corresponding content from the conclusion section.\nSecond: Modify the test report content resulting from changes to acceptance criteria, while keeping all other report content unchanged.\nThird: Generate the modified pharmaceutical certificate of analysis. No modification notes are required; generate the pharmaceutical certificate of analysis directly. Fourth: Include the following three items at the end of the pharmaceutical certificate of analysis:\n1. ****Number of Errors: []****, specifying the number of modified acceptance criteria within the brackets;\n2. ****Error Location: []****, detailing the modified acceptance criteria within the brackets;\n3. ****Correct Content: []****, documenting the deleted test item content within the brackets.*

Calculation Errors: *Modify the pharmaceutical certificate of analysis provided below according to the following requirements:\nFirst: Randomly alter 1–4 calculation results in the test report. For example, modify: \text{Loss on Drying (%)} = \left(\frac{1.825 - 1.821}{1.825}\right) \times 100 = 0.219 % to \text{Loss on Drying (%)} = \left(\frac{1.825 - 1.821}{1.825}\right) \times 100 = 0.719 %. The modified numbers may be chosen randomly. \nSecond: All other content in the test report remains unchanged. Third: Generate the modified Pharmaceutical Certificate of Analysis. No modification notes are required; generate the certificate directly. Fourth: Add the following three items at the end of the pharmaceutical certificate of analysis:\n1.****Number of Errors:[]****, specifying the number of modified test steps within [];\n2.****Error Location:[]****, specifying the modified test steps within [];\n3.****Correct Content:[]****, specifying the original calculation formula and result within [].*

A.5 Inter-Annotator Agreement and Annotation Guidelines

To ensure the reliability and consistency of manual annotation, we recruited three annotators familiar with both the Chinese Pharmacopoeia (ChP 2020) and the United States Pharmacopeia (USP 2023). Prior to annotation, each annotator received detailed training covering the definitions, examples, and output format for the five error types: Standard Error, Procedure Error, Step Omission, Calculation Error, and Logical Error. During the annotation process, the three annotators independently labeled LLM-generated PCoAs. Disagreements were resolved by majority vote, and cases with three-way disagreement were adjudicated by a senior pharmaceutical expert. Daily calibration sessions and consistency monitoring were conducted throughout the annotation process, and 10% of the samples were randomly selected for double-blind re-annotation.

Based on the collected annotation data, we computed inter-annotator agreement metrics. The aver-

age pairwise Cohen's k values were 0.87 for report authenticity classification, 0.84 for error type classification, 0.89 for error count, and 0.81 for both erroneous content and corrected content (after semantic normalization). The overall multi-annotator Fleiss' k was 0.85, indicating substantial agreement. To prevent annotation drift, weekly calibration meetings were maintained throughout the labeling phase, and retraining was triggered whenever the k value for any time segment fell below 0.80. Through this rigorous methodology, we ultimately constructed PCoAD, a high-reliability dataset comprising 4,272 manually verified Pharmaceutical Certificates of Analysis.

B Prompt

In this module, we will detail the prompts used throughout the model via the following prompts. The prompts for the Text Segmentation Agent are shown in Figure 10; The prompts for the adaptive retrieval agent are shown in Figures 11 and 12. Figure 11 displays the prompt used during the agent's initial evaluation, where each segment is used only once; Figure 12 shows the prompt used for re-evaluation after retrieval completion. The prompts for the inference agent are shown in Figures 13 and 14, where Figure 13 presents the prompt for segments, and Figure 14 shows the prompt after merging all segments.

C Detailed Experimental Setup

C.1 Compared Methods

To explore the advantages of MACAR, we compare it against ten baseline methods.

(1) Qwen3-14B (Yang et al., 2025) the Qwen model family, unifies thinking and non-thinking modes within a single framework featuring a dynamic thinking budget mechanism, supports 119 languages, delivers state-of-the-art performance on code, math, and agent tasks across dense and Mixture-of-Expert architectures from 0.6B to 235B parameters.

(2) Baichuan-M1-14B (Wang et al., 2025a) is a family of large language models trained from scratch on 20 trillion tokens to balance general-domain strengths (e.g., math and coding) with specialized medical expertise, and its open-sourced 14B version is publicly available.

(3) Retrieval-Augmented Generation(RAG) (Lewis et al., 2020) combines a pre-trained parametric sequence-to-sequence model with a non-parametric dense vector index of Wikipedia accessed via a neural retriever, enabling language generation to outperform parametric-only models on knowledge-intensive tasks.

(4) Chain-of-Note (CoK) (Li et al., 2024c) generates a sequence of reasoning steps after retrieving relevant knowledge, enabling a thorough assessment of the relevance between the retrieved knowledge and the given question, and integrates these reasoning steps to formulate the final answer.

(5) Summarizing Retrievals (SuRe) (Kim et al., 2024) constructs summaries for the retrieved passages corresponding to each candidate answer and identifies the most plausible answer by evaluating and ranking the quality of these generated summaries.

(6) Hypothesis Knowledge Graph Enhanced Framework (HyKGE) (Jiang et al., 2023) expands feasible exploration paths in the knowledge graph through "hypothesis outputs" and introduces a "HO Fragment Granularity-aware Rerank Module" to filter noise while maintaining a balance between diversity and relevance in the retrieved knowledge.

(7) Self-Reflective RAG (Self-RAG) (Asai et al., 2024) adaptively retrieves passages on demand and employs special "reflection tokens" during generation to simultaneously produce and reflect upon both the retrieved passages and its own outputs, thereby endowing the model with controllability during inference and enabling it to adjust its behavior according to diverse task requirements.

(8) Dynamic Retrieval-Augmented Generation based on Information Needs (DRAGIN) (Su et al., 2024) dynamically determines when to retrieve and what to retrieve by aligning retrieval decisions with the LLM's actual information needs during generation.

(9) Knowledge-driven Iterative RAG (KiRAG) (Fang et al., 2025) decomposes documents into knowledge triples and performs iterative retrieval over these triples, integrating reasoning into the retrieval mechanism to dynamically identify information gaps and fetch relevant knowledge to fill them.

(10) Turing-Complete RAG (TC-RAG) (Jiang et al., 2025) introduces a Turing-complete system to manage state variables via a memory stack architecture equipped with adaptive retrieval, reasoning, and planning capabilities; this ensures controllable termination of the retrieval process and prevents error propagation through explicit "push" and "pop" operations.

(11) Multi-Agent Debate (MAD) (Liang et al., 2024) involves multiple agents engaging in argumentative exchanges under a "tit-for-tat" protocol, with a designated "judge" agent overseeing the debate and synthesizing a final solution.

(12) Diversity of Thoughts (DoT) (Lingam et al., 2025) explicitly reduces redundant reflections to enhance exploration of the decision space and incorporates a task-agnostic memory module that enables the model to retrieve and reuse knowledge from previously solved tasks.

(13) Diverse Multi-Agent Debate (DMAD) (Liu et al., 2025a) encourages agents to adopt distinct reasoning strategies; by integrating multiple problem-solving approaches, each agent gains insights from unique perspectives, iteratively refining its own responses and collectively converging toward an optimal solution.

C.2 Evaluation Metrics

We further elaborate on the evaluation metrics of the PCoAD dataset in detail. For RA, EC, and NE, a strict evaluation criterion is adopted, requiring that the predicted results exactly match the corresponding ground-truth labels and that the model accurately verifies the PCoA content. For the evaluation metrics of IC and RC, we follow the research paradigm for multi-label classification tasks proposed in the literature (Godbole and Sarawagi, 2004; Feng et al., 2024). Unlike micro-F1 and macro-F1, this metric takes into account partially correct and partially incorrect content in IC and RC, thereby enabling a more objective assessment.

Furthermore, during actual inference, when the model generates IC and RC, no constraints are imposed on the generation span; the model is allowed to autonomously generate results based on prior reasoning and retrieved knowledge, thereby avoiding any interference from artificial span restrictions on the model's independent judgment. To handle possible duplicate content in IC and RC during evaluation, we adopt a one-by-one matching strategy: after each evaluation of an item in the IC/RC list generated by the large language model, that item is removed from the list. If a complete match with the ground-truth text is found, the matched item is also removed from the label set. This mechanism eliminates the interference of duplicate content on the final evaluation metrics. When determining the consistency between the generated IC/RC and the ground-truth labels, the encoding model used for similarity calculation is bge-large-v1.5, and the cosine similarity measure is employed.

C.3 Retrieval Strategies

In terms of retrieval strategies, MACAR combines BM25 and dense retrieval methods. The rationales for adopting this hybrid scheme are summarized as follows. First, pharmaceutical inspection criteria contain abundant retrievable keywords and relevant information. After eliminating the interference brought by lengthy texts and nested standard citations, traditional retrieval approaches are sufficient to acquire required external knowledge. Second, this scheme effectively reduces resource consumption for local deployment, while adopting more sophisticated retrieval techniques would substantially increase system complexity. Third, employing advanced retrieval methods may mislead readers into attributing performance gains merely to superior retrieval capabilities, thereby overlooking the core effectiveness of MACAR in addressing inherent challenges of the PCoAD task. Consequently, we adopt BM25 integrated with dense retrieval to ensure a fair performance comparison between MACAR and all baseline models. Specifically, we adopt bge-large-v1.5(Xiao et al., 2024a) as the encoder. Among them, bge-large-zh-v1.5 is used for Chinese data and bge-large-en-v1.5 for English data.

C.4 Experimental Implementation.

The hyperparameters in MACAR are set as follows: k_1=1.5, b=0.75, α=0.6, and a maximum retrieval count Z=5. To enhance response diversity, the temperature is set to 0.5. To ensure a fair comparison between MACAR and the baseline methods, both MACAR and all baseline approaches adopt Qwen3-14B(Yang et al., 2025) as the generation model, with the temperature of Qwen3-14B uniformly set to 0.5. Although some baseline approaches are amenable to training, we refrain from fine-tuning any model—including those baselines—because MACAR operates without any training, relying solely on the intrinsic capabilities of the underlying models. Moreover, model training incurs substantial computational costs and is incompatible with the practical constraints of validating PCoA in real-world scenarios. Hence, all methods are evaluated in a training-free setting to maintain comparability. The experiments are implemented in Python 3.9 using the PyTorch 2.1.0 framework and executed on an Ubuntu server equipped with four NVIDIA GeForce RTX 3090 GPUs and one In-

tel(R) Xeon(R) CPU.

We provide the following three retrieval tools for use by the baselines.

(1) Knowledge Graph Retrieval. This study uses three publicly available medical knowledge graphs as foundational data sources: CMeKG, CPubMed-KG, and Disease-KG. These graphs integrate massive medical textual information covering diseases, drugs, symptoms, diagnostic procedures, and therapeutic techniques. After fusion, the resulting knowledge graph contains 1,288,721 entities and 3,569,427 semantic relations. We further collect entity descriptions from authoritative Chinese sources—including Wikipedia, Baidu Baike, and Medical Baike—and uniformly incorporate them into the knowledge graph to enrich entity semantics.

(2) Document Retrieval. This work uses the original documents from the Chinese Pharmacopoeia (ChP, 2020) and the United States Pharmacopeia (USP, 2023) as the textual corpus, with document counts given in Table 5. For efficient semantic retrieval, we employ General Text Embeddings—specifically its "gte_sentence-embedding" variant—to generate vector representations of document content. During preprocessing, all documents are split into fixed-length chunks (128 tokens) with a 50-token overlap between adjacent chunks to preserve contextual continuity and reduce boundary information loss. This strategy improves the accuracy and robustness of subsequent retrieval and reasoning tasks.

(3) Web and Encyclopedia Retrieval. To enhance the system's knowledge acquisition in open-domain settings, we integrate a multi-source online retrieval mechanism. It accesses structured or semi-structured authoritative sources such as Wikipedia and MedNet Medical Encyclopedia, and also interfaces with major commercial search engines (Google and Bing) to capture broader unstructured web information through keyword- and entity-driven real-time retrieval. This hybrid paradigm balances authority and coverage breadth, providing multi-dimensional knowledge support for downstream reasoning and question-answering tasks.

D Additional Experiments

D.1 Cost Analysis

MACAR relies on a multi-stage agent and retrieval mechanism; therefore, it is important to further analyze the implications of latency, token usage, and retrieval cost for deployment. Regarding retrieval cost, since MACAR adopts a BM25 + dense retrieval approach, the main retrieval cost lies in dense retrieval. Given the stability of the pharmacopoeia knowledge base, where testing standards for specific drugs rarely change, we employ a vector knowledge base to store relevant data during retrieval to save retrieval time. According to our statistics, retrieval time accounts for only about 13.2% of the total runtime of MACAR, with most of the system time spent on LLM-generated content. Consequently, latency is mainly reflected in token usage, and we follow the comparative method of Liu et al to compare the token and time consumption of MACAR with baselines(Liu et al., 2025b).

Method	Avg. Token	Avg. Token
TC-RAG	100%	100%
DMAD	153.8%	167.4%
MACAR	113.2%	109.7%
MACAR-parallel	113.4%	78.3%

Table 7: Average token and time consumption.

The cost comparison between MACAR and baselines is shown in Table 7. MACAR increases token and time consumption by an average of 11.5% compared to TC-RAG, which is based on adaptive retrieval. Given an average performance improvement of 3.3%, this indicates that MACAR consumes more resources. This is because MACAR uses a text chunking method that increases the number of retrieval and inference steps, thus consuming more tokens and time overall. However, compared to the agent-based DMAD, MACAR has a clear advantage, as DMAD requires repeated debates and each debate round takes the entire context as input. It should be noted, however, that MACAR can decompose a PCoA into multiple subparts, and the retrieval for each subpart is independent. In contrast to serial-process methods such as TC-RAG and DMAD, MACAR enables parallel retrieval of subparts during the adaptive retrieval process without significant performance degradation. When parallel retrieval is adopted, although token consumption does not change significantly, time consumption is greatly reduced, leading to an average cost reduction of 4.2%. Therefore, MACAR still maintains a cost advantage over TC-RAG and DMAD.

Method	ChP(2020)						USP(2023)					
	0.90		0.95		0.98		0.90		0.95		0.98	
	IC	RC	IC	RC	IC	RC	IC	RC	IC	RC	IC	RC
CoK	50.49	48.34	50.15	48.18	49.99	47.91	55.18	51.25	54.93	51.06	54.66	50.85
SuRe	51.45	50.47	51.31	50.12	51.06	49.9	57.97	53.74	57.70	53.61	57.41	53.46
HyKGE	52.91	51.89	52.74	51.51	52.44	51.24	59.18	55.28	58.96	54.93	58.64	54.78
Self-RAG	53.6	52.24	53.41	52.08	53.25	51.85	59.7	55.18	59.26	54.81	58.92	54.71
DRAGIN	54.01	52.36	53.78	52.23	53.47	51.99	60.16	54.6	59.88	54.35	59.53	53.93
KiRAG	56.05	55.38	55.92	55.02	55.81	54.9	63.18	58.54	62.72	58.39	62.24	58.01
TC-RAG	56.58	55.14	56.16	54.86	55.81	54.44	62.83	57.97	62.64	57.87	62.2	57.57
MAD	48.86	47.88	48.69	47.48	48.4	47.33	51.05	49.74	50.84	49.36	50.53	48.88
DoT	51.5	51.16	51.20	50.68	50.84	50.55	55.35	53.91	54.89	53.66	54.53	53.39
DMAD	50.85	49.83	50.56	49.68	50.21	49.45	54.84	54.41	54.61	54.15	54.36	54.04
MACAR	**56.73**	**56.25**	**56.48**	**55.86**	**56.33**	**55.71**	**60.63**	**59.70**	**60.15**	**59.56**	**59.94**	**59.22**

Table 8: Experimental results of MACAR and baselines on threshold sensitivity. The best scores are displayed in bold.

D.2 Sensitivity Analysis of Threshold

For the evaluation metrics of IC/RC, we adopt a similarity-based method to determine the consistency between the generated text and the ground-truth labels, with the similarity threshold set at >0.95. To further analyze the sensitivity of this threshold, we set the threshold to 0.90, 0.95, and 0.98, and evaluate the performance of the baseline and MACAR on IC/RC accordingly. The experimental results are presented in Table 8. It can be observed that the choice of threshold has little impact on the IC/RC results of both the baseline and MACAR, and MACAR still demonstrates a clear advantage over the baseline.

To further justify our selection of the 0.95 threshold, we randomly selected 100 text items generated by MACAR that were manually verified as correct and 100 items verified as incorrect, and computed their semantic similarity with the corresponding ground-truth labels. The results are shown in Table 9. It is evident that, despite the small sample size, the semantic similarity between incorrect texts and the labels can reach above 0.90. This occurs because large language models may produce minor errors when generating numbers (e.g., 1.0g vs. 1.5g). Such small discrepancies account for a very small proportion of the entire sentence, and thus the generated text may still exhibit high semantic similarity with the label. To prevent interference from isolated erroneous content and to impose a stricter assessment of the model's verification and correction capabilities, we set the threshold to 0.95.

Type	Right	Error
<0.60	0	57
0.60-0.80	0	34
0.80-0.90	3	7
0.90-0.95	19	2
>0.95	78	0

Table 9: The relationship between the correctness of generated text and its semantic similarity to labels.

D.3 Sensitivity Analysis of Hyperparameter α

To investigate the impact of the hyperparameter α, used for retrieval in the Adaptive Retrieval Agent, on model performance, we conducted a sensitivity analysis of α with values set to 0.5, 0.6, and 0.7. The experimental results are presented in Table 10. As can be seen from the table, α = 0.6 achieves better performance. This is because PCoAs contains a large number of keywords, such as drug names, reagent names, testing methods, and testing standards, which can be effectively captured through term frequency-based retrieval. However, different formulations of the same drug, e.g., aspirin tablets and aspirin capsules, also share many similar keywords; thus, excessively increasing the weight of BM25 may introduce noise. Therefore, setting α = 0.6 in this paper is reasonably justified.

D.4 Influence of Drug Type

The inspection processes for different types of drugs may vary considerably, potentially affecting the performance of MACAR. Therefore, we selected drug categories with a large number of

α	ChP(2020)					USP(2023)				
	RA ACC	EC ACC	NE ACC	IC F1	RC F1	RA ACC	EC ACC	NE ACC	IC F1	RC F1
0.5	62.01	51.82	60.24	56.23	55.41	65.11	53.19	63.55	59.85	59.21
0.6	**62.34**	**52.12**	**60.54**	56.48	55.86	**65.36**	**53.49**	63.85	60.15	59.56
0.7	62.14	51.93	60.39	**56.61**	**56.02**	65.22	53.31	**63.98**	59.87	59.16

Table 10: Parameter sensitivity analysis of α on PCoAD. The best scores are displayed in bold.

Drug category	ChP(2020)					USP(2023)				
	RA ACC	EC ACC	NE ACC	IC F1	RC F1	RA ACC	EC ACC	NE ACC	IC F1	RC F1
Infectious Diseases	61.89	51.67	**61.35**	56.03	55.41	**66.17**	53.30	**64.66**	59.96	**59.57**
Cardiology	**62.78**	52.56	59.72	56.32	55.01	64.54	52.67	63.03	59.33	58.74
Gastroenterology	61.52	51.30	60.98	55.66	55.04	64.91	53.04	63.42	59.76	59.11
Endocrinology	63.15	**52.93**	60.09	**57.29**	**56.67**	65.10	**53.93**	64.29	**60.59**	59.47
MACAR	62.34	52.12	60.54	56.48	55.86	65.36	53.49	63.85	60.15	59.56

Table 11: Performance of MACAR across different drug categories on PCoAD. The best scores are displayed in bold, while the second-best results are underlined.

PCoAs from the ChP (2020) and USP (2023) and evaluated the performance of MACAR on these different drug types. The experimental results are shown in Table 11. It can be observed that for the five drug types with a relatively high proportion of the PCoAD, MACAR does not exhibit significant performance differences or distinct distribution characteristics. This is because the main challenges of PCoAD lie in long-text reasoning and knowledge retrieval; the difficulty of validating different PCoAs is influenced by the complexity of chemical substances, formulations, and specific inspection requirements of individual drugs, and does not show a strong correlation with drug type. Moreover, the drugs selected in PCoAD are common medications, for which inspection methods are typically designed to be more generalizable in order to accommodate large-scale inspections. This further mitigates the potential impact of drug type on MACAR's performance.

D.5 Influence of Report Generation Roles

As shown in Table 6, PCoAD mainly consists of reports generated by the Meticulous New Employee and the Rigorous Expert. Analyzing the impact of reports generated by different roles on the performance of MACAR can help us investigate the potential influence of PCoA writing style on automated validation. Due to the insufficient amount of data from the Careless New Employee and the Fatigued Expert, we only select the Meticulous New Employee and the Rigorous Expert for analysis. The dataset construction and the proportions among the five error types are the same as those in PCoAD. The results of the baselines and MACAR are shown in Table 12. The performance of both the baselines and MACAR on the Rigorous Expert is significantly better than that on the Meticulous New Employee. This is because the Rigorous Expert tends to describe operational procedures and calculations with more concise and precise text while reducing redundant steps. This helps the model improve its performance on RA, EC, and NE. However, for error identification and correction, which primarily rely on the model's ability to analyze external knowledge and reason, the impact of the role's expression style is relatively small.

D.6 Performance of fewer-parameter models

In real-world applications, PCoA often involves highly sensitive confidential information and personal privacy data, necessitating strict adherence to data security and regulatory compliance requirements. Consequently, directly invoking public APIs of commercial LLMs poses significant risks of data leakage and fails to meet legal and regulatory standards. Local deployment of open-source models offers a viable alternative. Although this study adopts Qwen3-14B(Yang et al., 2025) as the base model, its substantial computational demands-

Method	ChP(2020)					USP(2023)				
	RA ACC	EC ACC	NE ACC	IC F1	RC F1	RA ACC	EC ACC	NE ACC	IC F1	RC F1
Meticulous New Employee										
KiRAG	59.61	50.01	55.42	54.92	54.37	62.74	56.19	62.21	58.29	56.50
TC-RAG	60.27	51.05	55.66	54.36	53.46	62.83	53.22	61.94	57.37	57.17
DMAD	61.06	48.15	52.89	50.43	49.68	61.48	52.18	62.65	54.01	54.15
MACAR	61.84	51.67	60.04	56.08	55.71	64.86	52.99	63.35	59.65	59.56
Rigorous Expert										
KiRAG	60.61	50.61	56.12	54.87	54.37	63.74	57.19	63.22	58.89	56.50
TC-RAG	60.92	51.85	56.66	54.83	53.46	63.83	54.22	62.94	58.37	57.17
DMAD	61.60	49.15	53.89	51.06	49.68	62.48	53.18	63.65	54.91	54.15
MACAR	62.84	52.32	61.04	56.48	55.73	65.86	53.99	64.39	60.65	59.56

Table 12: Performance of MACAR and Baselines on Reports Generated by Different Roles.

Pharmacopoeia	Item	Qwen3-8B	Qwen3-4B
ChP (2020)	RA	58.14	53.94
	EC	47.62	44.30
	NE	53.66	49.36
	IC	51.30	47.81
	RC	50.17	46.97
USP (2023)	RA	59.01	55.03
	EC	51.33	47.86
	NE	52.43	49.75
	IC	50.87	48.03
	RC	49.47	46.72

Table 13: MACAR Performance on Qwen3-8B and Qwen3-4B.

typically requiring professional-grade GPUs for inference-present a significant deployment barrier for small and medium-sized enterprises with limited computational resources.

To enhance the practicality and accessibility of the MACAR framework and extend its applicability to resource-constrained environments, we further conduct a systematic evaluation using smaller-scale variants: Qwen3-8B and Qwen3-4B. As shown in Table 13, a clear performance degradation is observed as model size decreases. This decline stems from the fact that both the Adaptive Retrieval Agent and the Inference Agent in MACAR rely heavily on the reasoning capabilities of LLMs. As demonstrated by (Wei et al., 2022), such reasoning abilities are strongly correlated with model parameter scale. Therefore, improving MACAR's effectiveness when deployed with smaller-parameter models remains an important challenge for future research.

D.7 Error Analysis

Statistical analysis of error data from the baseline and MACAR models on the PCoAD dataset is summarized in Table 14. For NE tasks, FP indicates predictions below actual values, and FN indicates predictions exceeding actual values. In authenticity assessment, RAG-based methods were more inclined to classify PCoAD as TRUE, a bias caused by the models' over-acquisition of pharmaceutical inspection standards, which induces contextual rot during final judgment. Conversely, the agent-based approach showed the opposite trend, likely due to excessive deliberation in its multi-model debate framework. Regarding NE, the baseline and MACAR also exhibited significant discrepancies. RAG-based errors mainly caused by incomplete statistics that models halted reasoning upon detecting the first text error, without scanning the full report. In contrast, MACAR and agent-based methods exhibited reverse behavior, driven by their block-based. This occurs because when errors occur during reasoning, they propagate to the integration stage, thereby increasing the total error count. Typical error cases are presented in Table 16.

D.8 Classification Analysis

To conduct a more in-depth analysis of the verification performance of MACAR and the baselines on the five error types in PCoAD, we partition each error type into a separate sub-dataset, each comprising 50% correct samples and 50% erroneous samples. Dataset statistics are summarized in Table 15. During testing, models are not required to identify the specific error type; all other validation protocols and evaluation metrics remain consistent

Methods	RA		NE	
	FP	FN	FP	FN
ChP (2020)				
MACAR	36.52%	63.48%	23.70%	76.30%
TC-RAG	87.23%	12.77%	86.27%	13.73%
DMAD	28.21%	71.79%	31.38%	68.62%
USP (2023)				
MACAR	31.22%	68.78%	29.31%	70.69%
TC-RAG	85.36%	14.64%	89.59%	10.41%
DMAD	23.74%	76.26%	38.64%	61.36%

Table 14: Relative proportions of various error types in baseline and MACAR relative to total errors.

Pharmacopoeia	Classification	True Num	Error num
ChP(2020)	Standard Error	201	201
	Procedural Error	242	242
	Step Omission	230	230
	Calculation Error	202	202
	Logical Error	208	208
USP (2023)	Standard Error	208	208
	Procedural Error	206	206
	Step Omission	212	212
	Calculation Error	203	203
	Logical Error	224	224

Table 15: Sub-dataset Statistics.

across comparisons.

Tables 17 and 18 present the evaluation results of MACAR and the baselines on the sub-datasets derived from ChP (2020) and USP (2023), respectively. It is evident that both MACAR and the baselines achieve relatively high accuracy in identifying computational errors in PCoA and are particularly effective at locating and correcting such errors. This superior performance stems from the fact that LLMs are typically reinforced with extensive training in mathematical reasoning and code-related tasks to enhance their general reasoning capabilities, enabling them to accurately detect computational mistakes in reports. Moreover, unlike experimental procedure errors—which often require external knowledge for verification—computational errors can be resolved using the model's intrinsic reasoning abilities alone. Consequently, both RAG-based and agent-based approaches demonstrate effectiveness in handling this error type.

At the same time, we also observed lower accuracy in identifying standard errors, step omission, and procedural errors. This reflects the model's limitations when confronting long-text reasoning tasks like PCoA. On one hand, excessively lengthy texts reduce retrieval accuracy and introduce excessive noise into the reasoning process. On the other hand, the concatenation of extensive knowledge within long texts leads to context decay, impairing the model's reasoning capabilities. Identifying logical errors, however, poses the greatest challenge for both MACAR and the baseline model. This highlights the difficulty LLMs face in recognizing their own generated hallucinations. Furthermore, logical errors often contain mixed corrections and improvements, further confusing the model's judgment. Therefore, enhancing the model's ability to identify logical errors remains a critical issue for future research.

Type	Sample
Chunking error	**Text**: 2. COMPOUNDING PROCEDURE AND BATCH PREPARATION\n2.1 Formulation (Per·100·mL)\n · · ·\n3. ASSAY BY HIGH-PERFORMANCE LIQUID CHROMATOGRAPHY (HPLC)\n3.1 Objective · · · **Analysis**: COMPOUNDING PROCEDURE AND BATCH PREPARATION and ASSAY BY HIGH-PERFORMANCE LIQUID CHROMATOGRAPHY (HPLC) were incorrectly placed within the same text block.
Missing standard	**Text**: **PHARMACEUTICAL QUALITY CONTROL TESTING REPORT**\n**Drug Name:** Abiraterone Acetate\n · · · **Search results**: [<281>RESIDUE ON IGNITION, <197>SPECTROSCOPIC IDENTIFICATION TESTS, <921>WATER DETERMINATION, Abiraterone Acetate] **Analysis**: General Rule <621> CHROMATOGRAPHY is missing in the retrieval results.
Retrieve error	**Text**: **PHARMACEUTICAL QUALITY CONTROL TESTING REPORT**\n**Drug Name:** Cromolyn Sodium Ophthalmic Solution\n · · · **Search results**: [<71>STERILITY TESTS, <621>CHROMATOGRAPHY, <921>WATER DETERMINATION, Cromolyn Sodium Ophthalmic Solution] **Analysis**: Incorrect General Rule <921> WATER DETERMINATION was retrieved.
Text chunking inference error	**Text**: Chromatographic System	LC (<621>)\n	Detector	UV at 254 nm\n	Column	4.6 mm × 25 cm; 5 µm packing L1 (Agilent ZORBAX Eclipse Plus C18)\n	Flow Rate	2.2 mL/min · · · **Inference results**: Excessively high flow rate unreasonable. Conventional 4.6 mm C18 routine flow rate: 1.0–1.5 mL/min, 2.2 mL/min is too high. Column parameter non-standard writing - USP L1 C18 column: standard format is 4.6 mm i.d. × 250 mm, written as 25 cm is non-standard in pharmacopoeia documents. **Analysis**: An error regarding the non-standard specification of column parameters was constructed, resulting in additional error instances.
Final inference error	**Text**: Sub-section1: No errors exist; · · · Sub-section5: There is an error. Calculation Formula & Data Logic Errors · · · Sub-section6: We believe that canceling the concentration too arbitrarily may be a mistake, but according to the drug inspection standards, the final determination is not a mistake. · · · **Inference results**: ["label": false, "ans": {"Error Type": [Procedural error], "Number of Errors": [2], "Erroneous Content": [...], "Corrected Content": [...]}] **Analysis**: It increases the total number of errors by regarding the negated content in text chunk reasoning as erroneous information.

Table 16: Error Example of MACAR, with error parts marked in red.

Classification	Item	MACAR	KIRAG	TC-RAG	DoT	DMAD
Standard Error	RA	58.31	57.73	57.96	59.24	**59.97**
	NE	57.12	55.04	54.91	55.67	**57.43**
	IC	**55.61**	51.96	51.26	51.04	51.60
	RC	**54.20**	51.42	50.57	50.33	51.07
Procedural Error	RA	57.96	**58.43**	57.34	57.05	58.13
	NE	55.88	**56.01**	55.44	54.67	55.67
	IC	**54.63**	53.21	52.66	51.66	50.36
	RC	**54.76**	52.34	52.07	50.30	49.44
Step Omission	RA	**60.14**	58.41	59.15	57.64	59.02
	NE	58.47	57.73	**58.77**	55.71	57.82
	IC	**56.94**	52.68	53.51	52.80	53.04
	RC	**56.07**	51.83	52.86	52.51	52.31
Calculation Error	RA	**66.74**	63.98	63.46	65.41	65.49
	NE	**63.55**	59.11	60.23	57.18	57.81
	IC	**58.91**	57.01	56.87	54.19	53.77
	RC	**58.20**	56.22	56.11	53.54	52.97
Logical Error	RA	55.36	53.69	52.40	54.09	**55.79**
	NE	**53.97**	51.97	52.01	49.76	50.41
	IC	**51.68**	46.83	48.07	46.28	45.39
	RC	**51.04**	45.09	47.46	46.03	44.46

Table 17: Experimental results of MACAR and baseline on ChP (2020). The best scores are displayed in bold, while the second-best results are underlined.

Classification	Item	MACAR	KIRAG	TC-RAG	DoT	DMAD
Standard Error	RA	60.89	60.51	60.74	61.65	**62.12**
	NE	58.45	57.91	58.04	57.03	**58.86**
	IC	**57.78**	54.52	53.67	53.46	54.05
	RC	**57.69**	53.98	53.01	52.87	52.72
Procedural Error	RA	60.23	**61.28**	59.79	59.51	60.96
	NE	58.17	**58.99**	57.82	57.55	58.24
	IC	**56.95**	55.74	55.21	54.19	52.93
	RC	**56.25**	54.88	54.46	52.86	52.15
Step Omission	RA	**62.85**	60.94	61.82	60.56	61.79
	NE	61.01	60.17	**61.42**	58.19	60.48
	IC	**59.73**	56.32	56.29	55.72	55.66
	RC	**58.76**	55.69	55.34	55.18	55.33
Calculation Error	RA	**69.21**	66.55	65.89	68.02	68.11
	NE	**66.13**	61.94	62.68	59.77	60.62
	IC	**66.64**	59.65	59.63	56.72	56.41
	RC	**65.75**	58.84	58.96	56.31	55.58
Logical Error	RA	57.79	56.24	55.06	56.65	**58.33**
	NE	**56.62**	54.75	54.79	52.39	53.18
	IC	**57.41**	49.57	50.88	48.86	48.12
	RC	**56.58**	47.82	50.14	48.79	47.19

Table 18: Experimental results of MACAR and baseline on USP (2023). The best scores are displayed in bold, while the second-best results are underlined.

Sample of extracting inspection items

Prompt:
Based on the drug name and Monograph, extract the verbatim text of the relevant test items from the General Chapter. The extracted test items must include all details and be identical to the content in the General Chapter.

Drug Name: {*Abiraterone Acetate*}

Monograph: {*DEFINITION:\nAbiraterone Acetate contains NLT 98.0% and NMT 102.0% of abiraterone acetate (C26 H33 NO2), calculated on the as-is\nbasis. \nIDENTIFICATION\nChange to read:\n• A. SPECTROSCOPIC IDENTIFICATION TESTS〈197〉, Infrared Spectroscopy: 197K (CN 1-May-2020)\n• B. The retention time of the major peak of the Sample solution corresponds to that of the Standard solution, as obtained in the Assay. …*}

General Chapter: [
<281>RESIDUE ON IGNITION\nThe Residue on Ignition/SulfatedAsh test uses a procedure to measure the amount of residual substance not volatilized from a sample when the sample is ignited in the presence of sulfuric acid according to the procedure described below. …
<621>CHROMATOGRAPHY\nINTRODUCTION\nChromatographic separation techniques are multistage separation procedures in which the components of a sample are\ndistributed between two phases, one of which is stationary while the other is mobile. …
…]

Output: {*Infrared Spectroscopy : Preparation Method: Sample and USP Abiraterone Acetate RS were each mixed with KBr …*
Residue on Ignition : Procedure :Pre-ignited platinum crucible (600 ± 50°C) for 30 min. …}

Figure 6: Sample of Extracted Specific Inspection Items.

Sample of generating a table summary

Prompt: Extract the content in the picture and summarize it into a paragraph of text.

Input:

Time (minutes)	*Solution A* (%)	*Solution B* (%)	Elution
0	90	10	equilibration
0–3.5	90	10	isocratic
3.5–23.5	90→60	10→40	linear gradient
23.5–34.5	60→5	40→95	linear gradient
34.5–59.5	5	95	isocratic
59.5–60	5→90	95→10	linear gradient

Output: {*The table outlines a detailed chromatographic elution protocol over a 60-minute period, specifying the composition of two mobile phase solutions (A and B) and the corresponding elution mode at various time intervals. At time zero, the system is equilibrated with 90% Solution A and 10% Solution B. From 0 to 3.5 minutes, the system maintains this composition under isocratic conditions. Between 3.5 and 23.5 minutes, a linear gradient is applied, decreasing Solution A from 90% to 60% while increasing Solution B from 10% to 40%. From 23.5 to 34.5 minutes, another linear gradient continues, reducing Solution A from 60% to 5% and increasing Solution B from 40% to 95%. This is followed by an isocratic phase from 34.5 to 59.5 minutes, where the composition remains at 5% Solution A and 95% Solution B. Finally, from 59.5 to 60 minutes, a rapid linear gradient reverses the composition back to 90% Solution A and 10% Solution B for re-equilibration. This method likely aims to achieve effective separation of compounds with varying polarities by combining multiple isocratic and gradient steps, ensuring optimal resolution and peak detection.*}

Figure 7: Example of Generated Table Summary.

Certificate of Analysis (CoA) XXX, Oakwood, VA USA. Email: xxxxxx		
Name of the API	Ibuprofen Tablets	This lot was manufactured by
Batch/Lot Number	193253	xxxx
Manufacturing Date	08/01/2022	
Expiration Date	07/31/2027	
CAS	xxxxx-xx-x	
TESTS	**SPECIFICATIONS**	**RESULTS**
ASSAY ON ANHYDROUS BASIS	97.0 - 103.0 %	100.0 %
DESCRIPTION	White to off-white, crystalline powder, having a slight, characteristic odor.	CONFORMS
IDENTIFICATION A <197A>	IR: Reference to standard spectrum.	POSITIVE
IDENTIFICATION B <197U>	UV: <= 3.0 % (264 nm) UV: <= 3.0 % (273 nm)	1.3 %; POSITIVE 0.3 %; POSITIVE
IDENTIFICATION C	Reference to standard chromatogram.	POSITIVE
WATER <921>	<= 1.0 %	0.2 %
RESIDUE ON IGNITION <281>	<= 0.5 %	0.2 %
CHROMATOGRAPHIC PURITY	<= 0.3 % (Any individual impurity) <= 1.0 % (Sum of all the individual impurities)	< 0.05 % < 0.05 %
LIMIT OF IBUPROFEN RELATED COMPOUND C	<= 0.1 %	NOT DETECTED
ELEMENTAL IMPURITIES <232>	Meets the requirements	CONFORMS
RESIDUAL SOLVENTS <467>	Meets the requirements	CONFORMS
SOLUBILITY	Practically insoluble in water; very soluble in alcohol, in methanol, in acetone, and in chloroform; slightly soluble in ethyl acetate	
Created by: FH	Reviewed By: AH	Approved By: FR

(a) The current PCoA issued by competent governmental authorities.

PHARMACEUTICAL CERTIFICATE OF ANALYSIS REPORT
Drug Name: Ibuprofen Tablets
Batch Number: IBU-2024-087
Test Date: April 5, 2025
Analyst: Dr. Amina Patel, QC Analyst (License: USP-QC-1023)
Reviewed By: Dr. Robert Chen, Senior QC Supervisor
Laboratory: Central Pharmaceutical Testing Lab – USP Compliance Unit

Assay – HPLC Analysis (LC-UV 254 nm)
Chromatographic System
Instrument: Agilent 1260 Infinity II HPLC System
Column: Waters XBridge C18, 4.6 mm × 25 cm, 5 µm particle size (L1)
Detector: UV-Vis Diode Array Detector (DAD), wavelength set at 254 nm (Assay), 200–400 nm (Identification)
Flow Rate: 2.0 mL/min
Injection Volume: 10 µL
Temperature: Ambient (25°C)
Mobile Phase:
Dissolve 4.0 g chloroacetic acid in 400 mL water
Adjust pH to 3.0 ± 0.05 using ammonium hydroxide
Add 600 mL acetonitrile
Mix thoroughly; filter through 0.45 µm PTFE membrane before use
🔧 Note: Needle wash solvent: Methanol:Water (90:10) as recommended in auxiliary rules.
Solutions Prepared

Solution	Preparation
Standard Solution	10.0 mg/mL in Mobile Phase
Sample Solution	10.0 mg/mL nominal ibuprofen concentration • 10 tablets (total weight: 20.56 g) • Crushed in mortar, transferred to 100 mL volumetric flask • Added ~50 mL Mobile Phase • Shaken mechanically for 90 min until complete disintegration • Diluted to mark with Mobile Phase • Centrifuged at 3000 rpm for 10 min • Supernatant filtered through 0.45 µm PTFE filter
System Suitability Solution	10.0 mg/mL USP Ibuprofen RS + 0.01 mg/mL each of USP Ibuprofen Related Compound C RS and USP Ibuprofen Related Compound J RS in Mobile Phase
Sensitivity Solution	0.005 mg/mL USP Ibuprofen RS in Mobile Phase

System Suitability Testing

Parameter	Requirement	Result
Tailing Factor (Ibuprofen)	⩽ 2.5	1.84
RSD of Peak Area	⩽ 2.0%	1.23%
Resolution between Ibuprofen and Compound J	⩾ 2.5	3.12
Resolution between Ibuprofen and Compound C	⩾ 2.5	2.87
Signal-to-Noise Ratio (Sensitivity Solution)	⩾ 10	18.7
RSD of Impurity Peaks (Standard Solution)	⩽ 6.0%	4.12% (J), 3.98% (C), 4.35% (Ibuprofen)

✅ **System Suitability Passed — All criteria met.**
Chromatographic Run Results (Sample & Standard)

...

Assay Calculation

$$\text{Result}(\%) = \left(\frac{r_U}{r_S}\right) \times \left(\frac{C_S}{C_U}\right) \times 100$$

Where:

$r_U = 125{,}430$(Sample peak response)

$r_S = 126{,}210$(Stadard peak response)

$C_S = 10.0\ \text{mg/mL}$

$C_U = 10.0\ \text{mg/mL}$

$$\text{Result} = \left(\frac{125{,}430}{126{,}210}\right) \times 100 = 99.38\%$$

✅ Assay Result: 99.38%→ Within 97.0%–103.0% acceptance range

(b) The PCoAD contains detailed experimental and computational procedures.

Figure 8: The differences between the official PCoA and PCoAD.

Prompt for generating the pharmaceutical certificate of analysis

Role: You are an expert in pharmaceutical inspecting with a meticulous and rigorous approach. When conducting pharmaceutical certificate of analysis, you demand the strictest standards for all testing procedures, meticulously document every experimental step, and thoroughly analyze the causes of non-compliant results while proposing corrective actions.

Prompt: Extract the content in the picture and summarize it into a paragraph of text.

Input: {

"drug_name": "Abiraterone Acetate",

"methods": "DEFINITION:\nAbiraterone Acetate contains NLT 98.0% and NMT 102.0% of abiraterone acetate (C26 H33 NO2), calculated on the as-is\nbasis.\nIDENTIFICATION\nChange to read:\n• A. SPECTROSCOPIC IDENTIFICATION TESTS〈197〉, Infrared Spectroscopy: 197K (CN 1-May-2020)\n• B. The retention time of the major peak of the Sample solution corresponds to that of the Standard solution, as obtained in the Assay…",

"auxiliary_rules": [

"<281>RESIDUE ON IGNITION\nMethods:Portions of this general chapter have been harmonized with the corresponding texts of the European Pharmacopoeia and the Japanese Pharmacopoeia…",

"<621>CHROMATOGRAPHY\nMethods:This chapter describes general procedures, definitions, and calculations of common parameters and generally applicable requirements for system suitability…",

…]

}

Figure 9: Prompt for Generating PCoA.

Prompt for Text Chunking Agent

Prompt: Please divide the input text into several parts according to structure, paragraphs and importance. Each part should be coherent, and the content of the input text shall not be deleted or modified. Finally, output the result in JSON format.

Input: {*PHARMACEUTICAL QUALITY CONTROL TESTING REPORT** \n**Drug Name:** Abiraterone Acetate \n**Test Date:** April 5, 2025 \n**Report Number:** QCR-ABT-2025-0405 \n**Analyst:** Dr. Elena M. Rodriguez, QC Analyst (PharmD, USP-Certified) \n**Laboratory:** Central Quality Control Laboratory – Analytical Chemistry Division \n**Compliance with:** United States Pharmacopeia–National Formulary (USP-NF), Revision May 2020 (as harmonized)\n\n---\n\n## **1. INTRODUCTION**\n\nThis report details the comprehensive analytical testing performed on a batch of **Abiraterone Acetate** (Lot No.: ABT-2025-0401, Expiry: Dec 31, 2027) to ensure compliance with the official specifications outlined in the USP monograph. The testing includes: ⋯*}

Figure 10: Prompt for the Text Chunking Agent.

Prompt No. 1 for Adaptive Retrieval Agent: Determine whether to retrieve

Prompt:
Your knowledge may not be accurate, so you must use the provided tools to think and provide answers to users.
You have the following tools available:{"Specific Drug Testing Ruler Retrieval Tool", "General Ruler Retrieval Tool"}
The specific functions and detailed introductions of each tool are as follows:{
"Specific Drug Testing Ruler Knowledge Base": "Used to retrieve testing ruler for specific drugs, e.g., testing standards for aspirin, ibuprofen, etc.",
"General Ruler Retrieval Tool": "Used to retrieve ruler for general drug testing methods, e.g., testing standards such as General Rule 0831, General Rule 0704, etc."
}
Please determine whether you need to use tools based on the input text and the user's question. Tools are not necessarily required, but you must select a tool if you believe relevant knowledge is missing.Then Determine which tools to use, analyze what knowledge is needed, list all required knowledge using nouns present in the input text and existing knowledge, and output it in the form of a list.
Final output format:{"Whether to use tools":[],"Tools to use":[],"Required knowledge":[]}

User's question: "Please determine if the input drug test report is genuine"

Input text: {*Text chunking*}

Figure 11: Prompt No. 1 for Adaptive Retrieval Agent.

Prompt No. 2 for Adaptive Retrieval Agent: Determine whether to retrieve

Prompt:
You have now obtained some knowledge through tools, but this knowledge may not be complete. Please determine whether it is necessary to continue retrieving knowledge using tools and what knowledge needs to be retrieved.
The following tools are available for you:
You have the following tools available:{"Specific Drug Testing Ruler Retrieval Tool", "General Ruler Retrieval Tool"}
The specific functions and detailed introductions of each tool are as follows:{
"Specific Drug Testing Ruler Knowledge Base": "Used to retrieve testing ruler for specific drugs, e.g., testing standards for aspirin, ibuprofen, etc.",
"General Ruler Retrieval Tool": "Used to retrieve ruler for general drug testing methods, e.g., testing standards such as General Rule 0831, General Rule 0704, etc."
}
First, based on existing knowledge and input text, determine whether the user's question can be solved. If yes, answer "Yes" and stop subsequent work.
Second, based on the input text, the user's question, and existing knowledge, determine whether the existing knowledge is sufficient. If not, determine which tools to use, analyze what knowledge is needed, list all required knowledge using nouns present in the input text and existing knowledge, and output it in the form of a list.
Please analyze step by step, and finally output a summary in the following format:
{"Can the user's question be solved":[],"Whether to use tools":[],"Tools to use":[],"Required knowledge":[]}

User's question: "Please determine whether the input text is true. The requirement for being true is that every sentence is supported by evidence to prove its correctness."

Input text： {*Text chunking*}

Existing knowledge: {*Retrieved Passages*}

Figure 12: Prompt No. 2 for Adaptive Retrieval Agent.

Prompt No. 1 for Inference Agent

Prompt:
Perform the following actions based on the user's query, the input text, and existing knowledge:
1. Carry out step-by-step reasoning based on the input text, the user's question, and available knowledge.
2. During this step-by-step reasoning process, if you detect a lack of necessary knowledge, stop reasoning immediately. Analyze what knowledge is missing, list all required knowledge items, and use only nouns that appear in the input text and existing knowledge.
3. If an error is identified, perform the following steps:
3.1 Classify the type of error;
3.2 Extract the erroneous content;
3.3 Correct the erroneous content;
3.4 Count the total number of errors.

Error types: {
Standard error: The cited standard content or standard values are incorrect;
Procedural error: An experimental operation step is performed incorrectly;
Missing step: A test item specified in the testing standard is omitted;
Calculation error: The calculation result is incorrect;
Logical error: A test step has already yielded a result, but in subsequent text, through step-by-step logical analysis, the original result is altered, leading to an incorrect conclusion.
}

User's question: "Please determine whether the input text is true. The requirement for being true is that every sentence is supported by evidence to prove its correctness."

Input text：{*Text chunking*}

Existing knowledge: {*Retrieved Passages*}

Figure 13: Prompt No. 1 for Inference Agent.

Prompt No. 2 for Inference Agent

Prompt:
Summarize the input text based on the user's query and output the result in JSON format. If multiple errors exist, the erroneous content should be presented as a list. Correct content should also be provided in list format. The final output format should be as follows: ["label": [], "ans": {"Error Type": [], "Number of Errors": [], "Erroneous Content": [], "Corrected Content": []}]

User's question: "Please determine whether the input text is true. The requirement for being true is that every sentence is supported by evidence to prove its correctness."

Input text： {*Text chunking*}

Figure 14: Prompt No. 2 for Inference Agent.