PRGB Benchmark: A Robust Placeholder-Assisted Algorithm for Benchmarking Retrieval-Augmented Generation

Zhehao Tan*, Yihan Jiao*, Dan Yang†, Junwei Liu, Lei Liu
Jie Feng, Duolin Sun, Yue Shen, Jian Wang, Peng Wei, Jinjie Gu
Ant Group
{tanzhehao.tzh, jiaoyihan.yh, luoyin.yd}@antgroup.com

Abstract

Retrieval-Augmented Generation (RAG) enhances large language models (LLMs) by integrating external knowledge, where the LLM's ability to generate responses based on the combination of a given query and retrieved documents is crucial. However, most benchmarks focus on overall RAG system performance, rarely assessing LLM-specific capabilities. Current benchmarks emphasize broad aspects such as noise robustness, but lack a systematic and granular evaluation framework on document utilization. To this end, we introduce Placeholder-RAG-Benchmark, a multi-level fine-grained benchmark, emphasizing the following progressive dimensions: (1) ***multi-level filtering abilities***, (2) ***combination abilities***, and (3) ***reference reasoning***. To provide a more nuanced understanding of LLMs' roles in RAG systems, we formulate a novel ***Placeholder***-based approach to decouple the contributions of the LLM's parametric knowledge and the external knowledge. Experiments demonstrate the limitations of representative LLMs in the RAG system's generation capabilities, particularly in error resilience and context faithfulness. Our benchmark provides a reproducible framework to develop more reliable and efficient RAG systems.

Code — https://github.com/AQ-MedAI/PRGB
Datasets —
https://huggingface.co/datasets/AQ-MedAI/PRGB
Extended version — https://arxiv.org/abs/2507.22927

Introduction

Retrieval-Augmented Generation (RAG) has emerged as a transformative paradigm enabling large language models (LLMs) to integrate external knowledge. Within RAG systems, it is crucial for LLMs to generate more reliable responses based on a given query and retrieved documents.

Recently, diverse efforts have been conducted to evaluate RAG systems' performance across methodologies and metrics (Gan et al. 2025; Long et al. 2025). For example, RAG (Lewis et al. 2020), REALM (Lewis et al. 2020), and RETRO (Borgeaud et al. 2022) assess retrieval quality via exact match or F1 scores over retrieved passages and generation accuracy through QA-based metrics. Existing benchmarks mostly assess overall RAG performance, not the LLMs' generation capabilities based on the retrieved context.

Traditional methods typically focus on evaluating the retrieval process (Voorhees 1998), but recent studies indicate that retrieval quality alone cannot guarantee generation accuracy (Min et al. 2020). Therefore, it is very necessary to **evaluate the capabilities of generative models in RAG systems**. HIRAG(Jiao et al. 2025) trained the generation models in RAG systems, but it lacks more granular aspects about RAG-specific tasks. To investigate LLMs' capabilities to leverage RAG context, RAG-Bench (Friel, Belyi, and Sanyal 2025) systematically combined pre-defined RAG metrics and newly introduced LLM-specific metrics (*e.g.*, fluency, coherence, and factual consistency) to better describe the overall RAG system performance. RGB (Chen et al. 2023) tried to define basic abilities of RAG-based LLMs from the viewpoints of noise robustness and document integration. However, the dimensions of these benchmarks are relatively coarse, which is **insufficient to evaluate the capability of LLMs to effectively leverage the retrieved corpus within RAG systems**. The resulting capabilities that LLMs should possess in RAG environments are as follows: **(a) Multi-level Filtering** evaluates whether LLMs can accurately identify the relevant information from retrieved documents, filtering varying-degree noise. **(b) Combination** evaluates whether LLMs can recognize and retain all relevant data necessary for generating a complete and accurate response. **(c) Reference Reasoning** evaluates whether LLMs can perform multi-hop reasoning based on the information provided in retrieved documents and answer questions that go beyond direct answers on content retrieval (the above two abilities).

However, when evaluating whether LLMs could genuinely ground responses in external retrieval contexts, a critical challenge lies in **disentangling the contribution of LLMs' parametric knowledge**, since conflating these knowledge sources would obscure whether LLMs default to applying pre-trained parametric biases. For instance, when a model correctly answers a factual question using retrieved documents, it remains unclear whether the accuracy stems from genuine synthesis of external information or implicit recall of pre-trained knowledge. Recent solution POPQA

*These authors contributed equally.
† Corresponding Author.

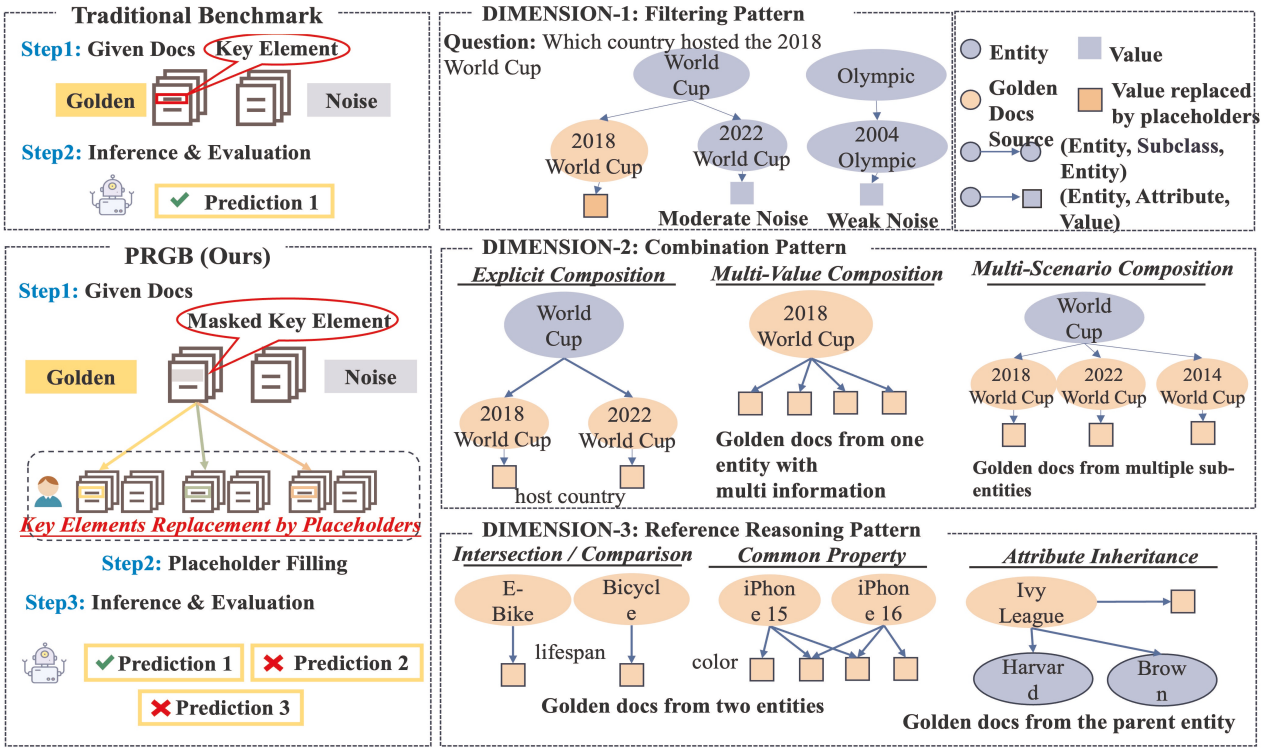

Figure 1: Visualization for Evaluation Dimensions of Placeholder-RAG-Benchmark. Starting from triplet-based metadata, three finer-grained evaluation tasks are formulated, including filtering, composition, and reasoning.

(Mallen et al. 2023) used less popular factual data for evaluation, because such data is more likely to be learned by LLMs. However, over time, newer LLMs are likely to learn from an increasing volume of online corpora, which means that information that was relatively less popular at the time may eventually be memorized by these models after a certain period. When evaluating datasets with RAG like TriviaQA (Joshi et al. 2017) and RGB(Chen et al. 2023), the majority of methods adopted a specific instruction template "If the information in the document does not contain the answer, you will generate I cannot answer the question because of insufficient information in the documents". However, instruction would influence the RAG evaluation, because LLMs with stronger instruction following ability are more likely to refuse to answer with "I cannot answer the question".

To address the above-mentioned issues, we propose a multi-level fine-grained benchmark named Placeholder-RAG-Benchmark, emphasizing the following progressive dimensions: multi-level filtering, cross-entity and multi-attribute composition, and multi-paradigm reasoning. Its construction is guided by principles from knowledge graphs(Liu et al. 2024). Besides, to decouple LLMs' parametric knowledge from the external knowledge, we formulate an innovative ***Placeholder***-based approach to decouple the contributions of the LLM's parametric knowledge and the external knowledge.

It is worth mentioning that, although our purpose is to evaluate the document utilization capability of the generative model in the RAG system, all the noise data associated with each piece of data can also be combined into a single library for evaluating retrieval models. This is because it involves many aspects (e.g., similar noise, multi-value, and multi-hop scenarios), which also present a significant challenge for retrieval models.

Our contributions are summarized as follows:

(1) A More Granular Evaluation Framework. We propose a novel Placeholder-RAG-Benchmark to enable fine-grained, multi-level assessments of LLMs within RAG systems. This framework introduces a more detailed dimensional structure for evaluating LLM capabilities, focusing on **(a) Multi-level Filtering**, **(b) Combination**, and **(c) Reference Reasoning** in effectively utilizing the retrieved corpus.

(2) A Robust Placeholder Algorithm. To achieve robust evaluations, a placeholder-based approach is formulated to mitigate the impact of the model's internal knowledge parameters, preventing LLMs from merely regurgitating answers and encouraging deriving conclusions from RAG-retrieved context.

(3) Comprehensive RAG-based LLM Assessments. Experiments demonstrate the limitations of representative LLMs in the RAG system's generation capabilities, particularly in error resilience and context faithfulness. Our benchmark provides a reproducible framework for developing more reliable and efficient RAG systems.

Related Work

RAG. RAG is a framework that enhances the capabilities of large language models by integrating external knowledge retrieval with text generation. The foundational work by (Lewis et al. 2020) introduced RAG, combining a pre-trained retriever with a sequence-to-sequence generator, allowing models to condition outputs on retrieved documents. Subsequent studies expanded RAG's applicability, such as REALM (Guu et al. 2020), which incorporates retrieval into pre-training, and variants like RAG-Sequence (Shi et al. 2024) for multi-hop question-answering. Besides, some studies have begun to investigate multi-round retrieval augmentation. For instance, ICRALM (Ram et al. 2023) and RETRO (Borgeaud et al. 2022) execute retrieval at the fixed token counts, while IRCot (Trivedi et al. 2022) triggers retrieval for each sentence. HANRAG adaptively adjusts the number of rounds based on the complexity of the question (Sun et al. 2025).

RAG Benchmark. Recent advancements in RAG evaluation have produced diverse automated frameworks, yet standardization remains elusive. RAGAS (Es et al. 2024) introduced LLM-based metrics (*e.g.*, context relevance, faithfulness) via GPT-3.5 prompting, while ARES (Saad-Falcon et al. 2023) proposed fine-tuned NLI models for similar assessments. Besides, (Chen et al. 2024) developed robustness tests through context perturbation. (Adlakha et al. 2024) formulated heuristic algorithms for faithfulness estimation. RAGBench(Friel, Belyi, and Sanyal 2025) is one of the few evaluation frameworks for large model generation that addresses aspects such as noise resistance, integration, and counterfactual detection. However, these areas are evaluated with relatively coarse granularity, and the distribution of topics within the data is often skewed.

Placeholder-RAG-Benchmark

In this section, we will introduce the details of the PRGB benchmark with the newly designed dynamic placeholder substitution strategy. Figure 1 illustrates the overall pipeline.

Triplet-based Meta-Data

We obtain several categories of popular entities from Wikipedia, such as Sports Events, Awards, Fictional Entities, and so on. From each category, a certain number of entities are selected as parent entities, denoted as E_p. To generate a rich collection of semantically related entities, we generate a set of semantic facets (D^p) for each parent entity ($e_p \in E_p$). For each pair of a parent entity p and one of its facets $d_i \in D^p$, we generate a set of child entities. The complete set of child entities is the union across all parent entities and their facets: $E_c = \bigcup_{e_p \in E_p} \bigcup_{d_i \in D^p} E_c^{(p,i)}$, all entities are denoted as $E = E_p \cup E_c$. For each entity, we extract a set of factual predicate-object pairs (r_j, o_j). These are then formulated into a set of triples $\Gamma_e = \{(e, rj, oj)\}$, the final triple set Γ for the entire dataset is the union of triples for all entities: $\Gamma = \bigcup_{e \in E}$. After manual verification, the dataset consists of 224 parent entities, 2,272 child entities, and 16,033 triplets.

Fine-grained Evaluation Dimension

To conduct a comprehensive assessment of model capabilities, we establish three fine-grained evaluation dimensions:**(a) Multi-level Filtering**, **(b) Combination**, and **(c) Reference Reasoning**. The evaluation instances for these tasks are systematically constructed from our dataset Γ.

Multi-Level Filtering. This dimension evaluates a model's ability to perform precise information extraction amidst varying levels of contextual noise. Formally, for a given child entity $e_c \in E_c^{(p,i)}$, we designate one of its attribute triplet, $\gamma_g \in \Gamma_c$, as the golden triplet. A corresponding query is then generated. We construct three distinct sets of distractor triplets (Γ_{noise}), each representing a different level of difficulty based on the semantic proximity of the noise source to the golden entity e_c. The noise levels are defined as follows:

Δ *Weak Noise*. Distractors are sourced from semantically distant entities. Specifically, this set comprises triplets whose subjects are child entities derived from a different parent entity.

$$\Gamma_{weak} = \{(e'_c, r', o') |\ e'_c \in E_c^{(p',i)} and\ e'_p \neq e_p\} \quad (1)$$

Δ *Moderate Noise*. Distractors originate from semantically similar entities. This set consists of triplets belonging to sibling entities-i.e., other child entities generated from the same parent entity e_p and via the same facet d_i.

$$\Gamma_{Mod} = \{(e'_c, r', o') |\ e'_c \in E_c^{(p,i)}\ and\ \ e'_c \neq e_c\} \quad (2)$$

Δ *Hard Noise*. The most challenging distractors are triplets representing generalized information that may conflict with the specific facts of ground truth. This set is composed of the attribute triplets of the parent entity e_p itself.

$$\Gamma_{Hard} = \{(e_p, r', o') |\ e_c \in E_c^{(p,i)}\} \quad (3)$$

Combination. This dimension assesses a model's capacity for knowledge combination, which requires aggregating information from multiple distinct triplets to formulate a comprehensive answer. We design three tasks with increasing compositional complexity.

Δ *Explicit Combination.* This task evaluates the model's ability to conjoin facts about two distinct but related entities. The golden triplets consist of two attribute triplets from different sibling entities that share the same predicate. Formally, we select two distinct child entities, $e_{c_1}, e_{c_2} \in E_c^{(p,i)}$, originating from the same parent entity e_p and facet d_i. The golden triplets are the set $\Gamma_g = \{(e_{c_1}, r, o_1), (e_{c_2}, r, o_2)\}$. The model is prompted with a query that requires both facts for a complete answer (e.g., " What were the host countries of the 2016 and 2020 Olympics, respectively?")

Δ *Multi-Value Combination.* This task assesses the model's ability to retrieve a complete set of objects for a single subject-predicate pair (a one-to-many relation). In our dataset, such relations are represented as multiple triplets sharing the same subject and predicate.The golden triplets, $\Gamma_g = \{(e_c, r, o_k) \in \Gamma_c |\ k = 1, ..., K\}$. The query prompts the model to list all associated values (e.g., "What were the

host cities of the 2018 FIFA World Cup?"), and a complete answer must include all objects $\{o_1, ..., o_K\}$.

Δ *Multi-Scenario Combination.* This task evaluates a model's ability to answer a general query about a parent entity by synthesizing information distributed across its multiple child entities. It tests for information summarization. The query is framed around the parent entity and a specific predicate, e.g., (e_p, r, o). The golden triplets, Γ_g, originate from the descendant entities of the e_p, rather than from e_p itself:

$$\Gamma_g = \{(e_c, r, o_c) | \ e_c \in E_c^{(p,i)}\} \tag{4}$$

Reference Reasoning. This mode mainly evaluates LLMs' ability to conduct multi-hop reasoning using information when answers are not directly provided in the source documents (previously mentioned abilities involve direct answering based on the documents).

Δ *Comparative Reasoning:* The model derives answers by comparing the values across two or more entities.

Δ *Deductive Reasoning:* This involves posing questions about attribute values through a major premise and a minor premise. The process can be divided into two types:

Deductive Reasoning-1: Inheritance-based Attribute Questioning
Question: How much does it cost to borrow books from Harvard?
Document: the book borrowing fees for Ivy League schools;
Explanation: Harvard belongs to the Ivy League schools, so the borrowing fees are subject to its parent.

During construction, inheritance-based attribute questioning identifies subclass entities that inherit attributes from parent class entities. The constructed questions focus on querying the inherited attributes of the subclass entity, with the golden document providing an explanation of the corresponding attribute of the parent class entity.

Deductive Reasoning-2: Relationship-based Value Questioning
Question: Is there a toll on the Guangzhou to Shenzhen expressway?
Document: Free expressways within the same provinces;
Explanation: The relation between Guangzhou and Shenzhen is the same province. So Free.

During evaluation, relationship-based value questioning requires the following steps: (1) retrieving the relevant context according to the parent attribute of value, (2) finding the relationship between entities via reasoning, and (3) obtaining the final answer via overall relationship-based reasoning.

Δ *Comparative Deductive Reasoning:* This form of reasoning involves posing questions about the attribute sets of different entities. It requires a higher-level generalization as the major premise to guide the comparison.

Comparative Deductive Reasoning
Question: What safety measures do Tesla and Mercedes have?
Document: Tesla equipped with airbags and automatic alarms; Mercedes equipped with airbags and automatic alarms;
Explanation: Airbags and automatic alarms belong to safety measures, and they are common features

Dynamic Placeholder Substitution

Given a query, dynamic placeholder substitution is to dynamically adjust the RAG contexts by changing the placeholder values. Then, we can observe whether LLMs can provide correct answers under these RAG contexts, thus effectively avoiding the internal knowledge of LLMs.

Placeholder-based Context Generation. We translate the structured triplets into natural language documents that serve as the context for our evaluation tasks. For a given triplet (e, r, o), we first create a template by masking its value: $(e, r, Placeholder)$. We then employ LLMs, specifically GPT-4o and Qwen-2.5-MAX, to author a document centered on the entity e. The generation is guided by two principles: (a) **Fact Embedding:** The document must contain the necessary information to correctly fill the placeholder, thereby embedding the original fact (e, r, o). For multivalued facts, multiple placeholders are used. (b) **Contextual Enrichment:** To create a realistic and complex context, the document is augmented with other true facts about the entity, drawn from its other triplets $(e, r', o') \in \Gamma$. The same procedure is used to generate golden documents(from golden triplets) and distractor documents (from noise triplets), ensuring stylistic consistency across all materials.

Based on $(e, r, Placeholder)$ and the question, we generate candidate values for the placeholder. These values must match the original value in the datatype. Furthermore, answers are constructed according to a set of placeholder values from the golden documents.

Methodological Rationale: Placeholder-based Evaluation. In fact, the dynamic placeholder substitution can enhance the robustness of the evaluation framework in two key ways: (a) By substituting critical information with placeholders, it reduces bias from the model's internal knowledge during evaluation. (b) By repeatedly testing the same sample with minimal changes, it lowers the probability of models guessing correct answers.

Benchmark Statistics

Statistical Information. We constructed an English dataset with 3,887 samples and a Chinese dataset with 3,387 samples. On average, the English dataset contains 1.74 golden documents per sample, while the Chinese dataset has 1.94. Regarding noise levels, the English dataset averages 5.80, 4.11, and 0.54 documents for weak noise, moderate noise, and hard noise, respectively. The corresponding values for the Chinese dataset are 1.94, 6.20, and 0.52. These ensure sufficient noise for robust experiments. Additionally, the English and Chinese datasets provide an average of 4.39 and 4.68 candidate placeholder values per sample, respectively, ensuring experimental stability. Further details

on subtasks are provided in the Appendix. Our data, built from diverse, multi-domain Wikidata entities, passed a style-transfer stress test across multiple prompts and LLMs. The consistent performance (e.g., GPT-4o accuracy at 0.67 ± 0.01) validates that the data is not heavily biased.

Data Quality Validation. Given the extensive use of GPT-4-based synthetic methods in our data construction pipeline, ensuring the quality of the generated data is of critical importance. Initially, we execute our evaluation pipeline using various state-of-the-art (SOTA) models under the simplest conditions—without introducing noise and with golden triplets provided as reference—to identify samples that multiple models fail to answer correctly (approximately 30% of the dataset). Subsequently, these problematic samples undergo manual validation and correction. The manual validation process focuses on two primary objectives: (1) ensuring that substituting specific values for placeholders in the document does not introduce contextual inconsistencies, and (2) verifying the accuracy of the relationships between placeholders and their corresponding answers.

Experiment

In this section, we conduct abundant experiments to show LLMs' performance under the recommended hyperparameter configurations, indicating the effectiveness of different proficiency tests. The highlighted conclusions are as follows.

Impact of Reasoning Modes: Working in "reasoning" versus "non-reasoning", LLMs exhibit huge performance variations across three evaluation dimensions. Analysis of reasoning chains indicates that reasoning models effectively process and reflect on noisy document hierarchies, thus aggregating all relevant information in combination tasks and performing complex reasoning in reference-reasoning tasks, while non-reasoning models are prone to selecting highly misleading answers without critical analysis.

Filter Ability vs. Model Size. Filter capability does not scale linearly with model size. Smaller models sometimes outperform larger ones in this task by directly extracting verbatim text from source documents, thereby matching key answer phrases more precisely. In contrast, larger models often paraphrase source text, occasionally omitting critical details. For instance, GPT-4 models tend to simplify date information, reporting only "month-year" while omitting "day," despite the day being explicitly provided in the source.

Performance on Combination and Inference Tasks. Larger models demonstrate clear advantages in combination and reference inference tasks, excelling at aggregating, reasoning, and constructing complete reasoning chains. Among the tested models, the Gemma series showed stronger reasoning capabilities, while the Qwen series exhibited superior performance in information integration and combination.

Experimental Setup

Hyper-parameters. We employed a configuration where both l1 noise doc and l2 noise doc were set to 4, and l3 noise doc was set to 1. Three placeholders were used for the evaluation in our main experiments. All models were tested on both Chinese and English datasets. For closed-source models, we accessed them via APIs, while for open-source models, experiments were conducted using 8 NVIDIA A100 GPUs with a batch size of 16. The VLLM framework was used to implement and run the open-source models.

Algorithm 1: Evaluation Pipeline

Input: LLM, Placeholder-RAG-Benchmark Dataset D, $n < N$ as candidate placeholder number
for each data point $d \in D$ **do**
  for i in range(n) **do**
    result$_i$=infer(LLM, $d\&P_i$)
  end for
  Score=AVG(metric(result$_{\{1,\ldots,N\}}$,GT))
end for
Calculate the final evaluation score, aggregated by different RAG tasks T_i
return Scores$_{T_i}$

Evaluated LLMs. We utilized recent strong LLMs with varying sizes, including GPT-4o, Claude 3.7, Gemini, Qwen Series and DeepSeek V3. Notably, Qwen3 series are evaluated under both enabled and disabled reasoning modes.

Evaluation Pipeline. Let D, P_N denote one data point(*i.e.*, query, RAG context, answer) and the placeholder set, respectively, where N denotes the number of candidate placeholder values. The evaluation pipeline is illustrated in Algorithm 1. During evaluation, hyperparameters are set to define the number of placeholders per data point and the configuration of noise documents. The algorithm iteratively samples data points by replacing placeholders, computes inference results for each placeholder, and calculates an average score per sample. Finally, it aggregates evaluation scores across different RAG tasks.

Evaluation Metrics. For evaluation, we mainly use two metrics: accuracy (Covered Exact Match) and GPT evaluation. Accuracy refers to whether the keywords appear in the document. We use logical operators "or (|)" and "and (&)" to improve the accuracy rate. The scenarios for using the "and" condition include situations where multiple possible answers must all be mentioned, and when a relatively long phrase is broken down into key, small elements. As for GPT evaluation, we directly let GPT4o judge whether the answer is correct according to the question.

Placeholders This part primarily examines the role of placeholders in evaluation methods. Experiments were conducted across different scenarios using three placeholders to test various models. The study explored the performance differences of multiple placeholders and the stability of the answers. We utilized models of varying sizes from the Qwen2.5 series to observe the outcome changes when three placeholders were modified for the same dataset. Specifically, these outcomes included scenarios where all answers were correct, all answers were incorrect, and some answers were incorrect. The results reveal that as model size increases, the proportion of completely correct answers rises, while the instances of partial errors and complete errors decrease; notably, the proportion of partial errors declines sig-

Models	ZH				EN			
	Overall	Multi-Level Filter	Combi-nation	Rea-soning	Overall	Multi-Level Filter	Combi-nation	Rea-soning
Gemini-2.5-pro-preview	**87.33**	**97.92**	**94.20**	70.18	**84.89**	**94.89**	**85.32**	**76.09**
Claude-3.7-sonnet	***85.74***	***97.62***	***90.59***	**70.39**	***82.96***	***93.18***	***82.13***	***76.51***
Gemini-2.5-flash-preview	81.85	93.92	88.54	63.86	79.20	90.69	80.30	67.90
Qwen3-235B-A22B	80.76	94.92	88.18	60.23	78.68	90.56	78.32	69.97
Qwen3-30B-A3B	80.45	95.87	86.11	61.42	79.09	91.01	78.01	71.78
Deepseek-V3(241226)	77.54	94.58	81.00	60.32	79.02	89.91	77.18	74.03
Qwen3-235B-A22B w/o think	75.20	91.50	79.67	57.14	70.27	83.95	66.37	67.15
Qwen-2.5-MAX	74.43	93.25	78.28	55.37	78.45	89.32	75.83	65.89
GPT4.1	71.15	88.08	73.28	56.76	72.27	83.95	69.41	68.03
Qwen3-30B-A3B w/o think	71.05	91.08	72.22	54.76	65.38	84.76	61.12	58.47
Gemma3_27b	70.24	92.21	73.09	50.24	79.18	92.03	78.00	71.33
GPT4o-1120	69.88	92.17	70.21	50.60	70.89	81.62	65.69	64.83
Qwen3_32B	69.69	89.75	75.74	46.70	78.05	90.69	77.23	69.65
Hunyuan-80B-A13B	68.84	93.50	68.94	50.64	73.42	86.89	71.58	66.38
Qwen2.5_72B	64.87	92.92	64.99	44.14	68.90	87.01	64.30	63.69
Gemma3_12b	64.10	60.20	89.92	50.52	72.35	87.42	68.46	68.12
Qwen3_8B	63.04	86.87	67.49	39.47	76.80	88.36	76.27	68.71
Qwen3_32B w/o think	60.73	89.50	59.53	41.30	68.30	84.35	63.74	64.59
Qwen2.5_32B	58.76	92.00	51.33	44.60	66.70	85.66	63.04	58.92
Qwen2.5_14B	55.94	89.42	52.69	35.87	63.29	84.40	57.35	58.34
Qwen2.5_7B	49.31	83.29	47.47	26.92	63.16	81.90	56.76	61.00
Qwen3_8B w/o think	50.02	83.96	47.83	28.17	64.71	83.21	58.93	61.52
Gemma3_4b	47.67	78.33	37.41	39.26	57.58	77.98	48.50	59.41

Table 1: Performance of Various State-of-the-Art Models in this benchmark. Results unavailable in public reports are marked as "–". Bold values indicate the best experimental results among small-scale models, italic bold values indicate the second-best experimental results, and underlined values denote the third-best experimental results.

nificantly slower than the proportion of complete errors.

Furthermore, analyzing placeholder-level performance, particularly for partially correct outcomes, reveals that the proportion of such answers remains relatively stable at around 30%. This indicates that placeholders influence models of all sizes. However, as shown in Figure 2, the proportion of partially correct answers increases steadily with model size, suggesting that larger models exhibit greater stability when handling variable RAG documents.

Figure 2: Qwen2.5 Series Model Performance Comparison, among these, the line graph for "partial correct score" represents the specific scores achieved in partially correct cases.

Experiment About Assessment Dimension

In this section, we examine the effectiveness of the proposed evaluation dimensions. For better control of variables, we conducted experiments using Qwen-2.5 models of varying sizes to analyze how performance changes across: **(a) Multi-level Filtering**, **(b) Combination**, and **(c) Reference Reasoning**, as reflected in our evaluation dataset.

For **Multi-Level Filtering** tasks, we assess how noise of varying difficulty levels affects the performance. To ensure controlled experimentation, we systematically adjust the proportions of noisy documents in the filter task. From Table 3 and 4, we observe that increasing moderate and hard noise levels degrade model performance on even simple filter tasks, while keeping the total number of noisy documents constant. This demonstrates the effectiveness of our multi-level noise design. Moreover, we observe that while the 7B model performs well in scenarios with only weak noise, its performance deteriorates significantly as noise becomes more challenging. In contrast, larger models demonstrate greater resilience to handling more complex noise.

In the **Combination** task, we experiment with three distinct task types. In Table 5, for L1 tasks, smaller models like Qwen2.5-7B perform well, whereas for the second and third task types, larger models exhibit a significant advantage.

The results about **Reference Reasoning** are shown in Table 6. There is no absolute measure of difficulty among its four aspects. However, we can observe that larger models generally possess stronger overall document reasoning capabilities. Nevertheless, the category of deductive reasoning is more challenging compared to other scenarios. On the one hand, in deductive reasoning, the model needs to identify

	Multi-Level Filtering	Combination	Reference Reasoning
Question	When did Mozart perform the Turkish March?	What is the shape of eucalyptus leaves?	Which country can the Best Mentor-Disciple Duo in East Asian martial arts visit?
Answer	**November 9, 1786**	**Lanceolate & Oval**	**the United States**
Documents	*Golden Document* Mozart's performance of the Turkish...it was brilliantly presented at the Vienna Conservatory on **November 9, 1786** *Noisy Document* The Turkish March,... When it was performed by Bobby McFerrin... The date of this performance was ***October 7, 1783***	*Golden Document* ...In addition, the blue gum tree has unique leaf shapes, including **lanceolate** ones. ...The leaves of eucalyptus trees in the Northwest region are usually **oval-shaped**	*Golden Document* The Best Mentor-Disciple Duo in Karate can travel to **the United States**. *Noisy Document* The Best Mentor-Disciple Duo in Wrestling can visit the Shaolin Temple in ***China*** for a pilgrimage.
Response	***October 7, 1783****(Noisy Error Answer)*	The leaves of the blue gum tree are mainly **lanceolate**(Partial Response).	***China****(Noisy Error Answer)*

Table 2: Error Cases of Three Dimension by GPT-4o-1120.

Noise Config Filtering-en	Config 1	Config 2	Config 3	Config 4
Qwen2.5 7B	89.01	83.74	85.09	84.72
Qwen2.5 32B	82.68	80.80	80.35	79.34
Qwen2.5 72B	89.09	88.15	87.58	87.42
Qwen2.5 MAX	85.38	84.84	84.68	84.56

Table 3: The model's performance under different noise configurations is evaluated on the filter task. Specifically, Config 1 represents the noise ratio of Weak:Moderate:Hard as 5:0:0. Config 2, Config 3, and Config 4 correspond to ratios of 3:2:0, 1:4:0, and 1:3:1, respectively.

Noise Config Filtering-zh	Config 1	Config 2	Config 3	Config 4
Qwen2.5 7B	95.33	84.54	83.92	80.83
Qwen2.5 14B	95.28	85.62	84.38	80.96
Qwen2.5 32B	95.64	88.93	88.24	85.13
Qwen2.5 72B	95.58	93.71	94.17	92.92
Qwen2.5 MAX	95.87	94.17	93.33	93.13

Table 4: Different noise ratios on the Chinese dataset.

the required attributes and then find the common attributes. It requires numerous cognitive leaps to arrive at the correct answer, posing a significant challenge to the model.

Conclusion

This paper introduces PRGB, a placeholder-based RAG benchmark to evaluate LLMs' information utilization capabilities from multiple dimensions. Starting from triples, we constructed two human-reviewed evaluation datasets: 3,887 English samples and 3,387 Chinese samples, synthesized using GPT-4. The benchmark comprehensively assesses models across three dimensions: multi-level filtering, complex composition, and multi-paradigm reasoning capabilities. To enhance evaluation robustness, we employ a dynamic placeholder strategy to replace critical information in reference documents during testing, mitigating the interference of knowledge in the model's internal parameters. In the experiments, we conducted a thorough evaluation of various SOTA models, offering insights into model selection for RAG scenarios. Looking ahead, we aim to optimize evaluation metrics and establish a more comprehensive framework to better benchmark RAG-based tasks.

Model	Overall	Combination		
	(%)	L1 (%)	L2 (%)	L3 (%)
Qwen2.5 7B	47.47	74.07	39.66	28.60
Qwen2.5 14B	52.69	79.27	47.00	31.73
Qwen2.5 32B	51.33	85.07	45.32	23.53
Qwen2.5 72B	64.99	84.00	56.36	54.53
Qwen2.5 MAX	78.28	86.53	73.06	75.20

Table 5: Performance of Qwen2.5 series models on combination tasks across various english datasets: L1 denotes explicit composite questions, L2 refers to multi-value questions, and L3 represents multi-scenario analysis Questions.

Model	Overall Reasoning	Comp (%)	Comp Deduc (%)	Deduc V1 (%)	Deduc V2 (%)
Qwen2.5 7B	26.9	33.0	20.3	26.1	41.8
Qwen2.5 14B	35.9	42.7	30.8	33.7	48.7
Qwen2.5 32B	44.6	53.1	36.8	43.3	62.0
Qwen2.5 72B	44.1	48.4	33.7	45.3	66.5
Qwen2.5 MAX	55.4	67.4	46.2	55.1	73.2

Table 6: Multi-Paradigm Reasoning Performance of Qwen2.5 Series. (Comp: Comparative; Comp Deduc: Comparative Deductive; Deduc V1: Inheritance-based; Deduc V2: Relationship-Based Deductive Reasoning)

References

Adlakha, V.; BehnamGhader, P.; Lu, X. H.; Meade, N.; and Reddy, S. 2024. Evaluating correctness and faithfulness of instruction-following models for question answering. *Transactions of the Association for Computational Linguistics*, 12: 681–699.

Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; Rutherford, E.; Millican, K.; Van Den Driessche, G. B.; Lespiau, J.-B.; Damoc, B.; Clark, A.; et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, 2206–2240. PMLR.

Chen, J.; Lin, H.; Han, X.; and Sun, L. 2023. Benchmarking Large Language Models in Retrieval-Augmented Generation. arXiv:2309.01431.

Chen, J.; Lin, H.; Han, X.; and Sun, L. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17754–17762.

Es, S.; James, J.; Anke, L. E.; and Schockaert, S. 2024. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, 150–158.

Friel, R.; Belyi, M.; and Sanyal, A. 2025. RAGBench: Explainable Benchmark for Retrieval-Augmented Generation Systems. arXiv:2407.11005.

Gan, C.; Yang, D.; Hu, B.; et al. 2025. POLYRAG: Integrating Polyviews into Retrieval-Augmented Generation for Medical Applications. arXiv:2504.14917.

Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; and Chang, M. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, 3929–3938. PMLR.

Jiao, Y.; Tan, Z.; Yang, D.; Sun, D.; Feng, J.; Wang, J.; and Wei, P. 2025. HIRAG: Hierarchical-Thought Instruction-Tuning Retrieval-Augmented Generation. arXiv:2507.05714.

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. arXiv:1705.03551.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474.

Liu, Z.; Hua, Y.; Chen, M.; Zhang, Y.; Chen, Z.; Liang, L.; Chen, H.; and Zhang, W. 2024. UniHR: Hierarchical Representation Learning for Unified Knowledge Graph Link Prediction. *arXiv preprint arXiv:2411.07019*.

Long, M.; Sun, D.; Yang, D.; Wang, J.; Shen, Y.; Wang, J.; Wei, P.; Gu, J.; and Wang, J. 2025. DIVER: A Multi-Stage Approach for Reasoning-intensive Information Retrieval. arXiv:2508.07995.

Mallen, A.; Asai, A.; Zhong, V.; Das, R.; Khashabi, D.; and Hajishirzi, H. 2023. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. arXiv:2212.10511.

Min, S.; Michael, J.; Hajishirzi, H.; and Zettlemoyer, L. 2020. AmbigQA: Answering ambiguous open-domain questions. *arXiv preprint arXiv:2004.10645*.

Ram, O.; Levine, Y.; Dalmedigos, I.; Muhlgay, D.; Shashua, A.; Leyton-Brown, K.; and Shoham, Y. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11: 1316–1331.

Saad-Falcon, J.; Khattab, O.; Potts, C.; and Zaharia, M. 2023. Ares: An automated evaluation framework for retrieval-augmented generation systems. *arXiv preprint arXiv:2311.09476*.

Shi, Z.; Sun, W.; Gao, S.; Ren, P.; Chen, Z.; and Ren, Z. 2024. Generate-then-ground in retrieval-augmented generation for multi-hop question answering. *arXiv preprint arXiv:2406.14891*.

Sun, D.; Yang, D.; Shen, Y.; Jiao, Y.; Tan, Z.; Feng, J.; Zhong, L.; Wang, J.; Wei, P.; and Gu, J. 2025. HANRAG: Heuristic Accurate Noise-resistant Retrieval-Augmented Generation for Multi-hop Question Answering. arXiv:2509.09713.

Trivedi, H.; Balasubramanian, N.; Khot, T.; and Sabharwal, A. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Voorhees, E. M. 1998. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 315–323.