OCEAN: Observation Consensus over Evidence with Dynamic Routing for Retrieval-augmented Generation

Anonymous authors
Paper under double-blind review

Abstract

Retrieval-Augmented Generation (RAG) has substantially enhanced the reasoning capabilities of Large Language Models (LLMs) by incorporating external evidence retrieved from knowledge sources. However, leveraging multiple pieces of retrieved evidence inevitably incurs additional computational overhead, which has motivated growing interest in evidence compression for RAG. Existing methods typically compress evidence by either extracting salient contexts or generating evidence summaries. Nevertheless, such compression is inherently lossy: it discards fine-grained evidence and, in the abstractive case, may introduce hallucinations, while prior work largely overlooks how the discarded information can be recovered at decoding time. To address these limitations, we propose Observation Consensus over Evidence with dynAmic routiNg (`OCEAN`) for RAG, which compensates for the information loss of compression-based RAG *at the decoding stage* rather than by enlarging the input. The core idea of `OCEAN` is to combine a question-conditioned evidence summary with complementary multi-view evidence through uncertainty-aware dynamic logit routing. Specifically, alongside the compressed summary, `OCEAN` exploits a small high-fidelity subset of semantically similar evidence and the model's parametric knowledge as complementary information sources, and integrates these multi-view distributions via uncertainty-aware dynamic logit routing to produce the final response. Additionally, extensive experiments on four open-domain QA benchmarks with two diverse backbone models demonstrate the effectiveness and robustness of `OCEAN`.

1 Introduction

Large language models (LLMs) (Achiam et al., 2023; Touvron et al., 2023; Zhao et al., 2026) have demonstrated remarkable capabilities across a wide range of natural language processing (NLP) tasks, including question answering (Lazaridou et al., 2022; Yasunaga et al., 2021; Kamalloo et al., 2023), complex reasoning (Huang & Chang, 2023; Wei et al., 2022), and text generation (Li et al., 2024; Xu et al., 2025). Through pre-training on large-scale corpora, LLMs acquire broad world knowledge and embed it in their parameters (Xiang et al., 2023; Yu et al., 2024), enabling them to serve as general-purpose models that answer diverse questions without accessing external knowledge sources at inference time (Naveed et al., 2025). Despite these advances, the knowledge encoded in LLM parameters is inherently constrained by the coverage and temporal scope of their pre-training corpora. Moreover, model parameters typically remain fixed during inference unless the model is further fine-tuned or updated. These limitations reduce the ability of LLMs to reliably answer up-to-date, specialized, and long-tail queries (Kandpal et al., 2023), making them more prone to generating plausible but factually incorrect responses, a phenomenon commonly referred to as hallucination (Liu et al., 2024; Xu et al., 2024; Huang et al., 2025).

To mitigate hallucinations for LLMs, prior studies (Huang et al., 2025) have explored data-centric approaches (Abbas et al., 2023; Tirumala et al., 2023), parameter-level interventions (Tian et al., 2024; Zhang et al., 2024), and inference-time augmentation (Lewis et al., 2020; Chuang et al., 2024). Among them, retrieval-augmented generation (RAG) has emerged as a widely adopted paradigm as it enables LLMs to access external knowledge without modifying their underlying parameters (Chen et al., 2024). A typical RAG system retrieves relevant passages from an external corpus and conditions the reader model on the retrieved

evidence to generate an answer (Gao et al., 2023). Despite its effectiveness, directly feeding all retrieved passages into the reader is often inefficient and potentially detrimental. Retrieved contexts commonly contain redundant, weakly relevant, or distracting information, resulting in longer inputs and increased inference costs. Additionally, excessive context may obscure critical evidence and make it more difficult for the reader to identify information that is truly useful for answering the query. Therefore, evidence compression has been introduced to reduce the retrieved context while retaining its most informative content, offering a practical balance between answer quality and computational efficiency (Xu et al., 2023; Verma, 2024).

Existing evidence compression methods generally adopt either extractive or abstractive strategies. Extractive methods retain a subset of tokens, sentences, or passages according to their estimated relevance to the query (Wang et al., 2023), whereas abstractive methods summarize or rewrite the retrieved evidence into a more compact representation (Chevalier et al., 2023; Qin et al., 2024; Rau et al., 2024). More recent approaches further incorporate query-aware or generation-aware signals to preserve information that is likely to benefit downstream generation. Despite their effectiveness in reducing context length, these methods still face two major limitations. First, compression inherently risks discarding fine-grained yet answer-critical information or preserving misleading details from imperfect retrieval results. Second, the compressed representation is typically treated as the sole evidence source throughout decoding, leaving the reader unable to detect or recover from incomplete, distorted, or off-target evidence (Shi et al., 2023). Consequently, even a locally plausible but misleading detail in the compressed context may silently derail the final answer.

Towards this end, we propose `OCEAN`, a training-free multi-view decoding framework for compression-based RAG. Our key observation is that the evidence for a query is spread across imperfect sources that rarely err in the same way. `OCEAN` therefore builds three complementary views of the query: a *compressed view* from the summary; a *complementary view* from a small high-fidelity subset of retrieved passages; and a *parametric view* from the reader queried without retrieval (Roberts et al., 2020). At each step, *uncertainty-aware dynamic logit routing* fuses the three next-token distributions, weighting each view by its entropy-based confidence so that the ensemble defers, token by token, to the most certain view. When the views disagree, a gated look-ahead rescores candidate tokens via short rollouts under the same policy, firing only at high-uncertainty steps. Operating entirely at decoding, `OCEAN` is orthogonal to the retriever, compressor, and reader, and plugs into any RAG pipeline without further training.

In conclusion, our contributions are summarized as follows:

❶ ***Novel Perspective.*** We show that compression methods inevitably incur information loss, which can be remedied during the decoding phase.

❷ ***New Methodology.*** We propose `OCEAN`, a training-free multi-view decoding framework that fuses compressed, complementary, and parametric views via uncertainty-aware dynamic logit routing with a gated look-ahead.

❸ ***Empirical Analysis.*** Across four open-domain QA benchmarks and two reader backbones, `OCEAN` consistently outperforms strong compression baselines and nearly matches full-context generation while preserving efficiency.

2 Related Work

Evidence Compression for RAG. RAG lead performance improvement in open domain question answering by leveraging the documents retrieved (Lewis et al., 2020). As the number of documents grows,the quadratic cost of self-attention makes this approach prohibitively expensive. (Vaswani et al., 2017). To mitigate this, prior work compresses the retrieved evidence before passing it to the reader which could be conducted at either the token level or the latent level.Methods at the token level operate in the discrete text space. Some are extractive, selecting salient tokens or sentences from the documents. LLMLingua (Jiang et al., 2023) prunes uninformative tokens by their perplexity, LLMLingua-2 (Pan et al., 2024) trains a token classifier for the same purpose, and the extractive variant of RECOMP (Xu et al., 2023) selects the most relevant sentences. Others are abstractive, using a language model to generate a condensed summary. The abstractive variant of RECOMP trains a summarizer conditioned on the query, CompAct (Yoon et al.,

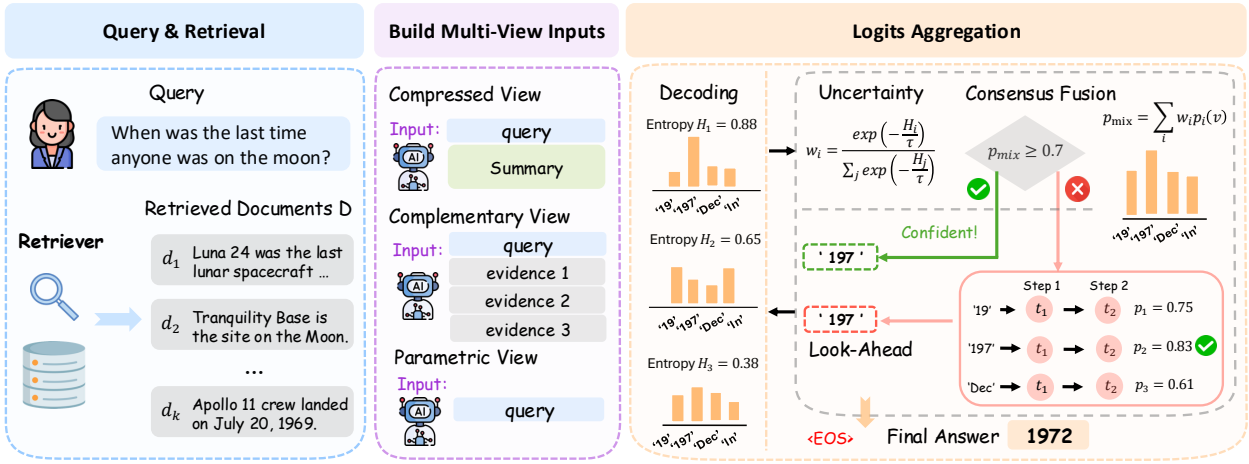

Figure 1: Overview of OCEAN. For a query, a retriever returns documents $\mathcal{D}$ and a compressor summarizes them. OCEAN then builds three views, a *compressed view* (query and summary), a *complementary view* (query and the top-3 retrieved passages), and a *parametric view* (query only). At each decoding step, the three views produce next-token distributions whose entropies H_1, H_2, H_3 measure their confidence. These are fused into a single distribution p_{mix} that favors the more confident views, from which OCEAN emits the next token.

2024) fine-tunes a language model to iteratively compress the context, and FaviComp (Jung et al., 2025) performs generative compression conditioned on the query.Methods at the latent level compress documents into continuous representations rather than text. xRAG (Cheng et al., 2024) encodes each document as a single token in the LLM's representation space, AutoCompressor (Chevalier et al., 2023) and ICAE (Ge et al., 2023) distill long contexts into a few soft summary vectors, and PISCO (Louis et al., 2025) distills each document into a small set of embeddings.

Ensemble Decoding. A complementary line of work combines multiple output distributions at the *decoding* stage to improve generation without modifying model parameters. These methods differ in *how* they combine the distributions. *Contrastive* methods subtract one distribution from another to amplify a desired signal. Contrastive decoding (Li et al., 2023) subtracts a weak amateur model from a strong expert, CAD (Shi et al., 2024) subtracts the parametric, context-free distribution to amplify the retrieved evidence, and DExperts (Liu et al., 2021) combines expert and anti-expert distributions in a product. These methods use a fixed contrast strength, which AdaCAD (Wang et al., 2025) relaxes by inferring the weight per instance from the conflict between the two distributions. *Mixture* methods instead additively blend distributions. FaviComp (Jung et al., 2025) averages the compressed evidence with the reader's parametric distribution, and PCED (Corallo & Papotti, 2026) aggregates per-document experts at decoding time. OCEAN is a mixture method, but it differs from prior work in two ways. First, it blends three complementary views, compressed, complementary, and parametric, so it aggregates multiple evidence sources and keeps the parametric prior as a fallback, while the views fail differently and cover one another. Second, whereas prior mixture methods fix their weights and contrastive methods tune a single contrast strength, OCEAN routes per token, deferring to whichever view is most confident. This lets OCEAN recover the evidence discarded by compression rather than merely re-weighting a single context channel.

3 Methodology

Problem Definition. Given a query q, a RAG system retrieves a set of documents $\mathcal{D} = \{d_1, d_2, \ldots, d_k\}$ relevant to q, and generates an answer a conditioned on $(q, \mathcal{D})$. As k grows, directly conditioning on the concatenated documents becomes resource-inefficient in both memory and latency, motivating the use of a compressed representation $\tilde{\mathcal{D}} := \mathcal{C}(q, \mathcal{D})$, produced by an off-the-shelf compressor $\mathcal{C}$. However, compression is inherently lossy: $\tilde{\mathcal{D}}$ retains strictly less information than $\mathcal{D}$, which upper-bounds the quality of any

answer derived solely from $\tilde{\mathcal{D}}$. This motivates the need to recover the discarded information at decoding time, without sacrificing the efficiency gains of compression.The goal of OCEAN is to approach the answer quality attainable from the full retrieved documents while incurring only the memory and latency cost of the compressed context.

3.1 Overview of OCEAN

We propose OCEAN, a multi-view decoding framework that recovers the information lost to compression at decoding time without enlarging the input. Rather than rely on a single compressed context, OCEAN decodes from three different views of the context. The *compressed view* conditions the reader on the summary, giving broad but lossy coverage at low cost. The *complementary view* conditions on a small high-fidelity subset of the retrieved passages, supplying precise evidence the summary may have dropped. The *parametric view* queries the reader without retrieval, drawing on its parametric knowledge as a fallback when the evidence is incomplete. At each decoding step, the three views yield next-token distributions whose entropies measure their confidence, and OCEAN fuses them into a single mixture weighted by that confidence. When the views disagree, the resulting flat mixture can mislead greedy decoding, so OCEAN applies a confidence-gated look-ahead that scores a few candidate tokens by short rollouts under the same mixing rule before committing. Figure 1 illustrates the overall pipeline of OCEAN.

3.2 Compression for RAG

Following recent compression-based RAG approaches (Wang et al., 2023; Xu et al., 2023; Jeong et al., 2025), we reduce the input length by replacing the raw retrieved documents with a compressed summary. Given a query q and the top-k retrieved documents $\mathcal{D} = \{d_1, \ldots, d_k\}$, a compressor $\mathcal{C}$ produces a summary

$$\tilde{\mathcal{D}} = \mathcal{C}(q, \mathcal{D}), \tag{1}$$

which condenses the relevant evidence into a much shorter context with $|\tilde{\mathcal{D}}| \ll |\mathcal{D}|$. Let $p_\theta(\cdot \mid \cdot)$ denote the next-token distribution of the reader LLM. Conditioning on the summary yields the *compressed view*,

$$P_1(\cdot) = p_\theta(\cdot \mid q, \tilde{\mathcal{D}}), \tag{2}$$

which serves as the primary, low-cost evidence channel in OCEAN. The exact prompt used by $\mathcal{C}$ is shown in Figure 2.

Compression Prompt

```
Generate a summary that is helpful for answering the question.  Contain information only related to the
question.Output nothing except the summary.
Question:  {q}
Passages:  {d_1, ...  d_k}
Summary:
```

Figure 2: Prompt template used by the compressor $\mathcal{C}$. Given the question q and the top-k retrieved passages $\{d_1, \ldots, d_k\}$, the compressor produces a concise summary $\tilde{\mathcal{D}}$ that retains only the evidence relevant to answering q.

3.3 Multi-view Evidence Ensembling

Compression yields a resource-efficient context for long-input scenarios, but the text-level eviction inevitably discards information that may be critical for answering the query. To compensate, we introduce two complementary views alongside the compressed view P_1 and combine all three at decoding time.

Complementary View. We additionally retain a small high-fidelity subset $\mathcal{D}_c = \{d_{i_1}, \ldots, d_{i_m}\} \subseteq \mathcal{D}$ with $m \ll k$, selected to provide precise evidence without re-introducing the original context length. Conditioning the reader on this subset yields the *complementary view*,

$$P_2(\cdot) = p_\theta(\cdot \mid q, \mathcal{D}_c). \tag{3}$$

Because $|\mathcal{D}_c|$ is small, the additional inference cost is negligible relative to processing the full $\mathcal{D}$. The context-augmented prompt in Figure 3 serves the compressed and complementary views:

Evaluation Prompt (with Context)

```
Answer the question based on the provided context. Output only the answer as a short name or phrase. No
explanation, no prefix, no full sentences.
Context: {context}
Question: {q}
Answer:
```

Figure 3: Evaluation prompt for the compressed and complementary views. The context is the compressed summary $\tilde{\mathcal{D}}$ for the compressed view and the high-fidelity subset $\mathcal{D}_c$ for the complementary view.

Parametric View. The reader LLM also encodes substantial world knowledge in its parameters, which can serve as a fallback when the retrieved evidence is incomplete or noisy. We expose this signal directly through the *parametric view*,

$$P_3(\cdot) = p_\theta(\cdot \mid q), \tag{4}$$

where $P_3(\cdot)$ denotes the reader's next-token distribution obtained by querying the reader on q alone, without any retrieved context. The output therefore reflects only the knowledge encoded in the model parameters θ. Furthermore, the closed-book prompt in Figure 4 serves the parametric view, which is shown as follows:

Evaluation Prompt (Closed-Book)

```
Answer the question from your own knowledge. Output only the answer as a short name or phrase. No
explanation, no prefix, no full sentences.
Question: {q}
Answer:
```

Figure 4: Answering prompt for the parametric view P_3. The reader is queried on the question q alone, with no retrieved context, so that the response reflects only the knowledge encoded in the model's parameters.

Multi-view Ensemble. At each decoding step, `OCEAN` aggregates the three distributions through an operator M,

$$a = M(P_1,\, P_2,\, P_3), \tag{5}$$

where M adaptively weights P_1, P_2, and P_3 based on their predictive entropy, allowing the model to defer to whichever view is most confident at the current generation step. We detail the entropy-based weighting scheme in Section 3.4.

3.4 Uncertainty-driven Dynamic Logit Aggregation

To merge the information from the three views, we introduce an uncertainty-driven logit aggregation algorithm. At each decoding step, each view produces a probability distribution over the vocabulary, and we quantify its confidence by the entropy of this distribution. Specifically, for view i with predictive distribution p_i, we compute

$$H_i = -\sum_{v \in \mathcal{V}} p_i(v) \log p_i(v), \tag{6}$$

where $\mathcal{V}$ is the vocabulary and H_i is the Shannon entropy of view i's next-token distribution. A larger H_i reflects a flatter distribution, indicating that view i is uncertain about which token to emit and should contribute less to the final decision. We therefore convert the entropies into mixing weights through a temperature-scaled softmax over their negations,

$$w_i = \frac{\exp(-H_i/\tau)}{\sum_j \exp(-H_j/\tau)}, \tag{7}$$

where $w_i \in [0,1]$ is the weight assigned to view i and satisfies $\sum_i w_i = 1$, and $\tau > 0$ is a temperature that controls how aggressively confident views are favored. A small τ makes the lowest-entropy view dominate, while a large τ recovers a uniform average over the three views. We then form the mixed distribution as the corresponding weighted average,

$$p_{\text{mix}}(v) = \sum_i w_i \, p_i(v), \tag{8}$$

where $p_{\text{mix}}(v)$ is the fused probability of token v. Since each p_i is a valid distribution and the weights satisfy $\sum_i w_i = 1$, the mixture p_{mix} is itself a normalized distribution over $\mathcal{V}$. The next token is finally emitted as $\arg\max_v p_{\text{mix}}(v)$.

3.5 Look-ahead Mechanism

The entropy-weighted mixture in Section 3.4 commits to $\arg\max_v p_{\text{mix}}(v)$ at every step. When the three views agree, p_{mix} is sharply peaked and this greedy choice is reliable; when they disagree, the mixture flattens and its top-1 token becomes sensitive to small differences in the per-view weights, so a single ambiguous step can derail the remainder of the generation. We mitigate this with a *gated look-ahead* that, only at such uncertain steps, defers the decision to a short rollout under the same ensemble policy.

Confidence Gate. Let $\pi = \max_v p_{\text{mix}}(v)$ be the peak mass of the mixed distribution, a proxy for the ensemble's confidence at the current step. Given a threshold $\gamma \in [0,1]$, `OCEAN` commits greedily whenever the ensemble is confident,

$$\pi \geq \gamma : \qquad v^\star = \arg\max_v p_{\text{mix}}(v), \tag{9}$$

and invokes look-ahead only when $\pi < \gamma$. The gate keeps the mechanism inexpensive in aggregate, as the rollout fires only on the minority of steps where the views genuinely conflict; setting $\gamma = 0$ disables look-ahead and recovers the pure entropy mixture of Section 3.4.

Candidate Set. At an uncertain step we restrict attention to a small set of plausible first tokens, formed as the union of the mixture's top-K tokens and each view's own most-likely token,

$$\mathcal{V}_{\text{cand}} = \text{Top-}K\big(p_{\text{mix}}\big) \;\cup\; \big\{\, \arg\max_v P_i(v) \,\big\}_{i=1}^{3}, \tag{10}$$

Including each view's argmax ensures that a confident minority view, whose preferred token may be suppressed in the averaged p_{mix}, is still admitted as a candidate.

Rollout Scoring. Each candidate $c \in \mathcal{V}_{\text{cand}}$ is forced as the next token of all three views, after which generation continues for up to d additional steps under the same entropy-mixing operator. At rollout step s we recompute the per-view distributions, their entropy weights w_i, and the mixture $p^{(s)}_{\text{mix}}$, then greedily emit $\hat{v}_s = \arg\max_v p^{(s)}_{\text{mix}}(v)$ and advance every view with this shared token, terminating early if $\hat{v}_s$ is an end-of-sequence token. The candidate is then scored by the length-normalized joint log-likelihood of the realized continuation,

$$S(c) = \frac{1}{L_c}\left(\log p_{\text{mix}}(c) + \sum_{s=1}^{L_c-1} \log p^{(s)}_{\text{mix}}(\hat{v}_s)\right). \tag{11}$$

Here, $p_{\text{mix}}(c)$ is the mixture probability of candidate c at the current step. The symbols $\hat{v}_s$ and $p^{(s)}_{\text{mix}}$ denote the emitted token and the mixed distribution at rollout step s, respectively. The normalizer $L_c \leq d+1$ is the number of scored tokens, namely the forced candidate together with its rollout continuation, which is truncated at an end-of-sequence token. Finally, `OCEAN` emits the highest-scoring candidate as the token for the current step,

$$v^\star = \arg\max_{c \in \mathcal{V}_{\text{cand}}} S(c), \tag{12}$$

and resumes mixture decoding from the next step. The rollout serves only to score candidates and does not commit any token beyond $v^\star$.

Algorithm 1: Overall pipeline of `OCEAN`

Input: query q, retrieved documents $\mathcal{D}$, reader p_θ, compressor $\mathcal{C}$, hyperparameters τ, γ, K, d, T
Output: answer a

Build the compressed view P_1, complementary view P_2, and parametric view P_3 from q and $\mathcal{D}$ (Eqs. 2–4);
Initialize the answer a as an empty sequence;
for $t \leftarrow 1$ **to** T **do**
  Compute the three views' next-token distributions and their entropies (Eq. 6);
  `// favor low-entropy views`
  Form p_{mix} by entropy-weighted averaging (Eqs. 7–8);
  if p_{mix} *is confident* (*peak* $\geq \gamma$, *Eq. 9*) **then**
    `// greedy commit`
    Emit the most likely token of p_{mix};
  else
    `// rerank candidates via short rollout`
    Form a candidate set from p_{mix} and each view's top token (Eq. 10);
    Score each candidate by a short d-step rollout under the same mixing rule (Eq. 11);
    Emit the best-scoring candidate;
  Append the emitted token to all three views;
  break if the emitted token is the end-of-sequence token;
return a;

3.6 Overall Pipeline of `OCEAN`

Algorithm 1 summarizes the decoding pipeline of `OCEAN`. We first build the compressed, complementary, and parametric views, each conditioning the reader on different evidence, and then decode the answer token by token. At each step, the three views yield next-token distributions whose entropy indicate their confidence, which we fuse into a single mixture by entropy-weighted averaging so that more confident views contribute more. If the mixture is confident, with its peak above the gate γ, we emit its top token. Otherwise we trigger a short look-ahead that scores a small candidate set with d-step rollouts under the same rule and emits the best one. The emitted token is appended to all three views, and decoding stops at the end-of-sequence token or after T steps. A step-by-step description for `OCEAN` is given by Algorithm 1.

4 Experiments

In this section, we conduct experiments to comprehensively evaluate the effectiveness, robustness, and efficiency of `OCEAN`. In particular, our empirical analysis is organized around the following five research questions:

- **RQ1:** How does `OCEAN` perform against representative compression-based RAG methods across different QA benchmarks and reader backbones?
- **RQ2:** How does each component of `OCEAN`, including each of the three evidence views, contribute to its overall performance?
- **RQ3:** How sensitive is `OCEAN` to key hyperparameters, including the rollout depth, candidate width, and routing temperature?
- **RQ4:** What trade-off does `OCEAN` achieve between answer quality and computational efficiency compared with compression-based and full-context methods?
- **RQ5:** How effectively can `OCEAN` correct errors caused by incomplete or misleading compressed evidence?

4.1 Experimental Setup

Benchmark Datasets. We evaluate OCEAN on four open-domain QA benchmarks: Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (TQA) (Joshi et al., 2017), WebQuestions (WebQ) (Berant et al., 2013), and PopQA (Mallen et al., 2023). Additionally, we retrieve top-50 passages from the 2018 Wikipedia corpus using an MS MARCO-trained dense retriever(Izacard et al., 2021) for each query.

Evaluation Metrics. Following prior work (Jung et al., 2025; Lee et al., 2025; Cheng et al., 2025; Nakka et al., 2026), we evaluate model performance using Exact Match (EM), token-level F1, and Accuracy (ACC). EM measures the proportion of predictions that exactly match their corresponding ground-truth answers, whereas token-level F1 measures token-level overlap between predicted and reference answers based on the harmonic mean of precision and recall, thereby assigning partial credit to partially correct predictions. Additionally, ACC measures the proportion of examples for which the model predicts the correct answer. For all three metrics, higher values indicate better performance.

Compared Baselines. For evaluation, we consider two categories of baselines. The first consists of compression-free methods, which serve as the upper and lower bounds: the raw-document setting and the closed-book setting, respectively. The second consists of compression-based methods, for which we evaluate Naive Summary, RECOMP (Xu et al., 2023), CompAct (Yoon et al., 2024), and FaviComp (Jung et al., 2025). Across all settings, we use Meta-Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct as the downstream readers, which define our two backbone families. The compressor varies by method. Baselines that provide a dedicated compression model use their released checkpoints, described below, whereas FaviComp and OCEAN pair each reader with a same-family 3B compressor, Llama-3.2-3B-Instruct for the Llama family and Qwen2.5-3B-Instruct for the Qwen family. The implementation details of the baseline methods are described as follows:

- RECOMP (Xu et al., 2023). We use the T5-large checkpoints released by the original authors. We adopt the per-dataset checkpoints for NQ and TQA. Since no dedicated checkpoint is released for WebQ or PopQA, we compress the retrieved evidence for these two datasets with the T5-large model fine-tuned on NQ.

- CompAct (Yoon et al., 2024). We use CompAct-7B, a Mistral-7B-Instruct model instruction-tuned by the original authors for evidence compression. We set the number of documents per segment to five and perform a single compression iteration.

- FaviComp (Jung et al., 2025). We follow the configuration in the original paper and use the same-family 3B compressor described above. The weighting coefficient α is set to 0.5.

Implementation Details of OCEAN. We instantiate OCEAN on two backbone families. For the Llama family we use Meta-Llama-3.1-8B-Instruct (Grattafiori et al., 2024) as the reader and Llama-3.2-3B-Instruct as the compressor, and for the Qwen family we use Qwen2.5-7B-Instruct (Qwen et al., 2025) and Qwen2.5-3B-Instruct, respectively. For every query we retrieve passages from the MS MARCO corpus with a TF-IDF retriever. The compressed view is a question-focused summary that the 3B compressor generates from the top-50 retrieved passages. The complementary view concatenates the top-3 retrieved passages as a small high-fidelity subset of the evidence. The parametric view queries the reader with no retrieval. All three views use the same reader and the same five-shot prompt shown in Table 5, and differ only in the context they condition on. At each step we fuse the three next-token distributions by entropy-based weighting with temperature $\tau = 1.0$ (Eq. 7). Following Eq. 9, the gated look-ahead is skipped when the fused distribution is already confident and is triggered otherwise, using a gate of $\gamma = 0.7$. When triggered, it rescores the top-3 candidate tokens via rollouts of depth 2 under the same fusion policy. The sensitivity of OCEAN to these hyperparameters is reported in Figure 5. Both compression and inference are run on a single NVIDIA GeForce RTX 4090 GPU.

Table 1: Main results on open-domain QA benchmarks for two reader backbones. **Bold**: best among compressed methods (within each backbone). Raw document serves as the uncompressed upper bound and is not directly comparable. †Uses uncompressed retrieved context; provided as a reference upper bound.

Method	Model	NQ			WebQ			TQA			PopQA			AVG		
		EM	F1	ACC	EM	F1	ACC	EM	F1	ACC	EM	F1	ACC	EM	F1	ACC
Raw document†	Qwen	37.1	49.7	45.2	23.2	39.6	34.9	63.0	71.7	69.1	41.6	47.3	45.8	41.2	52.1	48.8
	Llama	38.4	50.9	49.2	23.3	40.6	41.6	66.3	75.3	73.4	42.7	49.2	49.8	42.7	54.0	53.5
Zero-shot	Qwen	18.0	26.3	21.8	22.1	36.2	29.4	46.4	52.0	48.2	17.1	20.3	17.5	25.9	33.7	29.2
	Llama	22.4	33.3	29.9	20.8	38.2	33.5	54.9	62.8	59.7	21.9	26.6	24.6	30.0	40.2	36.9
Naive summary	Qwen	31.8	42.5	39.1	24.1	39.6	36.4	59.3	66.7	63.6	36.6	41.6	40.2	38.0	47.6	44.8
	Llama	30.6	44.7	43.3	21.4	39.1	37.4	57.1	67.5	66.0	33.7	41.9	42.4	35.7	48.3	47.3
RECOMP-Abstractive	Qwen	21.4	35.3	28.7	21.4	35.3	28.7	51.1	57.7	54.3	16.5	20.1	17.3	27.6	37.1	32.3
	Llama	25.4	35.1	30.2	21.6	38.1	31.0	57.4	64.6	60.5	22.8	27.1	24.3	34.8	45.9	40.6
CompAct	Qwen	21.1	30.0	**41.1**	11.4	21.9	37.3	55.5	63.7	63.8	29.5	35.4	42.6	29.4	37.8	46.2
	Llama	31.9	42.4	45.1	19.7	35.0	40.1	62.9	71.7	67.5	35.7	42.7	**45.6**	37.6	48.0	49.6
FaviComp	Qwen	29.8	43.5	39.9	23.4	41.1	**39.3**	60.0	69.0	66.9	37.8	**44.8**	**44.0**	37.8	49.6	**47.5**
	Llama	27.3	43.4	41.8	21.3	39.9	38.1	55.6	68.7	68.3	34.5	43.0	42.3	34.7	48.8	47.6
OCEAN	Qwen	**33.2**	**44.1**	39.4	**27.3**	**42.7**	39.2	**64.4**	**71.5**	**68.0**	**39.3**	43.6	41.4	**41.1**	**50.5**	47.0
	Llama	**33.3**	**47.7**	**45.7**	**23.9**	**42.4**	**40.7**	**63.8**	**73.3**	**71.4**	**38.3**	**45.6**	45.5	**39.8**	**52.3**	**50.8**

4.2 Main Results (RQ1)

Table 1 reports results on four open-domain QA benchmarks (NQ, WebQ, TQA, and PopQA) under exact match (EM), F1, and accuracy (ACC), on the Llama-3.1-8B and Qwen2.5-7B backbones. We compare OCEAN against trained compressors (RECOMP, CompAct) and training-free ones (Naive summary, FaviComp), with the uncompressed *Raw document* setting as the upper bound.

Across both backbones, OCEAN stands out along three dimensions. ❶ It is the most effective compression-based method, ranking first on nearly every metric and nearly closing the gap to the uncompressed upper bound, matching it on WebQ. ❷ It generalizes across readers, staying the top method on both backbones while CompAct and RECOMP, competitive on Llama, degrade sharply on Qwen with some falling near the closed-book floor. This shows that recovering evidence at decoding time transfers across model families rather than overfitting to a single reader. ❸ It is *efficient*, attaining this with only a lightweight compressor instead of the elaborate compression procedures that add latency in competing methods, a trade-off we examine in Table 3.

4.3 Ablation Study (RQ2)

To assess the contribution of each view, we remove one component at a time: *w/o summary* removes the compressed view, *w/o cosine retrieval* removes the complementary view, and *w/o zero-shot* removes the parametric view. All ablations use Llama-3.1-8B-Instruct as the reader and are evaluated on NQ, WebQ, TQA, and PopQA. As Table 2 shows, removing any view hurts performance on all four, but which view matters most varies across them.

The compressed view conditions on the broadest evidence, a summary of the top-50 passages. Although summarization is lossy, the summary preserves coverage of the retrieved passages, so removing it produces the largest drops on NQ and PopQA. This indicates the question-focused summary carries most of the evidence on these datasets. The complementary view retains a few high-fidelity passages, supplying precise facts that the lossy summary alone misses. Removing it produces the largest drop on TQA, indicating that TQA relies most on exact passage-level evidence. The parametric view relies on the reader's knowledge with no retrieval. Its effect is largest on WebQ and TQA, where this internal knowledge supplements the retrieved evidence, and negligible on NQ and PopQA. Crucially, every single-view removal lowers both EM and F1 across all four datasets, confirming that the three views carry complementary information.

Table 2: Ablation study of OCEAN's three view components on NQ, WebQ, TQA, and PopQA. **M1**: naive summary view (abstractive compression). **M2**: cosine-retrieval view (similarity-based sentence selection). **M3**: zero-shot view (direct generation without context compression). ✓ indicates the component is included, × indicates it is removed.

Method	Components			NQ			WebQ			TQA			PopQA		
	M1	M2	M3	EM	F1	ACC	EM	F1	ACC	EM	F1	ACC	EM	F1	ACC
OCEAN w/o zero-shot	✓	✓	×	32.2	46.9	**45.7**	22.6	40.3	40.0	61.1	71.2	69.7	36.5	44.7	45.2
OCEAN w/o cosine retrieval	✓	×	✓	31.9	45.8	43.9	23.3	41.9	40.2	60.8	70.6	69.0	36.1	43.3	43.2
OCEAN w/o summary	×	✓	✓	29.8	42.3	39.8	23.4	40.9	38.3	63.2	71.9	69.7	34.1	40.3	39.5
OCEAN	✓	✓	✓	**33.3**	**47.7**	**45.7**	**23.9**	**42.4**	**40.7**	**63.8**	**73.3**	**71.4**	**38.3**	**45.6**	**45.5**

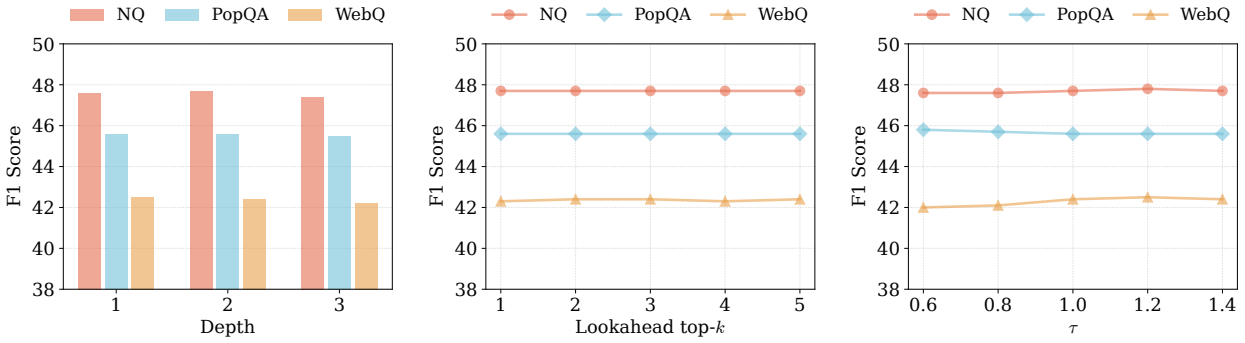

Figure 5: Sensitivity of OCEAN under F1 on NQ, PopQA, and WebQ. **Left:** rollout depth d. **Middle:** look-ahead candidate width K. **Right:** temperature τ.

4.4 Sensitivity Analysis (RQ3)

Influence of the Look-ahead Mechanism. We study OCEAN's sensitivity to the look-ahead mechanism's two hyperparameters, the rollout depth d and the candidate width K which indicates the number of top tokens to be rolled out. As shown in Figure 5, F1 stays stable across all three datasets. Increasing d from 1 to 3 (left) yields nearly identical scores, showing that once look-ahead triggers, how far it rolls out barely affects the answer. Varying K from 1 to 5 (middle) likewise leaves F1 essentially unchanged. Because the candidate set unions the mixture's top-K tokens with each view's top-1, the views' top-1 predictions already form a stable core, so larger K rarely adds distinct candidates. Overall, look-ahead is robust to both hyperparameters, requiring no careful per-dataset tuning. As Figure 6 shows, the trends are consistent, and OCEAN stays within a narrow band on NQ, PopQA, and WebQ as we vary the rollout depth d, the candidate width K, and the temperature τ, confirming its robustness to these hyperparameters.

Influence of the temperature hyperparameter τ. The temperature parameter τ in Eq. 7 controls how aggressively the routing favors confident views. Small τ pushes the mixture toward the lowest-entropy view, large τ toward a uniform average over the three views. As the rightmost panel of Figure 5 shows, OCEAN is largely insensitive across $\tau \in [0.6, 1.4]$, with F1 on all three datasets staying within roughly 0.5 points and showing no sharp optimum. The routing thus needs no careful temperature tuning, and we set $\tau = 1.0$ by default.

4.5 Efficiency Analysis (RQ4)

To evaluate the efficiency trade-off between simple and elaborate compression, we compare OCEAN against FaviComp under an identical setup: both use the same Compression Model (3B) and Reader Model (8B), so that any latency difference reflects the compression strategy rather than model size. Table 3 reports a per-

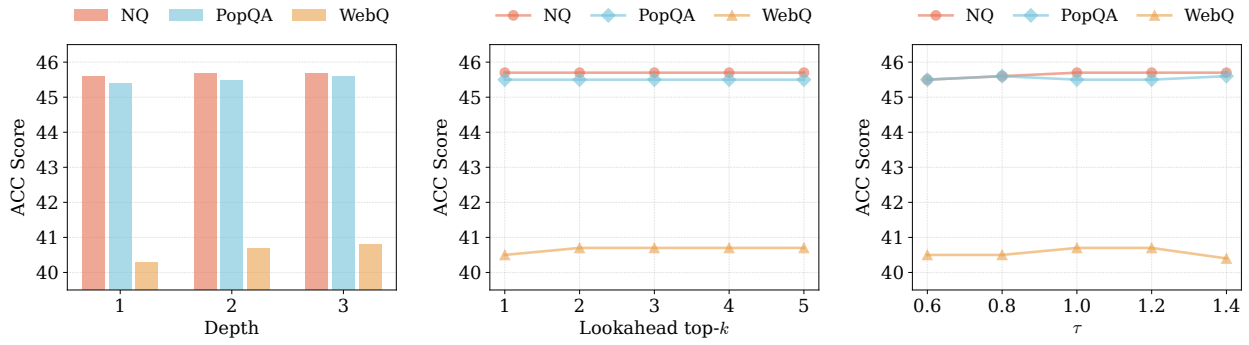

Figure 6: Sensitivity of OCEAN under ACC on NQ, PopQA, and WebQ. **Left:** rollout depth d. **Middle:** look-ahead candidate width K. **Right:** temperature τ.

Table 3: Per-sample compression latency on NQ, WebQ, and TQA (batch size 1, in ms). The Compressor and Reader columns report the latency of each model. Rel. is relative to OCEAN. OCEAN compresses with the Compression Model alone, whereas FaviComp also invokes the Reader Model, roughly doubling the total latency on every dataset.

Dataset	Method	Compressor	Reader	Total	Rel.
NQ	FaviComp	3591.0	3886.2	7477.2	2.11×
	OCEAN	3540.9	–	3540.9	1.00×
WebQ	FaviComp	3474.8	3706.2	7181.0	2.09×
	OCEAN	3440.8	–	3440.8	1.00×
TQA	FaviComp	3472.0	3685.6	7157.6	2.07×
	OCEAN	3459.8	–	3459.8	1.00×

sample latency breakdown on NQ at batch size 1, separating prefill from decode for the Compression Model and the Reader Model. Table 3 compares the per-sample compression latency of OCEAN and FaviComp on NQ, both using the same Compression Model (3B) and Reader Model (8B). OCEAN performs a single summarization pass with the Compression Model alone, whereas FaviComp also invokes the Reader Model during compression. As latency is dominated by decoding, with each model's decode far exceeding its prefill at batch 1, this additional reader pass roughly doubles FaviComp's compression latency (2.08×), almost entirely through the Reader Model's decode. OCEAN thus obtains its compressed context at roughly half the cost of FaviComp's procedure.

4.6 Case Study (RQ5)

Token-level Routing. Table 4 traces decoding for *"who was the ruler of England in 1616"* (gold: **James I**). On its own, the compressed view P_1 is confidently wrong, opening with *King* and later favoring the Scottish regnal form *VI*, yielding *King James*. The complementary and parametric views P_2 and P_3 instead favor *James* then *I*. Although P_1 is the most peaked view at t=0, *James* tops *both* P_2 and P_3, so under entropy weighting ($w_{1:3}$ = .39/.19/.42) their combined mass on *James* outweighs P_1's isolated vote for *King*. The same agreement at t=1 picks *I* over *VI*, and OCEAN recovers *James I*. A confident but isolated view cannot override two corroborating ones, so routing defers to agreement rather than raw peak confidence.

Overriding a Misleading Summary. Figure 7 shows a complementary failure mode on *"Where is the capital city of Alabama located?"* (gold: *Montgomery*), comparing OCEAN with FaviComp. The compressed view is misled by a salient but off-target phrase, *"located on the Alabama River"*, and returns a geographic descriptor rather than the city name. The complementary view, conditioned on the raw retrieved passages,

Table 4: Token-level case study for *"who was the ruler of England in 1616"* (gold: **James I**). $P_1/P_2/P_3$ are the compressed, evidence, and parametric views; cells show each view's top-1 next token and its probability. P_1 is confidently wrong at both decision points (dKing, dVI) and alone yields *King James*; entropy weighting lets P_3 (and P_2) dominate, so `OCEAN` recovers *James I*.

View	t=0	t=1	t=2	Answer
P_1	dKing (.73)	dVI (.59)	⟨eos⟩	*King James* ✗
P_2	James (.51)	I (.91)	⟨eos⟩	*James I* ✓
P_3	James (.72)	I (.97)	⟨eos⟩	*James I* ✓
`OCEAN`	**James**	**I**	⟨eos⟩	***James I*** ✓
$w_{1:3}$	.39/.19/.42	.25/.30/.45	.32/.32/.36	

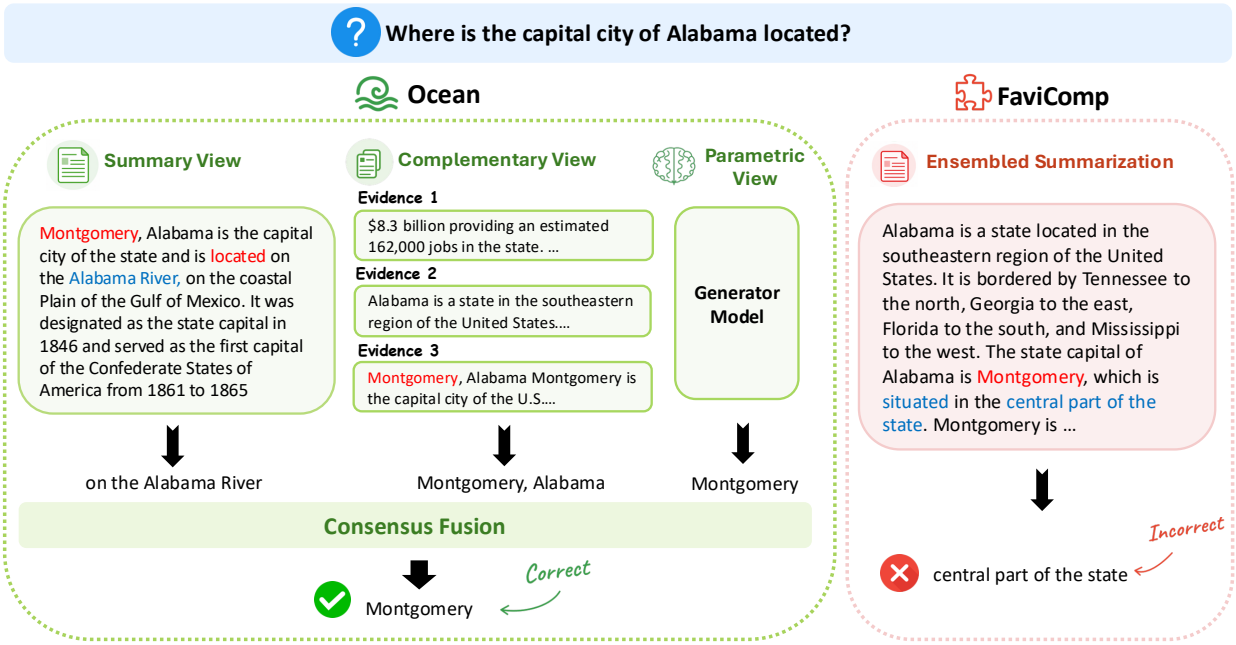

Figure 7: Case study on *"Where is the capital city of Alabama located?"* (gold: *Montgomery*). `OCEAN`'s compressed view is distracted into *"on the Alabama River"*, but its complementary and parametric views agree on *Montgomery*, which the fusion recovers; FaviComp, relying on a single summary, is misled into *"central part of the state"*.

and the parametric view, drawing on the reader's own knowledge, both agree on *Montgomery*With two of the three views in agreement, the fusion becomes confident in *Montgomery* and outputs the correct answer. By contrast, FaviComp feeds the reader a single ensembled summary with no complementary channel to correct it, is distracted in the same way, and answers *"central part of the state"*.

5 Conclusion

Here, we introduce `OCEAN`, a training-free decoding framework that recovers the information discarded by compression in RAG. `OCEAN` ensembles three complementary views, the compressed, complementary, and parametric, and fuses their next-token distributions by predictive uncertainty so that more confident views contribute more. It then resolves ambiguous steps through a gated look-ahead. As a result, `OCEAN` approaches the answer quality of full-context conditioning while retaining the efficiency of compressed RAG.

References

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023.

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

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1533–1544, 2013.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17754–17762, 2024.

Rong Cheng, Jinyi Liu, Yan Zheng, Fei Ni, Jiazhen Du, Hangyu Mao, Fuzheng Zhang, Bo Wang, and Jianye Hao. Dualrag: A dual-process approach to integrate reasoning and retrieval for multi-hop question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31877–31899, 2025.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. xrag: Extreme context compression for retrieval-augmented generation with one token. *Advances in Neural Information Processing Systems*, 37:109487–109516, 2024.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3829–3846, 2023.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. In *International Conference on Learning Representations*, volume 2024, pp. 54158–54183, 2024.

Giulio Corallo and Paolo Papotti. Parallel context-of-experts decoding for retrieval augmented generation. *arXiv preprint arXiv:2601.08670*, 2026.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. In *Findings of the association for computational linguistics: ACL 2023*, pp. 1049–1065, 2023.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2): 1–55, 2025.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

Yeonseok Jeong, Jinsu Kim, Dohyeon Lee, and Seung-won Hwang. Ecorag: Evidentiality-guided compression for long context rag. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 26607–26628, 2025.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 13358–13376, 2023.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics.

Dongwon Jung, Qin Liu, Tenghao Huang, Ben Zhou, and Muhao Chen. Familiarity-aware evidence compression for retrieval-augmented generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, Suzhou, China, November 2025.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st annual meeting of the Association for Computational Linguistics (volume 1: long papers)*, pp. 5591–5606, 2023.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *International conference on machine learning*, pp. 15696–15707. PMLR, 2023.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*, 2022.

Dahyun Lee, Yongrae Jo, Haeju Park, and Moontae Lee. Shifting from ranking to set selection for retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17606–17619, 2025.

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

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Pre-trained language models for text generation: A survey. *ACM Computing Surveys*, 56(9):1–39, 2024.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 12286–12312, 2023.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6691–6706, 2021.

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

Maxime Louis, Hervé Déjean, and Stéphane Clinchant. Pisco: Pretty simple compression for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15506–15521, 2025.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9802–9822, 2023.

Kiran Kumar Nakka, C Vamshi Krishna Reddy, Divyateja Pasupuleti, Prakhar Agarwal, Harpinder Jot Singh, Anshu Avinash, Nirav Pravinbhai Bhatt, et al. Browsenet: Graph-based associative memory for contextual information retrieval. In *The Fourteenth International Conference on Learning Representations*, 2026.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72, 2025.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, et al. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 963–981, 2024.

Guanghui Qin, Corby Rosset, Ethan Chau, Nikhil Rao, and Benjamin Van Durme. Dodo: Dynamic contextual compression for decoder-only lms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9961–9975, 2024.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report, January 2025. URL `http://arxiv.org/abs/2412.15115`. arXiv:2412.15115 [cs].

David Rau, Shuai Wang, Hervé Déjean, and Stéphane Clinchant. Context embeddings for efficient answer generation in rag. *arXiv preprint arXiv:2407.09252*, 2024.

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 5418–5426, 2020.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pp. 31210–31227. PMLR, 2023.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 783–791, 2024.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher Manning, and Chelsea Finn. Fine-tuning language models for factuality. In *International Conference on Learning Representations*, volume 2024, pp. 44851–44866, 2024.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari Morcos. D4: Improving llm pretraining via document de-duplication and diversification. *Advances in Neural Information Processing Systems*, 36: 53983–53995, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Sourav Verma. Contextual compression in retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2409.13385*, 2024.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Adacad: Adaptively decoding to balance conflicts between contextual and parametric knowledge. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 11636–11652, 2025.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. Learning to filter context for retrieval-augmented generation. *arXiv preprint arXiv:2311.08377*, 2023.

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

Jiannan Xiang, Tianhua Tao, Yi Gu, Tianmin Shu, Zirui Wang, Zichao Yang, and Zhiting Hu. Language models meet world models: Embodied experiences enhance language models. *Advances in neural information processing systems*, 36:75392–75412, 2023.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*, 2023.

Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. In *International Conference on Learning Representations*, volume 2025, pp. 33769–33789, 2025.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*, 2024.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 535–546, 2021.

Chanwoong Yoon, Taewhoo Lee, Hyeon Hwang, Minbyul Jeong, and Jaewoo Kang. Compact: Compressing retrieved documents actively for question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21424–21439, 2024.

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, et al. Kola: Carefully benchmarking world knowledge of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 44594–44637, 2024.

Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Lifeng Jin, Linfeng Song, Haitao Mi, and Helen Meng. Self-alignment for factuality: Mitigating hallucinations in llms via self-evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1946–1965, 2024.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, et al. A survey of large language models. *Frontiers of Computer Science*, 20(12):2012627, 2026.

A Appendix

A.1 Few-Shot Demonstrations

We use a fixed five-shot prompt for every dataset. The same demonstrations are reused across all prompting variants: when retrieved or compressed evidence is available it is supplied as the context, and for closed-book prompting (the parametric view) the context field is omitted. All answers are kept to short spans, consistent with our evaluation instruction. The demonstrations are listed in Table 5.

Question	**Answer**
Natural Questions (NQ)	
Who sings i've got to be me	Sammy Davis, Jr
Who wrote i will follow you into the dark	Ben Gibbard
Who won season 2 of total drama island	Owen (Scott McCord)
What part of the mammary gland produces milk	cuboidal cells
When did the golden compass book come out	1995
WebQuestions (WebQ)	
What character did natalie portman play in star wars?	Padmé Amidala
Who does joakim noah play for?	Chicago Bulls
Who did draco malfoy end up marrying?	Astoria Greengrass
What kind of money to take to bahamas?	Bahamian dollar
Where did saki live?	United Kingdom
TriviaQA (TQA)	
Who sang the theme for the James Bond film 'Thunderball'?	Tom Jones
A hendecagon has how many sides?	Eleven
In the 1968 feature film Chitty Chitty Bang Bang, of what country is Baron Bomburst the tyrant ruler?	Vulgaria
Artists Chuck Close, Henri-Edmond Cross, John Roy, Georges-Pierre Seurat, Paul Signac, Maximilien Luce and Vincent van Gogh painted in what style?	Pointillism
What is the study of the relation between the motion of a body and the forces acting on it?	Dynamics
PopQA	
What is Barack Obama's occupation?	politician
Who was the director of The Godfather?	Francis Ford Coppola
What genre is Bohemian Rhapsody?	rock
Who was the composer of Swan Lake?	Pyotr Ilyich Tchaikovsky
In what country is the Eiffel Tower?	France

Table 5: Five-shot demonstrations used for each dataset. The same set is shared across the compressed, complementary, and parametric views; for the parametric view the context is omitted.