Retrieval Heads are Dynamic

Yuping Lin[1]*, Zitao Li[2]†, Yue Xing[1], Pengfei He[1], Yingqian Cui[1], Yaliang Li[3], Bolin Ding[3], Jingren Zhou[3], Jiliang Tang[1],

[1]Michigan State University, [2]Zoom Communications, [3]Tongyi Lab, Alibaba Group,

{linyupin, xingyue1, hepengf1, cuiyingq, tangjili}@msu.edu,
zitao.li@zoom.us,
{yaliang.li, bolin.ding, jingren.zhou}@alibaba-inc.com

Abstract

Recent studies have identified "retrieval heads" in Large Language Models (LLMs) responsible for extracting information from input contexts. However, prior works largely rely on static statistics aggregated across datasets, identifying heads that perform retrieval on average. This perspective overlooks the fine-grained temporal dynamics of autoregressive generation. In this paper, we investigate retrieval heads from a dynamic perspective. Through extensive analysis, we establish three core claims: (1) Dynamism: Retrieval heads vary dynamically across timesteps; (2) Irreplaceability: Dynamic retrieval heads are specific at each timestep and cannot be effectively replaced by static retrieval heads; and (3) Correlation: The model's hidden state encodes a predictive signal for future retrieval head patterns, indicating an internal planning mechanism. We validate these findings on the Needle-in-a-Haystack task and a multi-hop QA task, and quantify the differences on the utility of dynamic and static retrieval heads in a Dynamic Retrieval-Augmented Generation framework. Our study provides new insights into the internal mechanisms of LLMs.

1 Introduction

Recently, there is a growing interest in Large Language Models (LLMs) (Radford et al., 2019; Brown et al., 2020; Vaswani et al., 2017; Chowdhery et al., 2023; Hoffmann et al., 2022; Touvron et al., 2023) to understand how they process context, particularly focusing on their ability to extract key information from the input. Prior work has shown that although LLMs demonstrate strong in-context learning abilities (Garg et al., 2022; Xie et al., 2021), their ability to utilize long contexts often needs improvement (Liu et al., 2024; Press et al., 2023). A line of mechanistic interpretability

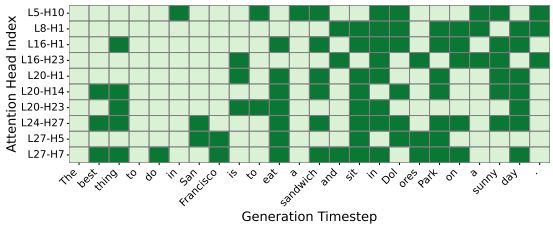


Figure 1: **Dynamism of Retrieval Heads.** The retrieval scores of individual attention heads fluctuate across the generation process. **Dark color** indicates heads having a retrieval score of 1, as defined by Equation (1). The x-axis denotes the generation step, labeled by the token generated at that step. The y-axis shows the 10 most variated retrieval heads, selected based on their retrieval score variance over the entire generation process. Lx-Hy denotes the y-th head (starting from 0) on the layer x (starting from 0).

works suggests that attention heads exhibit functional specialization (Voita et al., 2019; Michel et al., 2019; Elhage et al., 2021): For example, a pioneer work (Wu et al., 2024) analyzes the model from an attention-head perspective, identifying a specific set of heads termed "retrieval heads" that are responsible for the copy-paste behavior of LLMs from the given inputs. Recent studies (Zhang et al., 2025; Fu et al., 2024) provide further evidence of the existence of retrieval heads, even in tasks requiring complex reasoning.

While these works provide valuable insights on the mechanism of LLMs, they have been mainly constrained to a **fixed** subset of attention heads. For example, Wu et al. (2024) aggregated attention patterns across datasets to find the heads which frequently perform copy-paste operations, identifying a fixed set of heads for each model that perform retrieval on average. However, treating retrieval heads as a fixed set assumes that this average behavior is a heuristic approximation for the model's real-time operation. Given the autoregressive nature of LLMs, it is natural to question whether the retrieval heads should instead be a **dynamic** set conditioned

*Work done during internship at Tongyi Lab, Alibaba Group.

†Work done during at Tongyi Lab, Alibaba Group.

on the given context. Relying on static definitions risks oversimplifying the model's real-time behavior, as heads that are statistically dominant may not be active at every critical timestep, while "less significant" heads might play irreplaceable roles in specific contexts. As illustrated in Figure 1, the set of heads acting as retrieval heads actually fluctuates significantly across token generation steps. This observation challenges the completeness of static definitions, raising the following fundamental questions: *1) How does the set of retrieval heads evolve during generation? 2) Are these dynamic heads functionally interchangeable with static ones? 3) Is this dynamism predictable given a model and a context?*

To answer these questions, we present the first systematic study of retrieval heads from a **dynamic** perspective.

Our key contributions are as follows:

1. We demonstrate that the retrieval heads are highly **dynamic** that statistical methods fail to capture. (Claim 1 in Section 3.2)

2. Given the dynamic nature of the retrieval head pattern, we show that the specific retrieval heads at a given generation step are **not replaceable**, and ablating them causes severe performance degradation. (Claim 2 in Section 3.3)

3. We reveal that the model's final hidden state exhibits a strong **correlation** with future retrieval head patterns, revealing a **predictive mechanism** within LLMs. (Claim 3 in Section 3.4)

4. While the above claims are based on a Needle-in-a-Haystack task in Section 3, we validate them in a question-answering task where reasoning efforts are needed. (Section 4)

5. We use the dynamic and static retrieval heads in a Dynamic RAG scenario to compare their practical utility, demonstrating that dynamically selecting heads based on the current generative state significantly improves retrieval accuracy and downstream performance compared to static retrieval heads. (Section 5)

We hope these findings will serve as a foundation for future research in model interpretability and the development of more precise, state-aware intervention techniques.

2 Related Works

Mechanistic Interpretability Understanding the internal mechanisms of Transformer-based models (Vaswani et al., 2017) has been a focal point of recent research. Mechanistic interpretability has emerged as a principled approach to understanding neural networks beyond input-output behavior (Olah et al., 2020; Elhage et al., 2021). Early work by Olsson et al. (2022) identified Induction Heads, a specialized circuit responsible for in-context learning by copying previous tokens that follow similar patterns. Subsequent work further investigated induction-like circuits (Elhage et al., 2022; Wang et al., 2022). This laid the theoretical groundwork for understanding how attention heads perform "copy-paste" operations. In the context of long-sequence modeling, Xiao et al. (2023) discovered Attention Sinks, revealing that models dedicate massive attention to initial tokens (e.g., BOS) to maintain numerical stability. Related analyses have also examined positional and numerical artifacts in attention mechanisms (Press et al., 2021; Su et al., 2024a). Furthermore, studies on the "Lost in the Middle" phenomenon (Liu et al., 2024) have highlighted the non-uniform capability of models to access information across long contexts. These studies primarily focus on static circuit structures or attention biases, and do not fully explain how the model dynamically modulates its attention allocation step-by-step to perform precise retrieval during the autoregressive generation.

Retrieval Heads Information retrieval within LLMs has been studied both implicitly through attention mechanisms and explicitly through retrieval-augmented generation (RAG) frameworks (Lewis et al., 2020; Borgeaud et al., 2022). Building on the concept of functional specialization, recent studies have isolated specific attention heads responsible for information retrieval. Wu et al. (2024) pioneered this direction by identifying Retrieval Heads via the Needle-in-a-Haystack (NIAH) test, characterizing them as a sparse, intrinsic subset of heads that perform copy-paste operations from long contexts. Addressing the limitations of synthetic benchmarks, Zhang et al. (2025) proposed QRHead, which refines head detection using query-aware attention scores on realistic tasks to improve downstream retrieval and re-ranking performance. Similarly, Fu et al. (2024) introduced HeadKV, a method that leverages retrieval and reasoning importance scores to perform head-

level KV cache compression, significantly outperforming layer-level methods like SnapKV (Li et al., 2024), H2O (Zhang et al., 2023), and PyramidKV (Cai et al., 2024). A common limitation across these works is their reliance on a *static perspective*. However, this approach overlooks the *temporal dynamism* of the generation process.

3 Analysis of Dynamic Retrieval Heads

3.1 Setup

This section focuses on the traditional **copy-paste retrieval head** as in Wu et al. (2024). This type of retrieval head considers the exact copy-paste of the input tokens to the next generated token.

To better trace and analyze the exact copy-paste behavior, following Wu et al. (2024), we consider the Needle-in-a-Haystack (NIAH) task (Kamradt, 2023), which evaluates a model's ability to precisely retrieve the specific piece of information (the "needle") embedded at a random location within a long, distracting document (the "haystack").

Definition of Retrieval Head Following Wu et al. (2024), to define the copy-paste retrieval head, an attention head is considered to be performing a retrieval operation if and only if two conditions are satisfied: (1) at the current inference step, the generated token is identical to the token receiving the highest attention weight from that head, and (2) the token with the highest attention weight lies within the "needle" context, i.e., it is a *needle token*. When these conditions are satisfied, its retrieval score is set to be 1.[1]

Formally, let $i^* = \arg\max_i(\mathbf{a}_i^{h,t})$ be the index of the token that receives the maximum attention from head h at timestep t, where $\mathbf{a}^{h,t}$ is the vector of attention scores from the final token of the input x^t (as query) to all tokens in x^t (as keys) for head h, i.e., $\mathbf{a}^{h,t} = \text{AttnScore}^h[t, :]$. The retrieval score of head h on input x^t is then defined as:

$$S_{\text{copy-paste}}(x^t, h) = \mathbf{1}\left[i^* \in I_{\text{needle}} \wedge x_{i^*}^t = \hat{y}\right] \quad (1)$$

where I_{needle} is the set of indices for tokens within the needle, and $\hat{y}$ is the token predicted to be generated at the current timestep.

[1]The original work (Wu et al., 2024) normalizes this score by the length of the needle. We omit this normalization, assigning a binary score of 1 and 0, because our analysis is conducted at the token level, in contrast to the sample-level analysis of the original work.

Overview of Claims Given the above task description and definition of retrieval heads, we present our central claims:

- **Claim 1: Dynamism.** The patterns of retrieval heads are dynamic throughout the autoregressive generation process.
- **Claim 2: Irreplaceability.** The retrieval functionality of the specific retrieval heads at a given timestep cannot be replaced by other heads. If these heads are disabled, the model will suffer from performance degradation.
- **Claim 3: Correlation.** A strong correlation exists between the model's hidden state and the patterns of retrieval heads in the future.

3.2 Retrieval Heads are Dynamic

Different from existing literature (Wu et al., 2024; Zhang et al., 2025; Fu et al., 2024) where a large corpus of samples are collected to identify a set of statistically significant retrieval heads (i.e., **static retrieval heads**), we argue the existence of unique patterns of retrieval heads that emerge at individual timesteps during the generation process (i.e., **dynamic retrieval heads**). Specifically, the retrieval score of attention heads fluctuates across timesteps. Therefore, we hypothesize that the dynamic retrieval heads at a particular timestep do not always align with the static retrieval heads.

To verify our hypothesis, Figure 1 in Section 1 plots the retrieval scores calculated as per Equation (1) for some attention heads over the course of an autoregressive generation process for a given sample. The plot clearly demonstrates that the retrieval scores for individual heads fluctuate significantly across timesteps, confirming the dynamic nature of retrieval heads.

Furthermore, to rigorously quantify this dynamism, we conducted a statistical analysis. The results for all models are summarized in Table 1.

There are several observations from the table: **First,** the standard deviation in the number of active heads (**Mean** $\pm$ **Std**) indicates that the quantity of active retrieval heads varies across timesteps. **Second,** the **Unique / Total** active heads metric demonstrates a "long-tail" distribution. For instance, llama3.1-8b activates 238 distinct heads for retrieval over time, which exceeds the scope of the conventional static subset (e.g., top-20). **Third,** the Jaccard similarity (**"Jaccard w/ Static"**, ranging from 0.1845 to 0.4611) indicates that only a fraction of static heads are identified as retrieval heads

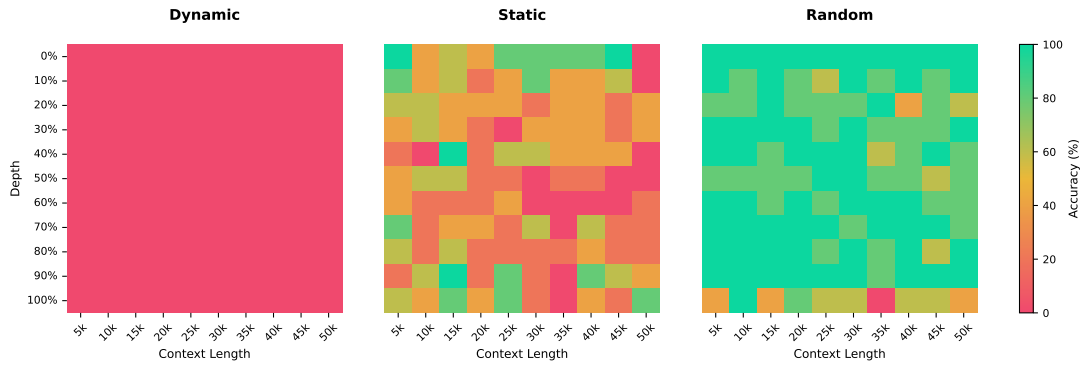


Figure 2: **Impact of Head Ablation on Retrieval Performance.** Comparison of NIAH test scores after masking three different sets of attention heads: dynamic retrieval heads, top-ranked static retrieval heads, and randomly selected heads on llama3.1-8b. The x-axis shows different haystack lengths. The y-axis shows the different locations ("depth") where the needle is inserted. The evaluation metric is Accuracy (exact string match). The average number of masked heads is kept consistent across all conditions. Masking dynamic heads (identified at each timestep via Eq. (1)) results in the most significant performance degradation, indicating their critical role in retrieval.

Model	Mean ± Std	Unique / Total	Jaccard w/ Static	Adj. Jaccard	Entropy
llama3.1-8b	12.97 ± 7.69	238 / 1024	0.3512	0.2793	3.8154
llama3.2-3b	9.69 ± 6.18	149 / 672	0.3126	0.3188	3.0083
qwen3-8b	20.18 ± 10.43	415 / 1152	0.4611	0.3668	4.1038
llama2-13b	6.20 ± 5.49	172 / 1600	0.2077	0.4979	4.8973
phi4-mini	6.13 ± 7.09	176 / 768	0.1845	0.5056	3.5532

Table 1: **Quantitative Statistics of Retrieval Head Dynamism. Mean ± Std**: Average number of dynamic retrieval heads per step and its standard deviation. **Unique / Total**: Total number of unique heads activated at least once during generation versus the total number of attention heads. **Jaccard w/ Static**: Similarity between dynamic retrieval heads and the top-20 static retrieval heads; lower values indicate fewer static heads are in the set of dynamic heads. **Adj. Jaccard**: Similarity of dynamic retrieval heads between consecutive steps. **Entropy**: Measure of distribution spread; higher values indicate broader head involvement in dynamic retrieval.

in a given generation step. **Fourth,** the **Adjacent Jaccard** similarity scores (ranging from 0.2793 to 0.5056) reveal a turnover rate that suggests the model frequently changes its active retrieval heads between consecutive tokens. **Finally,** the **entropy** corroborates this broad involvement. As a baseline, a uniform distribution over 20 heads yields an entropy of $\ln 20 \approx 2.99$. The observed entropy values exceed 3.0 (reaching 4.89). Together with the unique head counts, this indicates that dynamic retrieval is distributed across a wider set of heads rather than being confined to a small, fixed static subset. (See Appendix A.3 for detailed formulations).

Furthermore, the observed low similarity is robust to the choice of the static truncation threshold k. We conducted a sensitivity analysis by extending the static baseline to $k = 50$ and $k = 100$. As shown in Appendix B, the Jaccard similarity further decreases as k increases (e.g., dropping to 0.1236 at $k = 100$ for llama3.1-8b). This trend confirms that dynamic retrieval heads are not merely a rotation within a slightly larger fixed pool of heads, but rather represent a distinct and sparse distribution that remains largely uncovered even by broader static selections.

3.3 Dynamic Retrieval Heads are Irreplaceable

We further claim that the dynamic retrieval heads at a specific timestep **can not be replaced by the static retrieval heads**.

3.3.1 What will Happen without Dynamic Retrieval Heads?

To verify Claim 2, we conducted a head ablation study. The experiment procedure is as follows:

1. For each token generation step, we first execute a standard forward pass without any interventions. Discard the generated token.
2. We locate the retrieval heads at this step, as defined by Equation (1), and label them as the set of dynamic retrieval heads.
3. We mask all the dynamic retrieval heads of this timestep, and execute a second forward pass to re-generate the token for this timestep.

For comparison, we considered two baseline approaches. We masked heads drawn from either (a) the top-ranked static retrieval heads or (b) a randomly selected set of heads. For fair comparison, we mask the same number of static retrieval heads/random heads as the average number of masked dynamic retrieval heads. (Details in Appendix A.1)

This experimental design allows us to directly test our hypothesis: if dynamic retrieval heads indeed carry the primary functionality of retrieval at a given timestep, then masking them should cause a significantly greater performance degradation than masking any other set of heads.

Figure 2 presents the NIAH test results on *meta-llama/Llama-3.1-8B-Instruct* (llama3.1-8b). The results clearly show that masking the dynamic retrieval heads leads to the most severe degradation in retrieval performance, as the colors are almost red. This far exceeds the impact of masking an equal number of static or random heads, in which only some of the pieces in the heatmap are red. For ROUGE-L results and other models, see Appendix D.1. This verifies our hypothesis that the dynamic retrieval heads are not replaceable.

3.3.2 To What Extent do Static Retrieval Heads Help?

To further investigate the irreplaceability of dynamic retrieval heads, we also designed a progressive ablation study to analyze the model's compensatory mechanisms.

Our primary objective is to quantify the extent to which the model compensates for the loss of these optimal heads by activating additional strong static retrieval heads. To measure this, we first identify the set of *compensated heads* at each timestep, i.e., heads that become retrieval heads only after k dynamic heads are ablated. We then count how many of these compensated heads are in the top-20 static retrieval heads. Detailed experiment descriptions can be found in Appendix A.2.

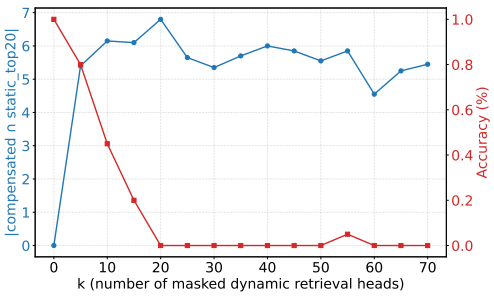


Figure 3: **Irreplaceability of Dynamic Retrieval Heads.** The plots show the degradation in NIAH performance as an increasing number (k) of dynamic retrieval heads are masked on llama3.1-8b. Even though the model compensates by activating top-20 static retrieval heads (blue line, left y-axis), the overall retrieval performance, measured by Accuracy (red line, right y-axis), continues to decline sharply. This demonstrates that static retrieval heads cannot effectively substitute for context-specific dynamic heads.

Figure 3 illustrates the result on llama3.1-8b, plotting the retrieval performance (red line, right y-axis) against the number of compensated heads that overlap with the top-20 static retrieval heads (blue line, left y-axis). The result reveals two critical observations. First, as the number of masked dynamic heads (k) increases, the model attempts to compensate by activating new heads as retrieval heads. A significant proportion of these compensated heads are indeed static retrieval heads, specifically, the overlap with the top-20 static set rises sharply to range between 4.5 and 7 for $k \geq 5$. Second, however, this compensation is insufficient. As shown by the red line, the overall retrieval performance degrades significantly. For instance, the Accuracy drops sharply from 1.0 to 0.0 as k reaches 20, suggesting that the function of dynamic retrieval heads is irreplaceable, and cannot be fully compensated for by static retrieval heads. For ROUGE-L metric and other models, see Appendix D.2.

3.4 Retrieval Scores are Correlated with Hidden States

Our final claim is that a **strong correlation** exists between the model's hidden state and its future retrieval activations, indicating that the model employs a **predictive mechanism** for its functional behavior, specifically retrieval in this case.

To validate this, we employed Canonical Correlation Analysis (CCA) with a **temporal offset**, denoted as k. Specifically, we measured the linear correlation between the **final hidden state** (the output embedding of the last input token at the last layer) at timestep n and the **retrieval scores** of all attention heads at a future timestep $n + k$. Detailed experiment settings can be found in Appendix A.4.

As shown in Figure 4, the canonical correlation decreases as the temporal offset k increases. At $k = 0$, the first canonical correlation is an exceptionally high 0.966, confirming a strong synchronous relationship between the hidden states and the retrieval scores. Besides, this correlation remains extremely high for future steps, at 0.931 for $k = 1$ and 0.915 for $k = 2$, indicating that the model's state acts as a predictive mechanism for retrieval operations several steps before they are executed.

While CCA demonstrates a strong linear relationship in this model, we additionally trained an MLP probe to further capture the non-linear relationships. We focused this analysis on the $k = 0$ case, i.e., the retrieval scores at the same timestep

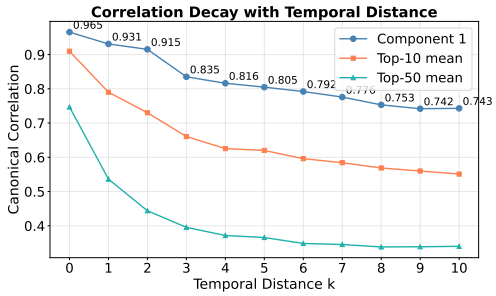


Figure 4: **Predictive Correlation Between Hidden States and Future Retrieval Scores.** Canonical Correlation Analysis (CCA) coefficients between the final hidden state at timestep n and the retrieval scores at a future timestep $n + k$. The plot shows the decay of the leading (Top-1) canonical correlation, as well as the average of the Top-10 and Top-50 correlations, as the temporal offset k increases. The high correlation at $k > 0$ demonstrates the predictive encoding of future retrieval activations.

as the hidden states. The probe's task was to learn the mapping from the hidden state to the specific retrieval score pattern of all heads. Detailed experiment settings can be found in Appendix A.5.

As shown in Table 2, the probes achieve strong performance across all models, with F1-scores ranging from 0.80 to 0.86 and AU-PRC values all exceeding 0.88, confirming that the dynamic retrieval head patterns are predictable.

LLM	F1	Precision	Recall	AUPRC
llama3.1-8b	0.8349	0.8344	0.8353	0.9173
llama3.2-3b	0.8456	0.8564	0.8351	0.9289
qwen3-8b	0.8566	0.8780	0.8362	0.9339
llama2-13b	0.8336	0.8455	0.8220	0.9183
phi4-mini	0.8038	0.8219	0.7865	0.8862

Table 2: **Performance of MLP Probes in Decoding Retrieval Scores.** The probes were trained to predict retrieval head scores from the final hidden state for various LLMs. The reported metrics (Precision, Recall, F1-Score) are calculated at the optimal decision threshold, which was determined by maximizing the F1-score on the validation set's Precision-Recall curve.

4 Generalizing Dynamic Retrieval Heads in a Question Answering Task

In Section 3, we systematically investigated three core claims of retrieval heads within the controlled experimental setting of the NIAH task. However, NIAH represents a simplified "copy-paste" scenario where the retrieved token is directly generated. To consider reasoning tasks, we need a broader definition of "retrieval" that based on attention allocation rather than token copying.

4.1 A Reasoning-Oriented Definition of Retrieval Score

In complex reasoning tasks such as Question Answering, the model's behavior goes beyond simple "copy-paste" operations. The model must integrate multiple supporting facts to derive an answer. Therefore, we propose a more generalized retrieval score, and correspondingly name the retrieval heads as **reasoning retrieval heads** whose retrieval score exceeds a pre-defined threshold. Based on the works of Fu et al. (2024); Zhang et al. (2025) with slight adaptation, we define the **reasoning retrieval score** $S_{\text{reasoning}}(x^t, h)$ for an attention head h at timestep t as the proportion of attention it allocates to all supporting facts (the "needles") relative to the total attention it distributes across the entire effective context (excluding the interference from attention sinks (Xiao et al., 2023) and local attention).[2] Formally,

$$S_{\text{reasoning}}(x^t, h) = \frac{\sum_{i \in I_{\text{needle}}} \mathbf{a}_i^{h,t}}{\sum_{j \in I \setminus \{I_{\text{sink}} \cup I_{\text{local}}\}} \mathbf{a}_j^{h,t}} \quad (2)$$

where I_{needle} is the set of indices for all supporting fact tokens, while I_{sink} and I_{local} represent the indices of the attention sink and local attention window, respectively. Intuitively, this score relaxes the strong copy-paste condition and measures a head's focus on the correct information at a given step, better aligning with how facts are used in the LLM reasoning process.

4.2 Experiments

We use the HotpotQA dataset (Yang et al., 2018) as our testbed. HotpotQA is a question-answering dataset that requires multi-hop reasoning, where the model must find and integrate multiple discrete supporting facts from the context to formulate a correct answer. To examine whether the three claims in Section 3 are still valid for the reasoning retrieval heads on the HotpotQA task, we adopt the analytical framework from Section 3.

Dynamism We first validate Claim 1. Figure 5 visualizes the retrieval scores of the top-10 variated retrieval heads during a generation. Consistent with the NIAH task, the retrieval head pattern is highly

[2]Following common practice, we exclude two types of attention patterns that are not directly related to long-range retrieval: (1) **Attention Sinks** (Xiao et al., 2023), where certain tokens (e.g., the initial BOS token) often receive high attention regardless of content, and (2) **Local Attention**, where heads focus on a small, fixed window of recent tokens.

dynamic: no single head dominates throughout the entire process. Instead, different heads operate as retrieval heads at distinct generation stages, confirming that the dynamic nature of retrieval heads is a general phenomenon that persists in complex reasoning tasks.

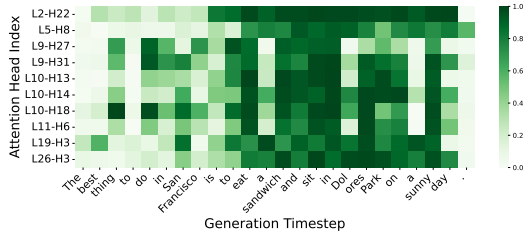


Figure 5: **Dynamic Pattern of Retrieval Heads in a Multi-Hop Reasoning Task.** The heatmap illustrates the retrieval scores (defined in Eq. 2) for ten active retrieval heads over the course of the generation process.

Irreplaceability Next, we validate Claim 2 (Irreplaceability) through the head ablation experiment described in Section 3.3. The results in both Figure 6 and Figure 7 show that masking the dynamic retrieval heads at each step (identified using Equation (2)) leads to a far more severe performance degradation on the HotpotQA task than masking an equivalent number of top static retrieval heads or random heads.

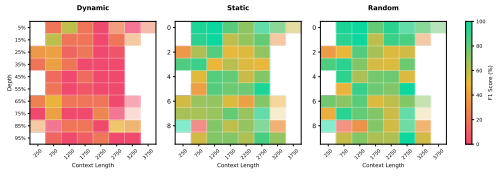


Figure 6: **Impact of Head Ablation on Multi-Hop Reasoning Performance.** F1-score comparison of HotpotQA test scores after ablating three different sets of attention heads on llama3.1-8b. The opacity of each cell corresponds to the number of valid samples it contains, with blank as no valid samples.

Correlation Finally, we validate Claim 3. Using the same Temporal Offset CCA, Figure 8 shows that the strong linear correlation between hidden states and retrieval scores persists in HotpotQA. In terms of the MLP experiment, unlike the NIAH task where retrieval is binary, in HotpotQA, the retrieval score (defined in Eq. 2) is a continuous value representing the intensity of attention on supporting facts. Consequently, we trained an MLP regressor rather than a classifier to predict the precise retrieval score vector for all heads from the final hidden state at the synchronous step ($k = 0$). As shown in Table 3, the probes achieve high R^2 scores (up to 0.81) across all models. This indicates that the information of retrieval scores is effectively encoded in the hidden state, verifying our claim of the strong correlation.

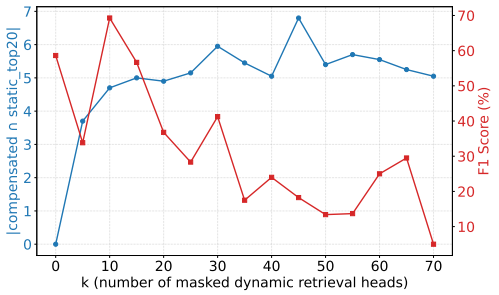


Figure 7: **Irreplaceability of Dynamic Heads in a Multi-Hop Reasoning Context.** The plots show the degradation in HotpotQA performance as an increasing number (k) of dynamic retrieval heads are ablated on llama3.1-8b.

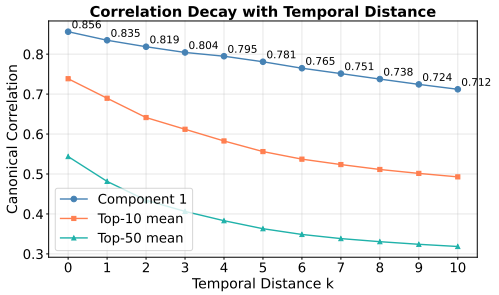


Figure 8: **Temporal Decay of Correlation on a Multi-hop Reasoning Task.**

Model	MSE (↓)	MAE (↓)	R^2 (↑)
llama3.1-8b	0.0023	0.0177	0.8120
llama3.2-3b	0.0036	0.0247	0.8015
qwen3-8b	0.0050	0.0255	0.7200
llama2-13b	0.0009	0.0121	0.7669
phi4-mini	0.0014	0.0109	0.7333

Table 3: **Performance of MLP Probes in Predicting Reasoning Retrieval Scores on HotpotQA.**

5 Case Study: Applying Dynamic Retrieval Heads to Dynamic RAG

In the previous sections, we verified that the LLM's retrieval behavior is better characterized by dynamic retrieval heads rather than a fixed set of static retrieval heads. However, while Sections 3 and 4 focus on comparing the properties of dynamic and static retrieval heads themselves, an important remaining question is how much practical advantage dynamic retrieval heads provide in real-world applications. Therefore, in this section, we designed

Model	Dynamic EM / F1	Static EM / F1	Dynamic Random EM / F1	Fixed Random EM / F1	w/o RAG EM / F1
llama3.1-8b	**0.456 / 0.5586**	0.398 / 0.5098	0.272 / 0.3670	0.272 / 0.3763	0.252 / 0.3257
llama3.2-3b	0.384 / 0.4993	**0.428 / 0.5386**	0.224 / 0.3143	0.226 / 0.3051	0.184 / 0.2439
qwen3-8b	**0.286 / 0.3580**	0.278 / 0.3429	0.210 / 0.2804	0.210 / 0.2804	0.220 / 0.2961
llama2-13b	**0.284 / 0.3838**	0.278 / 0.3789	0.276 / 0.3762	0.272 / 0.3751	0.192 / 0.2750
phi4-mini	**0.202 / 0.2690**	0.186 / 0.2505	0.082 / 0.1090	0.086 / 0.1111	0.172 / 0.2331

Table 4: Performance comparison of different retrieval strategies on the HotpotQA dataset (Exact Match (EM) / F1-Score (F1)). For each model, the better-performing strategy between Dynamic and Static is highlighted in **bold**.

an case study integrating them into an existing Dynamic Retrieval-Augmented Generation (Dynamic RAG) framework for question answering.

5.1 Task Description

Unlike traditional RAG that retrieves from external knowledge bases, retrieval heads are specialized for sourcing information already present within the model's input context. We therefore focus on an **in-context retrieval** task, evaluating the model's ability to accurately attend to and utilize key information provided in its input.

5.2 Method

We adapt DRAGIN (Su et al., 2024b), a prominent Dynamic RAG framework, for our in-context retrieval task. We make the following changes to integrate the retrieval heads into this framework. At each generation step, the model's access to the context is controlled via attention masks. When no retrieval is needed, the entire context is masked, while the question and the currently generated text remain visible. When retrieval is needed, we identify the active retrieval heads for that step, determine their top-k most attended-to positions by averaging their attention scores, cluster those positions and expand a fixed-size window around each cluster. In the subsequent regeneration step, only the tokens corresponding to these clusters are made visible to the model via the attention mask. [3] We use DRAGIN's RIND algorithm to determine when to retrieve, and the detailed pipeline can be found in Algorithm 1 in Appendix E.

5.3 Experiment Setup

Datasets Following Section 4, we employ the HotpotQA dataset (Yang et al., 2018) for better practical utility evaluation compared to NIAH.

[3]We choose to use attention masking over directly rewriting the context to minimize potential disruptions to the autoregressive process, thereby isolating the impact of the retrieved information.

Models We selected five popular open-source models of varying sizes and architectures to ensure the generalizability of our findings: *meta-llama/Llama-3.1-8B-Instruct* (llama3.1-8b), *meta-llama/Llama-3.2-3B-Instruct* (llama3.2-3b), *Qwen/Qwen3-8B* (qwen3-8b), *meta-llama/Llama-2-13b-chat-hf* (llama2-13b), and *microsoft/Phi-4-mini-instruct* (phi4-mini) (Dubey et al., 2024; Yang et al., 2025; Touvron et al., 2023; Abouelenin et al., 2025).

Baselines We compare five configurations of our adapted DRAGIN framework to evaluate the efficacy of different head selection strategies:

- **Dynamic**: Use dynamic retrieval heads identified by the MLP probe at each step. MLP probe from Section 4.2 is used to predict the top-5 dynamic retrieval heads.
- **Static**: Use 5 pre-identified static retrieval heads.
- **Dynamic Random**: Use a new set of 5 randomly selected heads at each retrieval step.
- **Fixed Random**: Use a fixed set of 5 randomly selected heads for the entire generation.
- **w/o RAG**: Perform no retrieval with no context provided, relying solely on the model's parametric knowledge.

5.4 Results

The results are summarized in Table 4. The observations indicate a superiority of dynamic retrieval heads over the other baselines: For the majority of the tested models (llama3.1-8b, qwen3-8b, llama2-13b, and phi4-mini), the Dynamic strategy using dynamic retrieval heads achieves higher or comparable performance in both EM and F1-score compared to the Static strategy. The advantage is particularly pronounced for llama3.1-8b, where the dynamic strategy's F1-score (0.5586) is nearly 10% higher than that of the static one (0.5098). This aligns with our findings in Sections 3 and 4,

suggesting that "expert" heads selected dynamically at each timestep can more precisely locate the information required for the current reasoning step than a fixed set of "generalist" heads. An exception is the llama3.2-3b, with the Static strategy (F1=0.5386) outperforming the Dynamic one (F1=0.4993). We hypothesize this may be related to the model size: compared to other models, this model has the least number of layers with the least attention heads, indicating a possibility that each head needs to perform multiple tasks.

To verify that the superiority of the dynamic approach is robust to the choice of the truncation threshold, we additionally evaluated the downstream performance across varying top-k constraints ($k \in \{5, 10, 20\}$). As detailed in Appendix C, the dynamic strategy consistently outperforms the static baseline across all settings, further validating the functional necessity of dynamically selecting retrieval heads.

6 Conclusion

This paper presents the first systematic study of retrieval heads from a dynamic perspective. Through extensive analysis on NIAH and HotpotQA tasks, we establish that retrieval head activation is highly dynamic, functionally irreplaceable, and correlated with the model's internal state. Furthermore, we demonstrate the practical utility of these insights by integrating dynamic retrieval heads into a Dynamic RAG framework, achieving significant performance gains compared to static retrieval heads.

Beyond these immediate gains, our dynamic perspective opens up concrete directions for future research. First, our findings suggest a path toward **dynamic KV cache compression**. Existing static compression methods typically prune heads based on average importance, which risks discarding "long-tail" retrieval heads that are rarely active but functionally irreplaceable. Future inference systems could leverage the predictive mechanism (e.g., via lightweight probes) to dynamically retain or fetch KV pairs only for the specific heads active at the current step, achieving high compression rates while preserving sparse retrieval capabilities. Second, the token-level predictive signal can enable **hallucination detection and precision RAG**. The absence of retrieval head activation during the generation of factual claims could serve as an intrinsic, interpretable indicator of potential hallucination. Conversely, systems can use this signal to trigger external retrieval strictly when the model indicates a genuine need for information, minimizing context pollution. Ultimately, we hope this work offers a more granular understanding of LLM internal mechanisms and advances the development of efficient and interpretable model steering.

Limitations

First, our Dynamic RAG experiment utilizes attention masking to simulate retrieval for validation purposes, rather than physically selecting and concatenating context as in standard production RAG pipelines; bridging this gap for practical deployment remains a direction for future work. Second, our method in the case study in Section 5 relies on a learned MLP probe to predict head activation. While the probe achieves high accuracy, it is not an oracle; any prediction errors imply that the identified heads may not perfectly match the true optimal dynamic retrieval heads, potentially introducing noise into the retrieval process. Finally, our analysis primarily focuses on retrieval-intensive QA tasks (NIAH and HotpotQA); whether these findings generalize to other long-context domains, such as summarization or long-context QA, warrants further investigation.

Acknowledgement

Yuping Lin, Pengfei He, Yingqian Cui, and Jiliang Tang are supported by the National Science Foundation (NSF) under grant numbers CNS2321416, IIS2212032, IIS2212144, IIS 2504089, DUE2234015, CNS2246050, DRL2405483 and IOS2035472.

Yue Xing is supported by NSF DMS 2515194, Open Philanthropy, NVIDIA Academic Grant Program and Google Cloud Research Credit.

References

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, and 1 others. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, and 1 others. 2022. Improving language models by retrieving from

trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and 1 others. 2024. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, and 1 others. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, and 1 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12.

Yu Fu, Zefan Cai, Abedelkadir Asi, Wayne Xiong, Yue Dong, and Wen Xiao. 2024. Not all heads matter: A head-level kv cache compression method with integrated retrieval and reasoning. *arXiv preprint arXiv:2410.19258*.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. *Advances in neural information processing systems*, 35:30583–30598.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Greg Kamradt. 2023. LLMTest_NeedleInAHaystack: A test to measure llm performance over long contexts. https://github.com/gkamradt/LLMTest_NeedleInAHaystack.

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

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

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

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, and 1 others. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

Ofir Press, Noah A Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Tal Ridnik, Emanuel Ben-Baruch, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. 2021. Asymmetric loss for multi-label classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 82–91.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024a. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024b. Dragin: dynamic retrieval augmented generation based on the information needs of large language models. *arXiv preprint arXiv:2403.10081*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2024. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Wuwei Zhang, Fangcong Yin, Howard Yen, Danqi Chen, and Xi Ye. 2025. Query-focused retrieval heads improve long-context reasoning and re-ranking. *arXiv preprint arXiv:2506.09944*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, and 1 others. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.

A Experimental Setup for Needle-in-a-Haystack Masking

For reproducibility, all experiments in this paper use greedy sampling as the LLM decoding strategy. Each experiment can be perform on a single NVIDIA H200 GPU.

A.1 Needle-in-a-Haystack Test Setting

The original Needle-in-a-Haystack (NIAH) test (Kamradt, 2023) was proposed in 2023, while most of the models we study in this work were released after this. This raises the possibility that these models were exposed to the NIAH data during their training phase. To prevent potential data leakage, we use **dynamically, randomly generated UUID strings** as the needle string as a substitute, to exclude the possibility that the model has seen the Needle data during its training phase. Its specific format is as follows:

- Needle: The magic word is [UUID].
- Question: What is the magic word?

For the experiment conducted in Section 3.3, the detailed experiment setting is as follows:

For evaluation criteria, we use two types: accuracy and ROUGE-L (Lin, 2004). Accuracy is to check whether the model's response completely contains the UUID string. If it does, the score is 1.0; otherwise, it is 0.0, even if there is only a one-character difference.

We extend the context length to the specified length by repeatedly concatenating and truncating the haystack text. Then, we randomly select a depth, backtrack to the nearest end of a sentence to insert the needle, and apply a dialogue template to construct the input. The dialogue template is as follows:

NIAH Test Prompt

System
You are a helpful AI bot that answers questions for a user. Keep your response short and direct.
User
Context:
{A Haystack with a Needle inserted in}

Question:
{Question}
Instruction:
Don't give information outside the document or repeat your findings.

To obtain the data for Figure 2, we conducted 5 independent runs for each grid cell and reported the average metric value.

To collect the data for Figure 3, we conducted 20 independent runs at intervals of $k = 5$, with the haystack length fixed at 5000 tokens. The reported results are the averages of these trials.

A.2 Detailed Setting for the Experiment in Section 3.3.2

In the ablation study involving the masking of k dynamic heads, our goal is to quantify the model's attempt to compensate using static retrieval heads. To do this, we track the compensated heads and measure their overlap with the Top-20 static retrieval heads.

A key methodological challenge is how to aggregate this metric over a full generation sequence. We observed that under heavy ablation (i.e., large k), the model's retrieval mechanism often collapses in later generation timesteps, resulting in zero retrieval heads. Consequently, a simple average across all timesteps would include these zero-values, artificially deflating the metric and failing to reflect the model's actual capacity to utilize compensated heads. To address this and robustly capture the model's peak compensatory effort, we record the maximum number of compensated heads observed at any single timestep within each sample's generation.

Formally, for each sample s, let $H_{s,t}$ be the set of dynamic retrieval heads at timestep t before masking, let $H'_{s,t}$ be the set of dynamic retrieval heads at timestep t after masking. Let $E_{s,t} = H'_{s,t} - H_{s,t}$ be the set of compensated heads at timestep t. We compute the maximum intersection with the top-20 static set, H_{static}, within the sample s: $m_s = \max_t |E_{s,t} \cap H_{\text{static}}|$. The final metric is the average of m_s over all samples.

A.3 Details for Entropy Metric in Dynamism Analysis

We calculate the entropy of the retrieval score distribution to measure how broadly retrieval responsibility is shared. Let p_h be the probability that head h is activated as a retrieval head across all timesteps. The entropy is defined as:

$$S = -\sum_h p_h \ln p_h \tag{3}$$

To provide a baseline for interpretation, consider a scenario where retrieval is exclusively and uniformly performed by the top-20 static heads. In this case, $p_h = \frac{1}{20}$ or these 20 heads and 0 for others. The resulting entropy would be:

$$S_{baseline} = -\sum_{i=1}^{20} \frac{1}{20} \ln\left(\frac{1}{20}\right) = \ln(20) \approx 2.9957 \tag{4}$$

Our observed entropy values are consistently higher than this baseline (e.g., 3.8154 for llama3.1-8b). Combining with the low Jaccard w/ Static values, this indicates that the effective number of heads participating in retrieval is significantly larger than 20.

A.4 Experimental Settings for Canonical Correlation Analysis

To analyze the correlation between the model's final hidden states and retrieval scores, we employed Canonical Correlation Analysis (CCA). The detailed experimental procedure is as follows:

Data Preprocessing We first standardized the retrieval scores (min 0, max 1) to ensure consistent scaling across attention heads. To improve computational efficiency and focus on the principal signal subspaces, we applied Principal Component Analysis (PCA) to both the hidden states and retrieval scores prior to CCA. We retained principal components explaining 95% of the variance for the hidden states and 99% for the retrieval scores.

CCA Configuration We set the number of canonical components to 50.

Temporal Offset We analyzed the correlation with a temporal offset k ranging from 0 to 10. For each offset k, we paired the hidden state at timestep n with the retrieval scores at timestep $n + k$. Any samples where $n + k$ exceeded the sequence length were excluded from the analysis.

A.5 Experimental Settings for MLP Probe Training

To investigate the fine-grained correlation of retrieval head patterns, we trained a Multi-Layer Per-

ceptron (MLP) probe for each LLM. The detailed configuration is as follows:

Model Architecture The probe is a feed-forward neural network consisting of three hidden layers with dimensions [8192,4096,4096]. We applied a dropout rate of 0.1 after each hidden layer to prevent overfitting. The input dimension matches the hidden size of the respective LLM, and the output dimension corresponds to the total number of attention heads.

Training Configuration The models were trained for 100 epochs with a batch size of 128. We used the Adam optimizer with a learning rate of 3×10^{-4} and a scheduler that reduces the learning rate upon a plateau in validation loss (patience set to 3 epochs). To handle the sparse retrieval score distribution of attention heads, we employed the Asymmetric Loss (Ridnik et al., 2021) as the objective function. Gradients were clipped at a maximum norm of 1.0 to ensure stability.

Data Split The dataset collected from the NIAH runs was split into training (70%), validation (20%), and testing (10%) sets. All reported metrics (Precision, Recall, F1, AUPRC) are evaluated on the held-out test set.

B Sensitivity Analysis of Jaccard Similarity

To address potential concerns regarding the arbitrary choice of the truncation threshold ($k = 20$) for identifying static retrieval heads, we report the Jaccard similarity between dynamic retrieval heads and static heads at varying thresholds ($k \in \{20, 50, 100\}$).

Table 5 summarizes the results across all evaluated models. We observe a consistent decrease in Jaccard similarity as k increases. This indicates that the expansion of the static set (the denominator in Jaccard similarity) significantly outpaces the inclusion of additional dynamic retrieval heads in the intersection. This result reinforces our finding that dynamic retrieval heads originate from a broad, long-tail distribution of attention heads rather than being confined to a slightly expanded static subset.

C Robustness of Dynamic Retrieval Across Different k Thresholds

In our primary Dynamic RAG case study (Section 5), we demonstrated the effectiveness of dynamically selecting retrieval heads. To further ensure that this advantage is not an artifact of a specific threshold, we conducted a robustness check on the llama3.1-8b model by varying the number of allowed retrieval heads, denoted as $k \in \{5, 10, 20\}$.

Model	Jaccard ($k = 20$)	Jaccard ($k = 50$)	Jaccard ($k = 100$)
Llama-3.1-8B	0.3512	0.2106	0.1236
Llama-3.2-3B	0.3126	0.1755	0.0964
Qwen3-8B	0.4611	0.3011	0.1799
Llama-2-13B	0.2077	0.1116	0.0612
Phi-4-mini	0.1845	0.1087	0.0606

Table 5: Jaccard similarity between dynamic retrieval heads and static retrieval heads at different truncation thresholds (k).

Table 6 presents the Exact Match (EM) and F1 scores across the dynamic, static, and random baseline strategies. We observe two key phenomena:

First, consistent superiority. The dynamic strategy consistently outperforms both the static baseline and the random baselines across all evaluated k values. This confirms that selecting heads based on the current timestep's predictive state is fundamentally more effective than relying on a fixed, historically aggregated static set.

Second, the sparsity constraint. Counterintuitively, the absolute performance for all methods drops as k increases from 5 to 10 and 20. This phenomenon aligns perfectly with our statistical findings in Section 3.2. As shown in Table 1, the average number of active dynamic retrieval heads per step for llama3.1-8b is relatively small and highly sparse (e.g., concentrated in a few heads at any exact moment). Forcing the model to utilize a strict top-10 or top-20 heads introduces "noise injection." In other words, when k exceeds the actual number of required retrieval heads for a specific step, the system artificially forces non-retrieval heads to participate in the retrieval operation, which interferes with the generation and degrades the overall downstream accuracy. This further highlights the necessity of dynamic sparsity over static inclusion.

Method	$k = 5$ (EM/F1)	$k = 10$ (EM/F1)	$k = 20$ (EM/F1)
Dynamic (Ours)	**0.456 / 0.5586**	**0.286 / 0.3877**	**0.292 / 0.3967**
Static	0.398 / 0.5098	0.280 / 0.3767	0.282 / 0.3810
Dynamic Random	0.272 / 0.3670	0.280 / 0.3807	0.274 / 0.3713
Fixed Random	0.272 / 0.3763	0.276 / 0.3711	0.276 / 0.3751

Table 6: Downstream QA performance (EM/F1) on llama3.1-8b under varying top-k constraints. Bold values indicate the best performance in each column.

D Additional Experiment Results

D.1 All Heads Ablation Results

See Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, Figure 14.

D.2 Different Numbers of Heads Ablation Results

See Figure 15, Figure 16, Figure 17, Figure 18, Figure 19, Figure 20.

E Algorithms for the Dynamic RAG Method in Section 5

See Algorithm 1, Algorithm 2.

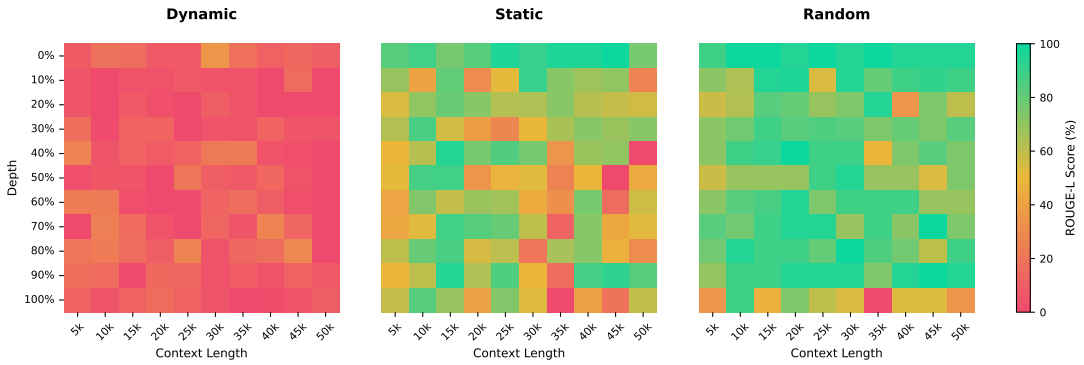


Figure 9: Head Ablation on NIAH test on llama3.1-8b. Using ROUGE-L as the metric.

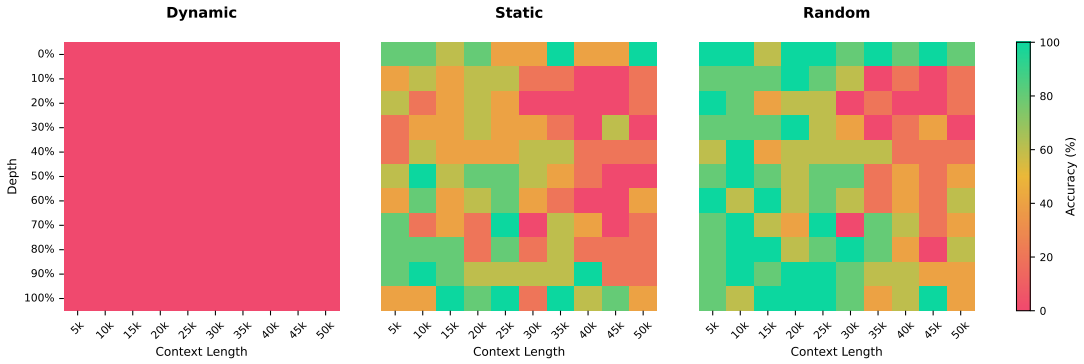
Algorithm 1 Dynamic RAG with In-Context Retrieval

Require: Context $\mathcal{C}$, Question $\mathcal{Q}$, Model $\mathcal{M}$

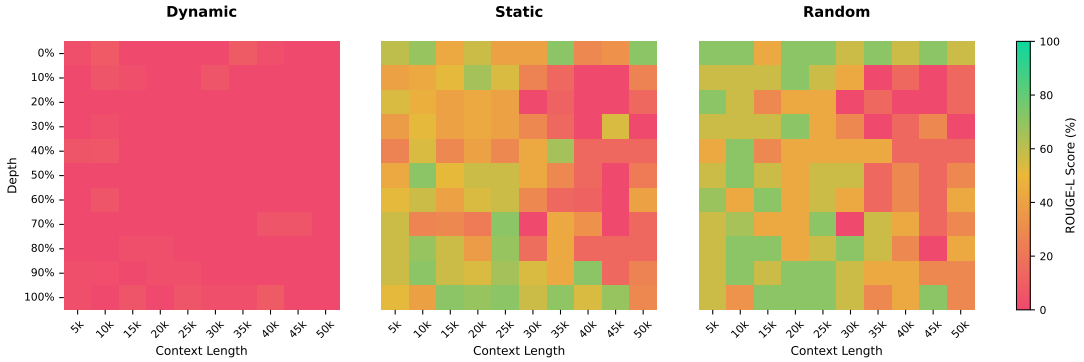
Generated Text $\mathcal{G} \leftarrow$ ''
Visible Mask $\mathcal{V} \leftarrow \text{MaskAll}(\mathcal{C})$ ▷ Initially mask context
while not finished **do**
  Input $\mathcal{I} \leftarrow \text{Concat}(\mathcal{C}, \mathcal{Q}, \mathcal{G})$
  Draft $\mathcal{D}$, Attentions $\mathcal{A} \leftarrow \mathcal{M}.\text{GenerateDraft}(\mathcal{I}, \text{mask} = \mathcal{V})$
  is_hallucination, pos $\leftarrow \text{RIND}(\mathcal{D})$
  if is_hallucination **then**
    $\mathcal{G} \leftarrow \text{Retract}(\mathcal{G}, \text{to sentence of pos})$
    $\mathcal{V} \leftarrow$ **Retrieve**$(\mathcal{C}, \mathcal{Q}, \mathcal{G})$ ▷ See Alg. 2
    Input $\mathcal{I} \leftarrow \text{Concat}(\mathcal{C}, \mathcal{Q}, \mathcal{G})$
    Rewritten $\leftarrow \mathcal{M}.\text{Generate}(\mathcal{I}, \text{mask} = \mathcal{V})$
    $\mathcal{G} \leftarrow \text{Append}(\mathcal{G}, \text{Rewritten})$
  else
    $\mathcal{G} \leftarrow \text{Append}(\mathcal{G}, \mathcal{D})$
    $\mathcal{V} \leftarrow \text{MaskAll}(\mathcal{C})$ ▷ Re-mask for next draft
  end if
end while
return $\mathcal{G}$

Algorithm 2 In-Context Retrieval via Attention Heads

function RETRIEVE($\mathcal{C}, \mathcal{Q}, \mathcal{G}$)
  UnMaskAll($\mathcal{C}$)
  Active Heads $\mathcal{H}_{dyn} \leftarrow \text{IdentifyHeads}(\mathcal{C}, \mathcal{Q}, \mathcal{G})$
  MaskAll($\mathcal{C}$)
  Avg Scores $\mathbf{s} \leftarrow \text{AverageAttention}(\mathcal{H}_{dyn})$
  Top-k Indices $\mathcal{K} \leftarrow \text{TopKIndices}(\mathbf{s}, k)$
  Index Clusters $\mathcal{C}_{idx} \leftarrow \text{ClusterIndices}(\mathcal{K})$
  Expanded Windows $\mathcal{W} \leftarrow \emptyset$
  for each cluster $c \in \mathcal{C}_{idx}$ **do**
    Representative Index $i_{rep} \leftarrow \text{GetRepresentative}(c)$
    $\mathcal{W} \leftarrow \mathcal{W} \cup \text{ExpandWindow}(i_{rep}, \text{size})$
  end for
  Final Windows $\mathcal{W}_{final} \leftarrow \text{MergeOverlapping}(\mathcal{W})$
  Visible Mask $\mathcal{V} \leftarrow \text{CreateMaskFromWindows}(\mathcal{W}_{final})$
  return $\mathcal{V}$
end function

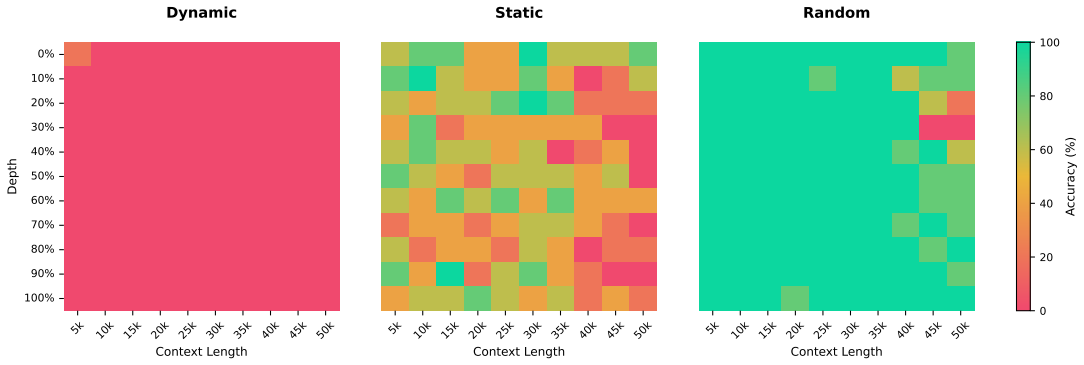


(a) Accuracy

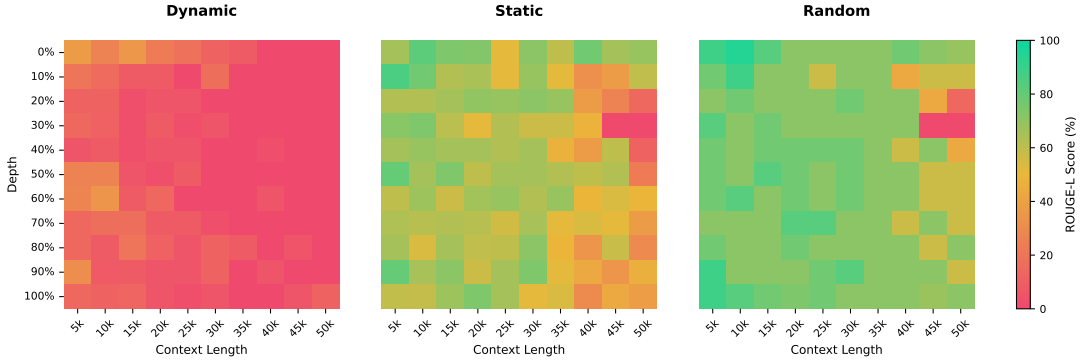


(b) ROUGE-L

Figure 10: Head Ablation on NIAH test on llama3.2-3b.



(a) Accuracy



(b) ROUGE-L

Figure 11: Head Ablation on NIAH test on qwen3-8b.

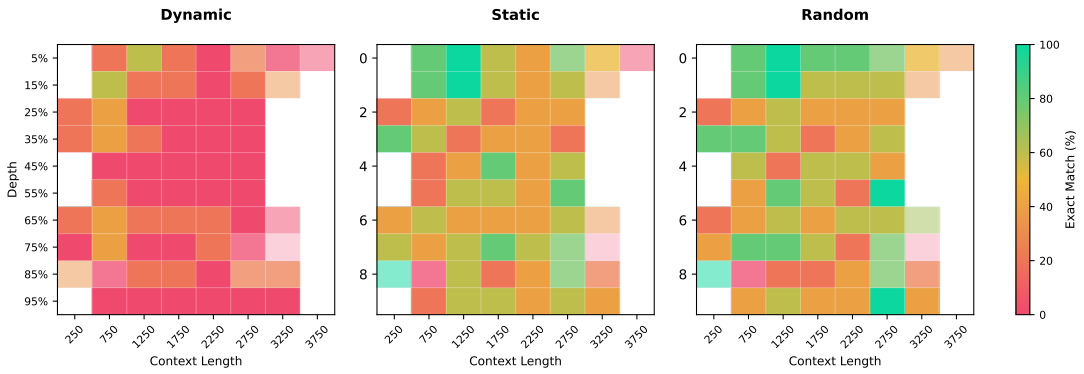
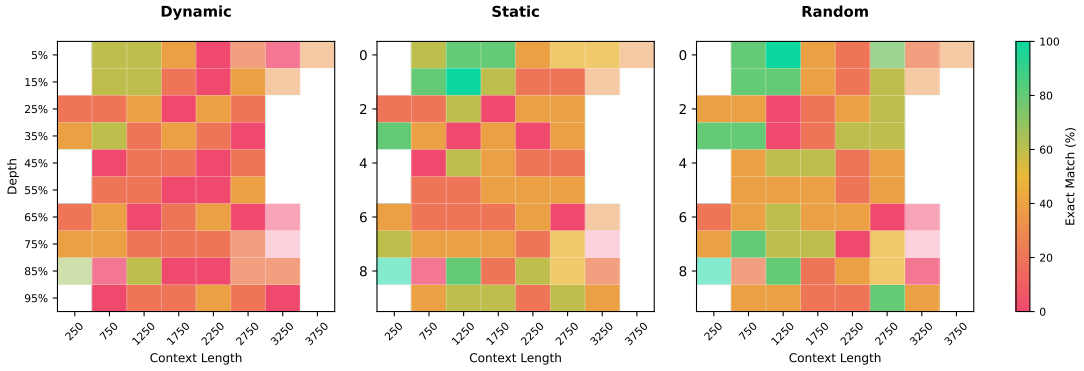
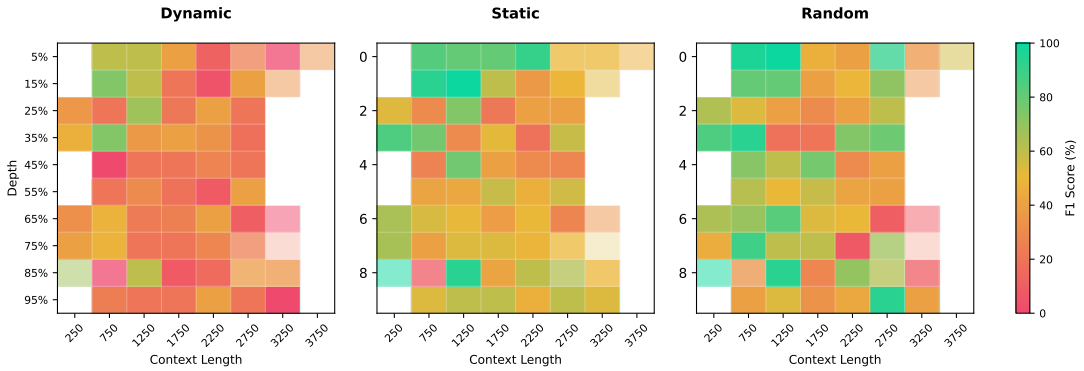


Figure 12: Head Ablation on HotpotQA test on llama3.1-8b. Using EM as the metric.

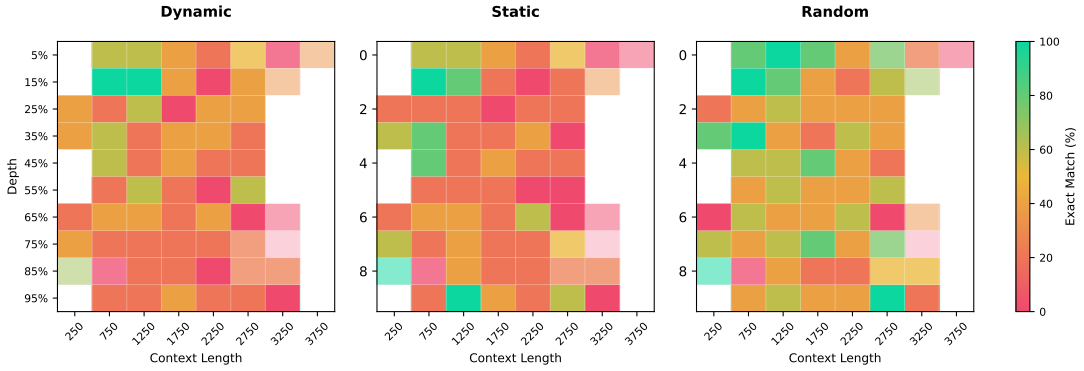


(a) EM

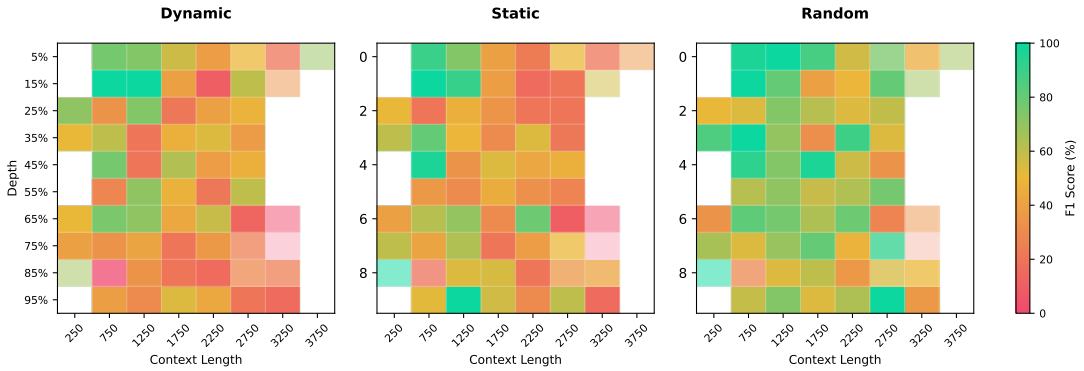


(b) F1

Figure 13: Head Ablation on HotpotQA test on llama3.2-3b.



(a) EM



(b) F1

Figure 14: Head Ablation on HotpotQA test on qwen3-8b.

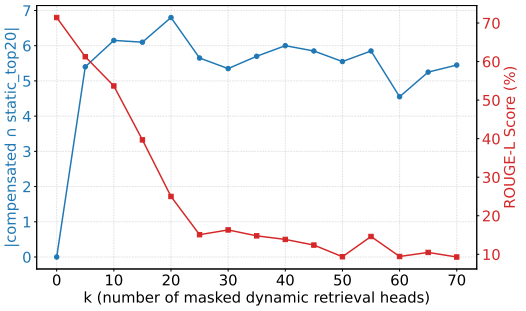
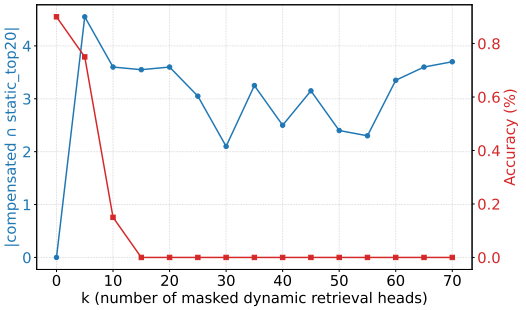
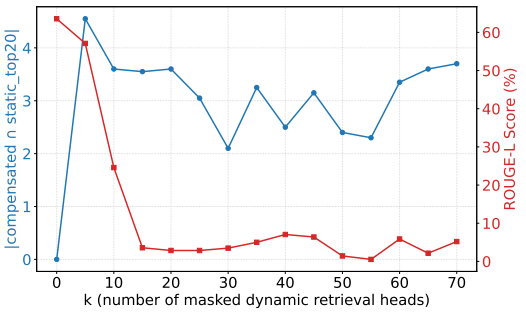


Figure 15: Different Numbers of Head Ablation on NIAH test on llama3.1-8b. Using ROUGE-L as the metric.

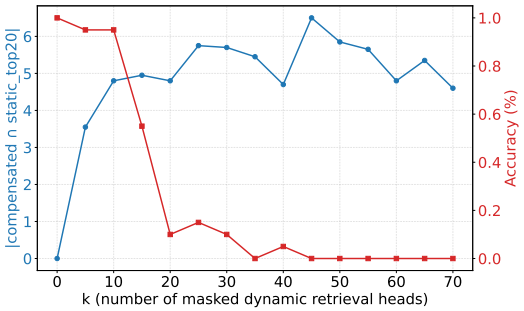


(a) Accuracy

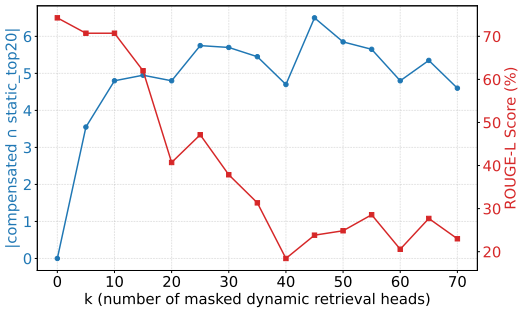


(b) ROUGE-L

Figure 16: Different Numbers of Head Ablation on NIAH test on llama3.2-3b.



(a) Accuracy



(b) ROUGE-L

Figure 17: Different Numbers of Head Ablation on NIAH test on qwen3-8b.

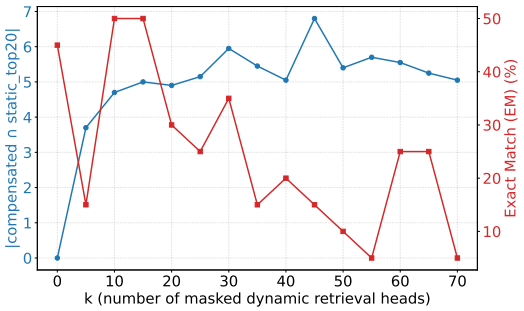
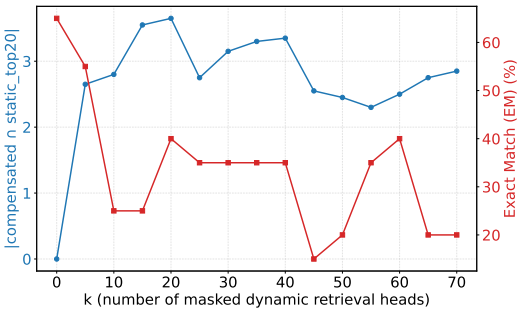
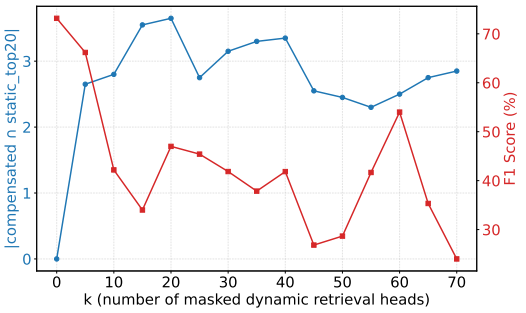


Figure 18: Different Numbers of Head Ablation on NIAH test on llama3.1-8b. Using EM as the metric.

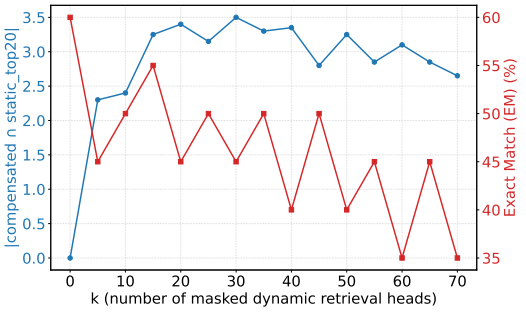


(a) EM

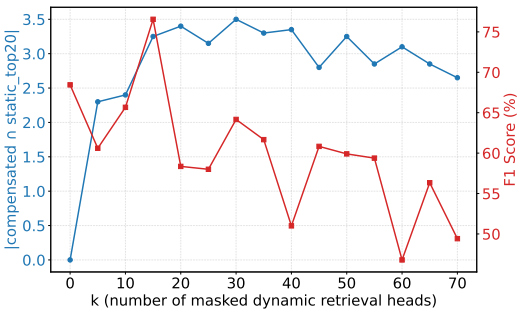


(b) F1

Figure 19: Different Numbers of Head Ablation on HotpotQA test on llama3.2-3b.



(a) EM



(b) F1

Figure 20: Different Numbers of Head Ablation on HotpotQA test on qwen3-8b.