Model Tells Itself Where to Attend: Faithfulness Meets Automatic Attention Steering

Anonymous ACL submission

Abstract

Large language models (LLMs) have demonstrated remarkable performance across various real-world tasks. However, recent studies reveal that LLMs often struggle to fully comprehend and effectively utilize their input contexts, resulting in responses that lack faithfulness or suffer from hallucination. This difficulty becomes particularly evident when the contexts are lengthy or contain distracting information, which can divert LLMs from fully capturing essential evidence. Most prior work focuses on designing effective prompts to guide LLMs in utilizing contextual information more faithfully. For instance, iterative prompting highlights key information through two high-level prompting steps that first ask the LLM to identify important pieces of context and then derive answers accordingly. However, prompting methods are constrained to highlighting key information implicitly in token space, which is often insufficient to fully steer the model's attention. To improve model faithfulness more reliably, we propose AutoPASTA, a method that automatically identifies contextual key information and explicitly highlights it by steering the model's attention scores. Similar to prompting, AutoPASTA is applied at inference time and does not require changing any model parameters. Our experiments on open-book QA demonstrate that AutoPASTA can effectively guide models to grasp essential contextual information, leading to substantially improved model faithfulness and performance, e.g., an average improvement of 11.26% for LLAMA3-8B-Instruct. Code will be publicly available.

1 Introduction

Large language models (LLMs) exhibit remarkable performance across various natural language processing (NLP) tasks and artificial intelligence (AI) applications (Brown et al., 2020; Touvron et al., 2023; OpenAI, 2023). Despite their remarkable capabilities, recent studies reveal that LLMs often encounter challenges in fully understanding their input contexts, overlooking or showing insensitivity to crucial contextual information (Kasai et al., 2023; Li et al., 2023; Si et al., 2023; Zhou et al., 2023; Yu et al., 2024; Zhang et al., 2024). Consequently, the models tend to fabricate answers (also known as hallucination), resulting in *unfaithful* responses that are inconsistent with the presented contexts (Zhou et al., 2023; Yu et al., 2024). This becomes particular problematic when models are presented prompts containing lengthy background contexts (Liu et al., 2023) or complex questions, such as open-book question answering (QA) (Kwiatkowski et al., 2019; Shi et al., 2023b; Peng et al., 2023). In these information-dense scenarios, lengthy contexts can overwhelm LLMs, which contain many details with varying degree of relevance (Wan et al., 2024; Zhang et al., 2024). Some sentences are crucial for providing the correct answer, while others, though irrelevant, can distract models from fully capturing the essential information.

To improve model faithfulness, most prior work explores well-designed prompts to guide the LLM to use contextual knowledge more reliably (Zhou et al., 2023; Wan et al., 2024; Radhakrishnan et al., 2023). In particular, *iterative prompting* in chain-of-thought (COT; Wei et al., 2022) fashion can help LLMs decompose complex task-solving into more interpretable and manageable intermediate steps, thus yielding better performance (Radhakrishnan et al., 2023). Motivated by this, it is natural to design multi-step iterative prompting to guide LLMs to pay more attention to relevant contextual parts and derive answers accordingly. Specifically, for open-book QA tasks iterative prompting can be decomposed into two steps: (i) *identifying key information* and (ii) *deriving answers using the key information*. This strategy can work effectively for black-box LLMs of significantly large sizes (e.g., >100B) (Radhakrishnan et al., 2023). However, for LLMs of smaller sizes (e.g., LLAMA3-

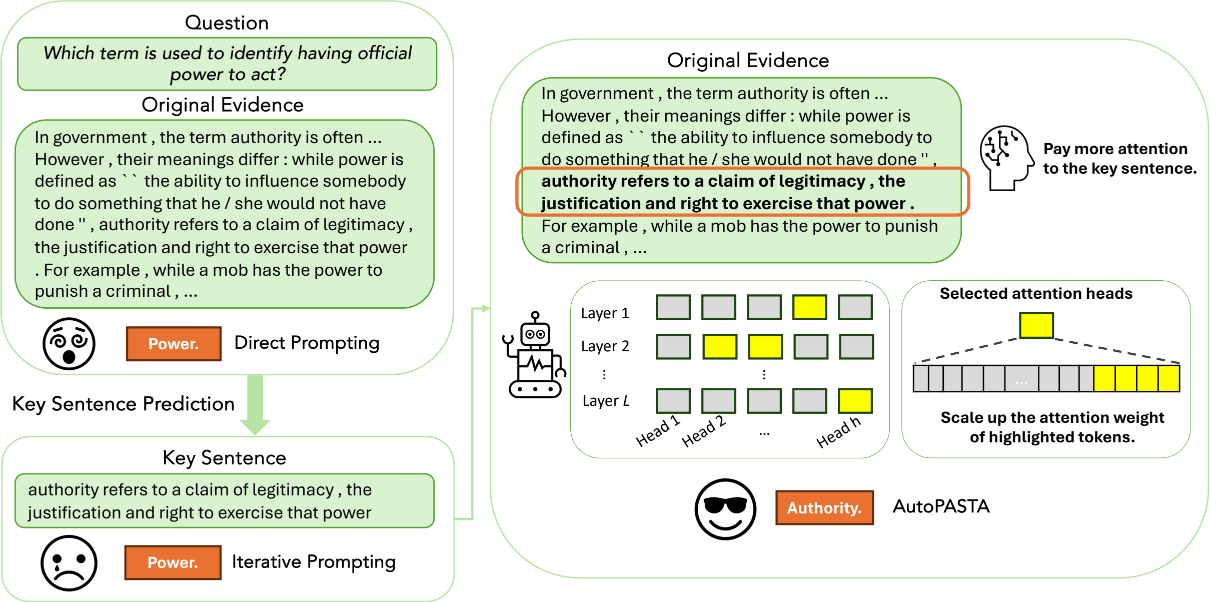

Question
Which term is used to identify having official power to act?
Original Evidence
In government , the term authority is often ... However , their meanings differ : while power is defined as `` the ability to influence somebody to do something that he / she would not have done '' , authority refers to a claim of legitimacy , the justification and right to exercise that power . For example , while a mob has the power to punish a criminal , ...
Power.
Direct Prompting
Key Sentence Prediction
Key Sentence
authority refers to a claim of legitimacy , the justification and right to exercise that power
Power.
Iterative Prompting
Original Evidence
In government , the term authority is often ... However , their meanings differ : while power is defined as `` the ability to influence somebody to do something that he / she would not have done '' ,
authority refers to a claim of legitimacy , the justification and right to exercise that power .
For example , while a mob has the power to punish a criminal , ...
Pay more attention to the key sentence.
Layer 1
Layer 2
Layer L
Head 1
Head 2
...
Head h
Selected attention heads
Scale up the attention weight of highlighted tokens.
Authority.
AutoPASTA

Figure 1: The illustration of AutoPASTA and alternative methods given a running example. Responses by Vicuna-7B are shown in red square where *Authority* is the label. Prompting methods (direct and iterative prompting) fail to guide a model to derive correct answers while AutoPASTA successfully steers it to answer correctly by explicitly highlighting identified key parts.

70B, Meta, 2024), it remains unclear if this strategy can guide models to fully attend to the extracted key information and subsequently improve performance. First, step-by-step generations typically result in longer contexts. However, key information is only highlighted in token space by appending the short predicted key sentences, which are often not strong enough to fully steer the model's attention. As illustrated in the left part of Figrue 1, even though the model correctly predicts the key sentence which is appened in the subsequent prompt, it still fails to provide the correct answer. Moreover, errors can propagate across steps, further compromising performance. Therefore, we aim to develop an alternative inference framework that emulates iterative prompting while addressing these limitations.

Motivated by this, we propose AutoPASTA, an inference-only approach that (i) *automatically* identifies key contextual parts, and (ii) *explicitly* highlights them through attention score manipulation for improving model faithfulness and performance on open-book QA tasks. Specifically, AutoPASTA integrates iterative question-decomposition prompting and attention steering approaches (Zhang et al., 2024). Given the original context and question, AutoPASTA first prompts an LLM to identify the key information (sentences) through free-text generation. Then, instead of appending those key sentences to the initial prompt, AutoPASTA maps those sentences back to the original context using semantic embeddings (Figure 1 right). By using the original sentences for highlighting, we avoid more lengthy input for the next step, and potentially reduce the unfaithful key sentences generations, mitigating the error propagation. Finally, to guide the model to attend to the selected key sentences, AutoPASTA highlights them through attention steering that upweights their corresponding attention scores at the selected attention heads as done by Zhang et al. (2024). Unlike existing attention steering work, our method does not necessitate human annotation on the highlighting part, rectifying its critical limitation. Additionally, we also design an efficient coarse-to-fine search scheme for identifying effective attention heads for steering, which reduces the searching overheads by 4.5× compared to the greedy method used by previous work (Zhang et al., 2024).

We conduct experiments to evaluate the effectiveness of AutoPASTA using Vicuna-7B (Chiang et al., 2023), LLAMA3-8B-Instruct, and LLAMA3-70B-Instruct (Meta, 2024) on both single- and multi-hop open-book QA tasks from Natural Questions (NQ; Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018b). AutoPASTA consistently provides significant performance improvements over baseline prompting strategies. For example, AutoPASTA achieves an average improvement of 18.28% on exact-match (EM) score over iterative prompting for LLAMA3-8B-Instruct across both tasks. Remarkably, the attention head sets obtained by AutoPASTA exhibit outstanding generalization ability, allowing them to be effectively steered across different tasks.

2 Background

Problem description. In standard LLM prompting, we are given a pre-trained LLM and a text prompt $\boldsymbol{x}$ consisting of n tokens. In the closed-book setting, the prompt $\boldsymbol{x}$ can only be a question or instruction to be completed by models. Relying solely on model parametric knowledge poses challenges in scenarios involving complex questions that entail new knowledge or private information (Zhou et al., 2023; Yu et al., 2024). Existing methods (Shi et al., 2023b; Peng et al., 2023) resort to augmenting the prompt with additional background contexts to facilitate question answering, i.e., open-book question answering. The following box presents a prompt template that we use for open-book QA:

A direct prompt template for open-book QA

Answer the question below, paired with a context that provides background knowledge. Only output the answer without other context words.

Context: {context}

Question: {question}

Answer:

Multi-head attention. A typical transformer model consists of L stacked layers, where each layer contains two submodules: a multi-head attention (MHA) and a fully connected feed-forward network (FFN). Given the input $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, MHA of the layer l performs the attention function in parallel H heads: $\text{MHA}^{(l)}(\boldsymbol{X}) = \text{Concat}(\boldsymbol{H}^{(l,1)}, ..., \boldsymbol{H}^{(l,H)})\boldsymbol{W}_o$ with

$$\boldsymbol{H}^{(l,h)} = \text{Softmax}(\boldsymbol{A}^{(l,h)})\boldsymbol{V}^{(l,h)}$$

where $\boldsymbol{A} = \frac{1}{\sqrt{d_h}}\boldsymbol{Q}\boldsymbol{K}^\top \in \mathbb{R}^{n\times n}$ is the scaled inner product between query $\boldsymbol{Q}$ and key $\boldsymbol{K}$. $\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_{q_h}, \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_{k_h}, \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_{v_h}$ and $\boldsymbol{W}_{q_h}, \boldsymbol{W}_{k_h}, \boldsymbol{W}_{v_h} \in \mathbb{R}^{d\times d_h}$ are learnable projection matrices of head h. d_h is typically set to d/H.

Post-hoc attention steering. Zhang et al. (2024) propose PASTA, an inference-only method that applies attention reweighting to steer model attention towards user-highlighted input sets, thereby improving instruction following and contextual comprehension. Specifically, given the index set of user-specified tokens as $\mathcal{G}$ ($\mathcal{G} \subset [n]$), PASTA highlight these tokens by upweighting their attention scores with a constant attention bias $\boldsymbol{B}^{(l,h)}$:

$$\boldsymbol{H}^{(l,h)} = \text{Softmax}(\boldsymbol{A}^{(l,h)} + \boldsymbol{B}^{(l,h)})\boldsymbol{V}^{(l,h)},$$
$$\boldsymbol{B}_{ij}^{(l,h)} = \begin{cases} -\delta & \text{if } (l,h) \in \mathcal{H} \text{ and } j \notin \mathcal{G} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

where δ is a positive constant. After $\text{Softmax}(\cdot)$, the attention scores of tokens not in $\mathcal{G}$ is scaled down by $\exp(\delta)$. Correspondingly, the others in $\mathcal{G}$ are upweighted due to the renormalization of Softmax[1], steering the model to pay more attention to the input spans of $\mathcal{G}$. Following Zhang et al. (2024), we set $\delta = \log 100$ in all of our experiments. $\mathcal{H}$ is an index set of attention heads selected for steering. Since various heads function diversely, steering different heads yields dramatically different performance. To identify the effective heads, Zhang et al. (2024) employ a greedy search approach that evaluates the steering performance of each head on multiple tasks and selects those with best accuracy. The resulting head set $\mathcal{H}$ can be generalized for steering across different tasks.

PASTA requires the access to user-annotated input spans for highlighting. In the case of context-specific tasks, it is generally prohibitively expensive to extract and annotate relevant sentences from lengthy contexts through humans. To address this critical limitation and improve the contextual faithfulness by automatic explicit highlighting, we introduce our method – AutoPASTA.

3 Method

Our proposed method – Automatic Post-hoc Attention Steering Approach (AutoPASTA), integrates iterative prompting and attention steering. This integration synergistically combines the advantages of both techniques while mitigating their respective limitations. For multi-step iterative prompting, incorporating attention steering externalizes the highlighting of key information through an inference-only operation, efficiently enhancing model faithfulness with improved reliability and controllability. For post-hoc attention steering, equipping it with iterative prompting enables the automatic identification of contextually relevant key information, thereby addressing its significant reliance on human annotations.

[1] (1) is a simplified formula from Equation (2) in Zhang et al. (2024), which we elaborate in Appendix A.

Algorithm 1 AutoPASTA

```
Input  a question $q$, a context $c$, the head set $\mathcal{H}$ of
    an LLM $\mathcal{M}$, prompt templates $\mathcal{P}_i, \mathcal{P}_d$ and $\delta$.
Generate $g_1 = \text{Generate}_{\mathcal{M}}(\mathcal{P}_i(q, c))$;
Calculate $s_k = \text{Match}_e(g_1, \{s_1, \ldots, s_m\})$;
Steer $g_2 = \text{Steer}_{\mathcal{H}, s_k}(\text{Generate}_{\mathcal{M}}(\mathcal{P}_d(q, c)))$;
Output:  The final answer $g_2$
```

3.1 Automatic Contextual Highlighting

In the open-book QA task, an LLM $\mathcal{M}$ is prompted to answer a question $\boldsymbol{q}$ paired with a background context $\boldsymbol{c}$ that consists of m sentences $\boldsymbol{c} = \boldsymbol{s}_1 || \ldots || \boldsymbol{s}_m$. Instead of directly prompting an LLM with $(\boldsymbol{q}, \boldsymbol{c})$, AutoPASTA first prompts the LLM to generate a key sentence from the context $\boldsymbol{c}$ that supports answering the question:

$$\boldsymbol{g}_1 = \text{Generate}_{\mathcal{M}}(\mathcal{P}_i(\boldsymbol{q}, \boldsymbol{c})), \qquad (2)$$

where $\mathcal{P}_i$ is the prompt template of key sentence identification that we show in Section 4.1. Then, AutoPASTA maps $\boldsymbol{g}_1$ back to a sentence from the original context $\boldsymbol{c}$ to avoid potential token-level generation errors in $\boldsymbol{g}_1$ and mitigate error propagation. Specifically, it employs a small encoder e to calculates the semantic embeddings of $\boldsymbol{g}_1$ and every $\boldsymbol{s}_i (1 \le i \le m)$, and pick the best-matching sentence $\boldsymbol{s}_k$ with the highest similarity to $\boldsymbol{g}_1$:

$$\boldsymbol{s}_k = \text{Match}_e\Big(\boldsymbol{g}_1, \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_m\}\Big) \subset \boldsymbol{c}. \qquad (3)$$

In the final step, AutoPASTA steers the attention scores of tokens in $\boldsymbol{s}_k$ based on (1) at the specific attention heads $\mathcal{H}$, when directly prompting the LLM $\mathcal{M}$ to derive the answer for $(\boldsymbol{q}, \boldsymbol{c})$:

$$\boldsymbol{g}_2 = \text{Steer}_{\mathcal{H}, \boldsymbol{s}_k}\Big(\text{Generate}_{\mathcal{M}}\left(\mathcal{P}_d(\boldsymbol{q}, \boldsymbol{c})\right)\Big) \qquad (4)$$

where $\mathcal{P}_d$ is the prompt template of direct answering as shown in Section 2, and $\text{Steer}_{\mathcal{H}, \boldsymbol{s}_k}(\cdot)$ is detailed by (1) with $\mathcal{G}$ as the index set of $\boldsymbol{s}_k$. As such, the identified key sentence $\boldsymbol{s}_k$ is explicitly highlighted through attention score upweighting, directing the model to grasp the key information and generate more faithful answers. Notably, AutoPASTA is applied at inference time and does not require changing any model parameters. More importantly, it does not involve human annotation on highlighted parts. The key information is automatically identified by iterative prompting the model $\mathcal{M}$, addressing the major limitation of existing attention steering approach.

3.2 Coarse-to-fine Model Profiling

AutoPASTA requires carefully selecting $\mathcal{H}$, the set of attention heads to be steered in (1), but finding these heads can be computationally intensive. Zhang et al. (2024) propose a greedy search strategy that evaluates the steering performance of each head on small validation sets of multiple tasks and selects the heads that yield the best performance. This greedy strategy requires evaluating $L \times H$ times, resulting in non-trivial overheads especially for large models. To improve the efficiency of searching heads, we propose an alternative coarse-to-fine model profiling scheme that searches from the layer level to head level. Specifically, we first evaluate the performance of steering all attention heads of one single layer, then pick the top-l layers, and further evaluate the steering performance of each head in these layers. The head set $\mathcal{H}$ is obtained by selecting the best-performing heads from top-l layers. Empirically, we find that a small l (e.g., $l = 6$ compared to $L = 32$) is sufficient for AutoPASTA to achieves superior performance and identify effective attention heads that can generalize across tasks, substantially reducing the searching overheads to $\frac{lH+L}{LH}$.

4 Experiments

We conduct experiments to evaluate the effectiveness of AutoPASTA using Vicuna-7B (Chiang et al., 2023), LLAMA3-8B-Instruct, and LLAMA3-70B-Instruct (Meta, 2024) on both single- and multi-hop open-book QA tasks including Natural Questions (NQ, (Kwiatkowski et al., 2019)) and HotpotQA (Yang et al., 2018b).

4.1 Experimental Setup

Dataset. We study 2 datasets: the MRQA version (Fisch et al., 2019) of Natural Questions (NQ) (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018a). Following the filtering procedures outlined by Yu et al. (2024), duplicated and low-quality questions are removed from the NQ dataset, resulting in 7,189 instances remaining in NQ, and 5,190 instances in HotpotQA. For each dataset, we randomly select 1,000 examples as the profiling set and keep the remaining examples as the test set (see breakdown in Table 5). For all the experiments, we present two evaluation metrics: Exact Match (EM), and Token-level F1 score. We apply greedy search decoding for all experiments.

Task	Prompt	Baseline	AutoPASTA
NQ	Answer the question below, paired with a context that provides background knowledge. Only output the answer without other context words. Context: Although the delegates were divided early on as to whether to break from Crown rule, **the second Continental Congress on July 2, 1776, passed a resolution asserting independence, with no opposing vote recorded**. The Declaration of Independence was issued two days later declaring themselves a new nation: the United States of America. It established a Continental Army, giving command to one of its members, George Washington of Virginia. It waged war with Great Britain, made a militia treaty with France, and funded the war effort with loans and paper money. Question: when did the continental congress vote to adopt the declaration of independence? Answer:	✗ The Continental Congress voted to adopt the Declaration of Independence on July 4, 1776.	✓ July 2, 1776. Label: July 2, 1776.
HotpotQA	Answer the question below, paired with a context that provides background knowledge. Only output the answer without other context words. Context: [1]: Branford, Connecticut - **Branford is a shoreline town located on Long Island Sound in New Haven County, Connecticut, 8 mi east of New Haven**. The population was 28,026 at the 2010 census. [2]: Long Island Sound - **Long Island Sound is a tidal estuary of the Atlantic Ocean, lying between the eastern shores of Bronx County, New York City, southern Westchester County, and Connecticut to the north, and the North Shore of Long Island, to the south**. From east to west, the sound stretches 110 miles (177 km) from the East River in New York City, along the North Shore of Long Island, to Block Island Sound. A mix of freshwater from tributaries and saltwater from the ocean, Long Island Sound is 21 miles (34 km) at its widest point and varies in depth from 65 to. Question: How long is the tidal estuary in which Branford is a shoreline town? Answer:	✗ Long Island Sound.	✓ 110 miles. Label:110 miles.

Table 1: Generation examples of a Vicuna-7B on NQ and HotpotQA. Texts in **bold** are predicted by the model for highlighting and texts in blue are highlighted by AutoPASTA.

Implementation Details. We use PyTorch to implement the evaluation pipeline and all methods (Paszke et al., 2019). Our implementation is based on the publicly available *Huggingface Transformers*[2] (Wolf et al., 2019). All the experiments are conducted on NVIDIA A6000 and A100 GPUs.

AutoPASTA Settings. For AutoPASTA, we utilize the following prompt template $\mathcal{P}_i$ to prompt a LLM $\mathcal{M}$ to identify the key information from the context that support answering the question.

Prompt template $\mathcal{P}_i$ of key sentence identification

A question, and a passage are shown below. Please select the key sentence in the passage that supports to answer the question correctly. Only output the exactly same sentence from the passage without other additional words.

Question: {question}

Passage: {context}

Sentence:

Then, we map the predicted key sentence $\boldsymbol{g}_1$ back to the original context by (3), which uses a small encoder models to calculate the semantic embeddings of the predicted key sentence $\boldsymbol{g}_1$ and every sentence $\boldsymbol{s}_i$ in the context $\boldsymbol{c}$. Specifically, we use a "`all-MiniLM-L6-v2`" model from Sentence-Transformer (Reimers and Gurevych, 2019) as the encoder to encode sentences. Then, we calculate the cosine similarity between semantic embeddings of $\boldsymbol{g}_1$ and each sentence $\boldsymbol{s}_i$ in the context, and select the contextual sentence $\boldsymbol{s}_k$ with the highest similarity score as the final key sentence prediction. For multi-hop question answering, such as HotpotQA, the key sentences are identified for each individual hop separately. Finally, we highlight $\boldsymbol{s}_k$ by (4) while directly prompting the model to answer the question paired by the context with the direct prompting template shown in Section 2.

Coarse-to-fine Model Profiling. For the coarse-to-fine search strategy outlined in Section 3.2, we consider all attention heads in the top-l layers as potential candidates for selection, where l is chosen from $\{3, 4, 5, 6\}$. Subsequently, we either select top-i heads from each individual layer, or top-j heads from the pool of head candidates. Top-i is chosen from $\{4, 6, 8\}$, and top-j is chosen from $\{16, 24, 32, 64\}$. The final head set utilized in the study is determined based on the highest token-F1 performance achieved on the profiling set.

Baselines. We evaluate three open-source LLMs: Vicuna-7B (Chiang et al., 2023), Llama3-8B-Instruct, and Llama3-70B-Instruct under direct prompting, iterative prompting, and direct prompting with AutoPASTA.

- *Direct prompting:* Models are prompted to directly answer the question $\boldsymbol{q}$ based on the provided context $\boldsymbol{c}$. The prompt template $\mathcal{P}_d$ is displayed in Section 2.
- *Iterative Prompting*: Models are first prompted to generate the key sentence that supports answering the question, using the same prompt template $\mathcal{P}_i$. For multi-hop question answering, such as HotpotQA, the key sentences are identified for each individual hop separately. The predicted key sentences are also mapped back to the original context, similar as that in AutoPASTA. Then, the model are prompted to answer the question with the key sentences appended to the context:

[2] https://huggingface.co/

Model	Method	NQ		HotpotQA		All
		EM	Token F1	EM	Token F1	Ave.
Vicuna-7B	Direct Prompting	8.13	33.79	18.11	38.77	24.70
	Iterative Prompting	4.36	31.48	14.04	34.99	21.22
	AutoPASTA$_{\text{out-of-domain generalize}}$	11.78	35.53	21.94	39.92	27.29
	AutoPASTA$_{\text{in-domain profiling}}$	19.77	46.72	29.54	47.51	35.89
LLAMA3-8B	Direct Prompting	8.68	41.55	10.55	49.34	27.53
	Iterative Prompting	13.21	47.28	27.39	62.25	37.53
	AutoPASTA$_{\text{out-of-domain generalize}}$	31.93	52.36	44.49	66.95	48.93
	AutoPASTA$_{\text{in-domain profiling}}$	29.34	51.60	47.82	66.39	48.79
LLAMA3-70B	Direct Prompting	17.33	53.50	31.45	69.49	42.94
	Iterative Prompting	13.97	53.12	17.71	65.49	37.57
	AutoPASTA$_{\text{out-of-domain generalize}}$	35.26	55.97	54.77	73.43	54.86
	AutoPASTA$_{\text{in-domain profiling}}$	34.74	57.65	54.40	71.90	54.67

Table 2: Evaluation results using Vicuna-7B, LLAMA-8B-Instruct, and LLAMA3-70B-Instruct on NQ and HotpotQA. "In-domain" means that the head set is selected based on the profiling set of the target task. "Out-of-domain" means that the head set is selected from the other dataset and the target task is unseen during the profiling.

Prompt Templates of Two-Round Iterative Prompting

[First Round]: A question, and a passage are shown below. Please select the key sentence in the passage that supports to answer the question correctly. Only output the exactly same sentence from the passage without other additional words.

Question: {Question}

Passage: {Evidence}

Sentence:

[Second Round]: Answer the question below, paired with a context that provides background knowledge, and a key sentence. Only output the answer without other context words.

Context: {Evidence}

Key Sentence:{Predicted key sentence}

Question: {Question}

Answer:

4.2 Main Result: AutoPASTA improves open-book QA.

To demonstrate the effectiveness of AutoPASTA, we evaluate its performance on NQ and HotpotQA. Specifically, there are two settings: in-domain and out-of-domain evaluation. In the in-domain setting, we evaluate its performance on a task, using the head set that is selected based on the performance on the profiling set of the same task. Differently, the out-of-domain setting assesses the generalization ability of AutoPASTA, where the head set $\mathcal{H}$ is selected from a different dataset, and the target task is totally unseen during the profiling.

In-domain Evaluation. The results in Table 2 suggest that, for all the models, AutoPASTA significantly improves the model performance compared with other baselines, regardless of model size and datasets. For example, AutoPASTA achieves 47.82% EM for LLAMA3-8B-Instruct on HotpotQA, yielding a significant 20.43% improvement compared to the best-performing baseline. We also observe that iterative prompting can mostly improve upon the direct prompting, showcasing the performance gains from identifying key sentences and appending them to contexts. However, in certain cases, such as Vicuna-7B and LLAMA3-70B-Instruct on HotpotQA, iterative prompting can actually underperform direct prompting. It suggests that highlighting in token space by appending key sentences is insufficient to fully steer a model's attention. In contrast, AutoPASTA shows a consistently substantial improvement over all baselines, demonstrating the effectiveness of automatic attention steering to improve model faithfulness. Table 1 further illustrates this by comparing the generation examples of AutoPASTA and direct prompting.

Out-of-domain Evaluation. In this setting, given an evaluation task (e.g., NQ), we employ the head sets selected from profiling on the profiling set of the other task (e.g., HotpotQA) for AutoPASTA to evaluate its generalization ability across different domains and tasks. The results in Table 2 indicate that AutoPASTA significantly outperforms all baseline methods for all models and all datasets, achieving better or comparable performance to that of in-domain profiling. Notably, for LLAMA3-8B/70B-Instruct on NQ, the cross-domain performance surpasses the in-domain performance, compellingly demonstrating the robustness and generalization proficiency of our approach.

5 Analysis

5.1 Isolating the effect of AutoPASTA's two components

AutoPASTA consists of two primary components: automatic key sentence identification, and explicit highlighting key sentences. To underscore the necessity of both components, we conduct the comparison using LLAMA3-8B-Instruct model between following methods: (i) direct prompting with the original context; (ii) direct prompting with the identified key sentences appended to the context; (iii) highlighting the entire context by attention steering approach but without key-sentence identification; (iv) AutoPASTA that highlights the identified key sentences.

The results in Table 3 indicate that both AutoPASTA and direct prompting can benefit from using the identified key sentence, yielding significant performance gains. Specifically, highlighting the entire context via attention steering can improve upon direct prompting but underperforms AutoPASTA, suggesting the importance of key sentence identification. Meanwhile, the comparison between (ii) and (iv) illustrates the performance gains yielded by explicitly highlighting via attention steering. Therefore, these results suggest that both components are essential for AutoPASTA to achieve its best performance.

Method	EM	Token F1
Direct prompting	10.55	49.34
Direct prompting w. key sentences	27.39	62.25
Highlight the entire context	36.00	60.19
Highlight identified key sentences	47.82	66.39

Table 3: Performance of LLAMA3-8B-Instruct on HotpotQA when highlighting different parts of contexts.

5.2 Comparison between profiling strategies

To illustrate the effectiveness of the coarse-to-fine profiling strategy introduced in Section 3.2, we evaluate several different profiling approaches as follows:

• Greedy search proposed by (Zhang et al., 2024): This strategy involves selecting the top-k heads from all the attention heads in the models. The evaluation times for this strategy is $L \times H$.

• Group search inspired by (Ainslie et al., 2023): Here, 8 adjacent heads from one layer form a group. Then, we evaluate them group-wise, and select the top-k head groups. The evaluate times for this strategy is $LH/8$.

• Coarse-to-fine search: This strategy initially selects the top-l layers and then chooses the head set only from the heads within these layers. The evaluation times for this strategy is $L + lH$.

where L is the number of layers, and H is the number of attention heads per layer. We compare them with a Vicuna-7B (Chiang et al., 2023) that has 32 layers, and 32 heads per layer. The results in Table 4 show that coarse-to-fine profiling significantly outperforms all the other strategies while reducing the total evaluation times by $4.5\times$ compared to the original greedy search in (Zhang et al., 2024).

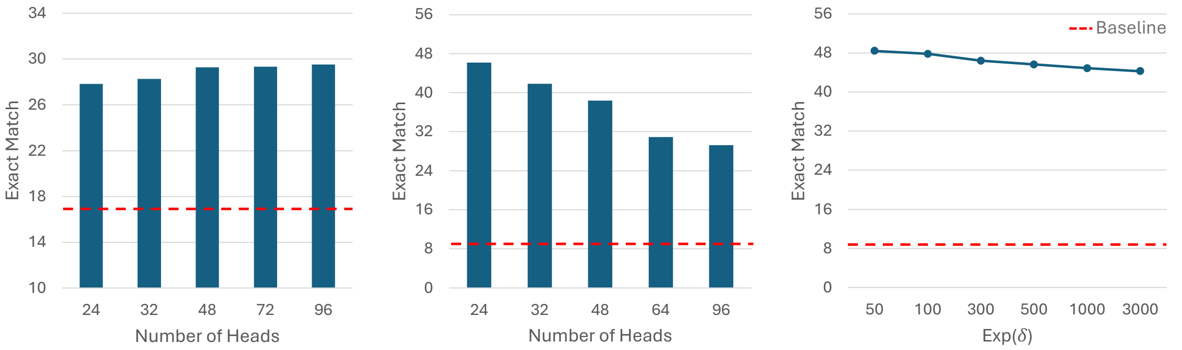
Method	# Eval	EM	Token F1
Baseline	N.A.	8.13	33.79
Greedy search all heads	1,024	14.81	35.63
Group search (size of 8)	128	12.12	36.13
Coarse-to-fine search	224	19.77	46.72

Table 4: Performance of AutoPASTA on NQ with Vicuna-7B when searching effective attention heads with different strategies. "# Eval" refers to the total evaluations with the profiling set.

5.3 Ablation study

We conduct ablation study to discuss the performance of AutoPASTA given different number of attention heads for steering and different δ.

Varying the number of steered heads. Figure 2a presents the performance variation of AutoPASTA with Vicuna-7B on HotpotQA dataset when steering different number of attention heads. Figure 2b illustrates the EM results for LLAMA3-8B-Instruct on the HotpotQA dataset under similar conditions. We see that steering more heads for AutoPASTA may result in slight performance degeneration, for example, the performance of LLAMA3-8B-Instruct on HotpotQA. This observation is similar to findings in previous work (see Figure 3 in Zhang et al. (2024)), where overemphasizing too many heads can lead models to focus on solely on highlighted information while ignoring other parts, potentially degenerating performance. In practice, we recommend applying AutoPASTA to steer a moderate number of heads. The optimal number of steered heads in our study is determined based on the performance metrics on the profiling data.



(a) Vicuna-7B on HotpotQA (b) LLAMA3-8B-Instruct on HotpotQA (c) LLAMA3-8B-Instruct on HotpotQA

Figure 2: Ablation study of AutoPASTA performance when steering different numbers of heads (2a and 2b) and setting different δ (2c). Dashed line in red refers to the baseline performance of direct prompting.

The sensitivity about δ. Figure 2c presents the sensitivity analysis for varying δ in (1) using LLAMA3-8B-Instruct on HotpotQA. We can see that the performance of AutoPASTA is not sensitive to the attention bias constant δ. Changing its logarithm values (i.e., the scaling-down coefficient for non-highlighted tokens as elobrated in Appendix A) from 50 to 3000 does not induce dramatic performance variation. Therefore, we set δ as its default value $\log(100)$, which is the same as Zhang et al. (2024).

6 Related Work

Large language models exhibit remarkable performance on (context-free) knowledge-intensive tasks, such as open-domain question answering (QA) (Kwiatkowski et al., 2019) and commonsense reasoning (Mihaylov et al., 2018; Clark et al., 2018), indicating that they encode substantial knowledge about open-world facts (Zhou et al., 2023) in their parameters. Despite their proficiency in memorization, different kinds of hallucinations in the output are observed, including factual knowledge hallucination (Huang et al., 2023; Yu et al., 2024), hallucination in summarization (Maynez et al., 2020; Pagnoni et al., 2021), hallucination in logical operations (Lyu et al., 2023; Huang et al., 2023). In this work, we focus on the factual knowledge hallucination due to models' unawareness of relevant knowledge or overlooking contextual information.

Retrieval-augmented LLMs. To address the problem of missing relevant knowledge, one popular method is to use retrieval-augmented LMs that supplement missing knowledge from external sources (Shi et al., 2023b; Peng et al., 2023). Retrieval augmentation requires that LLMs are sensitive to the input context and generate responses that are faithful. However, recent work shows that even if the relevant knowledge is presented, the model may still not be faithful to the given evidence (Zhou et al., 2023; Yu et al., 2024; Wan et al., 2024).

Prompt-based strategies. To improve the faithfulness of the models, various prompting strategies are designed to guide the model to detect the key information (Wei et al., 2022; Radhakrishnan et al., 2023), or focus on the given evidence (Zhou et al., 2023), while these extracted key information is only added as additional tokens in the input, and models may still not be faithful to these new tokens.

Model-based strategies. Besides using prompting to improve the faithfulness, Köksal et al. (2023) constructs counterfactual evidence to finetune models, and Shi et al. (2023a) proposes a context-aware decoding method to downweight the output distribution associated with the model's prior knowledge.

To the best of our knowledge, we are the first work to integrate key information prompting and explicit token highlighting during the inference without any additional training.

7 Conclusion

In this paper, we address the challenge of contextual faithfulness in open-book QA tasks and introduce AutoPASTA, an inference-only method that automatically identifies crucial information pieces within contexts and explicitly highlights them through steering a model's attention scores. AutoPASTA guides the model to focus on the essential information within contexts, leading to substantially improved model faithfulness and performance. Remarkably, by integrate iterative prompting and attention steering techniques, AutoPASTA synergistically combines their advantages while mitigating their respective limitations.

Limitations

First, while this study primarily examines the question answering scenario with passages of gold evidences provided, it is acknowledged that practical applications may present multiple passages, potentially enhancing retrieval recall. However, the performance of the proposed method in the absence of guaranteed gold evidence remains to be empirically validated. It is anticipated that our algorithm could still perform reasonably well when confronted with additional passages, though the exact impact of irrelevant or conflicting information requires further investigation.

Secondly, the efficacy of our algorithm is influenced by the accuracy of key sentence selection. While the mapping-back method offers a means to address certain propagation errors that may occur during intermediate stages, it is predicated on the assumption that the predicted key sentence closely aligns with the actual correct sentence. Future research endeavors may focus on refining techniques for key sentence prediction, potentially enhancing overall performance.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901.

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

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of 2nd Machine Reading for Reading Comprehension (MRQA) Workshop at EMNLP*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Jungo Kasai, Keisuke Sakaguchi, yoichi takahashi, Ronan Le Bras, Akari Asai, Xinyan Velocity Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. 2023. Realtime QA: What's the answer right now? In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Abdullatif Köksal, Renat Aksitov, and Chung-Ching Chang. 2023. Hallucination augmented recitations for language models. *arXiv preprint arXiv:2311.07424*.

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. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. 2023. Large language models with controllable working memory. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1774–1793, Toronto, Canada. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. *arXiv preprint arXiv:2301.13379*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *Preprint*, arXiv:1809.02789.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. *arXiv preprint arXiv:2104.13346*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *Preprint*, arXiv:2302.12813.

Ansh Radhakrishnan, Karina Nguyen, Anna Chen, Carol Chen, Carson Denison, Danny Hernandez, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Sam McCandlish, Sheer El Showk, Tamera Lanham, Tim Maxwell, Venkatesa Chandrasekaran, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. 2023. Question decomposition improves the faithfulness of model-generated reasoning. *Preprint*, arXiv:2307.11768.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Wen-tau Yih. 2023a. Trusting your evidence: Hallucinate less with context-aware decoding. *arXiv preprint arXiv:2305.14739*.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2023b. Replug: Retrieval-augmented black-box language models. *Preprint*, arXiv:2301.12652.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Lee Boyd-Graber, and Lijuan Wang. 2023. Prompting GPT-3 to be reliable. In *The Eleventh International Conference on Learning Representations*.

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

Alexander Wan, Eric Wallace, and Dan Klein. 2024. What evidence do language models find convincing? *arXiv preprint arXiv:2402.11782*.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018a. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018b. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xiaodong Yu, Hao Cheng, Xiaodong Liu, Dan Roth, and Jianfeng Gao. 2024. Reeval: Automatic hallucination evaluation for retrieval-augmented large language models via transferable adversarial attacks. *Preprint*, arXiv:2310.12516.

Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. 2024. Tell your model where to attend: Post-hoc attention steering for LLMs. In *The Twelfth International Conference on Learning Representations*.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14544–14556, Singapore. Association for Computational Linguistics.

A Derivation for Equation 1

In this section, we present the derivation to show why (1) is equivalent to equation (2) in Zhang et al. (2024).

For the token that are not highlighted $j \notin \mathcal{G}$, Zhang et al. (2024) downweight their attention scores by scaling down their scores post-softmax by a coefficient α ($0 \le \alpha \le 1$): $\alpha \cdot \text{Softmax}(\boldsymbol{A}_{i\cdot})_j / C_i$ where $C_i = \sum_{j \in \mathcal{G}} \text{Softmax}(\boldsymbol{A}_{i\cdot})_j + \sum_{j \notin \mathcal{G}} \alpha \cdot \text{Softmax}(\boldsymbol{A}_{i\cdot})_j$. Now we show that:

$$\alpha \cdot \text{Softmax}(\boldsymbol{A}_{i\cdot})_j / C_i = \frac{\alpha}{C_i} \frac{\exp(\boldsymbol{A}_{ij})}{\sum_{j'} \exp(\boldsymbol{A}_{ij'})} \tag{5}$$

$$= \frac{\exp(\boldsymbol{A}_{ij} + \log(\alpha))}{C_i \sum_{j'} \exp(\boldsymbol{A}_{ij'})} \tag{6}$$

For the tokens in $\mathcal{G}$:

$$\text{Softmax}(\boldsymbol{A}_{i\cdot})_j / C_i = \frac{\exp(\boldsymbol{A}_{ij})}{C_i \sum_{j'} \exp(\boldsymbol{A}_{ij'})} \tag{7}$$

Therefore, after the renormalization, it is equivalent to condut the Softmax among $\boldsymbol{A}_{ij} + \log(\alpha)$ for $j \notin \mathcal{G}$ and $\boldsymbol{A}_{ij}$ for $j \in \mathcal{G}$, which is our simplified equation in (1).

B Evaluation Details

B.1 Dataset Statistics

	Profiling	Test
Natural Questions	1,000	6,189
HotpotQA	1,000	4,190

Table 5: Natural Questions and HotpotQA data statistics after the preprocessing.

B.2 The detailed number of attention heads for steering

Model	**NQ**	**HotpotQA**
Vicuna-7B	top 64 heads from top 4 layers	top 96 heads from top 6 layers
LLAMA3-8B	top24 heads, 4 from each of top 6 layers	top24 heads, 4 from each of top 6 layers
LLAMA3-70B	top20 heads, 4 from each of top 5 layers	top 64 heads from top 5 layers

Table 6: The detailed number of attention heads for steering