SINKTRACK: ATTENTION SINK BASED CONTEXT ANCHORING FOR LARGE LANGUAGE MODELS

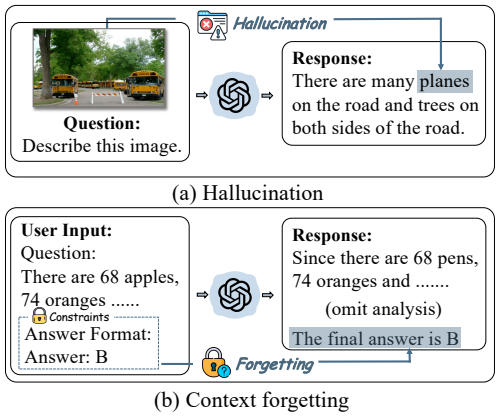
Xu Liu, Guikun Chen, Wenguan Wang†
The State Key Lab of Brain-Machine Intelligence, Zhejiang University

ABSTRACT

Large language models (LLMs) suffer from hallucination and context forgetting. Prior studies suggest that *attention drift* is a primary cause of these problems, where LLMs' focus shifts towards newly generated tokens and away from the initial input context. To counteract this, we make use of a related, intrinsic characteristic of LLMs: *attention sink* – the tendency to consistently allocate high attention to the very first token (*i.e.*, ⟨BOS⟩) of a sequence. Concretely, we propose an advanced context anchoring method, SINKTRACK, which treats ⟨BOS⟩ as an information anchor and injects key contextual features (such as those derived from the input image or instruction) into its representation. As such, LLM remains anchored to the initial input context throughout the entire generation process. SINKTRACK is *training-free*, *plug-and-play*, and introduces *negligible inference overhead*. Experiments demonstrate that SINKTRACK mitigates hallucination and context forgetting across both textual (*e.g.*, **+21.6**% on SQuAD2.0 with *Llama3.1-8B-Instruct*) and multi-modal (*e.g.*, **+22.8**% on M3CoT with *Qwen2.5-VL-7B-Instruct*) tasks. Its consistent gains across different architectures and scales underscore the robustness and generalizability. We also analyze its underlying working mechanism from the perspective of information delivery. Our source code is available at GitHub.

1 INTRODUCTION

Background. Large language models (LLMs) have demonstrated impressive capabilities in textual and multi-modal understanding (Bevilacqua et al., 2025; Xu et al., 2024; Guan et al., 2024; Kalyuzhnaya et al., 2025; Dong et al., 2023). However, their deployment in critical applications is constrained by two hard, long-standing problems: *hallucination* (Chang et al., 2024; Zhang et al., 2025b; Rawte et al., 2023) and *context forgetting* (Liu et al., 2024a; Ling et al., 2023; Li et al., 2025). Hallucination is the generation of content factually inconsistent with the provided input. For instance, when asked to describe an image of buses on the road, LLM might identify "planes" that do not exist in the image (*cf.* Fig. 1a). Context forgetting occurs during long-form tasks, where LLM may lose track of initial instructions, resulting in unexpected answers. In a multi-step reasoning dialogue, LLM might lose track of the initial instruction on answer formation (*cf.* Fig. 1b).



(a) Hallucination

(b) Context forgetting

Figure 1: Hallucination and context forgetting are pervasive obstacles in LLMs.

Motivation. Prior studies (Peysakhovich & Lerer, 2023; An et al., 2025; Liu et al., 2024a) identify *attention drift* as a direct cause of hallucination and context forgetting (*cf.* Fig. 2). As LLM generates a sequence, its attention progressively shifts towards newly created tokens, while attention on the initial tokens – which often contain the most critical input information, such as image

† Corresponding Author: Wenguan Wang.

features or user instructions – dramatically decays. Interestingly, we observe that this drift coexists with a countervailing phenomenon: the *attention sink* (Xiao et al., 2024; Kaul et al., 2025). While attention on most initial tokens fades, the very first token (*i.e.*, ⟨BOS⟩) intrinsically maintains a high degree of attention throughout generation, even though it is semantically sparse (*cf.* 1_{st} column in Fig. 2). These two patterns, one of decay and one of stability, present a synergistic opportunity. The stability of the attention sink provides a natural mechanism to counteract the information loss caused by attention drift. Based on this insight, our core idea is to transform the passive attention sink into an active mechanism for context anchoring (*i.e.*, using ⟨BOS⟩ as an information anchor), thereby mitigating the information decay caused by attention drift. Through injecting key contextual features (such as those derived from the input image or instruction) into the representation of ⟨BOS⟩, LLM remains anchored to the initial input context throughout generation.

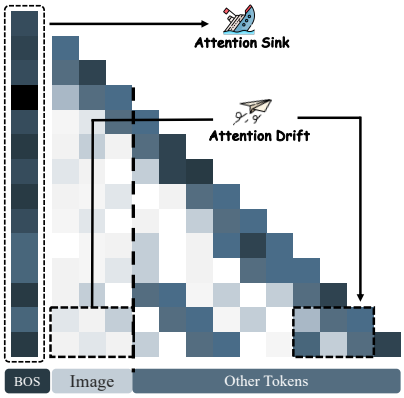


Figure 2: Attention drift and sink (the darker the color, the greater the weight).

Methodology. To implement the above idea, *i.e.*, using ⟨BOS⟩ as an information anchor, we make an iterative exploration. The exploration begins with a `hard injection` (§2.2), which directly replaces the Value vector of ⟨BOS⟩ in the KV cache with a mean-pooled information vector (*e.g.*, from image or instruction features). This implementation leads to a collapse, revealing that LLM's original computational flow should be preserved. We then investigate a `soft injection` (§2.3), which blends the pooled vector into LLMs using a weighted sum to respect the original flow. Despite effective, there are concerns regarding its robustness: it relies on manually tuned hyperparameters and injects all pooled representations into ⟨BOS⟩ *equally* – which might add noise and hurt performance on long contexts. Motivated by these limitations, our final solution is SINKTRACK (§2.4), which is built upon a dual-track cross-attention mechanism. In this design, one track enables ⟨BOS⟩ to adaptively query and retrieve relevant information from the context (*i.e.*, cross-attention), while a parallel self-attention track preserves LLM's native computational flow, leading to consistent performance gains.

Merits. Through our study of anchor formulation, injection strategies, and source integrity, we design SINKTRACK as a lightweight, adaptive enhancement with three benefits: First, it is *plug-and-play*. As a training-free, inference-time enhancement, SINKTRACK can be applied to any pre-trained LLM without the risks of fine-tuning, *e.g.*, catastrophic forgetting and model collapse. Second, it is *efficient*. Its core intervention is a one-off, lightweight operation performed prior to generation. Since the intervention can be stored in KV cache, SINKTRACK adds negligible computational overhead to the subsequent auto-regressive steps, thereby preserving LLMs' original inference speed. Third, it is *generalizable*. SINKTRACK makes use of *attention sink*, a common, intrinsic characteristic of decoder-only LLMs and multi-modal LLMs (MLLMs), to mitigate hallucination and context forgetting. This principle design makes it inherently generalizable across architectures, scales, and modalities, as validated in experiments (§3) on different LLMs and MLLMs from 3B to 12B sizes.

Results and Analysis. SINKTRACK demonstrates consistent performance improvements over both direct prompting and chain-of-thought (`CoT`) reasoning across various multi-modal and text-based benchmarks (§3.2). For instance, on the multi-modal M3CoT benchmark (Chen et al., 2024b), SINKTRACK increases the accuracy of *Qwen2.5-VL-7B* by **22.8**% over `CoT`, and on the reading comprehension benchmark, SQuAD2.0 (Rajpurkar et al., 2018), it improves *Llama3.1-8B*'s accuracy by **21.6**%. Qualitative analysis reveals that SINKTRACK enhances the logical structure and detail focus in LLM's reasoning process (§3.3). We also explore the underlying working mechanism of SINKTRACK (§B).

2 METHODOLOGY

We first review the phenomena of *attention drift* and *attention sink* that motivate our method (§2.1). Then, we present the exploration undertaken to use *attention sink* as a dynamic information anchor, beginning with an initial `hard injection` (§2.2), progressing to a `soft injection` (§2.3), and leading to the development of SINKTRACK, our core *dual-track* mechanism (§2.4).

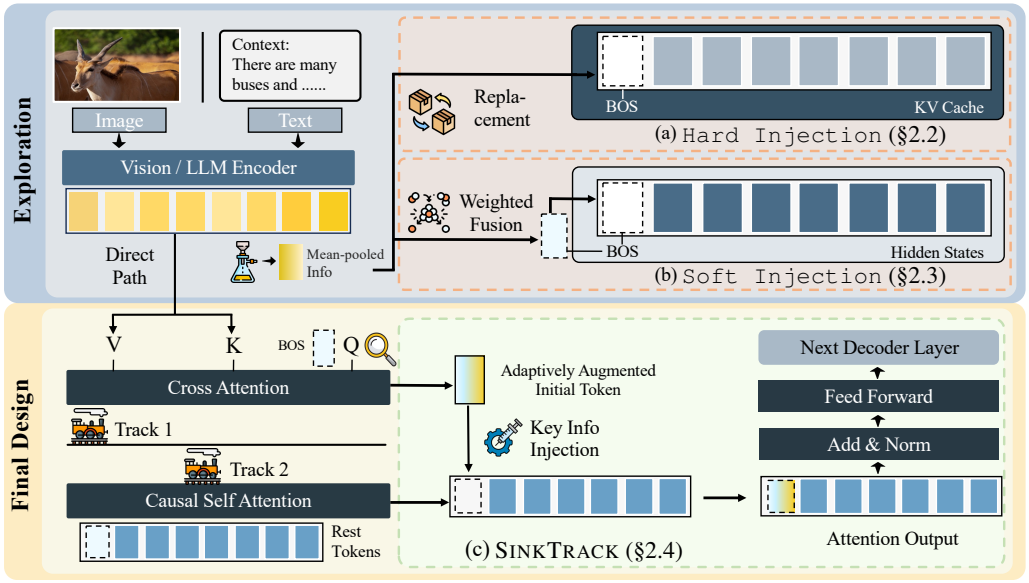


Figure 3: An overview of our methodology exploration.

2.1 Attention Patterns of LLMs

During auto-regressive generation, LLMs exhibit two fundamental and co-existing attention patterns that motivate our approach. The first is *attention drift* (Peysakhovich & Lerer, 2023; An et al., 2025; Liu et al., 2024a) (*cf.* Fig. 2), where LLMs' attention progressively concentrates on more recent tokens. As the generation proceeds, attention on the initial tokens, which contain the primary context (*e.g.*, instructions or image), dramatically decays. This progressive loss of focus on the original input is a primary cause of hallucination and context forgetting. The second pattern is *attention sink* (Xiao et al., 2024; Kaul et al., 2025). (*cf.* 1_{st} column in Fig. 2). In direct contrast to the general decay of attention on initial tokens, the very first token of the sequence (*i.e.*, ⟨BOS⟩) intrinsically and consistently maintains an exceptionally high degree of attention throughout the generation process. ⟨BOS⟩ is semantically sparse, yet it functions as LLMs' most stable point of attention.

These two patterns, one of decay and one of stability, operate in tandem. Surprisingly, while these phenomena have been studied independently, their synergistic potential has been overlooked. This presents a clear opportunity: rather than attempting to alter these inherent behaviors, this work, to our best knowledge, is the first to leverage their interplay. Our core idea is to repurpose the passive attention sink into an active mechanism to counteract drift. Detailed implementations are given next.

2.2 Hard Injection via Direct Replacement

This work targets **decoder-only LLM**, which is composed of L Transformer decoder blocks stacked sequentially. The l-th block processes the sequence of hidden states from the preceding block, $\boldsymbol{H}^{(l-1)} \in \mathbb{R}^{n \times d}$, where n is the sequence length and d is the hidden dimension. Each decoder block contains two main sub-layers. The first is a multi-head attention (MHA) sub-layer:

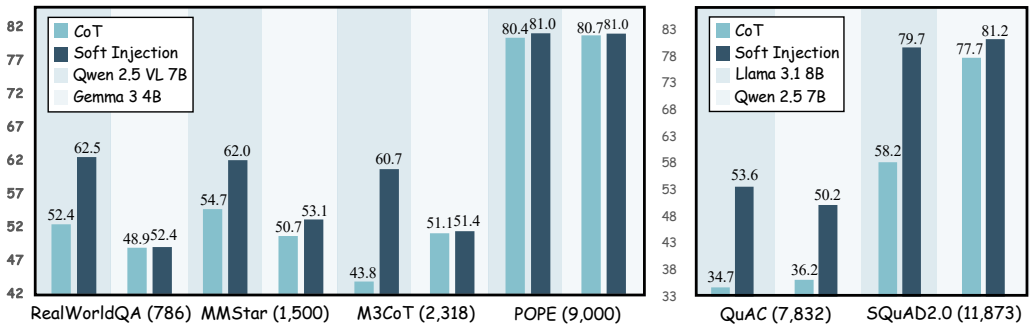
$$\boldsymbol{M}^{(l)} = \text{LayerNorm}\big(\boldsymbol{H}^{(l-1)} + \text{MHA}(Q = \boldsymbol{H}^{(l-1)},\ K = \boldsymbol{H}^{(l-1)},\ V = \boldsymbol{H}^{(l-1)})\big). \tag{1}$$

The second sub-layer is a feed-forward network (FFN):

$$\boldsymbol{H}^{(l)} = \text{LayerNorm}(\boldsymbol{M}^{(l)} + \text{FFN}(\boldsymbol{M}^{(l)})). \tag{2}$$

Implementation. Based on the characteristic of attention sink, the most direct and simple method to implement our idea is to place information at the representation of ⟨BOS⟩. Therefore, we first attempt `hard injection` (*cf.* Fig. 3a), which directly manipulates the embedding of ⟨BOS⟩. Concretely, we directly replace the Value vector (V in Eq. 1) corresponding to ⟨BOS⟩ in the KV cache with an external information vector $\boldsymbol{f}_{\text{info}}$. $\boldsymbol{f}_{\text{info}}$ is derived from the input context. For the textual task, it is the prompt embeddings; for the multi-modal task, it is the visual features from the MLLM's own vision encoder. It is then compressed via mean pooling to match the dimensionality.

Results and Analysis. Unfortunately, `hard injection` leads to a complete model collapse, yielding meaningless outputs. This indicates that LLM's internal computational pathways and



(a) Multi-modal Tasks (b) Text-based Tasks

Figure 4: `Soft injection` achieves considerable performance improvements (§2.3).

learned attention patterns are sensitive to direct manipulation. Abruptly overwriting values in the KV cache disrupts this finely-tuned structure, leading to a severe degradation. This finding establishes a core principle: any effective information injection must preserve the integrity of the LLM's native computational flow, motivating our development of the more nuanced `soft injection`.

2.3 Soft Injection via Static Fusion

Implementation. In contrast to the disruptive replacement used in `hard in jection`, `soft injection` fuses information into LLMs' computational flow. This is achieved by fusing a mean-pooled information vector with the original hidden state of ⟨BOS⟩ via weighted sum (*cf.* Fig. 3b):

$$\bar{\boldsymbol{H}}_0^{(l)} = \alpha \cdot \boldsymbol{H}_0^{(l)} + (1-\alpha) \cdot \boldsymbol{f}_{\text{info}}, \tag{3}$$

where $\boldsymbol{H}_0^{(l)}$ is the original hidden state of the first token ⟨BOS⟩ at injection layer l, which corresponds to the output of the FFN sub-layer (Eq. 2). $\boldsymbol{f}_{\text{info}} \in \mathbb{R}^d$ is the mean-pooled information vector, α is the injection strength, and $\bar{\boldsymbol{H}}_0^{(l)}$ is the updated hidden state after `soft injection`. Such an injection strategy holds two main hyperparameters: the injection layers and the injection strength α.

Results and Analysis. We evaluate `soft injection` in four multi-modal and two text-based tasks (§3.1). While soft injection achieves considerable gains (*cf.* Fig. 4) and validates our core premise, its static design reveals fundamental limitations. Its performance is contingent on a manually tuned hyperparameter (*i.e.*, injection strength α), restricting its applicability. Furthermore, by uniformly aggregating all contextual information, it lacks the capacity to adaptively select the most salient content. This indiscriminate fusion can introduce noise and even degrade performance, highlighting the need for a dynamic mechanism that can adaptively query the source context.

2.4 SinkTrack: Dual-Track Cross-Attention Mechanism

To overcome the limitations of soft injection, we introduce SinkTrack, an adaptive mechanism centered on a dual-track attention design. SinkTrack achieves dynamic information infusion by separating the attention computation for ⟨BOS⟩ from that of all subsequent tokens (*cf.* Fig. 3c). This separation allows SinkTrack to inject context adaptively while preserving LLM's native computational flow.

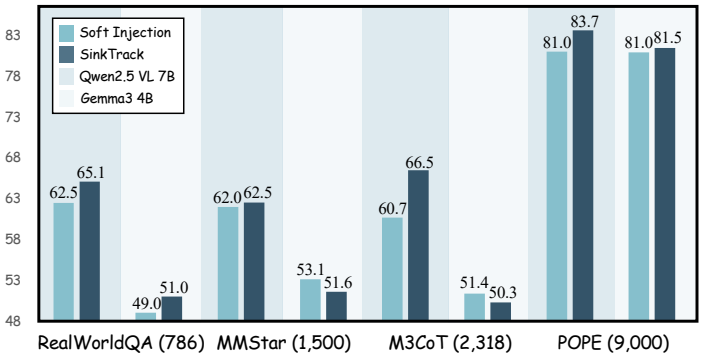


Figure 5: SinkTrack further improves performance.

Track 1: Adaptive Injection via Cross-Attention. The first track performs adaptive injection exclusively for ⟨BOS⟩. This is achieved by modifying MHA (Eq. 1) at designated injection layers. Concretely, the hidden state of ⟨BOS⟩ at injection layer l, $\boldsymbol{H}_0^{(l)}$, serves as the Query (Q), while the Key (K) and Value (V) are now derived from the external, mean-pooled information vector ($\boldsymbol{f}_{\text{info}}$).

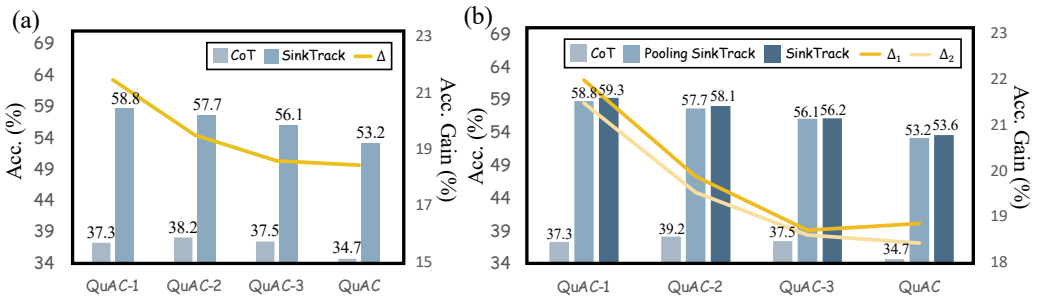


Figure 6: Effect of mean-pooling in information injection (Eq. 4). (a) Using the mean-pooled information vector; (b) using the original (unpooled) one.

This is implemented by cross-attention:

$$\bar{\boldsymbol{H}}_0^{(l)} = \text{MHA}(Q = \boldsymbol{H}_0^{(l)},\ K = \boldsymbol{f}_{\text{info}},\ V = \boldsymbol{f}_{\text{info}}), \tag{4}$$

where $\bar{\boldsymbol{H}}_0^{(l)}$ denotes the updated hidden state after cross-attention. While the cross-attention is designed for adaptive injection, one may wonder: how can the low-information query vector $\boldsymbol{H}_0^{(l)}$ retrieve information adaptively? We emphasize that this adaptive capability is a layer-wise information accumulation. At the initial injection layer, $\boldsymbol{H}_0^{(l)}$ aggregates a holistic representation of the global context from $\boldsymbol{f}_{\text{info}}$. This aggregation infuses $\boldsymbol{H}_0^{(l)}$ with context, thereby enabling it to formulate targeted queries in subsequent injection layers. This progression allows $\langle$BOS$\rangle$ to identify salient information dynamically, thus obviating the need for a manually tuned injection strength.

Track 2: Preserving Native Computation via Self-Attention. The second track ensures that LLM's computational flow remains undisturbed. All other tokens (*i.e.*, regular tokens) perform the standard causal self-attention computation, attending to the original, unmodified input sequence.

Sequence Reassembly. The two tracks operate in parallel within the same attention block. The final output is a concatenation of the updated $\langle$BOS$\rangle$ representation from Track 1 and the regular token representations from Track 2. This dual-track mechanism allows SINKTRACK to function as an adaptive context anchor, retrieving necessary information without compromising the foundational structure of LLMs. The superiority of this design is demonstrated by its consistent performance gains on long-context benchmarks like QuAC (*cf.* Fig. 5).

The Effect of Mean-Pooling. Despite outperforming `CoT`, we observe that SINKTRACK's performance gains on the long-dialogue QuAC dataset diminish as the context length grows (*cf.* Fig.6a). We identify the cause as the mean-pooling operation, a step inherited from `hard` and `soft injection`, which inevitably causes information loss by compressing a variable-length sequence into a fixed-size vector – a problem that intensifies with longer contexts. However, the cross-attention mechanism in SINKTRACK offers a structural advantage, as it does not require sequence length alignment between the Query and the Key/Value. We therefore eliminate the lossy pooling step, instead using the complete, uncompressed sequence of contextual features as the source. This modification effectively mitigates the trend of diminishing gains and preserves performance on long-context tasks (*cf.* Fig.6b). The complete process is described in Alg. S1 to ease understanding.

3 EXPERIMENT

3.1 EXPERIMENTAL SETTING

Datasets. We select four multi-modal datasets and two text-based datasets to assess the performance of SINKTRACK for different modalities.

• **Multi-modal Datasets.** Four multi-modal datasets are used for evaluation: **RealWorldQA** (X.AI, 2024), a visual question answering (QA) dataset designed to assess model robustness in complex, real-world scenarios; **MMStar** (Chen et al., 2024a), a comprehensive benchmark for assessing multi-modal capabilities that encompasses 6 core abilities across 18 main tasks; **M3CoT** (Chen et al., 2024b), a task collection focused on multi-modal, multi-hop, complex chain-of-thought reasoning; and **POPE** (Li et al., 2023b), a benchmark specifically designed to quantify object-level hallucination in multi-modal models.

Model	RealWorldQA (768)		MMStar (1,500)		M3CoT (2,318)		POPE (9,000)		Average	
	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1
Qwen2.5-VL-3B-Instruct (Bai et al., 2025)										
Direct	31.15±1.63	23.25±4.36	24.89±1.65	39.54±1.90	33.43±0.54	44.20±3.08	53.23±0.24	69.48±0.20	35.68	44.12
CoT	35.69±1.61	26.46±4.71	35.24±0.83	50.68±0.95	20.69±0.51	32.04±0.62	64.58±0.22	78.10±0.16	39.05	46.82
SINKTRACK	**48.28**±0.40	**28.42**±0.22	**47.24**±0.40	**61.38**±0.44	**48.25**±0.09	**58.73**±1.14	**77.69**±0.12	**87.44**±0.07	**55.37**	**58.99**
Gemma3-4B-Instruct (Team et al., 2025)										
Direct	51.33±0.06	**48.43**±1.10	35.47±0.20	52.17±0.12	27.71±0.20	33.81±0.62	83.98±0.04	91.30±0.02	49.62	56.43
CoT	49.41±1.02	44.53±5.27	51.36±1.53	67.52±1.32	50.96±0.32	**59.71**±1.03	82.01±0.38	90.04±0.24	58.44	65.45
SINKTRACK	**52.46**±0.43	47.39±1.59	**53.04**±0.53	**69.11**±0.48	**51.01**±0.78	59.18±1.20	**84.33**±0.02	**91.50**±0.01	**60.21**	**66.80**
Qwen2.5-VL-7B-Instruct (Bai et al., 2025)										
Direct	47.49±0.22	39.93±4.33	37.29±0.80	54.24±0.95	39.20±0.29	51.01±0.55	78.21±0.27	87.15±0.20	50.55	58.08
CoT	57.86±1.29	43.00±4.14	55.84±0.74	71.29±0.73	44.11±0.56	57.96±0.96	83.65±0.08	90.63±0.06	60.37	65.72
SINKTRACK	**65.49**±0.37	**52.81**±0.68	**63.78**±0.19	**77.86**±0.17	**66.94**±0.21	**79.13**±0.23	**85.47**±0.02	**91.67**±0.01	**70.42**	**75.37**
Gemma3-12B-Instruct (Team et al., 2025)										
Direct	57.34±0.44	50.96±3.74	44.58±0.55	60.89±0.59	41.90±0.40	51.46±0.87	84.59±0.07	91.65±0.04	57.10	63.74
CoT	57.69±0.25	49.92±2.00	60.56±0.91	75.09±0.67	60.32±0.30	**72.24**±0.83	84.53±0.33	91.61±0.19	65.78	72.22
SINKTRACK	**57.91**±0.33	**52.54**±0.77	**61.60**±0.46	**75.81**±0.51	**60.57**±0.12	70.97±0.67	**85.40**±0.13	**92.13**±0.05	**66.37**	**72.86**

Table 1: Performance of MLLMs on multi-modal tasks in terms of Accuracy (%) and Macro-F1. The **bold** indicates the best performance for the base LLM.

• **Text-based Datasets.** Two text-based datasets are used for evaluation: **QuAC** (Choi et al., 2018), a conversational QA benchmark, to measure the model's ability to handle long-context dependencies. Its structure, where questions build upon the dialogue history, allows us to analyze performance across progressive dialogue turns (QuAC-1, -2, -3) as the context lengthens, thereby assessing the mitigation of context forgetting; and **SQuAD2.0** (Rajpurkar et al., 2018), another QA benchmark to test LLMs' ability to abstain from answering when information is absent in the context, providing a direct measure of its capacity to avoid generating fabricated content.

Base LLMs. We validate the effectiveness and generalizability of SINKTRACK across mainstream, open-source LLMs and MLLMs. Specifically, *Qwen2.5* (Qwen et al., 2025), *MiniCPM3* (Hu et al., 2024), and *Llama3.1* (Grattafiori et al., 2024) are used for text-based tasks, while *Qwen2.5-VL* (Bai et al., 2025) and *Gemma3* (Team et al., 2025) are used for multi-modal tasks.

Competitors. Since SINKTRACK is training-free and plug-and-play, `Direct` queries with LLMs, and Chain-of-Thought (Wei et al., 2022) (`CoT`) are used as our primary competitors. Specifically, CoT guides the model to perform step-by-step reasoning by including instructions, "Let's think step by step", in the prompt, without applying any additional information injection mechanism. This comparison allows for a clear isolation of the performance gains attributable to SINKTRACK.

Implementation Details. We adopt an intermittent injection strategy for both `soft injection` and SINKTRACK. Specifically, the injection operation is applied at intervals of every 5 layers. This configuration is empirically determined based on our analysis (§3.4.1), where we observe that such intermittent injection yields better performance compared to dense injection.

3.2 QUANTITATIVE COMPARISON RESULT

To ensure reliability and rule out the influence of generation randomness, we conduct all evaluations using three distinct random seeds. Consequently, we report the performance as the mean ± variance across these independent runs. The quantitative results are presented in Table 1 for multi-modal tasks and Table 2 for text-based tasks. The findings show that SINKTRACK consistently outperforms both `Direct` and `CoT` by a clear margin across all tested LLMs, modalities, and benchmarks.

The results demonstrate that SINKTRACK consistently surpasses both `Direct` and `CoT` baselines across all tested LLMs (including Qwen2.5-VL and Gemma3) and benchmarks. Crucially, the low variance observed in SINKTRACK's performance indicates that anchoring the context via the attention sink not only improves accuracy but also enhances the stability of the model's reasoning process compared to standard prompting methods.

Multi-modal Tasks. On benchmarks designed to assess complex reasoning and object-level hallucination, SINKTRACK shows substantial gains. For instance, on M3CoT, SINKTRACK boosts the accuracy of *Qwen2.5-VL-7B* by **+22.8**% over `CoT`. On POPE, a benchmark specifically designed for hallucination detection, SINKTRACK consistently improves both accuracy and Macro-F1 scores

Model	QuAC-1 (1,000)		QuAC-2 (2,000)		QuAC-3 (3,000)		QuAC (7,354)		SQuAD2.0 (11,873)	
	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1
MiniCPM3-4B (Hu et al., 2024)										
Direct	52.82±0.62	52.79±0.40	49.97±0.34	50.94±0.23	48.13±0.36	49.53±0.23	47.17±0.19	48.07±0.05	62.93±0.15	**38.61**±0.08
CoT	42.69±0.94	45.45±0.71	41.03±0.33	43.35±0.73	39.39±0.27	41.19±0.74	37.68±0.18	36.50±0.20	**63.63**±0.11	23.73±0.04
SINKTRACK	**53.57**±0.41	**52.82**±0.43	**50.12**±0.26	**50.98**±0.25	**48.53**±0.15	**49.75**±0.21	**47.29**±0.27	**48.08**±0.30	62.74±0.04	36.83±0.09
Qwen2.5-7B-Instruct (Qwen et al., 2025)										
Direct	52.57±0.39	47.06±0.20	52.00±0.20	48.06±0.10	51.26±0.22	47.64±0.15	49.47±0.07	46.71±0.15	80.97±0.13	**33.39**±0.06
CoT	38.47±1.55	42.11±0.90	37.30±1.16	39.42±0.76	36.34±0.68	37.92±0.22	36.48±0.29	36.58±0.09	77.39±0.14	28.74±0.06
SINKTRACK	**52.93**±0.25	**47.52**±0.21	**52.98**±0.10	**48.08**±0.07	**52.01**±0.30	**47.77**±0.11	**50.25**±0.21	**46.73**±0.15	**81.04**±0.09	29.63±0.05
Llama3.1-8B-Instruct (Grattafiori et al., 2024)										
Direct	54.10±0.02	48.87±0.14	54.25±0.08	51.31±0.09	53.70±0.04	51.20±0.13	52.45±0.07	50.66±0.15	78.69±0.02	24.95±0.04
CoT	54.90±0.12	46.09±0.09	51.75±0.24	38.21±0.11	49.13±0.27	34.47±0.13	46.95±0.29	28.74±0.05	58.19±0.31	**28.86**±0.07
SINKTRACK	**59.40**±0.10	**54.05**±0.04	**58.05**±0.05	**54.25**±0.08	**56.20**±0.09	**53.45**±0.11	**53.51**±0.06	**51.56**±0.09	**79.83**±0.07	25.72±0.13

Table 2: Performance of LLMs on text-based tasks in terms of Accuracy (%) and Macro-F1. The **bold** indicates the best performance for the base LLM.

(*e.g.*, **85.5**% *vs* 83.7% accuracy and **91.8**% *vs* 90.6% F1 score compared to CoT), confirming its effectiveness in mitigating the generation of non-existent objects.

Text-based Tasks. We use long-context conversational QA to specifically measure the mitigation of context forgetting. As the conversational context lengthens from QuAC-1 to QuAC, the performance of baseline methods degrades due to attention drift. In contrast, SINKTRACK maintains a strong and often widening advantage. The most significant gains are seen with Llama3.1-8B, where on the longest-context QuAC setting (7,354 questions), SINKTRACK achieves an accuracy of **56.2**%, a **7.1**% increase over CoT. On SQuAD2.0, which tests the LLM's ability to abstain from answering unanswerable questions, SINKTRACK also yields consistent improvements.

Analysis of CoT Performance. It is worth noting that CoT sometimes underperforms Direct Prompting, an observation consistent with existing studies. This performance degradation is highly task-dependent. In long-context understanding tasks, such as summarization and QA, CoT can even degrade performance (Liu et al., 2025; Wang & Sun, 2025). Furthermore, while CoT excels in complex reasoning (Sprague et al., 2025) like M3CoT, in non-reasoning tasks, its extended chains exacerbate attention drift and raise the risk of error accumulation, where a single misstep derails the final answer (Shen et al., 2025; Liu et al., 2025).

Taking together, these results validate that SINKTRACK is a broadly effective and generalizable enhancement. It directly addresses both hallucination and context forgetting across different modalities, tasks, and model architectures without requiring any model training.

3.3 QUALITATIVE COMPARISON RESULT

We present qualitative comparisons to illustrate how SINKTRACK improves LLM's reasoning. In cases where CoT fails (Fig. 7, left), SINKTRACK succeeds by grounding its analysis in specific contextual details (*e.g.*, skis and poles) rather than ambiguous cues. This advantage also extends to failure cases where both methods produce an incorrect answer due to knowledge deficits (Fig. 7, right). Even then, SINKTRACK's reasoning chain remains internally consistent, whereas CoT exhibits context forgetting, where its final conclusion contradicts its own intermediate steps. This indicates that SINKTRACK enhances the generation process by enforcing closer adherence to the initial context and maintaining logical integrity, even when LLM's underlying knowledge is flawed.

3.4 ANALYSIS

3.4.1 INJECTION STRATEGIES

To study the effects of two main hyperparameters: the injection layers and the injection strength (§2.3), we analyze the performance of various `soft injection` using *Qwen2.5-VL-7B-Instruct* on M3CoT. The results in Table 3 highlight two critical findings: **i**) Injection Frequency. Intermittent injection (Inject Every 5 Layers) is more effective than continuous injection (Inject Every Layer), with a 9.0% absolute accuracy improvement. This suggests that overly frequent intervention might disrupt LLM's native computational flow to some extent. **ii**) Injec-

Method	Acc (%)
CoT (Baseline)	43.83
Inject Every Layer	51.02
Inject Every 5 Layers	60.04
Intermittent & Increasing	59.87
Intermittent & Decaying	**60.69**

Table 3: Performance of different `soft injection` variants(§2.3).

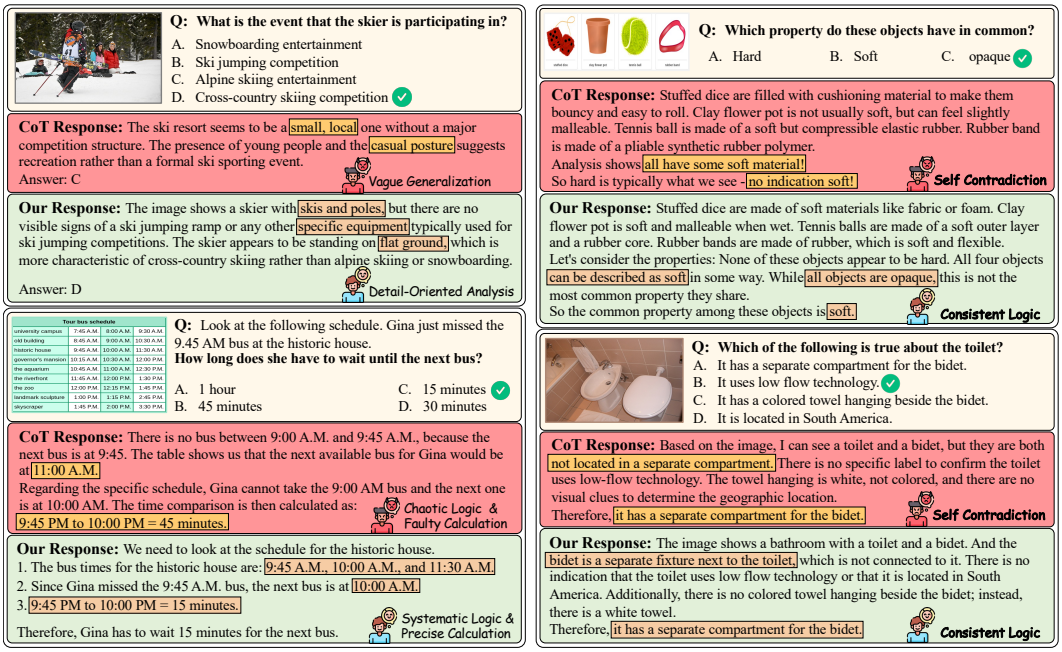


Figure 7: Qualitative comparison between CoT and SINKTRACK.

tion Strength. A decaying schedule, which applies stronger injection in earlier layers, yields better performance (60.7% *vs* 59.9%). This indicates that early contextual anchoring is more beneficial than later-stage interventions. In addition, the sensitivity to injection strength motivates us to develop an adaptive mechanism (§2.4).

3.4.2 HIERARCHICAL ATTENTION PATTERN

To analyze the structural impact of our injection, we quantify its effect on LLM's attention pattern. Concretely, we compute the Spearman's rank correlation between ⟨BOS⟩'s attention weight vectors across all layers, before and after applying SINKTRACK. The analysis reveals a near-perfect positive correlation ($\rho = 0.9985, p < 2.22 \times 10^{-17}$), as detailed in Appendix §B.1. This high correlation demonstrates that SINKTRACK preserves the relative importance of attention across different layers. Rather than disrupting LLM's pre-trained attention hierarchy, SINKTRACK enriches the information content of ⟨BOS⟩ while maintaining the integrity of the original computational flow.

3.4.3 INFORMATION FLOW

We further quantify the underlying working mechanism of SINKTRACK as an information anchor by analyzing its vertical and information flows.

Vertical Flow: Encoding via Value Vectors. The output of the attention mechanism is an aggregation of value vectors weighted by attention scores:

$$\boldsymbol{O}_t = \sum_{j=1}^{t} \alpha_{t,j} \cdot \boldsymbol{V}_j, \tag{5}$$

where $\alpha_{t,j}$ represents the attention weight and $\boldsymbol{V}_j$ denotes the value vector of the j-th token. Consequently, the information capacity of the ⟨BOS⟩ anchor can be quantified by calculating the L1 Norm of its value vector. As illustrated in Fig. 8(a), a significant difference value (yellow line) is observed between the original state (Before) and the injected state (After). This variation serves as a proxy for *Information Gain*. The distinct divergence in L1 Norm across layers confirms that the anchor information is successfully absorbed and encoded into the ⟨BOS⟩ representation, effectively transforming it from a standard placeholder into a semantic anchor.

Horizontal Flow: Sustained Attention Anchoring. To evaluate how this injected information is propagated during generation, we analyze the attention distribution of newly generated tokens. Specifically, we track the attention weights that subsequent tokens assign to the ⟨BOS⟩ anchor

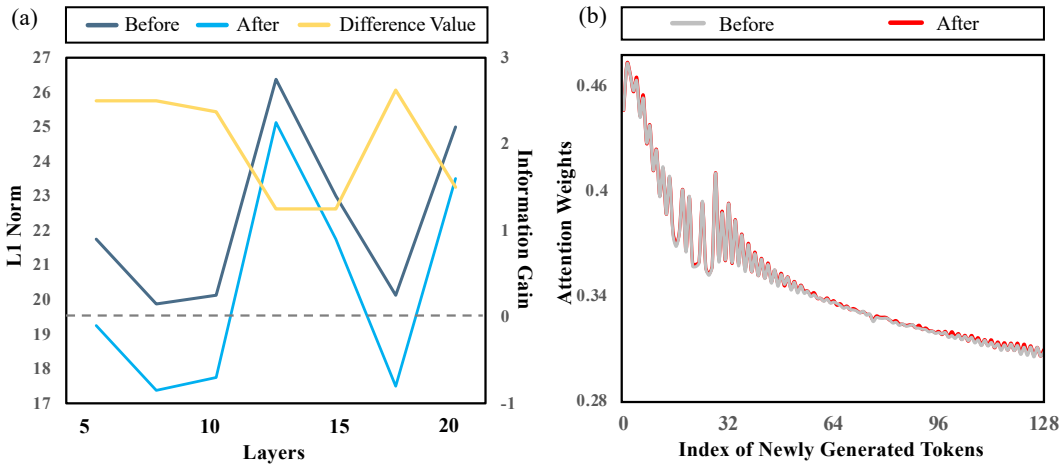


Figure 8: The illustration of information flow with and without SINKTRACK (§2.4).

($\alpha_{t,\text{BOS}}$). As shown in Fig. 8(b), the attention weights exhibit a natural decay but maintain a persistent magnitude, indicating that the generated sequence continuously attends to the ⟨BOS⟩ token. Crucially, the attention curves before and after injection (gray and red lines) overlap almost perfectly. This phenomenon demonstrates two key findings: **i**) SINKTRACK successfully establishes a long-term information bridge, as the anchor remains a focal point throughout generation; and **ii**) The injection is non-destructive, preserving the intrinsic attention patterns of the LLMs.

4 RELATED WORK

Training-free Inference-time Enhancement for LLMs. Developing plug-and-play enhancement methods to improve the capabilities of LLMs has received widespread attention from the community. These approaches can be broadly categorized into three main streams. The first category is *External Module Augmentation* (Wang et al., 2024; Ma et al., 2024; Feng et al., 2024; Chen et al., 2022). These methods couple an LLM with external components to address limitations such as outdated knowledge or the lack of specific skills, *e.g.*, retrieval-augmented (Wang et al., 2024; Feng et al., 2024; Yu et al., 2024a) and tool-augmented methods (Ma et al., 2024; Gou et al., 2024; Lu et al., 2023; Chen et al., 2022; Wei et al., 2022). The second category, *Internal Reasoning Process Enhancement* (Dhuliawala et al., 2024; Zhang et al., 2024c; Yao et al., 2023; Zhang et al., 2024a; 2025a), uses structured prompting strategies or adds reasoning steps to guide LLM, *e.g.*, decomposing complex problems into simpler sub-tasks (Khot et al., 2023; Chen et al., 2022; Wei et al., 2022; Ning et al., 2024), generating multiple candidate outputs with verification (Gou et al., 2024; Wang et al., 2023; Zhang et al., 2024b;a), and using non-linear reasoning structures to organize complex inference paths (Yao et al., 2023; Roy & Roth, 2015; Besta et al., 2024; Wei et al., 2022). The third category is *Internal Representation Intervention* (Li et al., 2023a; Zhu et al., 2025; Han et al., 2025; Turner et al., 2023; Liu et al., 2024c), where methods directly modify the LLM's internal states during inference to guide its behavior, *e.g.*, adding a targeted offset to LLM's hidden states (Li et al., 2023a; Subramani et al., 2022; Turner et al., 2023) and calibrating LLM's attention (Zhu et al., 2025; Yu et al., 2024b; Han et al., 2025; Lu et al., 2021). Despite impressive, these methods often require repetitive intervention, introducing considerable computational cost at each generation step.

Unlike methods that apply continuous modifications during the decoding loop, SINKTRACK achieves training-free enhancement through a one-time intervention prior to inference. The injection of key information into the representation of ⟨BOS⟩ is a one-off operation performed before generation begins. Once this information anchor is established, the subsequent auto-regressive process is identical to the standard decoding procedure, requiring no extra computational steps.

Attention Sink refers to the empirical observation that the initial token of a sequence (*i.e.*, ⟨BOS⟩) consistently receives a high degree of attention during auto-regressive generation. This phenomenon is considered an emergent property of Transformer architectures (Xiao et al., 2024; Kaul et al., 2025; Gu et al., 2025). Proposed causes include it being an artifact of the Softmax function within the attention mechanism (Barbero et al., 2025) or a consequence of the token's stable structural position in the sequence (Yu et al., 2024b; Ruscio et al., 2025; Cancedda, 2024). Existing efforts use the attention sink in two main ways. One line of work uses the initial token's stability for

tasks like KV cache management (Zhang et al., 2023) and key token protection (Liu et al., 2024b). Another research direction focuses on enhancing LLM's performance by modifying or calibrating the attention mechanism to synergize with this structural anchor (Han et al., 2025; Gu et al., 2025).

This work explores a new application branch for attention sink: repurposing it as an active mechanism for context anchoring through information injection. We do not alter the attention weights themselves; instead, we modify the representation of $\langle$BOS$\rangle$ to form a high-information anchor that provides LLMs with high-integrity context. This shift from indirect attention calibration to direct information supply reveals a new direct to address the issues of hallucination and context forgetting.

5 Conclusion

This work mitigates two long-standing issues (*i.e.*, hallucination and context forgetting) in decoder-only LLMs by exploiting the synergistic relationship between attention drift and attention sink. We propose SinkTrack, an efficient, plug-and-play method that transforms the passive attention sink into an active context anchor. Through an adaptive cross-attention mechanism, SinkTrack injects key contextual information into the first token's representation, thereby counteracting the information decay caused by attention drift during generation. Consistent performance improvements across diverse textual and multi-modal benchmarks validate its effectiveness and generalizability, and our analysis of its information delivery mechanism further substantiates the principles of SinkTrack.

Ethics Statement

In this study, we propose SinkTrack, a training-free mechanism designed to enhance the reliability and reasoning capabilities of LLMs. We acknowledge that any technique that improves the fundamental capabilities of LLMs – such as increasing their coherence or reducing factual errors – carries a dual-use potential. While our work is intended for benevolent purposes, more capable and robust models could theoretically be exploited by malicious actors to generate more persuasive and harder-to-detect misinformation.

Our research is focused exclusively on the positive application of mitigating object hallucinations and improving long-context reasoning on established, public academic benchmarks. The datasets and tasks used in our experiments do not involve sensitive personal data, human subjects, or the generation of socially harmful content. Our methodology is transparent and designed to be analyzed and understood by the research community.

Reproducibility Statement

All experiments are conducted using publicly available, open-source LLMs and MLLMs. Specific model checkpoints are obtained from the official Hugging Face Hub. The detailed formulation and algorithmic implementation of SinkTrack are presented in §2. A comprehensive description of our experimental setup, including the specific datasets, evaluation metrics, and baselines used, is provided in §3. Any additional implementation details are further detailed in the Appendix. To facilitate full reproducibility, our complete source code, along with the scripts required to run all experiments and reproduce the results reported in this paper, is available at GitHub.

Acknowledgement

This work was supported by National Science and Technology Major Project (No. 2023ZD0121300), National Natural Science Foundation of China (No. 62372405), Fundamental Research Funds for the Central Universities (226-2025-00057), Jiangsu Provincial Scientific Research Center of Applied Mathematics (No. BK20233002), and CCF-Tencent Open Fund.

References

Wenbin An, Feng Tian, Sicong Leng, Jiahao Nie, Haonan Lin, QianYing Wang, Ping Chen, Xiaoqin Zhang, and Shijian Lu. Mitigating object hallucinations in large vision-language models with

assembly of global and local attention. In *CVPR*, pp. 29915–29926, 2025.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Federico Barbero, Alvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Petar Veličković, Razvan Pascanu, and Michael M. Bronstein. Why do LLMs attend to the first token? In *COLM*, 2025.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *AAAI*, pp. 17682–17690, 2024.

Marialena Bevilacqua, Kezia Oketch, Ruiyang Qin, Will Stamey, Xinyuan Zhang, Yi Gan, Kai Yang, and Ahmed Abbasi. When automated assessment meets automated content generation: Examining text quality in the era of gpts. *ACM Transactions on Information Systems*, 43(2):1–36, 2025.

Nicola Cancedda. Spectral filters, dark signals, and attention sinks. In *ACL*, pp. 4792–4808, 2024.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? In *NeurIPS*, pp. 27056–27087, 2024a.

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. M^3CoT: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In *ACL*, pp. 8199–8221, 2024b.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. QuAC: Question answering in context. In *EMNLP*, pp. 2174–2184, 2018.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of ACL*, pp. 3563–3578, 2024.

Xin Luna Dong, Seungwhan Moon, Yifan Ethan Xu, Kshitiz Malik, and Zhou Yu. Towards next-generation intelligent assistants leveraging llm techniques. In *KDD*, pp. 5792–5793, 2023.

Shangbin Feng, Weijia Shi, Yuyang Bai, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. Knowledge card: Filling LLMs' knowledge gaps with plug-in specialized language models. In *ICLR*, 2024.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *ICLR*, 2024.

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

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. In *ICLR*, 2025.

Yanchu Guan, Dong Wang, Zhixuan Chu, Shiyu Wang, Feiyue Ni, Ruihua Song, and Chenyi Zhuang. Intelligent agents with llm-based process automation. In *KDD*, pp. 5018–5027, 2024.

Feijiang Han, Xiaodong Yu, Jianheng Tang, and Lyle Ungar. Zerotuning: Unlocking the initial token's power to enhance large language models without training. *arXiv preprint arXiv:2505.11739*, 2025.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

Anna Kalyuzhnaya, Sergey Mityagin, Elizaveta Lutsenko, Andrey Getmanov, Yaroslav Aksenkin, Kamil Fatkhiev, Kirill Fedorin, Nikolay O Nikitin, Natalia Chichkova, Vladimir Vorona, et al. Llm agents for smart city management: Enhancing decision support through multi-agent ai systems. *Smart Cities (2624-6511)*, 8(1), 2025.

Prannay Kaul, Chengcheng Ma, Ismail Elezi, and Jiankang Deng. From attention to activation: Unraveling the enigmas of large language models. In *ICLR*, 2025.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. In *ICLR*, 2023.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *NeurIPS*, pp. 41451–41530, 2023a.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *EMNLP*, pp. 292–305, 2023b.

Yubo Li, Xiaobin Shen, Xinyu Yao, Xueying Ding, Yidi Miao, Ramayya Krishnan, and Rema Padman. Beyond single-turn: A survey on multi-turn interactions with large language models. *arXiv preprint arXiv:2504.04717*, 2025.

Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, et al. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *ACM Computing Surveys*, 2023.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. In *TACL*, pp. 157–173, 2024a.

Ruikang Liu, Haoli Bai, Haokun Lin, Yuening Li, Han Gao, Zhengzhuo Xu, Lu Hou, Jun Yao, and Chun Yuan. IntactKV: Improving large language model quantization by keeping pivot tokens intact. In *Findings of ACL*, pp. 7716–7741, 2024b.

Ryan Liu, Jiayi Geng, Addison J. Wu, Ilia Sucholutsky, Tania Lombrozo, and Thomas L. Griffiths. Mind your step (by step): Chain-of-thought can reduce performance on tasks where thinking makes humans worse. *arXiv preprint arXiv:2410.21333*, 2025.

Shi Liu, Kecheng Zheng, and Wei Chen. Paying more attention to image: A training-free method for alleviating hallucination in lvlms. In *ECCV*, pp. 125–140, 2024c.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. In *NeurIPS*, pp. 43447–43478, 2023.

Yu Lu, Jiali Zeng, Jiajun Zhang, Shuangzhi Wu, and Mu Li. Attention calibration for transformer in neural machine translation. In *ACL*, pp. 1288–1298, 2021.

Yubo Ma, Zhibin Gou, Junheng Hao, Ruochen Xu, Shuohang Wang, Liangming Pan, Yujiu Yang, Yixin Cao, and Aixin Sun. SciAgent: Tool-augmented language models for scientific reasoning. In *EMNLP*, pp. 15701–15736, 2024.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. Skeleton-of-thought: Prompting LLMs for efficient parallel generation. In *ICLR*, 2024.

Alexander Peysakhovich and Adam Lerer. Attention sorting combats recency bias in long context language models. *arXiv preprint arXiv:2310.01427*, 2023.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *ACL*, pp. 784–789, 2018.

Vipula Rawte, Amit Sheth, and Amitava Das. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*, 2023.

Subhro Roy and Dan Roth. Solving general arithmetic word problems. In *EMNLP*, pp. 1743–1752, 2015.

Valeria Ruscio, Umberto Nanni, and Fabrizio Silvestri. What are you sinking? a geometric approach on attention sink. *arXiv preprint arXiv:2508.02546*, 2025.

Jianan Shen, Xiaolong Cui, Zhiqiang Gao, and Xuanzhu Sheng. Adaptive chain-of-thought distillation based on llm performance on original problems. *Mathematics*, 13(22), 2025.

Zayne Rea Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. In *ICLR*, 2025.

Nishant Subramani, Nivedita Suresh, and Matthew Peters. Extracting latent steering vectors from pretrained language models. In *Findings of ACL*, pp. 566–581, 2022.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Chupei Wang and Jiaqiu Vince Sun. Unable to forget: Proactive lnterference reveals working memory limits in llms beyond context length. In *ICML Workshop on Long-Context Foundation Models*, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023.

Zihao Wang, Anji Liu, Haowei Lin, Jiaqi Li, Xiaojian Ma, and Yitao Liang. Rat: Retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation. *arXiv preprint arXiv:2403.05313*, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, pp. 24824–24837, 2022.

X.AI. Grok-1.5 vision preview. `https://x.ai/blog/grok-1.5v`, 2024.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *ICLR*, 2024.

Jia Xu, Weilin Du, Xiao Liu, and Xuejun Li. Llm4workflow: An llm-based automated workflow model generation tool. In *ASE*, pp. 2394–2398, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*, 2023.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *EMNLP*, pp. 14672–14685, 2024a.

Zhongzhi Yu, Zheng Wang, Yonggan Fu, Huihong Shi, Khalid Shaikh, and Yingyan (Celine) Lin. Unveiling and harnessing hidden attention sinks: enhancing large language models without training through attention calibration. In *ICML*, 2024b.

Yongheng Zhang, Qiguang Chen, Min Li, Wanxiang Che, and Libo Qin. AutoCAP: Towards automatic cross-lingual alignment planning for zero-shot chain-of-thought. In *Findings of ACL*, pp. 9191–9200, 2024a.

Yongheng Zhang, Qiguang Chen, Jingxuan Zhou, Peng Wang, Jiasheng Si, Jin Wang, Wenpeng Lu, and Libo Qin. Wrong-of-thought: An integrated reasoning framework with multi-perspective verification and wrong information. In *Findings of EMNLP*, pp. 6644–6653, 2024b.

Yongheng Zhang, Xu Liu, Ruihan Tao, Qiguang Chen, Hao Fei, Wanxiang Che, and Libo Qin. Vitcot: Video-text interleaved chain-of-thought for boosting video understanding in large language models. In *ACM MM*, pp. 5267–5276, 2025a.

Yongheng Zhang, Xu Liu, Ruoxi Zhou, Qiguang Chen, Hao Fei, Wenpeng Lu, and Libo Qin. CCHall: A novel benchmark for joint cross-lingual and cross-modal hallucinations detection in large language models. In *ACL*, pp. 30728–30749, 2025b.

Yuansen Zhang, Xiao Wang, Tianze Chen, Jiayi Fu, Tao Gui, and Qi Zhang. P4: Plug-and-play discrete prompting for large language models personalization. In *Findings of ACL*, pp. 9129–9144, 2024c.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *NeurIPS*, pp. 34661–34710, 2023.

Younan Zhu, Linwei Tao, Minjing Dong, and Chang Xu. Mitigating object hallucinations in large vision-language models via attention calibration. *arXiv preprint arXiv:2502.01969*, 2025.

SUMMARY OF THE APPENDIX

- §A details the pseudo-code implementation of SINKTRACK.
- §B provides in-depth analyses concerning the stability of the hierarchical attention pattern, resistance to attention drift, and computational overhead.
- §C discusses the limitations of the current approach.

A PSEUDO-CODE FOR SINKTRACK

To provide a clear description of SINKTRACK, this section details the pseudo-code for it. As discussed in the main paper, the core idea is to create two parallel computational tracks within the attention module. *Track* 1 *(Injection Track)* uses cross-attention to inject external information into the initial token. *Track* 2 *(Original Track)* performs standard causal self-attention for the rest of the tokens to maintain the integrity of the pre-trained sequence structure. The complete process, including the conditional check for applying the injection, is described in Algo. S1.

Algorithm S1 Pseudo-code for the Dual-Track Cross-Attention Mechanism.

```
"""
h_ori: hidden states of the original sequence (L_ori × D_h)
h_info: hidden states of the external information (L_info × D_h)
cfg: configuration for injection rules
"""
def Hybrid_Attention(h_ori, h_info, cfg):
    #--- 1. Check if conditions for injection are met ---
    do_inject = Check_Conditions(cfg, h_info)

    if do_inject:
        #--- PATH A: HYBRID ATTENTION FOR INJECTION ---
        # 2a. Project inputs to Q, K, V spaces
        Q_ori, K_ori, V_ori = Project_QKV(h_ori)
        K_info, V_info = Project_KV(h_info)

        # 2b. Split the original Query for dual attention tracks
        q_first, Q_rest = Split_Original_Vector(Q_ori)

        # 3. Execute dual-track attention
        # Track 1: INJECTION via Cross-Attention
        o_first = Cross_Attention(Q=q_first, K=K_info, V=V_info)

        # Track 2: CAUSAL SELF-ATTENTION
        m = Create_Causal_Mask(Q_rest)
        o_rest = Causal_Self_Attention(Q=Q_rest, K=K_ori, V=V_ori, mask=m)

        # 4. Merge outputs from both tracks
        o_combined = Concatenate(o_first, o_rest)

    else
        #--- PATH B: STANDARD CAUSAL SELF-ATTENTION ---
        o_combined = Standard_Causal_Attention(h_ori)

    # 5. Apply final output projection
    h_out = Output_Projection(o_combined)

    return h_out
```

B IN-DEPTH ANALYSIS: MECHANISM AND EFFICIENCY

This section provides a deeper analysis of the underlying mechanism of SINKTRACK and its computational implications.

B.1 STABILITY OF HIERARCHICAL ATTENTION PATTERN

To analyze the structural impact of our injection, we quantify its effect on the LLM's hierarchical attention pattern. Concretely, we compute the Spearman's rank correlation between the ⟨BOS⟩'s attention weight vectors across all layers, before and after applying SINKTRACK. As shown in Fig. S1, the analysis reveals a near-perfect positive correlation ($\rho = 0.9985, p < 2.22 \times 10^{-17}$).

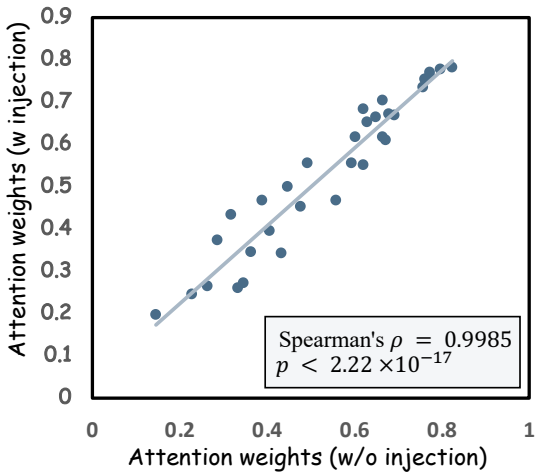


Figure S1: Comparison between the ⟨BOS⟩'s attention weight vectors before and after applying SINKTRACK.

This high correlation demonstrates that SINKTRACK preserves the relative importance of attention across different layers. Rather than disrupting the LLM's pre-trained attention hierarchy pattern, SINKTRACK enriches the information content of the initial token while maintaining the integrity of the original computational flow. This finding provides a quantitative explanation for why SINKTRACK can enhance performance without causing the model collapse observed in `hard injection`.

B.2 VERIFICATION OF SINKTRACK'S STABILITY

To further verify that SINKTRACK remains effective over long generation sequences, we conduct a "Drift Test" on Llama3.1-8B. We feed it an input context of 8k+ tokens and track the attention weights allocated to the ⟨BOS⟩ token by subsequently generated tokens (steps 1 to 1024).

As presented in Table S1, the attention weight to ⟨BOS⟩ remains dominant throughout the generation process. For instance, at generated token index 1024 (sequence position 9106), the attention to ⟨BOS⟩ is 0.582, which is approximately $14\times$ higher than the maximum attention paid to any other token. This confirms that SINKTRACK creates a robust attention sink that is immune to attention drift, ensuring long-term context retention.

Table S1: SinkTrack Drift Test on Llama3.1-8B. The attention to ⟨BOS⟩ consistently dominates other tokens even after generating 1024 new tokens.

Gen. Idx	Seq. Pos	Attn to ⟨BOS⟩	Max Attn (Others)	Ratio
1	8083	0.561	0.057	9.87x
64	8146	0.570	0.048	11.97x
128	5210	0.612	0.054	11.25x
256	8338	0.560	0.050	11.24x
512	8594	0.558	0.037	15.17x
1024	9106	0.582	0.042	13.96x

B.3 COMPUTATIONAL OVERHEAD

We measure the prefill stage latency on an NVIDIA GeForce RTX 4090 to evaluate the computational cost of SINKTRACK. The results, averaged over 500 samples, are presented in Table S2. Specifically, Llama3.1-8B shows a latency increase from 35.90 ms *(w/o Injection)* to 36.66 ms *(w/ Injection)*, representing a marginal overhead of only +0.76 ms. Similarly, Qwen2.5-VL-7B shows an increase of +3.13 ms. These findings confirm that SINKTRACK introduces negligible latency, as the core intervention is a one-off operation performed prior to the auto-regressive generation.

Table S2: Prefill stage latency measurement on NVIDIA GeForce RTX 4090 (averaged over 500 samples). SINKTRACK introduces negligible computational overhead.

Model	**w/o Injection**	**w/ Injection**	**Overhead**
Llama3.1-8B	35.90 ms	36.66 ms	+0.76 ms
Qwen2.5-VL-7B	107.13 ms	110.26 ms	+3.13 ms

C LIMITATION

This work proposes and validates SINKTRACK representing a solid step towards mitigating hallucination and context forgetting in LLMs. However, it is important to recognize several inherent limitations in our current approach that open avenues for future research. Specifically, SINKTRACK's core design anchors all contextual information to the representation of a single initial token. While our adaptive mechanism optimizes information retrieval, this setup introduces a potential information bottleneck, as a single fixed-dimension vector has a finite capacity for representation. This may not fully capture the nuanced informational needs of tasks requiring extremely long or dense contexts. For example, while effective, the diminishing performance gains we observe on the long-dialogue QuAC may be an early indicator of this bottleneck. This limitation suggests a need to explore future paradigms beyond a single anchor, such as distributed or dynamically positioned anchoring mechanisms.