The Fragile Truth of Saliency: Improving LLM Input Attribution via Attention Bias Optimization

Yihua Zhang **Changsheng Wang** **Yiwei Chen**
Chongyu Fan **Jinghan Jia** **Sijia Liu**
Michigan State University

Abstract

Input saliency aims to quantify the influence of input tokens on the output of large language models (LLMs), which has been widely used for prompt engineering, model interpretability, and behavior attribution. Despite the proliferation of saliency techniques, the field lacks a standardized and rigorous evaluation protocol. In this work, we introduce a stress-testing framework inspired by the needle-in-a-haystack (NIAH) setting to systematically assess the reliability of seven popular input saliency methods. Our evaluation reveals a surprising and critical flaw: existing methods consistently assign non-trivial importance to irrelevant context, and this attribution error worsens as input length increases. To address this issue, we propose a novel saliency method based on *Attention Bias Optimization* (ABO), which explicitly optimizes the attention bias associated with each input token to quantify its causal impact on target token generation. ABO robustly outperforms existing methods by $10 \sim 30\%$ in saliency accuracy across diverse NIAH tasks, maintains effectiveness up to 10K-token prompts, and enables practical applications including zero-shot detoxification, sentiment steering, and reasoning-error correction. Our findings highlight the limitations of prevalent attribution methods and establish ABO as a principled alternative for accurate token attribution.

1 Introduction

What makes a particular word in a prompt important to a large language model (LLM)? This question lies at the heart of input saliency analysis, a field that seeks to quantify how much each input token contributes to the model's output [1–8]. As LLMs become increasingly embedded in downstream tasks ranging from coding assistants [9, 10] to open-ended reasoning agents [11], understanding their decision-making process is essential for safety, interpretability, and controllability [12, 13]. Input saliency provides a foundation for prompt engineering by identifying which tokens most influence the model's behavior [14, 15], and it offers researchers a window into the inner workings of attention mechanisms and token interactions [16, 17]. As such, saliency analysis has emerged as a cornerstone in the interpretability toolkit for modern language models [18–20].

Diverse input attribution techniques already exists, spanning perturbation-based masking [7, 21], gradient and interaction methods [8, 12], vector-space decompositions [17, 22], Shapley values [23], and linear approximations [12, 24]. Each line of work claims to reveal the "true" importance of input tokens. Yet the field lacks a rigorous, model-agnostic benchmark for judging whether these scores separate meaningful signal from contextual noise, especially in the long-context prompts that define contemporary LLM use. Without such a benchmark, practitioners risk acting on attribution maps that are noisy, unstable, or flat-out wrong.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

To close this gap we design a *stress test* inspired by the classic "needle-in-a-haystack" (NIAH) evaluation [25]. We embed a single secret message inside a span of distractor text and ask the model to reproduce that message verbatim. Because the distractors have zero causal influence on the output, the ground-truth saliency distribution must concentrate on the secret message (and the final query). This controlled setting lets us audit attribution methods with surgical precision. Our evaluation of six popular techniques [1–6] uncovers a systemic flaw: every method assigns a substantial fraction of saliency to irrelevant tokens, which grows as the context length increases. The misattribution reaches over 90% of the total score for 10K-token prompts, implying that the interpretability tools most widely used today can be *actively misleading* in long-context scenarios.

Motivated by the observation that reweighting an input token's attention can deterministically sway the model's output when two contradictory messages are present, we recast attribution as an *optimization over attention biases*. ABO injects a small, learnable bias term into each attention head and solves for the minimal intervention that maximizes the probability of a target output token. The resulting bias magnitudes serve as faithful saliency scores: they are causal by construction, differentiable, and inexpensive to compute. Empirically, ABO withstands our NIAH stress test and remains robust up to 10K-token inputs. Beyond evaluation, ABO proves useful for zero-shot detoxification, sentiment steering, input token pruning, and rectifying errors in the reasoning traces. To summarize, our contributions are as follows:

• We introduce a **stress-testing framework** for LLM saliency based on the needle-in-a-haystack task, enabling fine-grained diagnosis under long-context inputs.

• We reveal **systemic reliability failures** in six widely-used attribution methods and show that misattribution to irrelevant tokens can exceed over 90 % in 10 K-token prompts, quantifying the brittleness of existing saliency tools.

• We propose **Attention Bias Optimization (ABO)**, a principled, optimization-based technique that delivers token-level saliency scores with superior causal fidelity and scalability to long-contexts.

• We demonstrate ABO's **broad practical utility** across sentiment control, toxic prompt detoxification, and LLM error correction, underscoring its value for both research and real-world deployment.

2 Related Work

LLM input saliency methods. Understanding how large language models (LLMs) attribute generated outputs to their preceding input tokens is crucial for model interpretability, debugging, and effective prompt engineering. Several attribution methods have been proposed to elucidate these relationships, typically categorized into gradient-based, vector-based, and perturbation-based approaches. Gradient-based methods measure token importance via gradients obtained through backpropagation. Examples include gradient norm [26–28], gradient-input production [29–31], Layer-wise Relevance Propagation (LRP-XAI) [32, 33], and Generic Attention Explainability (GAE) [8]. Although these methods have demonstrated their value across diverse NLP models and tasks, they predominantly rely on local linear approximations, potentially accumulating errors in deeply stacked transformer layers. Vector-based methods leverage internal model representations, typically attention weights, to infer token importance, exemplified by Attention Rollout [34], GlobEnc [35], and ALTI [36]. However, these approaches implicitly assume linearity in attention score combinations across layers, limiting their fidelity as model depth and complexity increase. Alternatively, perturbation-based techniques [23, 37, 38] measure token significance by assessing changes in model confidence upon token removal or masking. Representative methods include AtMan [6], which combines perturbation strategies with attention mechanisms specifically optimized for transformer models. The evaluations of existing methods typically involve short-context scenarios with limited causal interactions. Consequently, existing benchmarks neither sufficiently explore attribution accuracy under realistic long-context scenarios nor rigorously stress-test the methods' robustness against irrelevant contextual noise. Our work fills this critical gap by introducing a systematic long-context stress-test framework, providing insights into the limitations of current attribution methodologies when applied to LLMs.

Controlled generation and behavior steering for LLMs. A large body of work seeks to steer language–model behavior at *inference* time via prompt design. Manual and automated prompting techniques have shown striking gains on downstream tasks [39–41], and toolkits such as PROMPTSOURCE, RLPrompt, and AutoPrompt make cue construction more accessible [42–46]. Yet LLMs remain

[System Prompt] *You are a helpful AI chat bot that answers questions for a user. Keep your response short and direct. Don't present information outside the document below or repeat your findings . . .* ***[Long Irrelevant Context]*** *July 2006. . . imagine you were going to spend the weekend at a friend's house on a little island off the coast of Maine. . . . (irrelevant text continues). . . Quiet is another matter.* ***[Secret Message]*** *The password to get into the zone of Hagazin is "**Yin**".* ***[Irrelevant Text Resumes]*** *I realize it seems a bit eccentric to take earplugs on a trip to an island off the coast of Maine. If anywhere should be quiet, that should. But what if the person in the next room snored? . . . Pen and paper wick ideas.* ***[Query on Secret Message]*** *What is the password to get into the zone of Hagazin? Your answer is:*

Figure 1: **Color-coded needle-in-a-haystack (NIAH) [25] prompt.** System instruction, distractor text, secret message, and final query are visually distinguished. When calibrating an LLM saliency method with this example, *only saliency score assigned to the non-gray regions* (yellow, green, blue) is meaningful; any score allocated to gray tokens reflects incorrect attribution.

brittle: small lexical changes can flip predictions or derail chain-of-thought reasoning [47, 48]. Complementary to prompt engineering, *training-time* interventions, e.g. instruction finetuning [49, 50], RLHF and its variants [51–53], or loss terms that privilege user-marked tokens [54–57], improve overall steerability but require additional data or compute. Parallel work explores lightweight adaptation via low-rank or quantized updates (LoRA, AdaLoRA, QLoRA, TOAST) [58–61], and targeted knowledge editing with ROME, MEMIT, MEND, or REMEDI [62–65]. Our saliency-based Attention Bias Optimization is orthogonal to these approaches: it leaves model weights unchanged, requires no extra supervision, and instead manipulates generation by amplifying or suppressing precisely the tokens that the model itself deems most causal.

3 A Stress Test of LLM Input Saliency Methods

While input saliency methods have become ubiquitous tools for interpreting LLM predictions, it remains uncertain whether these attribution methods accurately capture token importance under realistic, long-context scenarios. Despite numerous techniques proposed in the literature, their evaluation often relies on relatively simplistic or short-text scenarios [1–3], leaving their generalizability and reliability largely unchecked. To bridge this gap, this section introduces a novel stress-testing framework that systematically probes the robustness and reliability of existing saliency methods under increasingly challenging input conditions. Our aim is twofold: first, to reveal critical limitations of current attribution techniques when exposed to long-context inputs; second, to motivate the necessity of a fundamentally more robust attribution strategy.

The needle-in-a-haystack (NIAH) test for saliency assessment. Inspired by the classic *needle-in-a-haystack* paradigm [25], originally developed to assess the information-retrieval capabilities of LLMs in long-context settings, we adapt this experimental setup explicitly for the evaluation of saliency methods. Our adapted NIAH scenario is carefully constructed to provide an explicit, ground-truth saliency distribution for rigorous evaluation. Specifically, as illustrated in **Fig. 1**, we craft prompts comprising four core components:

• a system prompt to establish the overall context, role, and behavior for the LLM's responses;

• a lengthy distractor passage sampled from diverse, public-domain texts (*e.g.*, news articles, Wikipedia entries), explicitly designed to provide irrelevant contextual noise;

• a short, deliberately constructed embedded secret message, which contain keywords (*e.g.*, "Hagazin" and "Yin" in Fig. 1) are intentionally chosen to be *out-of-vocabulary* words absent from standard LLM pre-training corpora, ensuring the uniqueness and singular salience of this message;

• a precise, targeted query placed at the end of the input, asking questions on the secret message hidden in the context.

Before the evaluation, we first make sure the LLM's answer to the query is correct. In the case above, the answer should be "Yin". Then, we use the saliency method to generate saliency scores to all the input tokens with respect to the target output token (*i.e., "Yin"*). The central evaluation metric employed throughout this analysis is the **Hit Ratio**:

$$\text{Hit Ratio} = \frac{\text{Saliency Score}(\text{system prompt} + \text{secret message} + \text{query})}{\text{Total Saliency}}. \tag{1}$$

Intuitively, a higher Hit Ratio approaching 1.0 indicates that a method effectively isolates truly salient tokens, while lower scores indicate a failure in accurately discriminating signal from noise. To ensure fair comparisons across varying methods (each with potentially different saliency scaling conventions), we normalize all saliency vectors to sum to one prior to aggregation.

Why is NIAH naturally a better fit for saliency calibration? Critically, the nature of this task involves pure information retrieval and no inferential reasoning is required. Consequently, the ground-truth saliency distribution should strictly allocate importance to three well-defined input regions: (1) the initial system prompt (if present), (2) the embedded secret message, and (3) the final query. All other input tokens, by experimental construction, are irrelevant and thus should be assigned negligible saliency scores. Furthermore, following the original NIAH setting [25], we systematically vary the length of the distractor text from 500 tokens (contexts) up to 4K tokens (contexts), effectively modulating the signal-to-noise ratio within the input while keeping the informative content fixed. This controlled manipulation of input length provides a straightforward yet rigorous means to evaluate each method's robustness to the increasing complexity of realistic long-context scenarios.

Examination of LLM input saliency via NIAH. To ensure comprehensive coverage, we evaluate a broad spectrum of **seven representative saliency methods** widely adopted in the literature. These methods span diverse methodological paradigms including gradient-based (GRADNORM [4], INPUT X GRAD [5]), attention-derived (ATMAN [6]), token distribution-based (TDD [2]), Shapley value-based (TOKENSHAP [1]), and linear modeling (AT2 [3]). As a reference, we also compare these methods with a random guess method (NAIVE), which assigns each input token equal saliency. Detailed algorithmic descriptions and hyperparameter selections for each method are provided in Appx. A to facilitate reproducibility and transparency. Our experiments are primarily conducted on a frozen LLaMA-2-7B [66] backbone to ensure controlled comparisons.

Fig. 2 presents a comprehensive visualization of Hit Ratio performance as a function of input length. Two critical observations immediately emerge: **First**, there is a steep accuracy collapse at long contexts. All evaluated methods experience a precipitous drop in attribution fidelity once the input exceeds approximately 1K tokens. Alarmingly, at input lengths beyond 2K tokens, several methods approach random performance (represented by the gray dashed baseline), highlighting their severe inability to cope with realistic, noisy, long-text inputs. **Second**, there is non-trivial misattribution even at short contexts. Surprisingly, even at the shortest tested context (100 tokens), over half of the attribution mass is consistently misallocated to irrelevant tokens. This persistent misattribution underscores that the root cause is not merely extreme input length, but a fundamental vulnerability in existing methods to structural or positional biases.

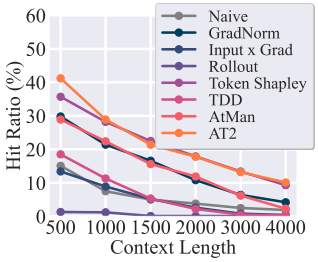

Figure 2: Hit ratio of existing saliency methods on the NIAH stress testing with different input length. Hit Ratio is the fraction of saliency scores assigned to the *true* causal tokens.

To understand the underlying causes of misattribution, **Fig. 3** provides a qualitative heatmap of saliency distributions for representative examples. Strikingly, misattribution does not manifest as

[System Prompt] _You _are _a _helpful _A I _chat _bot _that _answers _questions _for _a _user . _Keep _your _response _short _and _direct . _Don 't _present _information _outside _the _document _below _or _repeat _your _find ings ...
[Long Irrelevant Context] _July _2 0 0 6 ... _imagine _you _were _going _to _spend _the _week end _at _a _friend 's _house _on _a _little _island _off _the _coast _of _Maine ... _Qu iet _is _another _matter .
[Secret Message] _The _password _to _get _into _the _zone _of _H ag az in _is " _Yin " .
[Irrelevant Text Resumes] _I _realize _it _seems _a _bit _ecc ent ric _to _take _ear pl ugs _on _a _trip _to _an _island _off _the _coast _of _Maine . _If _anywhere _should _be _quiet , _that _should . _But _what _if _the _person _in _the _next _room _sn ored _? ..._Pen _and _paper _w ick _ideas .
[Query on Secret Message] _What _is _the _password _to _get _into _the _zone _of _H ag az in _? _Your _answer _is :

Figure 3: Token-level saliency heat-map for a NIAH prompt obtained using AT2 with the context length of $1K$. Each word is background-shaded according to the saliency score (darker=higher). The map reveals substantial saliency weight spilled onto distractor passage, notably high-frequency function words and punctuation.

[System Prompt] *You are a helpful . . . or repeat your findings . . .* ***[Long Irrelevant Context]*** *July 2006. . . imagine you . . . (irrelevant text continues). . . Quiet is another matter.* ***[Secret Message 1]*** ***[Irrelevant Text Resumes]*** *I realize it seems a bit eccentric . . . (irrelevant text continues) . . . Pen and paper wick ideas.* ***[Secret Message 2]*** ***[Irrelevant Text Resumes]*** *I hardly ever go back and read stuff . . . (irrelevant text continues) . . . way of having the next.* ***[Secret Message 3]*** ***[Irrelevant Text Resumes]*** *The best notebooks I've found . . . (irrelevant text continues) . . . out of space, like a Latin inscription.* ***[Query on Secret Message]*** *What is the password to get into the zone of Hagazin? Your answer is:*

Figure 4: Contradictory NIAH prompt used for the attention-bias study. The long input is color-segmented into a system instruction, alternating blocks of irrelevant context, and three mutually conflicting secret messages, message 1, message 2, and message 3, followed by the final query. The LLM chooses only one secret message to answer the query.

uniform randomness; instead, existing methods disproportionately latch onto structurally salient yet semantically irrelevant tokens, such as high-frequency function words ("the", "is", "of") or syntactic boundaries (sentence breaks, paragraph separations). This critical observation highlights a fundamental flaw: existing methods systematically confuse tokens that dominate attention structure due to frequent repetition or positional prominence with tokens genuinely contributing causally to the model's output. This exposes a systemic weakness of current input-saliency techniques: they conflate attention structure with causal contribution, and the error amplifies with additional distractors.

4 Saliency-Biased Attention for Token Attribution

In Sec. 3, we revealed a critical flaw underlying existing saliency methods: their interpretability severely degrades in long-context scenarios. This limitation naturally raises an important question: *Can we not only diagnose these attribution errors but actively intervene to rectify them?* To address this, we revisit our NIAH paradigm, this time introducing a novel and provocative modification by embedding multiple contradictory messages into the input.

Contradictory secrets reveal attention bias dynamics. We begin by embedding three *conflicting* secret messages into the input context: Message 1. *The password to get into the zone of Hagazin is "**Yin**"*, Message 2. *The password to get into the zone of Hagazin is "**Vin**"*, Message 3. *The password to get into the zone of Hagazin is "**Kin**"*. See **Fig. 4** as an illustration. At the conclusion of this deliberately confusing passage, we pose the query: "*What is the password to get into the zone of Hagazin?*" Interestingly, we observe a pronounced positional bias in the model's behavior: regardless of their semantic content, LLMs consistently favor the first encountered secret message. This phenomenon suggests that token saliency, model attention toward particular input tokens, may be implicitly influenced by their position rather than their semantic content alone.

Using attention bias to manipulate LLM's behaviors. Inspired by this intriguing observation, we probe deeper by deliberately manipulating the model's attention logits, *i.e.*, the intermediate values computed before the softmax normalization in the attention mechanism. Recall that in a standard transformer, the attention score (also called *attention logits*) between a query $\mathbf{q}_i$ and a key $\mathbf{k}_j$ is computed as: $L_{ij} = \dfrac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d_k}}$, where d_k is the key dimensionality. These logits are then passed through a softmax to yield attention weights. In order to investigate how attention scores might influence LLM's outputs, we inject an artificially positive attention bias toward tokens in the *second secret message*, gradually increasing its magnitude: $L_{ij} \leftarrow L_{ij} + \alpha_j$, where $[j]$ represents the input token IDs designed to be enhanced with this bias. **Fig. 5** vividly illustrates this experiment: as the bias towards the second secret grows, the probability of outputting the second message's password ("Vin") consistently increases, while the probability associated with the first message's password

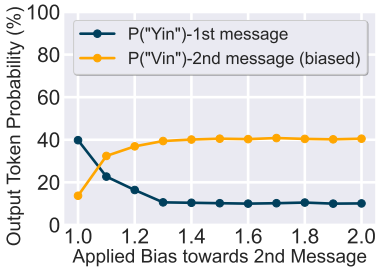

Figure 5: LLM's prediction can be shifted by biasing attention scores. Curves track the probability that the LLM will output the first-secret token "Yin" *vs.* the second-secret token "Vin", when a positive attention bias is gradually injected toward the second secret message. Increasing the bias drives *P*(Vin) upward from $\sim$14% to $\sim$40%, while *P*(Yin) falls symmetrically, and the two curves cross once the model's final answer flips from the first to the second.

("Yin") correspondingly decreases. Ultimately, the model's final prediction shifts from "Yin" to "Vin". This simple yet profound experiment reveals a deeper insight: **attention logits directly govern token saliency and thus strongly influence model outputs**. Consequently, attention biases can serve as a powerful lever for interpreting and controlling LLM behavior.

From manipulation to attribution: Attention bias optimization (ABO). Motivated by the above experiment, we propose ABO, a novel attribution framework. The central insight is straightforward yet compelling: if enhancing a token's attention logits reliably boosts the model's probability of generating a target output, it logically follows that the token is intrinsically salient to that output. Leveraging this principle, ABO reframes token attribution as a differentiable optimization problem over attention logits, thus providing a smooth and precise alternative to discrete masking or traditional gradient-based attribution approaches.

Formally, given an input sequence $\mathbf{x} = (x_1, x_2, \ldots, x_L)$ of length L and a target model-generated output token y_{target}, we introduce learnable attention biases. Specifically, we assign each input token x_i a scalar bias parameter α_i, shared across all attention heads and layers. After injecting these biases, we apply standard softmax normalization to yield final attention weights. This soft re-weighting ensures that no token is completely discarded, thus preserving the integrity of the full context. To derive meaningful, interpretable saliency scores, we optimize the parameters $\{\alpha_i\}$ by maximizing the biased model's probability of producing the target token y_{target}:

$$\mathcal{L}(\boldsymbol{\alpha}) = -\log p_{\text{bias}}\big(y_{\text{target}} \mid \mathbf{x}, \boldsymbol{\alpha}\big) + \lambda \sum_{i=1}^{L} \sigma(\alpha_i). \tag{2}$$

Here, the first term ensures tokens that genuinely contribute to generating y_{target} receive greater saliency. The second term is a carefully designed regularizer: by applying a sigmoid transformation to the biases, it penalizes excessive saliency across irrelevant tokens and promotes sparsity. This choice ensures numerical stability during optimization and yields a sparse, interpretable saliency distribution that highlights genuinely influential tokens. Unlike discrete token masking, ABO never removes tokens but instead smoothly reallocates attention weights across the entire input sequence. This soft, continuous approach provides clearer interpretability: increased bias for a token directly translates to increased model attention and influence. Furthermore, ABO addresses key drawbacks of previous methods, such as instability in gradient-based attribution and arbitrary saliency leakage, by directly coupling token saliency to the model's internal attention dynamics. Consequently, it consistently provides interpretable and robust attributions, even across extremely long contexts.

Isolating the learned bias yields a clearer and more causal attribution signal. An important design choice in ABO lies in measuring attribution through the optimized attention bias rather than the sum of the original attention value and bias. While both components jointly influence the model's output during inference, isolating the learned bias yields a cleaner and more causally interpretable signal. Original attention distributions span heterogeneous heads and layers, each capturing structural or positional regularities that may not directly reflect causal relevance. Aggregating them with the optimized bias would inevitably entangle genuine influence with such confounding patterns, thereby weakening attribution clarity. In contrast, the learned attention bias vector forms a unified and end-to-end optimized space that quantifies the minimal perturbation required for a token to affect the model's output. This separation allows ABO to measure causal sensitivity in a stable and comparable manner across contexts, avoiding complex normalization or scale mismatches that arise from mixing heterogeneous attention values.

Bias sharing across layers and heads stabilizes optimization and improves practicality. Another design decision concerns sharing the bias parameters across all attention heads and layers. Early variants of ABO explored assigning independent bias parameters to each head and layer, but this additional flexibility did not improve attribution fidelity. Instead, it introduced substantial optimization instability, as interactions among layers amplified local gradient noise and slowed convergence. Empirically, the shared-parameter formulation achieved more stable optimization dynamics and delivered consistently higher saliency precision under equal or lower computational cost. Beyond performance considerations, parameter sharing also enhances the method's practicality, reducing memory footprint, simplifying implementation, and enabling ABO to serve as a lightweight plug-in interpretability module applicable to diverse transformer architectures.

Table 1: Hit Ratio on the single output token NIAH stress test across different LLMs and context lengths. Each cell reports the percentage of saliency mass correctly assigned to the true causal tokens (system prompt + secret message + query), averaged over multiple secret-message insertion depths (from 10% to 90%) within each context and normalized to sum to 100. Higher is better.

Method	LLaMA2-7B [66]				Mistral-7B-Instruct-v0.3 [67]				DeepSeek-R1-Distilled-QWen-14B [68]			
	1K	2K	3K	4K	1K	3K	5K	10K	1K	3K	5K	10K
NAIVE	7.50	3.75	2.50	1.88	7.20	2.40	1.44	0.72	8.20	2.73	1.64	0.82
GRADNORM [4]	11.49	8.21	4.44	2.28	10.22	3.23	2.41	1.17	9.35	2.91	2.32	1.43
INPUT X GRAD [5]	12.20	4.44	3.49	1.68	13.58	3.31	2.28	1.81	10.27	2.96	2.56	1.45
TOKENSHAP [1]	17.73	14.82	9.91	5.24	15.54	10.27	6.37	3.73	16.73	11.43	5.59	3.81
TDD [2]	15.57	10.83	7.29	3.37	13.28	9.23	5.51	4.39	12.71	7.68	4.83	3.51
ATMAN [6]	19.94	16.69	12.67	4.34	20.22	13.33	7.18	3.12	18.88	12.48	6.54	2.83
AT2 [3]	21.15	18.27	12.28	6.62	20.43	14.41	10.03	5.59	21.76	13.65	9.86	4.99
ABO	**41.75**	**31.72**	**25.54**	**18.83**	**45.34**	**27.77**	**18.92**	**15.43**	**38.89**	**25.14**	**17.37**	**15.44**

5 Experiments

To systematically validate the accuracy, robustness, and utility of our proposed ABO method, we conduct systematic and comprehensive experimental evaluations. The first part (Sec. 5.1) focuses on quantitatively comparing the attribution accuracy of ABO against several strong baselines in controlled scenarios, including single-token and multi-token NIAH stress tests, as well as an input token pruning evaluation in the long-context setting. The second part explores broader applications of ABO, illustrating how accurate saliency attribution can facilitate model behavior manipulation (Sec. 5.2 and Sec. 5.3), and error correction on LLM's input on reasoning tasks (Sec. 5.4).

5.1 Saliency Attribution Accuracy

Single-token NIAH attribution analysis. First, we evaluate methods using the previously described single-token NIAH setup, measuring their Hit Ratio, *i.e.*, the fraction of saliency correctly attributed to truly causal tokens. As summarized in **Tab. 1**, ABO consistently achieves substantial improvements over all baselines across different models and context lengths. Crucially, ABO maintains a robust Hit Ratio of approximately $15 \sim 45\%$, even as input length expands from 1K to 10K tokens, while the strongest baseline (TOKENSHAP & AT2) rapidly deteriorates to below 5%. This stark contrast underscores ABO's unique resilience in handling long-context noise environments. Additionally, ABO's superior performance generalizes robustly across architectures, indicating that its advantage stems fundamentally from its principled attention bias optimization mechanism, rather than model-specific artifacts.

Multi-token NIAH attribution analysis. Second, to challenge the saliency methods further, we extend the NIAH setting to secret messages containing multiple target tokens, requiring methods to correctly distribute saliency across multiple output tokens. Results presented in **Tab. 2** affirm ABO's clear superiority: although moving from single-token to multi-token attribution naturally reduces Hit Ratios slightly for all methods, ABO experiences the mildest performance degradation (around $4 \sim 5\%$), maintaining approximately $13 \sim 40\%$ accuracy. In contrast, leading baselines suffer a more substantial drop towards below 10%, highlighting ABO's superior capability to capture meaningful attribution across multiple output tokens simultaneously. The persistent robustness of ABO even under increased complexity strongly advocates for its suitability in realistic applications, where multiple tokens frequently contribute jointly to model decisions.

Long-context token pruning evaluation. We further validate saliency accuracy in practical scenarios by conducting input token pruning experiments on the challenging `LongBench-2wikimqa_e` dataset [69]. Tokens are pruned in ascending order of estimated saliency, with QA accuracy measured at progressively higher pruning ratios (see **Tab. 3**). Remarkably, ABO outperforms all baselines, especially under aggressive pruning conditions. For example, at a 95% pruning ratio (retaining only 5% of tokens), ABO achieves a QA accuracy of 51.2% on QWen3-14B, significantly higher than the best baseline's 36.2%. Furthermore, ABO exhibits a flatter accuracy degradation curve across all models tested, demonstrating a stronger ability to reliably identify essential context information. These results confirm ABO's practical value: not only does it accurately reflect token importance,

Table 2: Hit Ratio on the NIAH stress test across with mutiple output tokens different LLMs and context lengths. Other settings follow Tab. 1.

Method	LLaMA2-7B [66]				Mistral-7B-Instruct-v0.3 [67]				DeepSeek-R1-Distilled-QWen-14B [68]			
	1K	2K	3K	4K	1K	3K	5K	10K	1K	3K	5K	10K
NAIVE	7.50	3.75	2.50	1.88	7.20	2.40	1.44	0.72	8.20	2.73	1.64	0.82
GRADNORM [4]	8.39	6.04	3.70	2.12	9.28	3.11	2.20	1.75	8.87	2.69	1.89	1.11
INPUT X GRAD [5]	8.73	5.45	3.51	2.03	8.85	2.98	1.92	1.56	9.17	2.82	2.15	1.15
TOKENSHAP [1]	12.13	9.91	4.12	3.98	12.25	10.38	5.73	3.31	11.42	5.51	4.11	2.35
TDD [2]	10.89	8.73	4.31	3.29	9.87	8.82	3.89	2.17	9.91	4.16	3.06	2.05
ATMAN [6]	13.27	10.29	4.98	3.71	15.55	12.27	7.73	5.41	10.40	4.41	3.71	1.51
AT2 [3]	15.37	12.48	5.49	4.41	13.39	10.12	5.32	3.19	16.72	6.75	4.99	2.86
ABO	**38.32**	**28.99**	**21.45**	**14.49**	**40.93**	**24.49**	**14.37**	**13.18**	**34.79**	**21.19**	**15.45**	**13.61**

but it also provides actionable insights enabling substantial runtime efficiency by safely pruning non-essential tokens.

Table 3: QA accuracy (%) on the `LongBench-2wikimqa_e` multi-choice task after pruning input tokens based on saliency scores. Each column indicates the pruning ratio.

Method	QWen3-8B					Mistral-Small-24B-Instruct-2501					QWen3-14B				
	10%	50%	90%	95%	97%	10%	50%	90%	95%	97%	10%	50%	90%	95%	97%
NAIVE	48.0	39.5	5.3	0.6	0.0	54.4	42.8	4.3	0.0	0.0	59.0	45.5	3.6	0.0	0.0
GRADNORM	58.6	56.9	35.4	8.9	3.2	66.0	65.3	38.1	9.4	3.6	80.5	79.6	41.0	10.1	4.2
INPUT X GRAD	58.4	56.7	36.0	8.5	2.8	66.9	65.1	38.5	9.0	3.3	80.2	79.3	41.2	9.8	3.8
TDD	58.5	55.6	33.2	7.2	2.4	67.6	63.9	36.8	8.3	2.9	78.4	77.3	39.1	9.1	3.5
ATMAN	59.1	58.6	44.8	28.2	5.2	67.1	66.5	49.5	31.3	8.3	81.8	80.9	54.1	34.1	9.1
AT2	59.4	58.9	46.3	30.0	7.8	67.4	66.9	51.0	32.5	8.7	82.1	81.2	56.0	36.2	9.5
Ours	**59.3**	**59.0**	**52.6**	**40.3**	**9.9**	**67.5**	**67.1**	**58.7**	**47.0**	**12.2**	**82.3**	**81.7**	**65.1**	**51.2**	**15.6**

5.2 LLM Saliency for Zero-Shot Toxicity Suppression.

We further demonstrate the practical utility of accurate saliency attribution through the task of zero-shot toxic language suppression. Despite recent advancements, LLMs remain prone to generating toxic outputs in response to certain inputs [70, 71]. Unlike prior reactive approaches aimed at mitigating toxicity post-generation [72–74, 74], our approach proactively identifies and removes toxic triggers directly from the prompt, ensuring safer downstream generation.

Table 4: Toxicity suppression results on REALTOXICPROMPTS [72] using Mistral-7B-Instruct-v0.3 [67]. Scores on six toxicity attributes provided by the Perspective API; lower values indicate better suppression.

Method	Toxicity	Severe Toxicity	Sexually Explicit	Threat	Profanity	Identify Attack
Original	0.53	0.24	0.37	0.23	0.42	0.19
TDD	0.36	0.21	0.29	0.18	0.33	0.14
TOKENSHAP	0.41	0.18	0.17	0.14	0.26	0.12
AT2	0.28	0.15	0.21	0.11	0.14	0.09
ABO	**0.24**	**0.12**	**0.14**	**0.08**	**0.07**	**0.05**

Experiment setup. Specifically, we utilize the REALTOXICPROMPTS [72] dataset, comprising 1,225 prompts known for eliciting highly toxic outputs, and evaluate our method using the Mistral-7B-Instruct-v0.3 model [67]. Given our rigorous assessment of saliency attribution methods in the previous subsection, we selectively benchmark against the top-performing saliency approaches, namely TOKENSHAP [1], AT2 [3], and TDD [2]. During the experiment, we first designate the predefined toxic words from WORDFILTER [72] as target tokens, identifying critical tokens (toxic triggers) via saliency scores, which we subsequently neutralize by replacing them with meaningless space tokens. Following the procedure of prior work [73, 75], we limit generations to 20 tokens and evaluate the resulting toxicity using six toxicity attributes provided by Perspective API, ensuring a clear and rigorous comparison of saliency-based toxicity suppression efficacy among our chosen baseline methods.

Results and analysis. As shown in **Tab. 4**, our method achieves the lowest toxicity scores across all six attributes, substantially outperforming all baselines. Compared to the original outputs, ABO

Table 5: Sentiment steering results on OPENWEBTEXT [76] using Mistral-7B-Instruct-v0.3 [67]. Generated outputs are classified via Huggingface sentiment analysis [77]; higher indicate stronger steering efficacy.

Method	Neutral → Negative Negative Ratio (%)	Neutral → Positive Positive Ratio (%)
Original	53.6	46.4
TDD	75.4	59.9
TOKENSHAP	80.2	68.3
AT2	69.3	66.7
ABO	**89.7**	**81.5**

reduces overall toxicity by more than half, and consistently yields the strongest suppression in severe categories such as *threat*, *profanity*, and *identity attack*. While AT2 emerges as the strongest baseline, it still lags behind across every metric. These results confirm that our saliency estimates more accurately pinpoint true toxic triggers, enabling more effective prompt sanitization and offering a robust foundation for safe LLM deployment.

5.3 LLM Saliency for Sentiment Steering

We next demonstrate ABO's applicability in zero-shot sentiment steering, an essential task for safely and reliably guiding LLM-generated content towards a desired sentiment polarity.

Experiment setup. To rigorously assess our approach, we use 5,000 neutral prompts sourced from the OPENWEBTEXT [76] corpus to feed the Mistral-7B-Instruct-v0.3 model [67]. Specifically, we designate negative sentiment words from SENTICNET [78] as target tokens and positive words as alternative tokens when steering towards positive sentiment, and reverse this assignment when steering negatively. Subsequently, we utilize saliency attribution methods, focusing exclusively on the highest-performing methods identified previously: TOKENSHAP [1], AT2 [3], and ATMAN [6] to pinpoint the single most salient sentiment cue within each prompt. This identified token is then directly replaced by the explicit keyword (either "positive" or "negative" accordingly) to guide the model's sentiment. Following prior work [79], each prompt undergoes exactly one token replacement due to their relatively short lengths. For evaluation, we exclusively report the sentiment distributions of generated outputs using the Huggingface sentiment analysis classifier [77], clearly reflecting the efficacy of each method in steering sentiment polarity.

Results and analysis. As shown in **Tab. 5**, a single ABO-guided token edit shifts *89.7%* of neutral prompts to negative and *81.5%* to positive sentiment, outperforming the strongest baseline (TOKENSHAP: $80.2\%/68.3\%$). The gap widens in the harder "neutral→positive" scenario, highlighting ABO's ability to pinpoint the *single* most causal sentiment cue. Together, these results confirm that bias optimization yields sharper, more actionable saliency than existing heuristics, enabling reliable zero-shot sentiment control with minimal input modification.

[Question] In a dance class of 20 students, 20% enrolled in contemporary dance, 25% of the remaining enrolled in jazz dance, and the rest enrolled in hip-hop dance. What percentage of the students enrolled in hip-hop dance?

[**Reasoning (Incorrect by Default)**] <think> To determine the percentage of students enrolled in hip-hop dance, let's break down the information provided: ... Students enrolled in Jazz Dance: (25%) of 20 students \[~~0.25 × 20 = 5~~ students \] ... <\think> Final Answer: \(\boxed55%\)

[Reasoning (Correct with Attention Bias)] <think> To determine the percentage of ... Students enrolled in Jazz Dance: (25%) of the **remaining** students after contemporary dance \[20 - 4 = 16 students \] Then, 25% of 16: \[**0.25 × 16 = 4** students \]... <\think> Final Answer: \(\boxed60%\)

Figure 6: Attention bias fixes a reasoning error. Highlighting the token "remaining" (yellow) and adding a positive bias to its attention logits redirects the model to the correct interpretation, changing the wrong 55% answer (left, red) to the correct 60% answer (right, green).

5.4 LLM Error Correction via ABO

Beyond interpretability and behavior steering, our ABO (attention bias optimization) approach provides a novel pathway for error correction during reasoning tasks performed by LLMs. Despite their impressive reasoning capabilities, even advanced reasoning-centric models such as DeepSeek-R1-Distilled-QWen-7B [68] occasionally commit errors due to misreading critical elements within problem statements. To illustrate this, consider a simple mathematical question from the GSM8K dataset [80] depicted in **Fig. 6**. Initially, the model erroneously overlooks the keyword *"remaining"*, consequently misunderstanding the problem statement and generating an incorrect answer. However, by explicitly introducing a positive attention bias to the token corresponding to the overlooked keyword, we effectively realign the model's attention, enabling the model to correctly interpret the question and subsequently correct its output.

Motivated by this observed phenomenon, we systematically examine a broader set of examples to quantitatively assess the efficacy of our attention-based error correction strategy. Specifically, we identify 50 distinct math reasoning problems from the GSM8K dataset initially answered incorrectly due to misreading or overlooking crucial textual cues. For each problem, we manually select and positively bias the attention logits corresponding to the critical tokens identified as frequently misinterpreted. Remarkably, after applying such targeted attention biases, 45 out of these 50 problems (90%) were subsequently answered correctly without any modifications to the input text itself. This result strongly indicates that the errors were indeed a consequence of attentional oversight rather than inherent model deficiencies. See Appx. B for more examples.

6 Limitation and Conclusion

We revisited *input saliency* for LLMs and showed that six popular attribution methods collapse in long-context settings, misplacing most of their mass on distractors. Our NIAH (needle-in-a-haystack) stress test makes this failure mode transparent and measurable. Building on the empirical insight that tiny shifts in attention logits can deterministically flip a model's choice among contradictory messages, we introduced ABO, which frames attribution as a lightweight per-token bias optimization on attention logits. We empirically demonstrated the accuracy of ABO's saliency scores and its ability in zero-shot detoxification, sentiment steering, and correction of reasoning errors.

Despite these promising results, this work also reveals a more fundamental limitation of current *input saliency* research for LLMs. Our evaluation shows that the gap between how saliency methods are *perceived* to work and how they actually behave in practice remains surprisingly large. While ABO substantially narrows this gap relative to prior methods, its absolute performance still leaves ample room for improvement. This persistent discrepancy stems from the inherent scale and complexity of modern LLMs, where attribution noise, token interactions, and non-linear dependencies grow rapidly with model size. Consequently, even small estimation errors in saliency can accumulate and distort the interpretation, posing a challenge that goes beyond the reach of any single method.

From an algorithmic perspective, ABO's per-example optimization introduces additional variance across runs and examples, especially when the target likelihood landscape is highly non-convex. This sensitivity makes results less stable in certain contexts and increases computational cost for large-scale deployment. Future work could explore amortizing bias inference through meta-learning or distillation-based techniques, allowing approximate but faster and more consistent saliency estimation. Furthermore, our current stress test mainly targets retrieval-style prompts; extending it to multi-step reasoning, dialogue, and multimodal tasks would provide a broader view of attribution fidelity under real-world conditions. Overall, ABO represents an encouraging step toward causally grounded saliency analysis for large language models. It not only improves attribution fidelity under long-context noise but also exposes deeper methodological gaps in how we interpret complex neural systems, gaps that will require sustained community effort to close.

Acknowledgment

This work was supported by the National Science Foundation (NSF) CISE Core Program Award IIS-2207052, the NSF Cyber-Physical Systems (CPS) Award CNS-2235231, the NSF CAREER Award IIS-2338068, the ARO Award W911NF2310343, the Cisco Research Award, the Amazon Research Award, and the IBM PhD Fellowship Award.

References

[1] R. Goldshmidt and M. Horovicz, "Tokenshap: Interpreting large language models with monte carlo shapley value estimation," *arXiv preprint arXiv:2407.10114*, 2024.

[2] Z. Feng, H. Zhou, Z. Zhu, J. Qian, and K. Mao, "Unveiling and manipulating prompt influence in large language models," *arXiv preprint arXiv:2405.11891*, 2024.

[3] B. Cohen-Wang, Y.-S. Chuang, and A. Madry, "Learning to attribute with attention," *arXiv preprint arXiv:2504.13752*, 2025.

[4] K. Yin and G. Neubig, "Interpreting language models with contrastive explanations," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 184–198. [Online]. Available: https://aclanthology.org/2022.emnlp-main.14

[5] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 3319–3328.

[6] B. Deiseroth, M. Deb, S. Weinbach, M. Brack, P. Schramowski, and K. Kersting, "Atman: Understanding transformer predictions through memory efficient attention manipulation," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[7] S. Feng, E. Wallace, A. Grissom II, M. Iyyer, P. Rodriguez, and J. Boyd-Graber, "Pathologies of neural models make interpretations difficult," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 3719–3728. [Online]. Available: https://aclanthology.org/D18-1407/

[8] H. Chefer, S. Gur, and L. Wolf, "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 397–406.

[9] D. Nam, A. Macvean, V. Hellendoorn, B. Vasilescu, and B. Myers, "Using an llm to help with code understanding," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–13.

[10] M. Kazemitabaar, R. Ye, X. Wang, A. Z. Henley, P. Denny, M. Craig, and T. Grossman, "Codeaid: Evaluating a classroom deployment of an llm-based programming assistant that balances student and educator needs," in *Proceedings of the 2024 chi conference on human factors in computing systems*, 2024, pp. 1–20.

[11] X. Huang, W. Liu, X. Chen, X. Wang, H. Wang, D. Lian, Y. Wang, R. Tang, and E. Chen, "Understanding the planning of llm agents: A survey," *arXiv preprint arXiv:2402.02716*, 2024.

[12] A. Ali, T. Schnake, O. Eberle, G. Montavon, K.-R. Müller, and L. Wolf, "XAI for transformers: Better explanations through conservative propagation," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 435–451. [Online]. Available: https://proceedings.mlr.press/v162/ali22a.html

[13] M. Geva, R. Schuster, J. Berant, and O. Levy, "Transformer feed-forward layers are key-value memories," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 5484–5495. [Online]. Available: https://aclanthology.org/2021.emnlp-main.446/

[14] M. Geva, A. Caciularu, K. Wang, and Y. Goldberg, "Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 30–45. [Online]. Available: https://aclanthology.org/2022.emnlp-main.3/

[15] G. Dar, M. Geva, A. Gupta, and J. Berant, "Analyzing transformers in embedding space," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 16 124–16 170. [Online]. Available: https://aclanthology.org/2023.acl-long.893/

[16] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 4190–4197. [Online]. Available: https://aclanthology.org/2020.acl-main.385/

[17] A. Modarressi, M. Fayyaz, Y. Yaghoobzadeh, and M. T. Pilehvar, "GlobEnc: Quantifying global token attribution by incorporating the whole encoder layer in transformers," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 258–271. [Online]. Available: https://aclanthology.org/2022.naacl-main.19/

[18] B. Cohen-Wang, H. Shah, K. Georgiev, and A. Madry, "Contextcite: Attributing model generation to context," *Advances in Neural Information Processing Systems*, vol. 37, pp. 95 764–95 807, 2024.

[19] T. R.-A. Generation, "Model internals-based answer attribution for trustworthy retrieval-augmented generation."

[20] J. Vig, S. Gehrmann, Y. Belinkov, S. Qian, D. Nevo, S. Sakenis, J. Huang, Y. Singer, and S. Shieber, "Causal mediation analysis for interpreting neural nlp: the case of gender bias (2020)," *arXiv preprint arXiv:2004.12265*, 2004.

[21] V. Prabhakaran, B. Hutchinson, and M. Mitchell, "Perturbation sensitivity analysis to detect unintended model biases," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5740–5745. [Online]. Available: https://aclanthology.org/D19-1578/

[22] J. Ferrando, G. I. Gállego, and M. R. Costa-jussà, "Measuring the mixing of contextual information in the transformer," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 8698–8714. [Online]. Available: https://aclanthology.org/2022.emnlp-main.595/

[23] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017, pp. 4768–4777.

[24] E. Wallace, J. Tuyls, J. Wang, S. Subramanian, M. Gardner, and S. Singh, "AllenNLP interpret: A framework for explaining predictions of NLP models," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, S. Padó and R. Huang, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 7–12. [Online]. Available: https://aclanthology.org/D19-3002/

[25] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," *arXiv preprint arXiv:2307.03172*, 2023.

[26] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: visualising image classification models and saliency maps," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014, pp. 1–8.

[27] J. Li, X. Chen, E. Hovy, and D. Jurafsky, "Visualizing and understanding neural models in NLP," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 681–691. [Online]. Available: https://aclanthology.org/N16-1082

[28] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein, "A diagnostic study of explainability techniques for text classification," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 3256–3274. [Online]. Available: https://aclanthology.org/2020.emnlp-main.263

[29] M. Denil, A. Demiraj, and N. De Freitas, "Extraction of salient sentences from labelled documents," *arXiv preprint arXiv:1412.6815*, 2014.

[30] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International conference on machine learning*. PMLR, 2017, pp. 3145–3153.

[31] E. Wallace, J. Tuyls, J. Wang, S. Subramanian, M. Gardner, and S. Singh, "AllenNLP interpret: A framework for explaining predictions of NLP models," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 7–12. [Online]. Available: https://aclanthology.org/D19-3002

[32] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[33] A. Ali, T. Schnake, O. Eberle, G. Montavon, K.-R. Müller, and L. Wolf, "XAI for transformers: Better explanations through conservative propagation," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 435–451. [Online]. Available: https://proceedings.mlr.press/v162/ali22a.html

[34] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 4190–4197. [Online]. Available: https://aclanthology.org/2020.acl-main.385

[35] A. Modarressi, M. Fayyaz, Y. Yaghoobzadeh, and M. T. Pilehvar, "GlobEnc: Quantifying global token attribution by incorporating the whole encoder layer in transformers," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 258–271. [Online]. Available: https://aclanthology.org/2022.naacl-main.19

[36] J. Ferrando, G. I. Gállego, and M. R. Costa-jussà, "Measuring the mixing of contextual information in the transformer," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 8698–8714. [Online]. Available: https://aclanthology.org/2022.emnlp-main.595

[37] S. Feng, E. Wallace, A. Grissom II, M. Iyyer, P. Rodriguez, and J. Boyd-Graber, "Pathologies of neural models make interpretations difficult," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 3719–3728. [Online]. Available: https://aclanthology.org/D18-1407

[38] V. Prabhakaran, B. Hutchinson, and M. Mitchell, "Perturbation sensitivity analysis to detect unintended model biases," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5740–5745. [Online]. Available: https://aclanthology.org/D19-1578

[39] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[40] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *arXiv preprint arXiv:2107.13586*, 2021.

[41] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.

[42] H. Strobelt, A. Webson, V. Sanh, B. Hoover, J. Beyer, H. Pfister, and A. M. Rush, "Interactive and visual prompt engineering for ad-hoc task adaptation with large language models," 2022.

[43] S. H. Bach, V. Sanh, Z.-X. Yong, A. Webson, C. Raffel, N. V. Nayak, A. Sharma, T. Kim, M. S. Bari, T. Fevry *et al.*, "Promptsource: An integrated development environment and repository for natural language prompts," *arXiv preprint arXiv:2202.01279*, 2022.

[44] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "Autoprompt: Eliciting knowledge from language models with automatically generated prompts," *arXiv preprint arXiv:2010.15980*, 2020.

[45] M. Deng, J. Wang, C.-P. Hsieh, Y. Wang, H. Guo, T. Shu, M. Song, E. P. Xing, and Z. Hu, "Rlprompt: Optimizing discrete text prompts with reinforcement learning," *arXiv preprint arXiv:2205.12548*, 2022.

[46] C. Singh, J. X. Morris, J. Aneja, A. M. Rush, and J. Gao, "Explaining patterns in data with language models via interpretable autoprompting," 2023.

[47] A. Webson and E. Pavlick, "Do prompt-based models really understand the meaning of their prompts?" *arXiv preprint arXiv:2109.01247*, 2021.

[48] Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp, "Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity," *arXiv preprint arXiv:2104.08786*, 2021.

[49] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," *arXiv preprint arXiv:2109.01652*, 2021.

[50] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma *et al.*, "Scaling instruction-finetuned language models," *arXiv preprint arXiv:2210.11416*, 2022.

[51] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, "Fine-tuning language models from human preferences," *arXiv preprint arXiv:1909.08593*, 2019.

[52] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[53] H. Lee, S. Phatale, H. Mansoor, K. Lu, T. Mesnard, C. Bishop, V. Carbune, and A. Rastogi, "Rlaif: Scaling reinforcement learning from human feedback with ai feedback," *arXiv preprint arXiv:2309.00267*, 2023.

[54] A. S. Ross, M. C. Hughes, and F. Doshi-Velez, "Right for the right reasons: Training differentiable models by constraining their explanations," *arXiv preprint arXiv:1703.03717*, 2017.

[55] L. Rieger, C. Singh, W. J. Murdoch, and B. Yu, "Interpretations are useful: penalizing explanations to align neural networks with prior knowledge," *arXiv preprint arXiv:1909.13584*, 2019.

[56] P. Schramowski, W. Stammer, S. Teso, A. Brugger, F. Herbert, X. Shao, H.-G. Luigs, A.-K. Mahlein, and K. Kersting, "Making deep neural networks right for the right scientific reasons by interacting with their explanations," *Nature Machine Intelligence*, vol. 2, no. 8, pp. 476–486, 2020.

[57] S. Krishna, J. Ma, D. Slack, A. Ghandeharioun, S. Singh, and H. Lakkaraju, "Post hoc explanations of language models can improve language models," *arXiv preprint arXiv:2305.11426*, 2023.

[58] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[59] Q. Zhang, M. Chen, A. Bukharin, P. He, Y. Cheng, W. Chen, and T. Zhao, "Adaptive budget allocation for parameter-efficient fine-tuning," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=lq62uWRJjiY

[60] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," 2023.

[61] B. Shi, S. Gai, T. Darrell, and X. Wang, "Refocusing is key to transfer learning," *arXiv preprint arXiv:2305.15542*, 2023.

[62] K. Meng, D. Bau, A. Andonian, and Y. Belinkov, "Locating and editing factual associations in gpt," *Advances in Neural Information Processing Systems*, vol. 35, pp. 17 359–17 372, 2022.

[63] K. Meng, A. S. Sharma, A. Andonian, Y. Belinkov, and D. Bau, "Mass-editing memory in a transformer," *arXiv preprint arXiv:2210.07229*, 2022.

[64] E. Mitchell, C. Lin, A. Bosselut, C. Finn, and C. D. Manning, "Fast model editing at scale," 2022.

[65] E. Hernandez, B. Z. Li, and J. Andreas, "Inspecting and editing knowledge representations in language models," 2023.

[66] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[67] F. Jiang, "Identifying and mitigating vulnerabilities in llm-integrated applications," Master's thesis, University of Washington, 2024.

[68] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[69] Y. Bai, X. Lv, J. Zhang, H. Lyu, J. Tang, Z. Huang, Z. Du, X. Liu, A. Zeng, L. Hou, Y. Dong, J. Tang, and J. Li, "Longbench: A bilingual, multitask benchmark for long context understanding," 2023.

[70] B. Wang, W. Chen, H. Pei, C. Xie, M. Kang, C. Zhang, C. Xu, Z. Xiong, R. Dutta, R. Schaeffer *et al.*, "Decodingtrust: A comprehensive assessment of trustworthiness in gpt models," *arXiv preprint arXiv:2306.11698*, 2023.

[71] Y. Liu, Y. Yao, J.-F. Ton, X. Zhang, R. G. H. Cheng, Y. Klochkov, M. F. Taufiq, and H. Li, "Trustworthy llms: a survey and guideline for evaluating large language models' alignment," *arXiv preprint arXiv:2308.05374*, 2023.

[72] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, "RealToxicityPrompts: Evaluating neural toxic degeneration in language models," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 3356–3369. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.301

[73] M. Geva, A. Caciularu, K. Wang, and Y. Goldberg, "Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 30–45. [Online]. Available: https://aclanthology.org/2022.emnlp-main.3

[74] E. Reif, D. Ippolito, A. Yuan, A. Coenen, C. Callison-Burch, and J. Wei, "A recipe for arbitrary text style transfer with large language models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 837–848. [Online]. Available: https://aclanthology.org/2022.acl-short.94

[75] T. Schick, S. Udupa, and H. Schütze, "Self-Diagnosis and Self-Debiasing: A Proposal for Reducing Corpus-Based Bias in NLP," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1408–1424, 12 2021. [Online]. Available: https://doi.org/10.1162/tacl_a_00434

[76] A. Gokaslan and V. Cohen, "Openwebtext corpus," 2019.

[77] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.6

[78] E. Cambria, R. Speer, C. Havasi, and A. Hussain, "Senticnet: A publicly available semantic resource for opinion mining," in *2010 AAAI fall symposium series*, 2010.

[79] A. Liu, M. Sap, X. Lu, S. Swayamdipta, C. Bhagavatula, N. A. Smith, and Y. Choi, "DExperts: Decoding-time controlled text generation with experts and anti-experts," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 6691–6706. [Online]. Available: https://aclanthology.org/2021.acl-long.522

[80] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman, "Training verifiers to solve math word problems," *arXiv preprint arXiv:2110.14168*, 2021.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the main claims in the abstract and introduction accurately reflect the paper's contributions and scope, including the contributions made in the paper and important assumptions and limitations.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: limitations are discussed in Sec. 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: the paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Detailed setups and hyper-parmaeters can be found in Appx. A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The data used in this paper are fully open datasets. The code and instructions are included in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper is inference-time saliency analysis, so does not involve data spliting. Detailed experiment setups are reported both in Sec. 5 and Appx. A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Although the paper does not explicitly report error bars for each data point, this is understandable given the large volume of quantitative results, where including error bars for every entry could significantly reduce readability. However, the authors state that all reported results are averaged over three independent runs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Detailed computational resources are reported in Appx. A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors confirm that in every respect, the research is not against NeurIPS code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The societal impact statement is included in Appx. C.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: the paper poses no such risks

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the open datasets and models are properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: the paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: the core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

A Experiment Setup

All saliency methods were normalized to ensure that their respective attribution scores sum to one.

Gradient Norm (GRADNORM) computes saliency scores based on the ℓ_2 norm of the gradient of the model's output with respect to the input token embeddings. Specifically, the score for each token x_i is calculated by taking the L1 norm of the gradient $\nabla_{x_i} q(y_t|x)$, where $q(y_t|x)$ is the model's predicted probability for token y_t given input x.

Gradient × Input (INPUT×GRAD) modifies the Gradient Norm approach by calculating the dot product between each input embedding x_i and its gradient $\nabla_{x_i} q(y_t|x)$, thereby emphasizing tokens highly expressed in the model's predictive process.

Attention Rollout (ROLLOUT) tracks the flow of attention information through transformer layers by recursively multiplying attention matrices across layers. Given raw attention matrices $A(l_i)$ at each layer l_i, attention from higher layers is multiplied downwards layer by layer, creating cumulative attention maps that reflect the propagation of token importance from input layers up to the output layer (`https://github.com/hmohebbi/context_mixing_toolkit`).

TokenSHAP (SHAP) estimates Shapley values for tokens via Monte Carlo sampling. The importance of each token is determined by evaluating its marginal contribution to the model's output across randomly sampled subsets of tokens, combining essential combinations with randomly sampled ones to ensure both computational efficiency and accurate estimation (`https://github.com/ronigold/TokenSHAP`).

Token Distribution Dynamics (TDD) analyzes token saliency using token distribution projections within the embedding space via the language model head. Variants such as forward, backward, and bidirectional TDD measure token saliency through their distributional shifts and causal contributions to model predictions (`https://github.com/zijian678/TDD`).

Attention Manipulation (ATMAN) generates relevance maps through memory-efficient perturbations of transformer attention mechanisms. It applies token-level perturbations directly within the attention score space, reducing computational overhead and facilitating efficient interpretation of model decisions (`https://github.com/Aleph-Alpha/AtMan`).

Attribution with Attention (AT2) treats attention weights from individual heads as learnable features. This method uses ablation-based training to learn attention-head coefficients, enabling efficient yet accurate token attribution across different contexts and significantly reducing computational requirements (`https://github.com/MadryLab/AT2`).

ABO: For ABO, we optimized the bias parameters $\{\alpha_i\}$ using the Adam optimizer with a typical learning rate of 0.01, applying a decay schedule after a fixed number of iterations. The optimization was capped at 50 to 200 steps, depending on the context length, with early stopping triggered by the convergence of the target token probability. To promote sparsity and focus in the saliency distribution, we applied a sigmoid-transformed ℓ_1 regularizer with weight λ ranging between 10^{-1} and 1. All bias parameters were initialized randomly to avoid introducing inductive bias at initialization.

A.1 Compute Environment

All experiments were conducted exclusively for evaluation purposes, with no model training or fine-tuning involved. We utilized NVIDIA H100 GPUs for all runs, and the total compute consumption was approximately 1000 GPU hours. This estimate includes forward passes, saliency score extraction, attention bias optimization, and repeated runs for multi-seed robustness analysis across multiple saliency methods and model backbones.

B Additional Experiment Results

In this section, we provide additional qualitative examples (Fig. A1∼Fig. A5) to illustrate how BPO can be used to correct model reasoning errors by redirecting attention to overlooked yet critical tokens. These examples are drawn from the GSM8K dataset and demonstrate instances where the

[Question] Paul went to a shop to buy some groceries. He bought some bread for 2, butter for 3, and juice for two times the price of the bread. He had 15 for his shopping. How much money did Paul have left?

[**Reasoning (Incorrect by Default)**] <think> Let's determine how much money Paul has left after his shopping step by step. The bread costs 2. ... The juice costs two times the price of the ~~butter~~ ... <\think> Final Answer: \(\boxed4\)

[Reasoning (Correct with Attention Bias)]<think> Let's determine how much money Paul has left after his shopping step by step. The bread costs 2. ... The juice costs two times the price of the **bread** ... <\think> Final Answer: \(\boxed6\)

Figure A1: **Example 1.** Initially, the model misidentifies "juice costs twice the butter" instead of "twice the bread". With attention bias on the word "bread," the model corrects its interpretation and computes the correct remaining balance.

[Question] There are some jelly beans in a jar. Three fourths of the red jelly beans are red, and one quarter of the jelly beans are coconut flavored. If 750 jelly beans are coconut flavored, how many jelly beans are there in the jar?

[**Reasoning (Incorrect by Default)**] <think> Let's solve the problem step by step. Given: ... Since one quarter of the ~~total jelly beans~~ equals 750, I can set up the equation: ... <\think> Final Answer: \(\boxed3000\)

[Reasoning (Correct with Attention Bias)]<think> ... That means if I have a total number of jelly beans, say, T, then **the number of red jelly beans** would be (3/4) * T. ... <\think> Final Answer: \(\boxed4000\)

Figure A2: **Example 2.** The model initially misattributes the fraction to total jelly beans. Highlighting "red" refocuses attention on the subject of the fraction, enabling correct inference of the jar size.

model initially produced incorrect answers due to misinterpretation. By applying positive bias to the appropriate input tokens, we successfully guide the model to revise its understanding and produce the correct answers—without modifying the input text. Each example shows the original faulty reasoning on the left and the corrected reasoning with attention bias on the right. These results showcase the potential of attention-level interventions for LLM robustness and post hoc debugging.

C Broader Impact

This work advances the field of trustworthy AI by proposing Attention Bias Optimization (ABO), a principled method for accurate token attribution in large language models (LLMs). By identifying and correcting systemic flaws in six widely-used saliency methods, our work enables more faithful, scalable, and robust interpretability, especially in long-context settings.

From an ethical standpoint, ABO enhances transparency in LLM behavior, reducing the risk of misinterpretation in high-stakes applications such as legal analysis, education, and healthcare. However, accurate attribution also raises dual-use concerns: malicious actors could leverage improved interpretability to reverse-engineer sensitive prompts or fine-tune manipulative behavior. Future research should explore safeguards that balance interpretability with misuse prevention.

At the societal level, our method supports the development of safer AI systems by enabling applications such as zero-shot toxicity suppression, error correction, and sentiment steering. These capabilities can improve the reliability and controllability of AI assistants and mitigate harms caused by misaligned or biased outputs. Still, we caution that attribution is not equivalent to accountability; robust evaluation and user education remain necessary.

[Question] A company pays each of its employees 600 in a month. The company has a policy of increasing the salaries of each of its employees by 10% of the initial salary every year for those who've stayed in the company for five years. If Sylvie just clocked 5 years in the company last December, what's her annual salary after three more years of service?

[**Reasoning (Incorrect by Default)**] <think> Alright, so I have this problem about Sylvie's salary ... First, it says that the company pays each employee ~~600 per year~~. ...<\think> Final Answer: \(\boxed780\)

[Reasoning (Correct with Attention Bias)]<think> First, I need to determine Sylvie's current monthly salary. Since she has been with the ... Starting salary: **600 per month**. ... <\think> Final Answer: \(\boxed10,541.52\)

Figure A3: **Example 3.** The model mistakes the salary to be yearly instead of monthly. Biasing the word "month" shifts the model toward the correct interpretation and output.

[Question] Brian's friend Bobby has 5 fewer than 3 times as many video games as Brian does. If Brian has 20 video games but lost 5 right after the comparison was made, how many does Bobby have?

[**Reasoning (Incorrect by Default)**] <think> Okay, let's tackle this problem step by step. ... Determine the current number of video games Brian has: Originally, Brian has 20 video games. He lost 5 video games. ~~Current number of Brian's games= $20 - 5 = 15$~~ ... <\think> Final Answer: \(\boxed40\)

[Reasoning (Correct with Attention Bias)]<think> Alright, let's try to solve this problem step by step. ... The phrase implies that the comparison occurred, and then Brian lost 5. So, it seems like the comparison was done when Brian had 20 games, **and after that, he lost 5**. ...<\think> Final Answer: \(\boxed55\)

Figure A4: **Example 4.** The model mistakenly adjusts the number of video games after the comparison. Emphasizing the word "after" leads to the correct timing of the event and the correct answer.

We encourage continued efforts to integrate interpretability into broader frameworks of AI alignment, governance, and fairness auditing to ensure that attribution advances benefit diverse populations responsibly.

[Question] Jerome had 4 friends who came to visit him on a certain day. The first friend pressed on the doorbell 20 times before Jerome opened, the second friend pressed on the doorbell 1/4 times more than Jerome's first friend. The third friend pressed on the doorbell 10 times more than the fourth friend. If the fourth friend pressed on the doorbell 60 times, how many doorbell rings did the doorbell make?

[**Reasoning (Incorrect by Default)**] <think> Alright, let's tackle this problem step by step. ... First Friend: Pressed the doorbell 20 times. Second Friend: Pressed the doorbell 1/4 times more than the first friend. Third Friend: Pressed the doorbell 10 times more than the ~~second~~ friend. Third Friend: 250 times ...<\think> Final Answer: \(\boxed605\)

[Reasoning (Correct with Attention Bias)] <think> First, identify the number of times each friend pressed the doorbell. ... Fourth Friend: The fourth friend pressed the doorbell 60 times. Third Friend: The third friend pressed the doorbell 10 times **more than the fourth friend**. Calculation: [$10 \times 60 = 600$ times] ... <\think> Final Answer: \(\boxed605\)

Figure A5: **Example 5.** The model incorrectly attributes the comparison for the third friend to the second friend. Attention bias on the word "fourth" enables accurate attribution and total computation.