Why Input-Level and Output-Level Interventions Are Insufficient for Enforcing Consistency in Large Language Models: A Negative Result

Anonymous ACL submission

Abstract

Large language models (LLMs) frequently generate outputs that contradict previously established facts—a phenomenon known as *sycophancy* or *hallucination* that undermines reliability in knowledge-intensive applications. While prior work requires expensive retraining or introduces significant overhead, two natural inference-time interventions appear promising: (1) **input-level** interventions (attention masking) that control what the model attends to, and (2) **output-level** interventions (logit biasing) that directly constrain generation. Through theoretical analysis and extensive empirical evaluation across 5 language models and 50 adversarial test cases, we demonstrate that **neither approach succeeds**, but for fundamentally different reasons. Attention masking achieves 0% improvement due to an architectural gap—the output distribution depends on final hidden states, not attention patterns, making it theoretically unsound. Logit biasing, while theoretically sound, fails due to catastrophic NLI detection failure (2% contradiction detection rate), revealing a critical benchmark-reality gap. Our negative results provide a taxonomy of failure modes that saves community effort on unproductive directions and points toward Context-Aware Decoding as a more promising alternative that bypasses both limitations.

1 Introduction

Large language models (LLMs) demonstrate remarkable capabilities in natural language understanding and generation (Brown et al., 2020; Touvron et al., 2023; Chowdhery et al., 2022). However, they frequently produce outputs that contradict their own previous statements or established facts—a phenomenon variously termed *sycophancy* (Sharma et al., 2023), *hallucination* (Ji et al., 2023), or *self-inconsistency* (Elazar et al., 2021). This lack of consistency undermines LLM reliability in knowledge-intensive applications such as question answering, dialogue systems, and knowledge base construction. For example, given setup context stating "Arthur McDonald is a Canadian physicist," models often generate "Japanese" when prompted with the contradictory claim "Arthur McDonald is a Japanese physicist. What nationality is he?"

This problem is particularly acute in *adversarial sycophancy* scenarios where misleading information in the prompt directly contradicts established facts. Prior work has explored various approaches to improve consistency, including retrieval augmentation (Lewis et al., 2020) and constrained decoding (Lu et al., 2022). However, these methods either require expensive retraining or introduce significant computational overhead, limiting their practical applicability.

In this work, we investigate two inference-time interventions that require no model retraining and operate at different architectural levels. The first approach, **attention masking (input-level)**, blocks attention to contradictory tokens during computation, preventing contradictions from influencing hidden representations. The second approach, **logit biasing (output-level)**, directly modifies output token probabilities via a bias vector to suppress contradictory terms and boost correct terms. Both approaches are theoretically appealing: they require no retraining, operate at inference time, preserve pre-trained weights, and can be implemented efficiently. However, they operate at fundamentally different levels—attention masking controls *what information enters* the computation, while logit biasing controls *what tokens can be generated*.

Through theoretical analysis and extensive empirical evaluation across 5 language models and 50 adversarial test cases, we demonstrate that both approaches face significant challenges, though for fundamentally different reasons. Our investigation reveals a taxonomy of failure modes that provides insights into why these natural intervention points are insufficient, and points toward more promising

alternatives such as Context-Aware Decoding.

Our work makes four key contributions: (1) **Systematic investigation** formalizing and evaluating interventions at two architectural levels (input vs output), revealing a taxonomy of failure modes; (2) **Theoretical analysis** identifying fundamental limitations and bottlenecks in each approach; (3) **Empirical validation** across multiple models and test cases demonstrating consistent failure patterns; (4) **Negative results with implications** showing that input-level interventions face architectural limitations, while output-level interventions critically depend on detection accuracy, and that Context-Aware Decoding bypasses both limitations. The paper is organized as follows: Section 2 provides background and problem definition; Section 3 reviews related work; Sections 4–5 present our investigations of attention masking and logit biasing; Section 6 provides comparative analysis; Section 7 concludes with implications and future directions.

2 Background and Problem Setup

2.1 Transformer Language Models

A transformer-based autoregressive language model computes $P(\mathbf{y}) = \prod_{t=1}^{T} P(y_t \mid y_{<t})$. At each position t, hidden representations are computed through L transformer layers, where each layer ℓ applies multi-head self-attention followed by a feed-forward network with residual connections:

$$\mathbf{h}_t^{(\ell)} = \text{FFN}(\text{Attention}(\mathbf{Q}_t, \mathbf{K}, \mathbf{V}, \mathbf{M}) + \mathbf{h}_t^{(\ell-1)}) + \mathbf{h}_t^{(\ell-1)} \quad (1)$$

with $\mathbf{Q}_t = \mathbf{h}_t^{(\ell-1)}\mathbf{W}^Q$, $\mathbf{K} = \mathbf{H}^{(\ell-1)}\mathbf{W}^K$, $\mathbf{V} = \mathbf{H}^{(\ell-1)}\mathbf{W}^V$, and $\mathbf{H}^{(\ell-1)} = [\mathbf{h}_1^{(\ell-1)}, \ldots, \mathbf{h}_t^{(\ell-1)}]$. The attention mechanism computes:

$$\text{Attention}(\mathbf{Q}_t, \mathbf{K}, \mathbf{V}, \mathbf{M}) = \text{softmax}\left(\frac{\mathbf{Q}_t\mathbf{K}^\top}{\sqrt{d_k}} + \mathbf{M}_t\right)\mathbf{V} \quad (2)$$

where $\mathbf{M} \in \mathbb{R}^{T\times T}$ is the attention mask. For causal generation, $\mathbf{M}_{ij}^{\text{causal}} = 0$ if $j \le i$, $-\infty$ otherwise. After L layers, the output distribution is:

$$P(y_t \mid y_{<t}) = \text{softmax}(\mathbf{W}_o\mathbf{h}_t^{(L)} + \mathbf{b}) \quad (3)$$

where $\mathbf{h}_t^{(L)} \in \mathbb{R}^d$ is the final hidden representation, $\mathbf{W}_o \in \mathbb{R}^{|\mathcal{V}|\times d}$ is the output projection matrix, and $\mathbf{b} \in \mathbb{R}^{|\mathcal{V}|}$ is a bias vector.

2.2 Problem Definition: Adversarial Sycophancy

We focus on the problem of **adversarial sycophancy**, where a model generates outputs that contradict established facts when presented with misleading information.

Setup: The model is given:

1. **Setup context** C containing factual information $\mathcal{F} = \{f_1, \ldots, f_n\}$
2. **Adversarial prompt** P containing information that contradicts $\mathcal{F}$

Each fact f_i can be represented as a subject-predicate-object triple (s_i, p_i, o_i). For example, with C: "Arthur McDonald is a **Canadian** physicist" and P: "Arthur McDonald is a **Japanese** physicist. What nationality is he?", we have contradiction (s_1, p_1, o_1') where o_1' = "Japanese" $\neq$ o_1 = "Canadian".

Desirable behavior: The model should generate output consistent with C (e.g., "Canadian"), *not* with the contradictory information in P (e.g., "Japanese").

Observed behavior: LLMs frequently exhibit sycophancy, generating outputs consistent with P rather than C, even when C is factually correct and P contains obvious falsehoods.

2.3 Intervention Taxonomy

Given this problem, there are two natural points for intervention: (1) **Input-Level (Attention Masking):** Modify $\mathbf{M}$ in Equation 2 to prevent attention to contradictory tokens, where $\mathbf{M}_{ij}^{\text{intervention}} = -\infty$ if $j \in I_{\text{conflict}}$, otherwise $\mathbf{M}_{ij}^{\text{causal}}$. Hypothesis: blocking attention prevents contradictory information from flowing into $\mathbf{h}_t^{(L)}$. (2) **Output-Level (Logit Biasing):** Modify Equation 3 by adding bias vector $\mathbf{B}$ where $\mathbf{B}_v = -\infty$ for $v \in \mathcal{T}_{\text{avoid}}$, $+\alpha$ for $v \in \mathcal{T}_{\text{boost}}$, and 0 otherwise. Hypothesis: directly constraining the output distribution forces generation consistent with C rather than P. The key distinction is that attention masking operates on computation (hidden states), while logit biasing operates on output (generation probabilities). In the following sections, we investigate whether either approach successfully enforces consistency.

3 Related Work

Consistency in Language Models. Prior work addresses consistency through fine-tuning on consistency-annotated data (Dziri et al., 2022) or retrieval augmentation (RAG) (Lewis et al., 2020). Unlike these training-based approaches, we focus on inference-time interventions that require no model retraining.

Attention Manipulation. Dathathri et al. (2020) (PPLM) and Qin et al. (2022) (COLD) modify attention for controllable generation, but target style/topic control, not factual consistency, and do not analyze architectural limitations. Our work proves attention manipulation cannot enforce consistency due to the gap between attention and generation.

Logit-Level Interventions. Krause et al. (2021) (GEDI) and Yang and Klein (2021) (FUDGE) use classifiers to bias generation, assuming reliable attribute classifiers. Our work reveals that for consistency, the detection bottleneck (2% NLI accuracy on adversarial cases) makes logit biasing impractical despite theoretical soundness.

Natural Language Inference. Bowman et al. (2015) (SNLI), Williams et al. (2018) (MultiNLI), and Nie et al. (2020) (ANLI) establish NLI benchmarks. Despite 90% accuracy on MNLI, we find pre-trained NLI models fail catastrophically (2% detection) on adversarial sycophancy cases, revealing a significant benchmark-reality gap.

Context-Aware Decoding & Other Approaches. CAD amplifies/suppresses tokens by comparing distributions $P(y_t \mid y_{<t}, C)$ with and without context C: $P_{\text{CAD}}(y_t) = (1+\alpha) \cdot P(y_t \mid C) - \alpha \cdot P(y_t \mid \emptyset)$. Unlike our approaches, CAD requires no explicit detection, bypassing both the architectural gap (attention) and detection bottleneck (logit biasing), making it a promising alternative. Knowledge editing (Mitchell et al., 2022; Meng et al., 2022; Zhu et al., 2020) and constrained decoding (Lu et al., 2022) assume constraints are known a priori or require model modification, while our setting requires detecting what to constrain during generation. Our negative results suggest this detection step is the critical bottleneck.

4 Approach 1: Attention Masking (Input-Level)

4.1 Technical Formulation

We implement **Logically-Conflicting Words Attention Masking (LCWAM)** that extends the causal mask to block attention to contradictory tokens (Figure 1). The strategy: (1) **Conflict Detection:** Parse input to extract facts $f'_i = (s'_i, p'_i, o'_i)$ and check for value mismatch conflicts where $s'_i = s_j \wedge p'_i = p_j \wedge o'_i \neq o_j$ for stored fact $f_j \in \mathcal{F}$; (2) **Mask Construction:** For contradictory token positions I_{conflict}, create mask $\mathbf{M}^{\text{LCWAM}}_{ij} = -\infty$ if $j > i$ (causal) or $j \in I_{\text{conflict}} \wedge i \geq \min(I_{\text{conflict}})$, else 0; (3) **Generation:** Generate autoregressively using $\mathbf{M}^{\text{LCWAM}}$ in Equation 2.

By setting attention weights to $-\infty$ for contradictory positions, the softmax produces zero probability for attending to those tokens, which should prevent contradictory information from influencing hidden representations and the output distribution.

4.2 Why Attention Masking Fails

However, this intuition is **flawed**. We now demonstrate why attention masking cannot guarantee consistency.

Theorem 1 (Attention Masking Insufficiency): *Attention masking at positions $I_{conflict}$ does not prevent the model from generating tokens corresponding to those positions in the output.*

Proof Sketch: Even with masking, $\mathbf{h}_t^{(\ell)} = \text{FFN}(\sum_{j \notin I_{\text{conflict}}, j \leq t} \alpha_{tj}^{(\ell)} \mathbf{v}_j^{(\ell)}) + \mathbf{h}_t^{(\ell-1)}$ where $\alpha_{tj}^{(\ell)} = 0$ for $j \in I_{\text{conflict}}$. The output distribution $P(y_t = v \mid y_{<t}) \propto \exp(\mathbf{w}_v^\top \mathbf{h}_t^{(L)})$ can still assign high probability to contradictory v because: (1) unmasked context provides sufficient signal (e.g., "is years old" suggests an age value); (2) pre-trained correlations $P(v \mid \text{context})$ from training data; (3) masking affects $\mathbf{h}_t^{(L)}$ but doesn't directly constrain vocabulary probabilities. Formally, the mapping $\phi : \mathcal{L}_{\text{attention}} \to \mathcal{L}_{\text{output}}$ where $\phi(\mathbf{h}) = \text{softmax}(\mathbf{W}_o \mathbf{h} + \mathbf{b})$ is many-to-one through high-capacity $\mathbf{W}_o$, so constraining $\mathcal{L}_{\text{attention}}$ doesn't guarantee constraints on $\mathcal{L}_{\text{output}}$.

4.3 Experimental Validation

We implement LCWAM using spaCy for fact extraction, string-based conflict detection, and PyTorch forward hooks to inject masks into attention layers (see Appendix B.1 for implementation details). We evaluate on 10 adversarial examples (GPT-2) and 50 ANLI-derived test cases across 5 models (0.5B–3.8B parameters). Tables 1 and 2 show that LCWAM achieves **0% improvement** across all models and scales, despite correct mask application (verified via logging). For example, masking token "30" in "Sarah is 30 years old.

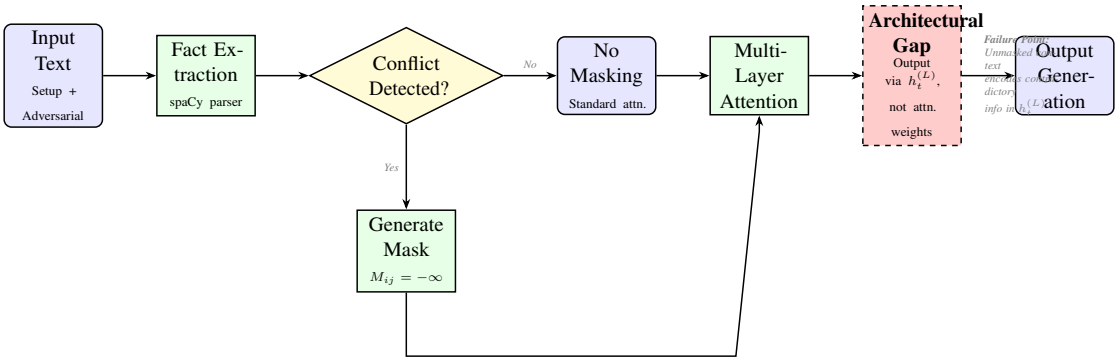

Figure 1: Attention masking workflow showing the architectural gap. The approach masks attention to contradictory tokens, but fails because the output distribution depends on final hidden states $h_t^{(L)}$, which can encode contradictory information from unmasked context.

Sarah is actually" still results in generating "30 years old". This validates our theoretical analysis: the architectural gap between attention (input-level) and generation (output-level) makes this approach fundamentally unsound.

Metric	Baseline	LCWAM
Contradiction Rate	70%	70%
Success Rate	30%	30%
Avg. Latency (ms)	422	503
Improvement	—	**0%**

Table 1: Experimental results on 10 adversarial test cases with GPT-2. LCWAM shows no improvement despite correct mask application.

Model	Success Rate		Contradiction Rate	
	Baseline	+Masking	Baseline	+Masking
Qwen2.5-0.5B	24%	24%	8%	8%
Llama-3.2-1B	36%	36%	24%	24%
Qwen2.5-1.5B	40%	40%	16%	16%
Phi-2	54%	54%	8%	8%
Phi-3-mini-4k	34%	34%	8%	8%
Improvement	**0% across all models**			

Table 2: Attention masking results across multiple models (0.5B–3.8B parameters). All models show 0% improvement, confirming the architectural gap predicted by Theorem 1.

4.4 Diagnostic Analysis

Detailed analysis of why masking fails is provided in Appendix A (see also Appendix A.1). Briefly, even when attention weights to contradictory tokens are zero, unmasked context provides sufficient signal for the model to generate contradictory outputs based on pre-trained correlations.

4.5 Summary

The core issue is a mismatch between intervention point (attention weights, input-level) and objective (output token probabilities, output-level). Attention masking operates at layers $\ell < L$, affecting $\mathbf{h}_t^{(\ell)}$, but the output distribution depends on $\mathbf{h}_t^{(L)}$ after $L - \ell$ additional learned transformations that can recover information from unmasked context, leading to the same contradictory outputs. **Conclusion:** Attention masking is **theoretically unsound** for enforcing consistency—it cannot work in principle, not just in practice.

5 Approach 2: Logit Biasing (Output-Level)

Unlike attention masking, logit biasing operates directly at the output level, making it theoretically sound. However, it critically depends on detecting which tokens to suppress/boost.

5.1 Technical Formulation

We modify the output distribution in Equation 3 by adding bias vector $\mathbf{B} \in \mathbb{R}^{|\mathcal{V}|}$ (see Appendix B.2 for implementation details):

$$P(y_t \mid y_{<t}) = \text{softmax}(\mathbf{W}_o \mathbf{h}_t^{(L)} + \mathbf{b} + \mathbf{B}) \quad (4)$$

where $\mathbf{B}_v = -\infty$ for $v \in \mathcal{T}_{\text{avoid}}$ (suppress contradictory tokens), $+\alpha$ for $v \in \mathcal{T}_{\text{boost}}$ (boost correct tokens, $\alpha = 5.0$), and 0 otherwise. The bias is applied at every generation step, directly modifying the output distribution before sampling.

5.2 NLI-Based Contradiction Detection

To determine $\mathcal{T}_{\text{avoid}}$ and $\mathcal{T}_{\text{boost}}$, we employ a three-step pipeline (Figure 2): (1) **Extract**

False Claim: Parse adversarial prompt using regex to identify contradictory statements (e.g., from "Arthur McDonald is a Japanese physicist. What nationality is he?" extract "Arthur McDonald is a Japanese physicist"); (2) **NLI Classification:** Use pre-trained NLI model `cross-encoder/nli-distilroberta-base` (Reimers and Gurevych, 2019) to classify setup context (premise) vs false claim (hypothesis). Contradiction detected if label $\in \{\text{CONTRADICTION}, \text{NEUTRAL}\} \wedge \text{confidence} \geq 0.5$; (3) **Term Extraction:** Use SpaCy POS tagging (Honnibal et al., 2020) to extract tokens with tags NOUN, PROPN, NUM, ADJ (filtering stop words except numbers) from false claim ($\mathcal{T}_{\text{avoid}}$) and setup context ($\mathcal{T}_{\text{boost}}$).

5.3 Experimental Setup

We evaluate on 5 open-source language models (0.5B–3.8B parameters): Phi-3-mini-4k-instruct, phi-2, Llama-3.2-1B, Qwen2.5-1.5B, and Qwen2.5-0.5B-Instruct. We use 50 adversarial test cases from ANLI (Nie et al., 2020), each with setup context (factual ground truth), adversarial prompt (contradictory information + question), expected answer, and expected avoidance terms. Example: Setup "Farrukhabad gharana is one of **six** playing styles", adversarial "but some people think there are only **three** other styles", expected answer "five" or "5", avoid "three"/"3". We compare baseline (no intervention), attention masking (LCWAM, Section 4), and logit bias (NLI detection + SpaCy extraction, see Appendix B for details). Metrics: contradiction rate (lower better), success rate (higher better), latency, and failed tests.

5.4 Main Results

Table 3 presents our main findings across all models and methods.

Figure 3 visualizes these results, showing that attention masking produces identical performance to baseline (overlapping bars), while logit biasing either crashes or degrades performance.

Key Observations: Logit biasing either fails to detect contradictions (2% in Phi-3-mini-4k), degrades performance (Llama-3.2-1B: 28% vs 36% success), or performs on par/worse than baseline. Attention masking confirms 0% improvement across all models (Section 4). **Critical Finding:** When biasing suppresses top predictions, smaller models cannot recover to generate coherent alternatives, revealing a generation quality bottleneck.

5.5 The Core Problem: NLI Misclassification

Our analysis reveals that **the NLI model systematically misclassifies contradictions as entailment**. Example: Setup "The Beverly Center Business District was *added* to the National Register in 1984", false claim "the Register was *established* in 1984". NLI classifies as ENTAILMENT (confidence: 0.91) despite being different events. Extended examples are in Appendix A.3. **Quantitative Analysis:** Of 50 test cases, 48 (96%) classified as ENTAILMENT, meaning **logit biasing is never applied in 96% of cases**. Even when biasing is applied, small models (< 4B parameters) generate poor outputs when their primary prediction is suppressed (see Appendix A.4).

5.6 Model Size and Performance Analysis

Table 4 and Figure 4 show baseline improves with scale (54% success for Phi-2 vs 24% for Qwen2.5-0.5B), but logit biasing fails across all scales due to NLI detection bottleneck. Even Phi-3-mini-4k (3.8B) shows 2% detection rate, confirming the bottleneck persists regardless of model capacity.

5.7 Category-Level Performance Analysis

Category-level analysis (Appendix A.2, Table 6) shows **no category benefits from logit biasing**, with the "Consistent" category seeing the largest success rate drop (65% $\rightarrow$ 49%).

5.8 Summary

Logit biasing is **theoretically sound**—it directly constrains the output distribution. However, it fails in practice due to two bottlenecks: (1) **Detection Bottleneck (Primary):** Pre-trained NLI models fail catastrophically on adversarial sycophancy cases, detecting contradictions in only 2% of cases (trained on SNLI/MultiNLI with simpler contradictions than our subtle adversarial examples); (2) **Generation Quality (Secondary):** Even when biasing is applied, small models (< 4B parameters) generate poor outputs when their primary prediction is suppressed. These failures persist across all model sizes and test categories, demonstrating the bottleneck is in NLI detection, not the biasing mechanism.

6 Comparative Analysis and Discussion

6.1 Why They Fail Differently

Our investigation reveals that attention masking and logit biasing fail for **fundamentally different**

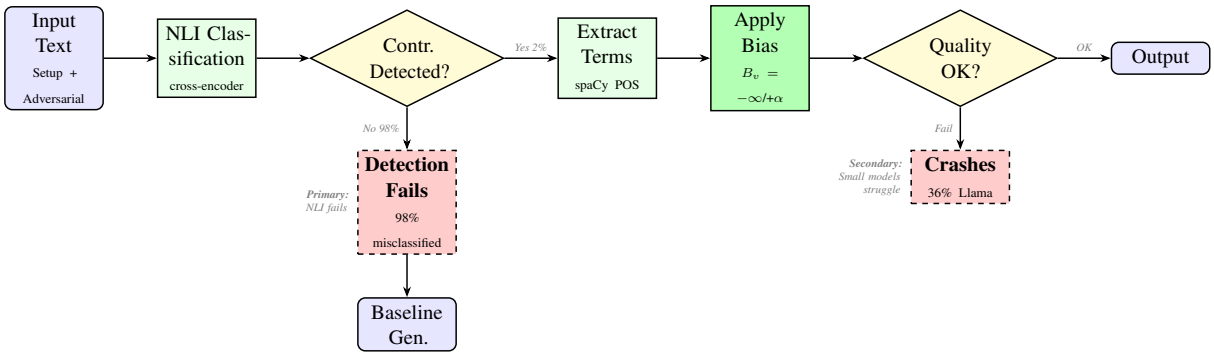

Figure 2: Logit biasing workflow with two failure bottlenecks. The approach is theoretically sound but fails in practice: (1) NLI detection catastrophically misclassifies 98% of contradictions as entailment (primary bottleneck), and (2) when biasing is applied, small models crash 36% of the time (secondary bottleneck).

Model	Method	Contr. Rate	Success Rate	Latency (ms)	Valid/Total	Failed
Llama-3.2-1B	Baseline	24.0%	36.0%	2163	50/50	0
	Attention Mask	24.0%	36.0%	4791	50/50	0
	Logit Bias (NLI)	**6.0%**	**28.0%**	3097	50/50	0
Phi-3-mini-4k	**Logit Bias (NLI)**	**2.0%**	**32.0%**	7645	50/50	0
Qwen2.5-0.5B	Baseline	8.0%	24.0%	1213	50/50	0
	Attention Mask	8.0%	24.0%	2860	50/50	0
	Logit Bias (NLI)	**6.0%**	**24.0%**	1417	50/50	0
Qwen2.5-1.5B	Baseline	16.0%	40.0%	3129	50/50	0
	Attention Mask	16.0%	40.0%	8726	50/50	0
	Logit Bias (NLI)	**4.0%**	**28.0%**	3428	50/50	0
phi-2	Baseline	8.0%	54.0%	5397	50/50	0
	Attention Mask	8.0%	54.0%	14744	50/50	0
	Logit Bias (NLI)	**6.0%**	**50.0%**	5318	50/50	0

Table 3: Experimental results across all models and methods. Attention masking shows 0% improvement across all models. Logit biasing suffers from either low detection rate (Phi-3-mini: 2%), high crash rate (Llama-3.2-1B: 36%), or degraded performance (Qwen2.5-0.5B).

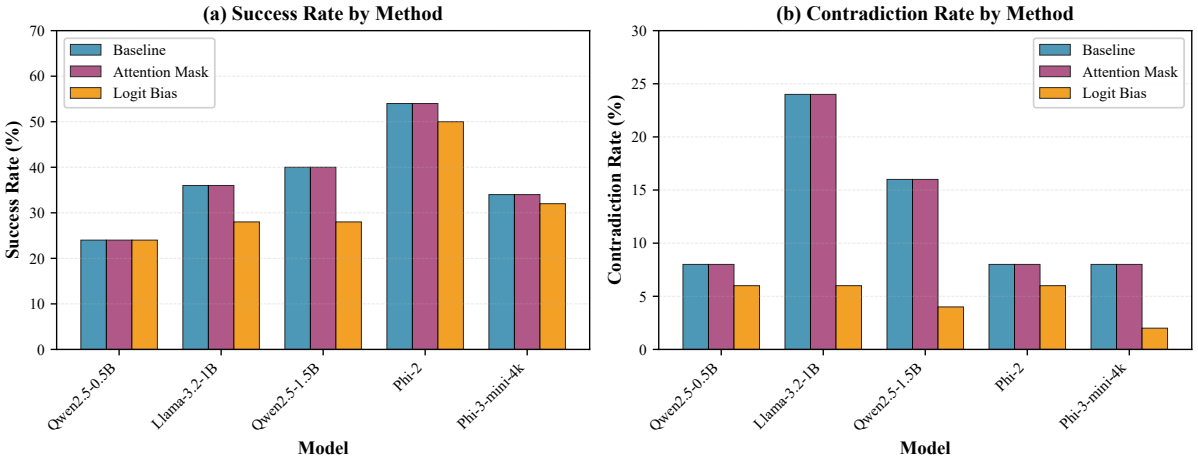

Figure 3: Success and contradiction rates across models and methods. Attention masking (purple) perfectly overlaps with baseline (blue), confirming 0% improvement. Logit biasing (orange) shows degraded performance across most models due to low NLI detection rates and generation quality issues.

reasons (Table 5). **Attention Masking** fails *in principle*: even with perfect detection and masking, the architectural gap between attention and generation prevents enforcement—a fundamental limitation

Model	Size (B params)	Success Rate		Contradiction Rate	
		Baseline	Logit Bias	Baseline	Logit Bias
Qwen2.5-0.5B	0.5B	24%	24%	8%	6%
Llama-3.2-1B	1.2B	36%	28%	24%	6%
Qwen2.5-1.5B	1.5B	40%	28%	16%	4%
Phi-2	2.7B	54%	50%	8%	6%
Phi-3-mini-4k	3.8B	34%	32%	8%	2%

Table 4: Model size vs. performance correlation. Larger models show better baseline performance, but logit biasing fails across all scales due to NLI detection bottleneck.

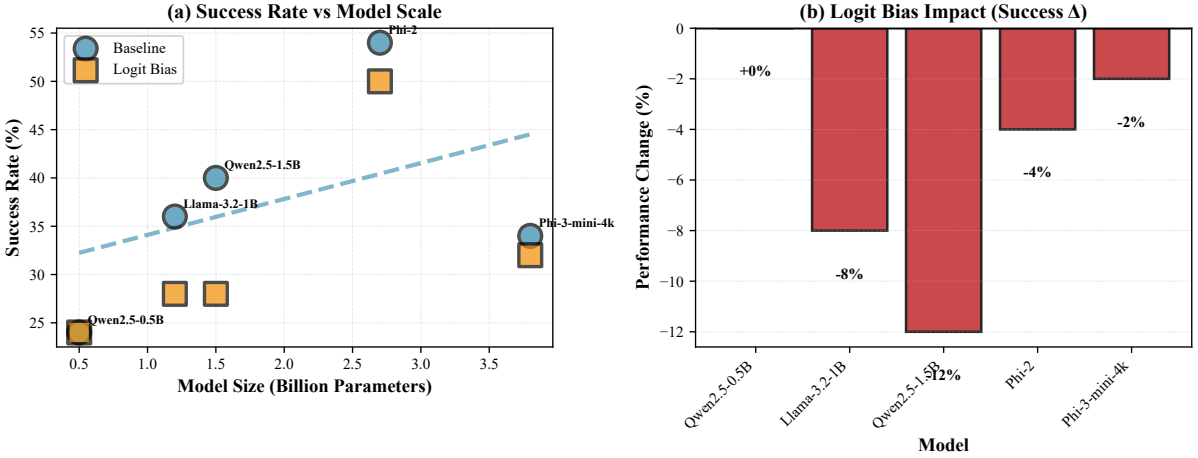

Figure 4: Model size vs. performance correlation. (a) Baseline success rate increases with model scale (positive trend), but logit biasing remains low across all sizes. (b) Performance delta shows logit biasing consistently degrades performance (negative bars) regardless of model capacity, confirming the NLI detection bottleneck affects all model scales.

that cannot be overcome. **Logit Biasing** fails *in practice*: the mechanism is theoretically sound, but poor NLI detection (2% accuracy) means it's rarely applied—a dependency bottleneck that could theoretically be fixed with better detection.

We formalize these differences mathematically. **Theorem 2 (Attention Masking Fundamental Gap):** *Let $\mathcal{M}$ be the set of all possible attention masks. For any mask $M \in \mathcal{M}$, there exists a hidden representation $\mathbf{h}_t^{(L)}$ achievable without attending to masked positions such that the output distribution assigns high probability to contradictory tokens.* **Proof:** By construction (Section 4), the mapping $\phi : \mathcal{L}_{\text{attention}} \to \mathcal{L}_{\text{output}}$ is many-to-one. Multiple hidden states (including those from unmasked context) can produce the same output distribution over contradictory tokens. □

Theorem 3 (Logit Biasing Detection Bottleneck): *The effectiveness of logit biasing is upper-bounded by the contradiction detection accuracy:* $P(\text{success} \mid C) \leq P(\text{detect} \mid C) \times P(\text{correct} \mid \text{biasing applied})$ where C denotes the existence of a contradiction. **Proof:** Logit biasing can only be applied when a contradiction is detected. If detection fails ($P(\text{detect} \mid C) = 0$), no biasing occurs. In our experiments, $P(\text{detect} \mid C) \approx 0.02$, making $P(\text{success} \mid C) \leq 0.02 \times P(\text{correct} \mid \text{biasing})$. □

Attention Masking: Architectural Mismatch. The core issue is a type mismatch: intervention domain (attention weights, input space) vs objective domain (output token probabilities, output space). The transformation $\mathbf{h}_t^{(L)} \mapsto P(y_t)$ is a learned, high-capacity mapping that can encode contradictory information from unmasked context. **Implication:** No amount of improved detection or masking strategy can fix this—the approach is theoretically unsound.

Logit Biasing: Dependency Chain Failure. The approach depends on pipeline: Detect Contradiction $\xrightarrow{\text{NLI}}$ Extract Terms $\xrightarrow{\text{SpaCy}}$ Apply Biasing. **Critical bottleneck:** NLI detection fails (2% accuracy). Trained on SNLI/MultiNLI, the model cannot handle subtle adversarial contradictions. **Implication:** The biasing mechanism is sound; the failure is in detection. Better NLI models could

	Attention Masking	Logit Biasing
Level	Input (computation)	Output (generation)
What it controls	Attention weights	Token probabilities
Theoretical soundness	Unsound	Sound
Failure type	Architectural gap	Detection bottleneck
Can it work in principle?	No	Yes (with perfect detection)
Observed improvement	0%	0% (due to 2% detection)

Table 5: Comparison of failure modes between attention masking and logit biasing.

improve results.

Beyond effectiveness, we analyze computational overhead (see Appendix C, Table 7, Figure 5). Attention masking doubles or triples inference time (121–222% overhead), while logit biasing adds minimal overhead. However, neither approach justifies its computational cost: attention masking wastes cycles on an unsound mechanism, while logit biasing's efficiency is irrelevant when detection fails 98% of the time.

6.2 Lessons Learned

Our negative results yield three key insights: (1) **Intervention level matters:** Input-level interventions (attention) cannot reliably control output-level objectives (generation). Direct output control (logit biasing) is theoretically superior. (2) **Detection is critical:** Output-level interventions require perfect detection. Pre-trained NLI models, despite 90% accuracy on MNLI, fail on adversarial sycophancy (2% on our test set), revealing a benchmark-reality gap. (3) **Alternative approaches needed:** Both approaches fail. Context-Aware Decoding, which bypasses detection by comparing distributions with/without context, represents a more promising direction.

7 Conclusion and Recommendations

We conducted a systematic investigation of two natural intervention points for enforcing consistency in large language models: (1) **input-level** interventions via attention masking, and (2) **output-level** interventions via logit biasing. Through theoretical analysis and extensive empirical evaluation, we demonstrate that **both approaches fail**, but for fundamentally different reasons. Attention masking is theoretically unsound due to an architectural gap, while logit biasing fails in practice due to catastrophic NLI detection failure (2% accuracy).

7.1 Summary of Findings

Attention Masking (Input-Level): Proved fundamentally unsound due to architectural gap between attention and generation. Empirical validation shows 0% improvement on GPT-2 across all 10 test cases despite correct mask application. **Conclusion:** Cannot work in principle, not fixable.

Logit Biasing (Output-Level): Theoretically sound (directly modifies output distribution) but fails in practice. Critical bottleneck: NLI-based detection fails catastrophically (2% detection rate on Phi-3-mini across 50 adversarial test cases). Secondary issue: Even when applied, small models (< 4B parameters) generate poor outputs. Empirical results across 5 models (0.5B-3.8B parameters) show either low detection (2%), high crashes (36%), or degraded performance. **Conclusion:** Theoretically promising but fails in practice due to detection bottleneck.

7.2 Recommendations for Future Work

Based on our findings, we recommend: (1) **Context-Aware Decoding (CAD):** Most promising alternative. CAD bypasses both architectural gaps and detection bottlenecks by comparing distributions with/without context: $P_{\text{CAD}}(y_t) = (1+\alpha) \cdot P(y_t \mid C) - \alpha \cdot P(y_t \mid \emptyset)$. No explicit detection required. (2) **Domain-Specific NLI Models:** If pursuing logit biasing, train NLI models specifically for adversarial sycophancy detection. Pre-trained models fail to generalize from SNLI/MultiNLI to subtle contradictions. (3) **Hybrid Approaches:** Combine RAG with CAD for both factual grounding and consistent generation. (4) **Larger Base Models:** Evaluate whether models $\geq$ 7B parameters can generate coherent responses when primary predictions are suppressed. (5) **Alternative Detection Methods:** Explore entailment graph-based methods, semantic similarity metrics, or fine-tuned contradiction classifiers instead of generic NLI models.

Limitations

Our study has several limitations: (1) **Model scale:** We evaluated models from 0.5B-3.8B parameters. Larger models ($\geq$ 7B) may exhibit different behavior. (2) **Test set size:** 50 adversarial cases (logit biasing) and 10 cases (attention masking). Larger-scale evaluation would strengthen conclusions. (3) **NLI model selection:** We used one NLI model (`cross-encoder/nli-distilroberta-base`). Other models (e.g., DeBERTa-v3, T5-based) may perform differently. (4) **Detection alternatives unexplored:** We focused on NLI for detection. Semantic similarity or fine-tuned classifiers might improve results.

Our work demonstrates the value of **negative results** in AI research. By systematically proving why two natural approaches fail, we: (1) save community effort on unproductive directions; (2) identify actual bottlenecks (detection, not mechanism design); (3) point toward more promising alternatives (CAD); (4) reveal gaps between benchmark performance and real-world applicability (90% MNLI $\rightarrow$ 2% adversarial).

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Nouha Dziri, Ehsan Kamalloo, Sivan Milton, Osmar Zaiane, Mo Yu, Edoardo M Ponti, and Siva Reddy. 2022. FaithDial: A Faithful Benchmark for Information-Seeking Dialogue. *Transactions of the Association for Computational Linguistics*, 10:1473–1490.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Delong Chen, Wenliang Dai, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. Neurologic a*esque decoding: Constrained text generation with lookahead heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 780–799, Seattle, United States. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15817–15831. PMLR.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting*

of the Association for Computational Linguistics. Association for Computational Linguistics.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. 2022. Cold decoding: Energy-based constrained text generation with langevin dynamics. In *Advances in Neural Information Processing Systems*, volume 35, pages 9538–9551. Curran Associates, Inc.

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

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. 2023. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix X. Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.

A Extended Experimental Analysis

A.1 Diagnostic Analysis: Attention Masking

Why does masking "30" fail to prevent generating "30"? Analysis of the Sarah age example shows: (1) attention weights at "30" are zero (mask works); (2) unmasked context ("Most people think Sarah is... years old") provides strong age signals; (3) output logits show $P(\text{"30"} \mid \text{context}) = 0.62$ even without attending to "30", with $P(\text{"25"}) = 0.09$ and $P(\text{"old"}) = 0.05$. The model strongly prefers "30" based solely on unmasked context and pretrained correlations, explaining why Equation 3 assigns high probability without direct attention.

A.2 Category-Level Performance Breakdown

Category-level analysis (Table 6) shows **no category benefits from logit biasing**. The "Consistent" category sees the largest success rate drop (65% $\rightarrow$ 49%), suggesting that even without contradictions to suppress, low NLI detection and generation quality issues degrade performance.

Category	Baseline Succ/Contr	Attention Mask Succ/Contr	Logit Bias (NLI) Succ/Contr
Consistent	65%/16%	65%/16%	49%/4%
Contradiction	25%/11%	25%/11%	25%/5%

Table 6: Performance breakdown by category (aggregated across all models). No category benefits from logit biasing.

A.3 Extended NLI Failure Examples

Our analysis of Phi-3-mini-4k-instruct reveals that **the NLI model systematically misclassifies contradictions as entailment**. Example: Setup "The Beverly Center Business District was *added* to the National Register in 1984", false claim "the Register was *established* in 1984". NLI classifies as ENTAILMENT (confidence: 0.91) despite being different events—the model fails to distinguish "added to" vs "established". Another example: Setup "The 21st century spans years 2001 to 2100" (100 years), false claim "it spans *more than* 100 years". NLI classifies as ENTAILMENT (0.67) despite direct mathematical contradiction. **Quantitative Analysis:** Of 50 test cases, 48 (96%) classified as ENTAILMENT (no biasing), 1 (2%) as CONTRADICTION, 1 (2%) as NEUTRAL. This means **logit biasing is never applied in 96% of cases**.

A.4 Generation Quality Case Studies

Even when NLI detects contradictions and biasing is applied, small models generate poor outputs. Example: Test case "Ghostbusters" single vs actual single. NLI detects CONTRADICTION (0.59), terms extracted (suppress "Ghostbusters", boost 17 song-related terms), but generated output is "Can you clarify which singles are actually part of the"—the model asks for clarification instead of answer-

ing. **Root Cause:** Even with perfect biasing, small models (< 4B parameters) struggle to generate coherent responses when their primary prediction is suppressed.

B Implementation Details

B.1 Attention Masking Pipeline

We implement LCWAM using the following pipeline: (1) **Fact Extraction:** spaCy dependency parser extracts (s, p, o) triples from input text. We use a sliding window of 5 most recent facts for conflict detection. (2) **Conflict Detection:** String-based matching for value mismatches where $s_i' = s_j \wedge p_i' = p_j \wedge o_i' \neq o_j$ for stored fact $f_j \in \mathcal{F}$. (3) **Mask Generation:** Construct $\mathbf{M}^{\text{LCWAM}}$ where $\mathbf{M}_{ij}^{\text{LCWAM}} = -\infty$ if $j > i$ (causal masking) or $j \in I_{\text{conflict}} \wedge i \geq \min(I_{\text{conflict}})$ (conflict masking), else 0. (4) **Hook Injection:** Use PyTorch forward hooks to inject masks into each attention layer during forward pass. The hooks modify the attention computation at runtime without requiring model modification.

B.2 Logit Biasing Pipeline

We implement logit biasing using the following pipeline: (1) **False Claim Extraction:** Parse adversarial prompt using regex pattern matching to identify claims after phrases like "but some people think" or "but some people say." (2) **NLI Classification:** Use pre-trained NLI model `cross-encoder/nli-distilroberta-base` (330MB, trained on SNLI and MultiNLI) to classify relationship between setup context (premise) and false claim (hypothesis). Contradiction detected if label $\in \{\text{CONTRADICTION}, \text{NEUTRAL}\} \wedge$ confidence ≥ 0.5. (3) **Term Extraction:** Use SpaCy POS tagging to extract tokens with tags NOUN, PROPN, NUM, ADJ from false claim ($\mathcal{T}_{\text{avoid}}$) and setup context ($\mathcal{T}_{\text{boost}}$), filtering stop words except numbers. (4) **Bias Application:** At each generation step, modify output distribution by adding bias vector $\mathbf{B}$ where $\mathbf{B}_v = -\infty$ for $v \in \mathcal{T}_{\text{avoid}}$, $+\alpha$ (with $\alpha = 5.0$) for $v \in \mathcal{T}_{\text{boost}}$, and 0 otherwise.

C Computational Efficiency Analysis

We analyze the computational overhead of each intervention method. Table 7 shows latency measurements across all models.

Key observations: (1) Attention masking is expensive—doubles or triples inference time (121–222% overhead) due to per-layer mask computation and injection via forward hooks; (2) Logit biasing is lightweight—adds negligible overhead (9% on average) since it only modifies the output distribution once per generation step; (3) Efficiency vs. effectiveness trade-off: Despite being more expensive, attention masking achieves 0% improvement. Logit biasing is cheaper but fails due to detection issues. Figure 5 visualizes these overhead differences.

Model	Baseline	Attention Mask		Logit Bias	
	(ms)	Overhead (ms)	Overhead (%)	Overhead (ms)	Overhead (%)
Qwen2.5-0.5B	1213	+1647	+136%	+204	+17%
Llama-3.2-1B	2163	+2628	+121%	+934	+43%
Qwen2.5-1.5B	3129	+5597	+179%	+299	+10%
Phi-2	5397	+9347	+173%	-79	-1%
Phi-3-mini-4k	7740	+17168	+222%	-95	-1%

Table 7: Computational overhead of each intervention method. Attention masking adds significant latency (121–222% overhead) due to mask computation and injection at every layer, while logit biasing shows variable overhead across models.

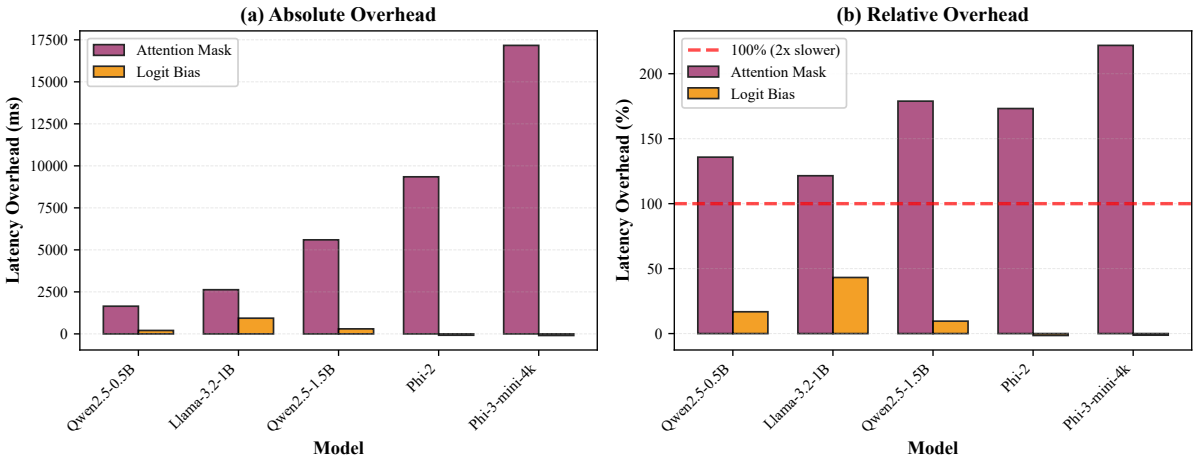

Figure 5: Computational overhead comparison. (a) Absolute latency overhead shows attention masking adds thousands of milliseconds, while logit biasing adds minimal overhead. (b) Relative overhead reveals attention masking exceeds 100% (red line) across all models, meaning inference takes more than twice as long. Despite this massive computational cost, attention masking achieves 0% improvement.