COMPASS: Context-Modulated PID Attention Steering System for Hallucination Mitigation

Kenji Sahay, Snigdha Pandya, Rohan Nagale, Anna Lin, Shikhar Shiromani, Kevin Zhu, Sunishchal Dev

Abstract

Large language models (LLMs) increasingly operate as autonomous agents—reasoning, planning, and interacting with humans and external tools. Yet these agentic systems often exhibit ungrounded or inconsistent behavior, undermining trust and alignment. We introduce COMPASS (Context-Modulated PID Attention Steering System), a lightweight and interpretable control framework for enhancing reliability in agentic LLMs. COMPASS embeds a model-based feedback loop directly within decoding, using a transparent metric, the Context Reliance Score (CRS), to quantify how attention heads ground decisions in contextual evidence. A PID controller dynamically adjusts internal attention distributions in real time, steering the model toward factually and semantically consistent reasoning without retraining or multi-pass inference. Across reasoning and retrieval-augmented benchmarks (HotpotQA, XSum, HaluEval, RAGTruth), COMPASS reduces hallucination rates (2.8–5.8exposing interpretable control signals that reveal which attention heads drive trustworthy behavior. By coupling interpretability with control-theoretic feedback, COMPASS provides a pathway for verifiable, steerable, and value-aligned agentic LLMs, advancing the goals of trustworthy autonomous AI.

Introduction

LLMs exhibit strong reasoning capabilities but often produce *contextual hallucinations*, where outputs conflict with the input context despite relevant evidence being present [8, 16]. These errors typically arise when the model over-relies on its parametric knowledge or generated history rather than the provided prompt.

Beyond mitigation, COMPASS is designed as a scientific probe of how LLMs use context. Each component,the Context Reliance Score, the classifier, and the PID controller, offers a transparent mapping between internal attention signals and model behavior. Rather than treating interpretability as a post-hoc visualization problem, we embed interpretability in the generation loop itself, allowing real-time observation and modulation of evidence use. This framework provides a principled way to study and steer complex model dynamics. Existing mitigation strategies include *contrastive or context-aware decoding* [10], which reweight token probabilities using an auxiliary distribution, and *attention-based diagnostics* such as Lookback Lens [2], which train classifiers on attention ratios to detect hallucinations and then guide decoding through candidate re-ranking. More recent approaches, such as DAGCD [3], intervene directly in the attention mechanism, but often rely on multi-pass decoding or pre-specified head selections, introducing latency and limiting flexibility. Our contributions are four fold:

- **Context-Modulated PID Attention Steering System (COMPASS):** A decoding-time intervention that adjusts attention heads on-the-fly via a *pre-softmax, context-key–only* bias using a real-time diagnostic signal, with no *base-model* retraining or multi-pass decoding.
- **Context Reliance Score (CRS):** The *logit* of the attention mass on *context* keys (last query row), a reformulation of the "lookback ratio" from Lookback Lens [2], used as an online per-head context-sensitivity signal for dynamic head selection.
- **Classifier-Guided Conditional Scaling:** Heads are modulated only when a hallucination detector indicates elevated risk, preserving fluency and minimizing unnecessary interventions.
- **Efficient, Interpretable Control:** COMPASS operates within a single decode stream; attentions are read every k tokens and adjusted via a *pre-softmax, context-key–only* bias, yielding fine-grained, interpretable head-level control.

Methods

Problem Setting and Notation

We study contextual hallucinations: unsupported or factually incorrect tokens given a supplied context. Consider an auto-regressive Transformer [13] with L layers and H heads per layer. At generation step t, the prompt is partitioned as a fixed *context* C (tokens $1{:}|C|$) followed by a fixed *question* Q (tokens $|C|{+}1{:}|C|{+}|Q|$); the model has produced $t{-}1$ output tokens thereafter. Let $\mathcal{K}_t = \{1, \ldots, |C|{+}|Q|{+}t{-}1\}$ denote key positions, with context keys $\mathcal{K}_C = \{1, \ldots, |C|\}$ and non-context

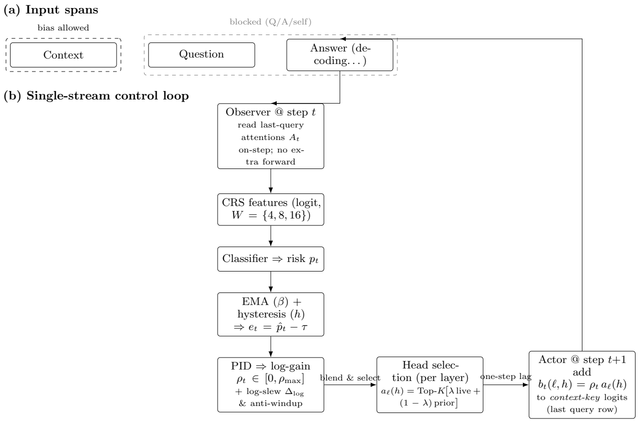

(a) Input spans
bias allowed
blocked (Q/A/self)
Context
Question
Answer (decoding…)
(b) Single-stream control loop
Observer @ step t read last-query attentions A_t on-step; no extra forward
CRS features (logit, $W = \{4, 8, 16\}$)
Classifier ⇒ risk p_t
EMA (β) + hysteresis (h) $\Rightarrow e_t = \hat{p}_t - \tau$
PID ⇒ log-gain $\rho_t \in [0, \rho_{\max}]$ + log-slew $\Delta_{\log}$ & anti-windup
blend & select
Head selection (per layer) $a_\ell(h) = \text{Top-}K[\lambda\,\text{live} + (1-\lambda)\,\text{prior}]$
one-step lag
Actor @ step t+1 add $b_t(\ell,h) = \rho_t\, a_\ell(h)$ to *context-key* logits (last query row)

Figure 1: Single-stream control loop and inputs.

keys $\mathcal{K}_G = \{|C|+1, \ldots, |C|+|Q|+t-1\}$ (which include the question and past outputs). For head h in layer ℓ, let $A_t(\ell,h) \in \mathbb{R}^{|\mathcal{K}_t|}$ be the (causal-masked) attention distribution for the *last query row* at step t, and $Z_t(\ell,h) \in \mathbb{R}^{|\mathcal{K}_t|}$ its pre-softmax logits.

Our goal is to bias attention toward the prompt context only when the model is likely to hallucinate, while leaving behavior unchanged when it is grounded. To this end, we design a decode-time intervention that (i) measures each head's reliance on context vs. non-context via a Context Reliance Score (CRS), (ii) predicts token-level hallucination risk from windowed CRS features, and (iii) uses a PID controller to apply a small *pre-softmax, context-key–only* additive bias to selected heads. Formally, when the gate is active we modify

$$Z'_t(\ell,h,k) \;=\; Z_t(\ell,h,k) \;+\; b_t(\ell,h)\,\mathbf{1}[\,k \in \mathcal{K}_C\,],$$

on the last-query row only, where $b_t(\ell,h) \geq 0$ is set by the PID output and head selection; all non-context keys and all non-last-query rows remain unchanged.

Context Reliance Score (CRS)

We quantify each head's context reliance as the fraction of its attention mass on the *prompt-context* keys. For head (ℓ,h) at step t, with attention distribution $A_t(\ell,h) \in \mathbb{R}^{|\mathcal{K}_t|}$ over all keys in the *last query row* (where $|\mathcal{K}_t| = |C|+|Q|+t-1$), define

$$p_{\text{ctx}}(t,\ell,h) \;=\; \sum_{i\in\mathcal{K}_C} A_t(\ell,h)[i] \;\in\; [0,1], \qquad (1)$$

i.e., the total softmax weight on context tokens. For numerical stability and an unbounded signal, we apply a logit transform with clipping:

$$\tilde{p}_{\text{ctx}} = \text{clip}\big(p_{\text{ctx}},\, \varepsilon,\, 1-\varepsilon\big), \qquad \varepsilon = 10^{-6}, \qquad (2)$$

$$\text{CRS}(t,\ell,h) = \log\frac{\tilde{p}_{\text{ctx}}}{1-\tilde{p}_{\text{ctx}}}. \qquad (3)$$

In preprocessing, we compute CRS for all (ℓ,h) across answer time steps and store tensors of shape $[L, H, T_{\text{ans}}]$ *in logit space* (per-head logit of context mass); summary statistics (mean, std., quantiles) are also recorded.

At runtime, we maintain a per-head history *in logit space* (logit of p_{ctx}) and use per-layer z-scores to rank heads; this live score can be blended with an optional offline prior.

Feature Extraction. We compute the Context Reliance Score (CRS) for each head as the fraction of attention mass that the last query places on *prompt-context* keys (i.e., within $\mathcal{K}_C$). For modeling, we apply a logit transform to CRS and compute windowed statistics per head (mean, standard deviation, last-minus-first *delta*) over $W \in \{4, 8, 16\}$, yielding a feature vector of size $3 \cdot |W| \cdot L \cdot H$ (e.g., 9,216 for LLaMA-2-7B). We do not globally standardize these features; runtime head selection uses per-layer z-scores of the *live* CRS in probability space, while the classifier consumes the raw windowed features (in logit space).

Token-Level Hallucination Risk via Logistic Classifier

We train a token-level classifier that maps windowed CRS features to a hallucination probability. Our primary model is XGBoost with a logistic objective. Inputs are sliding-window statistics of each head's recent CRS *logits*: mean, standard deviation, and last-minus-first *delta* which is then computed over $W \in \{4, 8, 16\}$ (concatenated in increasing W) and concatenated across all heads, yielding a feature vector of size $3 \cdot |W| \cdot L \cdot H$ (no global standardization).

Classifier Training. We use **XGBoost** with a logistic objective to map windowed CRS features to hallucination risk. XGBoost handles nonlinear interactions among heads/layers, runs efficiently for repeated decode-time queries, and provides per-feature importances that we aggregate into per-(layer, head) weights. These weights serve as a *static prior* for the online modulator and are *blended* with live per-layer z-scores during head selection. Data is split 70/10/20 into Train/Validation/Test by example id to prevent leakage across partitions.

Table 1: Classifier AUROC Performance (Hallucination = Positive Class)

Model	Dataset	AUROC
Qwen-2.5-7B-Instruct	HotpotQA	0.839
	XSum	0.953
	RAGTruth	0.789
	HaluEval	0.886
LLaMA-2-7B	RAGTruth	0.858
LLaMA-2-13B	RAGTruth	0.873
Mistral-7B	RAGTruth	0.912

Runtime use. The online modulator constructs the same windowed feature vector from live CRS histories (for all W) and queries the classifier every k tokens to obtain $p_t \in [0,1]$, which feeds the EMA+hysteresis-gated PID loop.

```
Algorithm 1: Head Selection & Context-Key Bias (per risk step)
Params : layer range $[\lfloor L/2 \rfloor,\, L)$; keep per layer $K$; blend $\lambda$
Input  : on-step attentions $A_t$ (output_attentions=true); prior $w(\ell,h)$; gain $\rho_t$
Output : pre-softmax bias on context keys (last query row)
if $\rho_t = 0$ then
    return (no intervention)
for $\ell \in [\lfloor L/2 \rfloor,\, L)$ do
    compute live CRS logits $v_\ell(h)$ from $A_t$
    $z_\ell(h) \leftarrow \max\{0, \mathrm{zscore}_{\text{per-layer}}(v_\ell(h))\}$
    $\tilde{a}_\ell(h) \leftarrow \lambda\, \mathrm{norm}_{[0,1]}(z_\ell(h)) + (1-\lambda)\, w(\ell,h)$
    $S_\ell \leftarrow$ Top-$K$_$h\, \tilde{a}_\ell(h)$;
     $a_\ell(h) \leftarrow \tilde{a}_\ell(h) / \sum_{h \in S_\ell} \tilde{a}_\ell(h)$
    for $h \in S_\ell$ do
        $Z_t(\ell,h)[1{:}|C|] \leftarrow Z_t(\ell,h)[1{:}|C|] + \rho_t\, a_\ell(h)$
         // context keys, last query row
/* $z(\cdot)$ is per-layer $z$-score; $\mathrm{norm}_{[0,1]}$
   rescales over heads in a layer.          */
```

```
Algorithm 2: PID–Gated Log–Gain (Controller)
Params : $\tau$ (target), $h$ (hysteresis), $\beta$ (EMA),
         $(K_p, K_i, K_d)$,
         $\rho_{\max}$ (cap), $\Delta_{\log}$ (log-slew), $\varepsilon$ (small)
Input  : $(\hat{p}_{t-1}, I_{t-1}, \rho_{t-1})$, new risk $p_t$
Output : $(\hat{p}_t, I_t, \rho_t)$
$\hat{p}_t \leftarrow \beta\, \hat{p}_{t-1} + (1-\beta)\, p_t$
$e_t \leftarrow \hat{p}_t - \tau$;   if $|e_t| \le h$ then
    $e_t \leftarrow 0$
$P \leftarrow K_p e_t$
if $(\rho_{t-1} = 0 \wedge e_t < 0) \;\vee\; (\rho_{t-1} = \rho_{\max} \wedge e_t > 0)$ then
    $I \leftarrow I_{t-1}$
else
    $I \leftarrow I_{t-1} + K_i e_t$
$D \leftarrow K_d(\hat{p}_t - \hat{p}_{t-1})$
$\rho_{\text{raw}} \leftarrow \mathrm{clip}(P + I + D,\, 0,\, \rho_{\max})$
$\ell_{\text{prev}} \leftarrow \log(\rho_{t-1} + \varepsilon)$;   $\ell_{\text{raw}} \leftarrow \log(\rho_{\text{raw}} + \varepsilon)$
$\ell \leftarrow \ell_{\text{prev}} + \mathrm{clip}(\ell_{\text{raw}} - \ell_{\text{prev}}, -\Delta_{\log}, \Delta_{\log})$
$\rho_t \leftarrow e^{\ell} - \varepsilon$;   $I_t \leftarrow I$
/* Outputs nonnegative log{gain $\rho_t$ with
   anti-windup and slew limiting.          */
```

Head Selection and Scaling

Algorithms 1–2 summarize the controller and per-step head modulation. In brief, we rank heads within a mid-to-upper layer range by blending per-layer z-scored live CRS with a static prior $w(\ell,h)$, keep the top-K per layer (set via `--keep-per-layer`; K=16 in our runs), renormalize the weights, and add a *pre-softmax, context-only* bias of magnitude $\rho_t\, a_\ell(h)$ to the last-query row. Non-context keys and all non-last-query rows remain unchanged

Pre-Softmax Attention Bias

We add a context-only, last-query-row bias to attention logits. For each selected head (ℓ,h) and context index $i \in C$,

$$\tilde{Z}_t(\ell,h)[i] = Z_t(\ell,h)[i] + b_t(\ell,h), \qquad b_t(\ell,h) \doteq \rho_t\, a_\ell(h),$$

while non-context keys remain unchanged, $\tilde{Z}_t(\ell,h)[j] = Z_t(\ell,h)[j]$ for $j \notin C$. The updated attention is $\tilde{A}_t(\ell,h) = \mathrm{softmax}(\tilde{Z}_t(\ell,h))$.

Adding the bias in logit space multiplies the affected unnormalized weights by $\exp(b_t(\ell,h))$, preserving softmax normalization:

$$\exp\{\tilde{Z}_t(\ell,h)[i]\} = e^{\,b_t(\ell,h)} \exp\{Z_t(\ell,h)[i]\}, \quad i \in C, \tag{4}$$

$$\alpha_t(\ell,h) \doteq \exp\big(\rho_t\, a_\ell(h)\big), \tag{5}$$

$$\tilde{Z}_t = Z_t + \log\alpha_t \cdot . \tag{6}$$

We reset the bias each step and only apply it when the controller is active; non-context keys and all non-last-query rows are never modified.

Full Decoding-Time Algorithm

Inputs. Model f; prompt context C; detector f_{det} (XGBoost) with threshold τ; hysteresis h; PID gains (k_P, k_I, k_D); layer subset $\mathcal{L}^\star$; update cadence k (compute risk/selection every k tokens); window set W for features (default $W = \{4, 8, 16\}$); per-layer head budget K (default $K = 16$); prior–blend weight λ (default $\lambda = 0.3$); log-space slew limit $\Delta_{\log}$ (default 0.20); log-gain cap $\rho_{\max}$ (default 1.0).

Per-step loop for $t = 1, 2, \ldots$

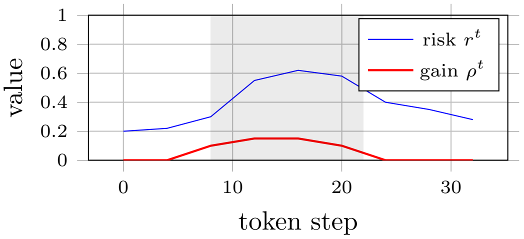
1. **Attention read (every k tokens).** When t mod $k = 0$, enable `output_attentions` and read the *last-token query row* per head *on the same forward pass* (no extra forward). Otherwise, reuse the last risk and head selection.
2. **CRS & features.** From that last-query row, compute $\mathrm{CRS}_t(\ell,h)$ as the fraction of attention mass on context keys, then form sliding-window features in the *logit* domain (mean, std, and end–minus–start trend) for $W \in \{4, 8, 16\}$; concatenate across windows.
3. **Risk prediction.** Feed features to f_{det} to obtain p_t; apply EMA smoothing to get $\hat{p}_t$ and a dead-band $|\hat{p}_t - \tau| \le h$ (hysteresis).
4. **PID update (nonnegative log-gain).** If outside the dead-band, update (P, I, D) on the error $e_t = \hat{p}_t - \tau$ and produce a nonnegative log-gain ρ_t. Apply a *log-space* slew limit with step size $\Delta_{\log}$ and clamp to $[0, \rho_{\max}]$; set $\alpha^t = \exp(\rho_t)$.
5. **Head selection.** For each $\ell \in \mathcal{L}^\star$, z-score the live head vector per layer, clamp negatives to 0, min–max normalize to $[0, 1]$, blend with the prior via $a = \lambda\,\mathrm{live} + (1-\lambda)\,\mathrm{prior}$, and keep the top-$K$ heads.
6. **One-step-lag actuation (pre-softmax).** At step $t{+}1$, add the bias $\log\alpha^t$ to the *context-key* logits of the selected heads on the last-query row only; non-context keys and all non-last-query rows remain unchanged.

Complexity. COMPASS reuses attention tensors produced on the same decode step when `output_attentions` is enabled (every k tokens); no extra forward pass is introduced. The additional work per risk step is: computing CRS from the read attentions, a small set of vector ops for windowed features, a lightweight classifier call, and a few scalar updates for the PID and log-slew. The only tensor write is adding the pre-softmax bias at selected context indices. Empirically the overhead is modest on 7B models, and decoding remains a single stream.

Risk Calculation Details

For clarity, we summarize the end-to-end loop executed at *risk steps* ($t \bmod k = 0$):

1. **Attention read & CRS:** On the same forward pass, enable `output_attentions` and read the *last-query row.* For each $\ell \in \mathcal{L}^\star$ and head h, compute the context mass on keys $1{:}|C|$ and its logit to obtain $\mathrm{CRS}_t(\ell, h)$; form sliding-window CRS-*logit* features (mean, std, end–minus–start) over $W \in \{4, 8, 16\}$ and concatenate across windows (no dataset-wide standardization).
2. **Predict risk:** Feed the windowed features to f_{det} to obtain $p_t \in [0, 1]$ and compute the smoothed score $\hat{p}_t$ via EMA; apply a hysteresis dead-band $|\hat{p}_t - \tau| \le h$.
3. **PID update (nonnegative log-gain):** If outside the dead-band, update (P, I, D) on $e_t = \hat{p}_t - \tau$ with anti-windup, producing a nonnegative log-gain ρ_t; apply a log-space slew limit $\Delta_{\log}$ and clamp to $[0, \rho_{\max}]$.
4. **Head selection:** For each layer, z-score the live CRS vector per layer, clamp negatives to 0, min–max normalize to $[0, 1]$, blend with the static prior $w(\ell, h)$ via $a = \lambda\,\text{live} + (1 - \lambda)\,\text{prior}$, select the top-$K$ heads, and renormalize to weights $a_\ell(h)$.
5. **Actuation (pre-softmax, one-step lag):** At step $t{+}1$, add a context-only bias of magnitude $\rho_t\, a_\ell(h)$ to the selected heads' logits on the last-query row: $Z_{t+1}(\ell, h)[1{:}|C|] \leftarrow Z_{t+1}(\ell, h)[1{:}|C|] + \rho_t\, a_\ell(h)$. Non-context keys and all non–last-query rows remain unchanged.
6. **Generate token:** Finish the forward pass and sample the next token as usual.



This procedure is well-posed: CRS is bounded and interpretable; the detector outputs calibrated probabilities under a logistic objective; head influence is transparent via $a_\ell(h)$; and the PID (with EMA, hysteresis, and log-space slew) responds to sustained risk while suppressing transient noise.

Hyperparameters. Unless noted, the classification threshold τ is tuned on a dev split (maximize F_1) and hysteresis width is $h = 0.01$. PID gains: $k_P = 0.8$, $k_I = 0.2$, $k_D = 0.0$; EMA $\beta = 0.8$; update cadence $k = 1$ (compute risk/selection every token by default); head budget $K = 16$ per layer; windows $W = \{4, 8, 16\}$; prior blend $\lambda = 0.3$; log-space slew limit $\Delta_{\log} = 0.20$; log-gain cap $\rho_{\max} = 1.0$. By default we act on mid-to-upper layers (upper half of the stack); a different subset $\mathcal{L}^\star$ can be provided via configuration.

Ablations. We ablate: (i) no PID (threshold+gate only), (ii) no classifier (heuristic CRS-based risk), (iii) layer range choices (last layer only vs. mid-to-upper vs. all layers), (iv) keep-per-layer $K \in \{4, 8, 16, 32\}$ and prior blend $\lambda \in [0, 1]$, (v) log-gain parameters $\rho_{\max}$ and $\Delta_{\log}$, and (vi) update cadence $k \in \{1, 2, 4\}$.

Latency. COMPASS reuses attentions produced on the same decode step whenever `output_attentions` is enabled (every k tokens); no extra forward pass is introduced. Per risk step, the additional work is: computing CRS from the read attentions, a small set of vector ops to form windowed features, a lightweight classifier call, and a few scalar PID/log-slew updates. The only tensor write is adding a pre-softmax bias at selected context indices.

Experimental Setup

Our experiments proceed in two stages. Phase 1 constructs a high-precision hallucination detector that operates during decoding using only attention-based features. Phase 2 integrates this detector into the generation loop and dynamically rescales attention heads that are automatically selected at runtime as context-reliant using the Context Reliance Score (CRS), applying modulation only when hallucination risk is high. We obtain the needed attention signals with occasional auxiliary reads every k tokens (via `--risk-step`), minimizing overhead. Together, these stages test whether lightweight, real-time control of internal attention can improve factual faithfulness without training or multi-pass decoding.

Phase 1: Detector Construction and Validation

Data & Labeling. We generate answers with LLaMA-2-7B, LLaMA-2-13B, Mistral-7B-Instruct, and Qwen-2.5 on four faithfulness-critical benchmarks: HotpotQA (open-domain QA), XSum (single-sentence summarization), HaluEval (hallucination evaluation in QA and summarization), and RAGTruth (adversarial fact-checking). Each answer is split into span-level substrings. An automatic verifier (Gemini 2.5-Flash, structured JSON schema) labels each example with `is_hallucination` and up to 0–5 `unsupported_spans`, plus a brief analysis and confidence; adjudication uses the same model at temperature 0.0. We manually annotated a random sample of 100 spans and found a 93% agreement between Gemini 2.5-Flash's annotations and human judgments, confirming high consistency.

Phase 2: Modulating Attention Heads

Head Selection. At runtime we compute a blended score per head:

$$s_t(\ell, h) = \lambda\, z\mathrm{CRS}_t(\ell, h) + (1 - \lambda)\, \mathrm{prior}(\ell, h), \quad (7)$$

where the live term uses per-layer z-scores and the prior comes from exported head importance. We then select the top-k heads per layer over a default mid-to-upper range (layers 16–31 in a 32-layer LLaMA-2-7B), configurable via `--layers`.

Control Loop. Every k tokens we read on-step attentions (`output_attentions=true`; no extra forward), compute a token-level risk p_t (EMA-smoothed) with hysteresis gating, and pass it to a PID controller to produce an intensity $\rho_t \in [0, \rho_{\max}]$. We then add a pre-softmax bias for the *context-key* logits of the selected heads at the last query row (equivalent to log-gains), leaving non-context keys untouched.

Models and Baselines. Modulation is evaluated on LLaMA-2 7B and Mistral 7B. Baselines include (i) identical sampling path with mitigation disabled, (ii) Lookback Lens guided re-ranking, (iii) Contrastive Decoding [6], and (iv) random-head scaling (same α but on a random head subset).

Table 2: Results (Hallucination Reduction)

Model	**Dataset**	**MR↓**	**SD↓**	**CO↑**
Qwen-2.5-7B-Inst.	HotpotQA	4.2%	-14.2%	+0.06
	XSum	**2.8%**	-11.4%	+0.04
	RAGTruth	3.1%	**-16.7%**	**+0.08**
	HaluEval	5.8%	-13.8%	+0.05
LLaMA-2-7B	RAGTruth	**4.2%**	-18.3%	+0.09
LLaMA-2-13B	RAGTruth	5.8%	**-22.4%**	**+0.12**
Mistral-7B	RAGTruth	4.9%	-20.1%	+0.11

Results

We evaluated COMPASS, our Context-Modulated PID Attention Steering System, on LLaMA-2-7B, LLaMA-2-13B, Mistral-7B, and Qwen-2.5-7B across four benchmarks probing different aspects of contextual hallucination: RAGTruth, HotpotQA, XSum, and HaluEval. We evaluate hallucination reduction using three complementary metrics to show that reductions in hallucination come from better grounding rather than blunt suppression:

- **Mitigation Rate (MR):** The absolute reduction in hallucination rate compared to the unmodified baseline model.
- **Span Density (SD):** The number of unsupported spans per 100 generated tokens. A span is considered *unsupported* if it lacks a 3–5-gram match in the provided context and fails the sentence-level factual verifier.
- **Context Overlap (CO):** The fraction of generated tokens whose aligned n-grams appear in the retrieved context, serving as a proxy for grounding distinct from surface-level repetition.

These results indicate that COMPASS consistently reduces hallucination rates relative to unmodified baselines, with absolute reductions ranging from approximately 2.8% to 5.8% depending on the model and dataset. For instance, on RAGTruth, LLaMA-2-13B achieved a 5.8% reduction in hallucination rate, while LLaMA-2-7B saw a 4.2% decrease. HotpotQA and XSum also showed improvements in multi-hop reasoning accuracy and summarization faithfulness, respectively. Span density of unsupported content decreased across all datasets. Context overlap (CO)increased or remained stable suggesting that COMPASS preserves model grounding without excessively perturbing attention. Across the board, larger models (e.g., LLaMA-2-13B, Mistral-7B) benefited more from attention modulation, likely reflecting richer redundancy and more exploitable head-level structure. These findings support the feasibility of lightweight, real-time attention modulation for mitigating contextual hallucinations without multi-pass decoding or retraining. To keep comparisons compute-fair while covering diverse failure modes, we evaluate Qwen-2.5-7B across all four datasets and use RAGTruth as a shared grounded-QA setting for the LLaMA-2 and Mistral models.

Discussion

Lookback Lens detects and mitigates contextual hallucinations by monitoring the “lookback ratio” (attention to source context vs. newly generated tokens) and guiding decoding with a lightweight classifier; it transfers across tasks/models and reports measurable reductions (e.g., 9.6% on XSum) without model retraining. By contrast, COMPASS preserves the full prompt and steers head-level attention using a PID controller keyed to an online context-reliance signal. Empirically (table 1), COMPASS achieves lower hallucination rates alongside higher CO and lower SD, indicating that steering internal attention during generation is competitive with (and complementary to) Lookback Lens’s classifier-guided decoding approach. Compared with other approaches, decoding-only tweaks (temperature/nucleus/repetition) do not explicitly target evidence alignment and yield smaller or inconsistent mitigation; self-consistency can help but multiplies decoding cost; static head ablations capture some benefit but cannot adapt to example-specific evidence patterns. COMPASS achieves single-pass mitigation via per-token head modulation, which we observe as reduced SD without sacrificing CO.

Limitations

While our dynamic attention head modulation framework reduces hallucinations, several constraints remain. First, head-importance estimation is driven by short-horizon, per-step last-query attention signals and may under-perform in extremely long-context or multi-turn settings where risk accrues gradually without strong local cues. Second, gating decisions are made per step without global discourse awareness, so suppression of longer narrative drifts is limited. Third, our context-modulated PID attention steering system introduces sensitivity to controller hyperparameters: EMA smoothing and hysteresis (β, h), PID gains (k_P, k_I, k_D), and the log-space slew limit and cap $(\Delta_{\log}, \rho_{\max})$; poor tuning can cause oscillatory activation, unstable convergence, or oversuppression of useful heads. Finally, although decoding remains single-stream and attentions are read on-step every k tokens (no second forward pass), computing CRS features and per-layer top-K selection adds modest but non-negligible overhead that can impact latency on smaller GPUs or very long sequences. Our approach also assumes that head importance can be estimated reliably in real time from live CRS (optionally blended with a prior), an assumption that may degrade in highly non-stationary domains. Evaluation to date focuses on standard benchmarks; open-world and adversarial settings remain to be tested.

Related Work

LLMs frequently generate fluent but factually incorrect content ("hallucination"). Prior strategies include grounding with retrieval [4], structured knowledge graphs, or external consistency classifiers [16]. Reinforcement learning from human feedback [1] further improves reliability [9]. However, most approaches act post hoc, correcting outputs after hallucinations emerge, rather than intervening in the model's internal reasoning. Transformer heads vary widely in function and importance [14]. Some heads are strongly tied to factual grounding, while others promote hallucinations [2] Work on pruning and masking [5, 15] shows selective head control can shift model behavior, but interventions are static. Our method instead uses dynamic, classifier-informed modulation, adjusting hallucination-prone heads online during decoding. Control theory, particularly PID feedback, has been widely applied in dynamical systems [12] , optimization [7] , and reinforcement learning [4]. We frame hallucination control as a feedback problem: a classifier monitors drift, while a PID loop gates attention heads in real time. Unlike retrieval-based grounding [11][4] [1] [9] , or static head pruning [14, 15], our contribution is the *first closed-loop framework for hallucination mitigation that unifies detection and internal modulation via PID control.*

Conclusion

COMPASS introduces a lightweight, interpretable, and real-time approach to mitigating contextual hallucinations in LLMs. By embedding a PID-controlled feedback loop into the decoding process and leveraging the Context Reliance Score as a per-head grounding signal, the system achieves preliminary reductions of 2.8–5.8% in hallucination rate, decreases unsupported-span density, and improves context overlap metrics without retraining or multi-pass decoding. The results suggest that attention-level control can complement traditional post hoc mitigation methods, providing a fine-grained mechanism for enhancing factual accuracy during generation.

Mathematically, COMPASS demonstrates that closed-loop feedback applied to attention logits, modeled as:

$$\tilde{Z}_t(\ell, h)[i] = Z_t(\ell, h)[i] + \rho_t a_\ell(h), \quad i \in C,$$

can dynamically steer model outputs toward contextually supported tokens while leaving other attention weights unchanged. This approach highlights the potential for control-theoretic methods in LLM alignment, offering an interpretable, modular alternative to more opaque interventions like fine-tuning or contrastive decoding.

Future work will explore: (i) the integration of richer detector signals that capture semantic coherence, and (ii) formalizing stability guarantees for PID-controlled attention modulation. Overall, these preliminary results validate the feasibility of feedback-driven attention control as a scalable, low-overhead strategy for reducing hallucinations in modern LLMs.

Ethics Statement

This study makes exclusive use of open-source datasets and pre-existing model checkpoints; no personally identifiable information was collected or processed. All resources were accessed under their respective licenses and applied only for research purposes. Our approach aims to strengthen factual grounding in language models, with potential benefits for downstream systems that rely on trustworthy text generation. Nonetheless, inherent risks of bias, misinformation, or offensive outputs remain, underscoring the need for careful monitoring and responsible deployment.

References

[1] Christiano, P. F.; Leike, J.; Brown, T. B.; Martic, M.; Legg, S.; and Amodei, D. 2023. Deep Reinforcement Learning from Human Preferences. https://arxiv.org/abs/1706.03741.

[2] Chuang, Y.-S.; Krishna, R.; and Hashimoto, T. 2024. Lookback Lens: Detecting and Mitigating Contextual Hallucinations in LLMs Using Only Attention Maps. In *Findings of EMNLP 2024*.

[3] Huang, Y.; Zhang, Y.; Cheng, N.; Li, Z.; Wang, S.; and Xiao, J. 2025. Dynamic Attention-Guided Context Decoding for Mitigating Context-Faithfulness Hallucinations in Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2025*.

[4] Lewis, F. L.; and Vrabie, D. 2009. Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control. *IEEE Circuits and Systems Magazine*.

[5] Li, G.; Chen, Y.; and Tong, H. 2025. Taming Knowledge Conflicts in Language Models. https://arxiv.org/abs/2503.10996.

[6] Li, X. L.; Holtzman, A.; Fried, D.; Liang, P.; Eisner, J.; Hashimoto, T.; Zettlemoyer, L.; and Lewis, M. 2023. Contrastive Decoding: Open-ended Text Generation as Optimization. https://arxiv.org/abs/2210.15097.

[7] Lin, T.; and Jordan, M. I. 2021. A Control-Theoretic Perspective on Optimal High-Order Optimization.

[8] Matarazzo, A.; and Torlone, R. 2025. A Survey on Large Language Models with Some Insights on Their Capabilities and Limitations. https://arxiv.org/html/2501.04040v1.

[9] Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training Language Models to Follow Instructions with Human Feedback. https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

[10] Shi, W.; Han, X.; Lewis, M.; Tsvetkov, Y.; Zettlemoyer, L.; and tau Yih, W. 2023. Trusting Your Evidence: Hallucinate Less with Context-aware Decoding. https://arxiv.org/pdf/2305.14739.

[11] Shuster, K.; Poff, S.; Chen, M.; Kiela, D.; and Weston, J. 2021. Retrieval Augmentation Reduces Hallucination in Conversation. https://arxiv.org/pdf/2104.07567.

[12] ström, K. J. A.; and Murray, R. M. 2008. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton, NJ: Princeton University Press. ISBN 9780691135762.

[13] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, ; and Polosukhin, I. 2017. Attention Is All You Need. https://arxiv.org/abs/1706.03762.

[14] Voita, E.; Talbot, D.; Moiseev, F.; Sennrich, R.; and Titov, I. 2019. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *2019 Conference*. See Lena Voita blog: “The Story of Heads”.

[15] Zhang, Y.; and Li, X. 2022-2023. Works on Selective Masking/Gating Heads.

[16] Zhao, Z.; Monti, E.; Lehmann, J.; and Assem, H. 2024. Enhancing Contextual Understanding in Large Language Models through Contrastive Decoding. https://arxiv.org/html/2405.02750v1.