CoRect: Context-Aware Logit Contrast for Hidden State Rectification to Resolve Knowledge Conflicts

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) often struggles with **knowledge conflicts**, where model-internal parametric knowledge overrides retrieved evidence, leading to unfaithful outputs. Existing approaches are often limited, relying either on superficial decoding adjustments or weight editing that necessitates ground-truth targets. Through layer-wise analysis, we attribute this failure to a **parametric suppression** phenomenon: specifically, in deep layers, certain FFN layers overwrite context-sensitive representations with memorized priors. To address this, we propose CoRect (**Co**ntext-Aware Logit Contrast for Hidden State **Rect**ification).By contrasting logits from contextualized and non-contextualized forward passes, CoRect identifies layers that exhibit high parametric bias without requiring ground-truth labels. It then rectifies the hidden states to preserve evidence-grounded information. Across question answering (QA) and summarization benchmarks, CoRect consistently improves faithfulness and reduces hallucinations compared to strong baselines.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in knowledge-intensive tasks (Brown et al., 2020; Touvron et al., 2023).However, because much of their knowledge is encoded in static pre-training corpora, LLMs often hallucinate or produce outdated statements when queried about evolving facts.Retrieval-Augmented Generation (RAG) mitigates this issue by conditioning generation on external evidence retrieved at inference time, thereby improving factuality and robustness (Lewis et al., 2020; Guu et al., 2020). Despite its promise, RAG often fails under *knowledge conflict*—where the retrieved evidence contradicts the model's parametric prior knowledge (Longpre et al., 2021; Chen et al., 2022). In

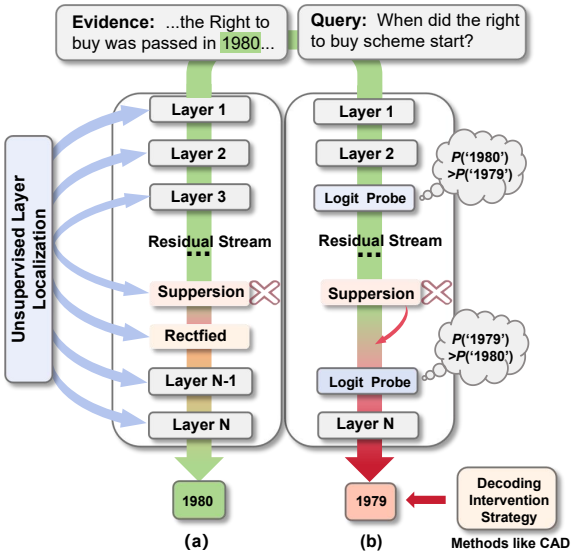

Figure 1: Comparison of intervention strategies. (a) Our method localizes and rectifies parametric suppression layers within the model's internal residual stream. (b) Baseline methods treat the model as a black box by intervening only at the output stage.

such settings, LLMs tend to under-utilize the provided context, favoring internally memorized but incorrect knowleage. As illustrated in Figure 1, when queried about the start date of the right to buy scheme, LLMs favor internal biases 1979 even when context specifies the correct fact 1980.

Recent work has proposed decoding-time corrections to address this failure mode. Prominent approaches include adjusting the final token distribution by subtracting logits attributed to parametric priors (Shi et al., 2024; Wang et al., 2025), or utilizing information entropy constraints to dynamically balance context and memory (Yuan et al., 2024). While effective in many scenarios, these methods largely treat the model as a *black box* and intervene only at the output layer. Such late interventions are weakly grounded in the model's internal computation: they do not prevent intermediate representations from being progressively steered toward the

parametric answer. Moreover, aggressive manipulation of the output distribution can disrupt the model's inherent linguistic priors, often leading to semantic instability or quality degradation.

In this work, we shift from output-level corrections to a mechanistic view of where knowledge conflicts emerge inside the model. Using a layer-wise analysis of the residual stream via Logit Lens (nostalgebraist, 2020), we observe a consistent *parametric suppression* phenomenon: under conflicts, although early layer representations initially reflect the retrieved evidence, deeper layers are progressively overwhelmed by internal priors, effectively suppressing the context to revert toward the and incorrect answer.Inspired by mechanistic interpretability and model editing studies, we hypothesize that this drift is driven by a subset of Feed-Forward Network (FFN) layers that act as key-value memories for parametric facts (Geva et al., 2021; Dai et al., 2022). Concretely, while attention mechanisms successfully incorporate external context, certain FFN layers overwrite this evidence-consistent state with parametric memories, thereby enforcing the prior over the context.

Once the failure is attributed to FFNs, a natural idea is to directly modify these modules. Indeed, ROME-style editing methods update weights in selected FFN layers to rewrite factual associations (Meng et al., 2022a,b). However, such approaches crucially assume access to the answer, which is unrealistic for open-ended generation in RAG. At inference time, the system typically does not know the single gold answer in advance. This limitation motivates a different goal: resolving conflicts without target answer during inference.

We therefore propose a training-free framework that mitigates knowledge conflicts via targeted hidden-state intervention. Our key idea is a dynamic layer localization strategy that identifies conflict-inducing layers where parametric suppression originates—without the correct answer. Once localized, we apply an inference-time intervention that blocks the propagation of conflict-induced parametric information and restores evidence-following behavior, while preserving the model's general generation capabilities.Our contributions are summarized as follows:

- **Unveiling the Parametric Suppression Phenomenon.** We identify a phenomenon termed Parametric Suppression where deep FFN layers override representations with memorized priors. This insight isolates the specific internal components responsible for unfaithful generation.
- **Target-Agnostic Inference-Time Intervention.** We propose a novel intervention framework that localizes conflict-inducing layers without target answers, achieving high alignment with over **70%** recall relative to critical layers identified by ROME. By rectifying hidden states within the identified layers during inference, our approach effectively mitigates knowledge conflicts and reduces hallucinations while preserving fluency.
- **Effective Mitigation of Knowledge Conflicts** Extensive evaluations across Question Answering and Summarization tasks demonstrate the effectiveness of our framework, showing that it outperforms state-of-the-art baselines in resolving knowledge conflicts.

2 Related Work

We tackle RAG knowledge conflicts via decoding, model editing, and mechanistic interpretability.

Decoding-Time Interventions. Knowledge conflicts occur when models prioritize parametric priors over contradictory retrieved evidence (Longpre et al., 2021; Xie et al., 2023). To mitigate this, strategies like CAD (Shi et al., 2024) and AdaCAD (Wang et al., 2025) adjust the final token distribution, typically by contrasting context-augmented logits against a context-free baseline.By treating the model as a black box, they fail to arrest the internal propagation of incorrect parametric information, potentially leading to generation instability despite maintaining surface-level fluency.

Mechanistic Localization and Editing. Going deeper than the output layer, mechanistic studies identify Feed-Forward Networks (FFNs) as key-value memories storing factual associations (Geva et al., 2021; Dai et al., 2022). Methods like ROME (Meng et al., 2022a) and MEMIT (Meng et al., 2022b) leverage this insight to permanently edit FFN weights. However, these approaches require the target answer to compute updates, rendering them unsuitable for open-ended RAG where the correct answer is unknown. Furthermore, permanent parameter modifications lack the flexibility to handle transient, query-specific conflicts.

Inference-Time Activation Intervention. Alternatively, activation engineering methods like ITI (Li et al., 2023) and Representation Engineering (Zou et al., 2023) steer model behavior by injecting static vectors derived from trained probes.

While effective, they typically rely on labeled data for training and apply global, static interventions. In contrast, our framework is **fully training-free and instance-specific**. Instead of relying on static directions or supervised weight updates, we utilize the Logit Lens (nostalgebraist, 2020) to dynamically localize the specific layers where parametric suppression originates for each query, enabling targeted hidden-state intervention that resolves conflicts without pre-trained classifiers.

3 Preliminaries

3.1 Residual Decomposition for Transformer

We analyze a standard L-layer decoder-only LLM, denoted as F_{LLM}, with a vocabulary $\mathcal{V}$. In the prevalent pre-layer normalization (Pre-LN) architecture, the model structure can be formalized as:

$$F_{\text{LLM}} = W_U \circ f_{\text{LN}} \circ f_L \circ \cdots \circ f_1 \circ W_E, \quad (1)$$

where $W_U \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the unembedding matrix, $W_E \in \mathbb{R}^{d \times |\mathcal{V}|}$ is the embedding matrix, and f_{LN} denotes layer normalization. The l-th Transformer layer f_l comprises an attention module and a feed-forward network (FFN), formulated as:

$$f_l = f_{\text{ffn}}^{(l)} \circ f_{\text{LN}} \circ f_{\text{attn}}^{(l)} \circ f_{\text{LN}}, \quad (2)$$

where $f_{\text{ffn}}^{(l)}$ and $f_{\text{attn}}^{(l)}$ denote the FFN and attention modules of the l-th layer, respectively. Let $h_l \in \mathbb{R}^d$ denote the residual stream state at layer l. The stream is updated additively:

$$h_l = h_{l-1} + a_l + u_l, \quad (3)$$

where $a_l = f_{\text{attn}}^{(l)}(f_{\text{LN}}(h_{l-1}))$ and $u_l = f_{\text{ffn}}^{(l)}(f_{\text{LN}}(h_{l-1} + a_l))$. The final logits z_L are obtained by applying a decoding head to the normalized final state: $z_L = W_U f_{\text{LN}}(h_L)$.

To analyze intermediate representations, we employ the Logit Lens technique (nostalgebraist, 2020). By treating Layer Normalization as a locally linear operation or absorbing it into the projection, we can approximate the final logits as a linear decomposition of components from all layers:

$$z_L \approx W_U h_0 + \sum_{l=1}^{L} W_U a_l + \sum_{l=1}^{L} W_U u_l. \quad (4)$$

Eq. (4) interprets the prediction as a superposition of the initial embedding and the outputs from attention heads and MLP blocks.

3.2 Weight Editing as Residual Steering

Most knowledge editing methods modify the parameters of Feed-Forward Network (FFN) modules, operating on the hypothesis that FFNs serve as key-value memories for factual knowledge (Meng et al., 2022a). In the context of residual networks, the output of these modules, u_l, plays a critical role: it is added directly to the residual stream, thereby propagating the encoded factual information as a linear contribution to the final prediction.

General Framework. Given a prompt and a desired target token t^*, the objective of editing is to amplify the logit of t^*. Functionally, parameter edits can be abstracted as injecting a *steering vector* δ_l into the residual pathway:

$$\widehat{u}_l = u_l + \delta_l. \quad (5)$$

Let w_t represent the row vector of W_U corresponding to token t^*. Adopting the view of the residual stream as a linear communication channel (Elhage et al., 2021), and following the decoding analysis in (Geva et al., 2021), we approximate the cumulative effect of the edit as the projection of the steering vector onto the target's unembedding direction(A detailed derivation based on linear channel assumption is provided in Appendix A):

$$\Delta z_L(t^*) \approx \sum_{l \in \xi} w_{t^*}^\top \delta_l, \quad (6)$$

where ξ indexes the edited layers. Eq. (6) reveals that the optimal steering direction is explicitly aligned with w_{t^*}, making target-guided residual rectification inherently label-dependent.

Analysis of ROME. Formally, we define the *post-activation key vector* m_l as the intermediate activation produced by the up-projection:

$$m_l = \sigma\left(W_{\text{up}}^{(l)} f_{\text{LN}}(h_{l-1} + a_l)\right). \quad (7)$$

Consequently, the output of the FFN at layer l, which propagates to the residual stream, is computed as:

$$u_l = W_{\text{down}}^{(l)} m_l. \quad (8)$$

where σ is the activation function. ROME (Meng et al., 2022a) instantiates this abstraction by realizing δ_l via a rank-one weight update. ROME computes a parameter update $\Delta W^{(l)}$:

$$\delta_l = \Delta W^{(l)} m_l. \quad (9)$$

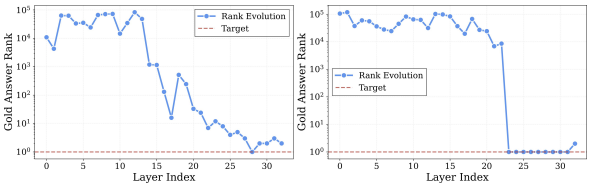

Figure 2: **Rank evolution analysis.** (Top) Middle layer flip pattern. (Bottom) Last layer flip pattern.

ROME(Detailed proofs are provided in Appendix B) constructs $\Delta W^{(l)}$ specifically to align the induced δ_l with the target direction w_{t^*}. This highlights a critical limitation: existing weight-editing approaches heavily rely on an *oracle target* t^* to determine the update direction. This dependency precludes their application in autonomous correction settings where the ground truth label is unavailable.

4 Methodology

4.1 Parametric Suppression

To understand why the model fails to generate the correct answer despite potentially encoding it, we investigate the internal representation dynamics using the Logit Lens technique (nostalgebraist, 2020). Formally, we project the hidden state h_l of each layer l directly onto the vocabulary space $\mathcal{V}$ using the pre-trained language modeling head W_U. This allows us to interpret the model's instantaneous prediction preference at specific depths of the network.We observe a counter-intuitive phenomenon where the gold answer token often achieves a high probability in intermediate layers, only to be suppressed in the final output. We term this phenomenon **Parametric Suppression**.

Qualitative Analysis. As illustrated in Figure 2, we analyze the rank evolution of the gold answer for the query: *"How many episodes in season 5 of Curse of Oak Island?"*. In the intermediate layers, the model's internal states strongly favor the correct answer 111, However, as the information propagates to the final layers, the rank of 111 drops significantly, and the model ultimately outputs the incorrect token 5. This discrepancy suggests that the correct knowledge was successfully retrieved and encoded in the middle layers but was subsequently overridden during the forward pass.

To verify the prevalence of this phenomenon, we conducted a systematic evaluation on 500 sampled cases from the Natural Questions (NQ) training set that involve knowledge conflicts. We tracked the rank evolution of the gold answers across all layers. Our analysis reveals that **282 out of 500 cases** exhibit this flip phenomenon: the gold answer reaches Rank 1 at some point but is not the final prediction. Specifically, we categorize these into two patterns: Middle Flip (154 cases):The gold answer surfaces in the intermediate layers but is lost in the later stages. Last Layer Flip (128 cases):The gold answer remains dominant until the very last projection layer, where it is suddenly suppressed.The high prevalence of parametric suppression indicates that the model often possesses the capability to solve the problem, yet fails to manifest it in the final generation. While we acknowledge that some rank fluctuations may reflect legitimate reasoning steps, the fact that over 56% of error cases contain the correct answer internally serves as a strong motivation for our approach. It suggests that instead of injecting new knowledge, we can improve performance by intervening in the decoding process to release the suppressed correct information.

4.2 Transition: From Supervised Overwriting to Autonomous Rectification

The foregoing analysis underscores that conventional editors operationalize editing as *supervised overwriting*: an externally specified t^* is used to construct a parameter patch that compels the model toward a designated output. In contrast, our approach departs from this paradigm along two axes—*how the target is obtained* and *how the intervention is applied*. we introduce a two-stage procedure: (1) **Token Selection** We infer a reliable target token $\tilde{t}^*$ from the input prompt itself, thereby removing the requirement for oracle target; (2) **Hidden State Rectification** Instead of forcefully steering generation toward $\tilde{t}^*$, we apply a minimal intervention to the hidden representation u_l that selectively attenuates components that *inhibit* $\tilde{t}^*$. Conceptually, we frame this as disinhibition rather than injection. By removing suppressive components, we aim to preserve the model's native generative dynamics while selectively mitigating hallucinations.The overall architecture of our proposed model is Figure 3.

4.3 Stage 1: Trustworthy Token Selection

Simple subtraction of logits often amplifies noise alongside the correct answer. To select a robust target token $\tilde{t}^*$ at current step, we integrate Contextual Mutual Information with attention-based evidence.

Contextual Mutual Information Estimation We perform two forward passes: one with the full

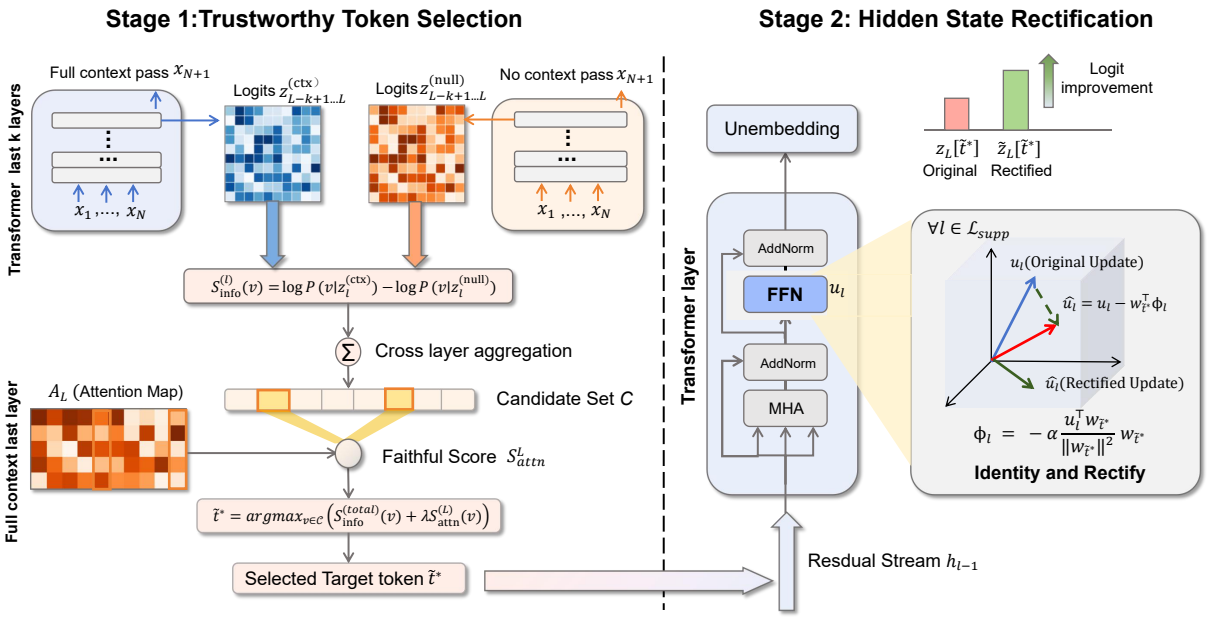

Figure 3: The overall architecture of our proposed model.

input prompt and one with a null context consisting only of the instruction tokens, so that the null pass captures generic prior preferences independent of the specific query. Let $z_l^{(\text{ctx})}$ and $z_l^{(\text{null})}$ denote the logits at layer l, obtained by projecting the layer's hidden state onto the vocabulary using the logit lens. For each token $v \in \mathcal{V}$, we compute the layer-wise $S^{(l)}_{\text{info}}(v)$ which captures the boost in likelihood for token v provided by the context:

$$S^{(l)}_{\text{info}}(v) = \log P(v \mid z_l^{(\text{ctx})}) - \log P(v \mid z_l^{(\text{null})}), \tag{10}$$

where $P(\cdot \mid z) = \text{softmax}(z)$ represents the probability distribution over the vocabulary given logits z. To capture the evolution of information, we focus on the last k layers. We aggregate these scores via the mean:

$$\tilde{S}_{\text{info}}(v) = \frac{1}{k} \sum_{l=L-k+1}^{L} S^{(l)}_{\text{info}}(v). \tag{11}$$

To ensure numerical stability, we apply sign-preserving max-normalization. The global information score is defined as:

$$S^{(\text{total})}_{\text{info}}(v) = \frac{\tilde{S}_{\text{info}}(v)}{\max_{v' \in \mathcal{V}} \left|\tilde{S}_{\text{info}}(v')\right| + \epsilon}. \tag{12}$$

where ϵ is a small scalar for numerical stability.

Candidate Selection & Attention Filter. We define the candidate set $\mathcal{C}$ as the top-M tokens ranked by $S^{(\text{total})}_{\text{info}}$. Then we employ an attention-based verification mechanism that favors candidates grounded in the input context. For each candidate $v \in \mathcal{C}$, we perform a token matching step to identify its occurrences in the context $\tilde{\mathbf{x}}$ of the input $\mathbf{x}$. Let $\mathcal{I}(v) = \{j \mid \tilde{x}_j = v\}$ denote the set of position indices where the token v appears. We then quantify the evidence for v by aggregating the attention mass the model allocates to these matched positions:

$$\tilde{S}^{(L)}_{\text{attn}}(v) = \begin{cases} \dfrac{1}{H} \displaystyle\sum_{h=1}^{H} \sum_{j \in \mathcal{I}(v)} A^{(\text{curr},j)}_{L,h}, & \text{if } \mathcal{I}(v) \neq \emptyset \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

where H represents the total number of attention heads. The term $A^{(\text{curr},j)}_{L,h}$ denotes the attention weight assigned by the h-th head at layer L, This formulation essentially averages the attention mass that the model's final layer directs toward the candidate v.This metric serves as a *Faithfulness Score*, ensuring that the selected target is not only informative but also supported by the model's internal attention to the context. We apply the same sign-preserving max-normalization (as defined in Eq. 12) to the raw score $\tilde{S}^{(L)}_{\text{attn}}(v)$ to obtain $S^{(L)}_{\text{attn}}(v)$. At current autoregressive decoding step t, the step-wise target token $\tilde{t}^*_t$ is selected to maximize the following joint objective:

$$\tilde{t}^* = \underset{v \in \mathcal{C}}{\text{argmax}} \left(S^{(\text{total})}_{\text{info}}(v) + \lambda S^{(L)}_{\text{attn}}(v) \right), \tag{14}$$

where λ balances contextual induction with explicit evidence support, thereby offering the flexibility to accommodate varying demands ranging from extractive grounding to logical deduction.

4.4 Stage 2: Hidden State Rectification

Given the target unembedding vector $w_{\tilde{t}^*}$, we identify the *suppressive layers* $\mathcal{L}_{supp}$ where the MLP's direct contribution negatively impacts the target:

$$\mathcal{L}_{supp} = \left\{ l \in \{1, \ldots, L\} \;\middle|\; u_l^\top w_{\tilde{t}^*} < 0 \right\}. \quad (15)$$

To mitigate this, we construct an additive patch ϕ_l to cancel the projection of u_l onto the $w_{\tilde{t}^*}$ direction. We define the rectification patch:

$$\phi_l \;=\; -\alpha \frac{u_l^\top w_{\tilde{t}^*}}{\|w_{\tilde{t}^*}\|^2}\, w_{\tilde{t}^*}, \quad \forall l \in \mathcal{L}_{supp}. \quad (16)$$

This formulation offers a tunable spectrum of intervention controlled by α. Specifically, setting $\alpha = 1$ reduces the term to zero, strictly *neutralizing* the suppression. This corresponds to the pure "disinhibition" paradigm discussed in Sec. 4.2, where the blockage is removed without imposing new directional force. In contrast, setting $\alpha > 1$ flips the sign of the contribution from negative to positive, transitioning the operation from mere removal of inhibition to active *promotion*of evidence for $\tilde{t}^*$. Regardless of the specific regime, the net change in the final logit is geometrically guaranteed to be positive:

$$\Delta z_L(\tilde{t}^*) \triangleq \sum_{l \in \mathcal{L}_{supp}} w_{\tilde{t}^*}^\top \phi_l = -\alpha \sum_{l \in \mathcal{L}_{supp}} w_{\tilde{t}^*}^\top u_l > 0. \quad (17)$$

To verify the optimality of our identified layer l compared to ROME's $l^\star$(the derivation of which is detailed in Appendix C), we analyze the editing performance across different layers. Figure 4 illustrates the trends of F1 score and QA accuracy as we vary the intervention layer.

5 Experiments

We evaluate our proposed method on both Question Answering (QA) and Summarization tasks. Our experiments aim to demonstrate the effectiveness of our method in mitigating hallucinations and prioritizing contextual faithfulness compared to strong inference-time intervention baselines.Detailed formalizations of the decoding-time correction baselines and the specific hyperparameter configurations for all methods are provided in Appendix E and Appendix F, respectively.

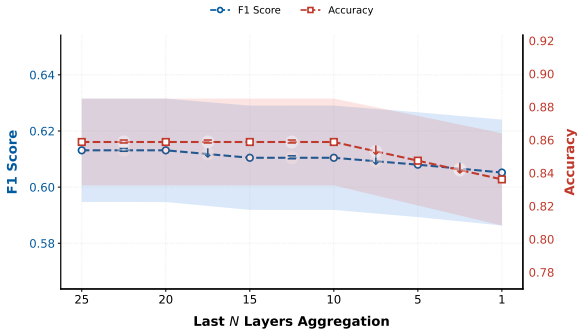

Figure 4: Performance metrics across different layer aggregation scales. The F1 score (blue) validates the localization accuracy by comparing ROME-identified layers l^* with our defined layers l, while the accuracy (red) illustrates the performance stability after applying our correction method.

5.1 Experimental Setup

Datasets and Metrics. To ensure a comprehensive evaluation, we test CoRect across a wide spectrum of tasks. For **QA and Reasoning**, we cover NQ (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), TabMWP (Lu et al., 2022), TriviaQA (Joshi et al., 2017), SQuAD (Rajpurkar et al., 2016)and standard/high-conflict benchmarks NQ-Swap (Longpre et al., 2021). For **Summarization**, we include short/long-form synthesis XSum (Narayan et al., 2018), CNN-DM (Hermann et al., 2015)) and topic-aligned evaluation (TofuEval (Tang et al., 2024). We report Exact Match (EM) for QA, alongside ROUGE-L (Lin, 2004), BERTScore-P (Zhang et al., 2019), and AlignScore (Zha et al., 2023) to rigorously assess faithfulness. Detailed dataset statistics and metric formalizations are provided in Appendix D.

5.2 Main Results

Question Answering. Table 1 summarizes the performance across QA benchmarks. CoRect achieves consistently strong EM scores on TriviaQA and SQuAD, while demonstrating superior evidence extraction on NQ, even within long-context documents. Most notably, on NQ-Swap—a high-conflict stress test—CoRect yields substantial gains, showcasing its robustness against direct contradictions between context and priors. Furthermore, CoRect consistently improves performance on reasoning-intensive tasks like HotpotQA and TabMWP across various backbones.

Summarization and Alignment. We evaluate CoRect on summarization benchmarks (Table 2). CoRect consistently outperforms strong baselines

Model	Method	NQ	NQ-Swap	HotpotQA	TriviaQA	TabMWP	SQuAD	Avg
Qwen2.5-7b-instruct	Greedy	69.49	70.54	43.30	76.85	59.90	81.27	66.89
	CAD	67.28	79.37	40.31	76.75	58.90	84.43	67.84
	COIECD	59.87	78.56	41.90	**80.75**	64.35	88.90	69.06
	AdaCAD	68.54	73.14	43.82	76.90	60.00	81.20	67.27
	CoRect	**72.74**	**80.15**	**45.67**	79.60	**70.60**	**88.93**	**72.95**
LLaMA3-8b-instruct	Greedy	68.72	60.67	39.47	80.30	47.40	81.93	63.08
	CAD	68.17	**80.10**	36.44	81.20	48.90	**84.93**	66.62
	COIECD	60.23	70.39	37.10	80.95	51.95	84.40	64.17
	AdaCAD	67.57	67.37	40.43	80.30	47.30	81.87	64.14
	CoRect	**71.22**	79.32	**41.15**	**83.00**	**52.60**	83.70	**68.50**
Mistral-7b	Greedy	46.34	44.51	31.60	72.10	25.20	63.93	47.28
	CAD	33.50	65.72	29.17	72.50	35.40	72.90	51.53
	COIECD	36.85	42.96	29.23	76.35	36.50	74.20	49.35
	AdaCAD	44.56	63.99	34.69	72.00	25.40	63.87	50.75
	CoRect	**56.31**	**70.37**	**36.06**	**78.60**	**38.50**	**74.33**	**59.03**
LLaMA2-13b-chat	Greedy	44.15	49.29	21.27	55.60	17.90	52.17	40.06
	CAD	46.88	74.59	23.53	63.55	**27.30**	68.97	50.80
	COIECD	43.11	54.41	20.83	60.35	26.70	58.67	44.01
	AdaCAD	43.40	57.69	29.13	55.05	17.70	51.80	42.46
	CoRect	**49.01**	**76.52**	**29.45**	**68.15**	27.20	**70.70**	**53.51**

Table 1: Comparison of different methods across multiple models on NQ, NQ-Swap, HotpotQA, TriviaQA,TabMWP and SQuAD.

Method	CNN-DailyMail			XSum			TofuEval (AlignScore)	
	ROUGE-L	BERT-P	AlignScore	ROUGE-L	BERT-P	AlignScore	Overall	Main / Marginal
Greedy	21.08	85.83	91.48	16.42	86.56	77.65	43.65	49.92 / 37.39
CAD	18.40	84.66	76.91	15.25	84.30	67.00	50.33	53.42 / 47.24
COIECD	21.17	85.86	91.63	15.77	86.48	81.06	49.61	64.26 / 34.95
AdaCAD	21.27	85.63	91.87	15.81	86.43	82.02	57.11	66.33 / 47.89
CoRect	**21.97**	**86.03**	**92.88**	**20.04**	**87.30**	**83.30**	**69.45**	**73.69 / 65.21**

Table 2: Abstractive summarization performance on LLaMA-3-8b. CoRect significantly improves factual alignment (AlignScore) while maintaining generation quality.

across XSum, CNN/DailyMail, and TofuEval, demonstrating superior generalization across varying summary lengths and topic constraints.

5.3 Ablation Studies

Q1: How Effective is the Token Selection Strategy? We compared our token selection strategy against baseline methods. As shown in Figure 5(a), when the number of intervened layers k=1, our token selection strategy begins to above the baseline method, In this setting, our method demonstrates competitive performance on both the **NQ** and **XSum** datasets, confirming the effectiveness of our token selection strategy.

Q2: Does the number of intervened layers k matter? We analyzed the performance on NQ by varying the number of last layers $k \in \{1, 5, 10, 15, 20, 25\}$. As shown in Figure 5(a), performance peaks at $k = 10$, but slightly degrades as we intervene with too many layers. Additionally, we conducted similar experiments on the XSum dataset. As shown in Figure 5(b), performance was maximized at k = 5, where the summary quality metrics reached their highest values. These findings suggest that, while increasing the number of layers can help capture more nuanced information, excessive layer interventions may lead to the inclusion of irrelevant or noisy tokens, negatively affecting model performance on both NQ and XSum.

Q3: Is the Attention Filter Useful? We applied the Attention Filter component on both the NQ and XSum datasets. As shown in Figure 5(c)(d), the results demonstrate that the attention weight, denoted by λ, plays a crucial role in performance

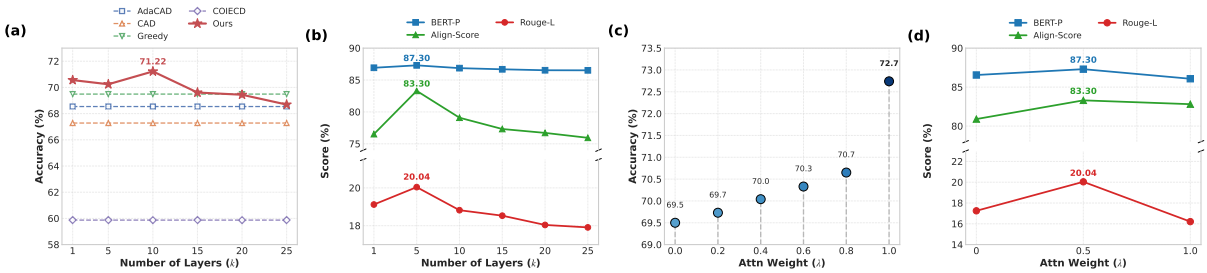

Figure 5: **Hyperparameter sensitivity and performance analysis. (a)** Accuracy with varying number of layers K on NQ. **(b)** Effect of layer depth on generation scores on XSUM. **(c)** Impact of attention weight (λ) on model accuracy on NQ. **(d)** Impact of attention weight (λ) on generation scores on XSUM.

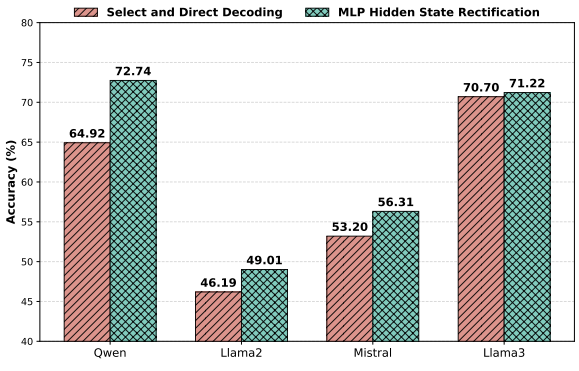

Figure 6: Accuracy comparison of different decoding strategies. Our Hidden State Rectification consistently outperforms Direct Decoding across all backbones.

optimization. On the **NQ** dataset, performance improves gradually as the attention weight increases, reflecting the benefit of utilizing more context into the model's responses. On the **XSum** dataset, the performance peaked at $\lambda = 0.5$. This difference can be attributed to the nature of the task. Unlike QA, where answers are typically extracted directly from the context , the summarization task in XSum requires the model to generate responses based on a broader understanding of the content. High attention weights in this case may cause the model to focus too heavily on specific parts of the input, which could limit its ability to generate a coherent and high-quality summary. Therefore, a moderate attention weight of $\lambda = 0.5$ strikes the optimal balance between context understanding and generating summaries with appropriate semantic depth.

Q4: Is Hidden State Rectification Effective? To evaluate the effectiveness of the hidden state rectification component, we compare the performance of directly decoding the target token selected in Stage 1 with that of using hidden state rectification. As shown in Figure 6, our method outperforms the direct decoding baseline. The core

Context: Suddenly is a song from Xanadu... performed as a duet between Olivia Newton-John and Cliff Richard...

Q: who sang the song suddenly with olivia newton john?
Gold: Cliff Richard
Direct Decoding: Ciff Richard sang the song...
Ours (Rect.): Cliff Richard sang the song...

Figure 7: Our method rectifies the hidden state to generate the (Cliff) where direct decoding fails (Ciff).

reason for this improvement lies in the fact that while the target token $\tilde{t}^*$ selected in Stage 1 provides a reliable direction, it is not always the optimal final token4. Direct decoding forces the model toward $\tilde{t}^*$, which can sometimes lead to semantic incoherence. In contrast, our rectification mechanism treats the selected token as a guiding vector to neutralize internal parametric suppression. This approach allows the model to maintain its native generative capabilities while surfacing the correct information, resulting in higher accuracy and better linguistic quality.A representative case demonstrating the efficacy of our approach is illustrated in Figure 7.

6 Conclusion

In this work, we unveil Parametric Suppression, a phenomenon where deep FFN layers in LLMs override retrieved evidence and revert to memorized priors under knowledge conflicts. To address this, we propose CoRect, a target-agnostic hidden-state rectification method that localizes conflict-inducing layers and applies targeted interventions to neutralize parametric bias. Extensive evaluations demonstrate that improving RAG faithfulness is more effectively achieved via internal mechanistic corrections rather than via superficial output-level calibration. Ultimately, CoRect offers a practical, interpretable and training-free framework for solving knowledge conflicts in real-world.

7 Limitations

While CoRect effectively mitigates knowledge conflicts, it introduces non-negligible computational overhead compared to standard decoding. The framework necessitates a multi-pass inference scheme and stage-wise hidden state rectification, which entails frequent access to residual streams across layers. To alleviate this temporal burden, KV cache can be reused across passes to minimize redundant prefix computations, preventing a strictly linear growth in latency. While CoRect still introduces additional latency compared to vanilla decoding, this trade-off facilitates a more mechanistically interpretable and context-faithful resolution of knowledge conflicts.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Hung-Ting Chen, Michael Zhang, and Eunsol Choi. 2022. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2292–2307.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, and 1 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra 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. *Advances in neural information processing systems*, 33:9459–9474.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2022. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

nostalgebraist. 2020. Interpreting GPT: the logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. Accessed: 2026-01-05.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791.

Liyan Tang, Igor Shalyminov, Amy Wong, Jon Burnsky, Jake Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, and 1 others. 2024. Tofueval: Evaluating hallucinations of llms on topic-focused dialogue summarization. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4455–4480.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025. Adacad: Adaptively decoding to balance conflicts between contextual and parametric knowledge. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11636–11652.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2023. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. *arXiv preprint arXiv:2305.13300*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Xiaowei Yuan, Zhao Yang, Yequan Wang, Shengping Liu, Jun Zhao, and Kang Liu. 2024. Discerning and resolving knowledge conflicts through adaptive decoding with contextual information-entropy constraint. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 3903–3922.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Derivation of the Optimal Steering Direction

In this section, we provide a formal justification for Eq. (6) and the optimality of aligning δ_l with w_{t^*}.

A.1 First-Order Approximation of Logit Shift

Let h_L be the final hidden state of the Transformer before the unembedding layer. The logit for the target token t^* is given by the inner product $z_L(t^*) = w_{t^*}^\top h_L$. When we introduce perturbations $\{\delta_l\}_{l \in l^*}$ at intermediate layers, the change in the target logit $\Delta z_L(t^*)$ can be estimated using a first-order Taylor expansion:

$$\Delta z_L(t^*) \approx \sum_{l \in l^*} \left(\nabla_{h_l} z_L(t^*)\right)^\top \delta_l \tag{18}$$

where h_l is the hidden state at layer l. Applying the chain rule, the gradient with respect to h_l is:

$$\nabla_{h_l} z_L(t^*) = \left(\frac{\partial h_L}{\partial h_l}\right)^\top \nabla_{h_L} z_L(t^*) = \mathbf{J}_{l \to L}^\top w_{t^*} \tag{19}$$

where $\mathbf{J}_{l \to L} = \frac{\partial h_L}{\partial h_l}$ is the Jacobian matrix representing the transformation of the residual stream through subsequent layers.

A.2 Linear Communication Channel Assumption

Following (Elhage et al., 2021), the linear communication channel hypothesis assumes that the residual stream acts as a high-fidelity conveyor where information is primarily transmitted linearly. Mathematically, this implies that for the relevant steering

components, the Jacobian matrix is approximately identity:

$$\mathbf{J}_{l \to L} \approx \mathbf{I} \tag{20}$$

Substituting this into the expansion, we obtain the approximation used in the main text:

$$\Delta z_L(t^*) \approx \sum_{l \in l^*} w_{t^*}^\top \delta_l \tag{21}$$

A.3 Proof of Optimality

To find the optimal steering direction δ_l, we consider the optimization problem under a fixed norm constraint $\|\delta_l\| = \epsilon$ for each layer:

$$\max_{\delta_l} \quad w_{t^*}^\top \delta_l \quad \text{s.t.} \quad \|\delta_l\| = \epsilon \tag{22}$$

By the Cauchy-Schwarz inequality:

$$|w_{t^*}^\top \delta_l| \leq \|w_{t^*}\| \|\delta_l\| \tag{23}$$

The equality (and thus the maximum value) is achieved if and only if δ_l is linearly dependent on w_{t^*} with a positive scalar:

$$\delta_l \propto w_{t^*} \tag{24}$$

This confirms that under the linear channel assumption, the optimal direction to rectify the residual stream for a target token t^* is to align the perturbation with its corresponding unembedding vector w_{t^*}.

B Derivation of Rank-One Model Editing

This appendix details the derivation of the ROME update rule referenced in Section 3.2. We explicitly show how the optimization objective translates into the rank-one weight update and connect this back to the residual steering framework introduced in Eq. (9).

B.1 Constrained Optimization Problem

Let W_0 be the original weight matrix of the FFN down-projection at layer l. In the notation of the main text, the input to this layer is the post-activation vector m_* (associated with the subject token), and the original output is $u_* = W_0 m_*$.

ROME seeks an updated matrix $\widehat{W}$ that satisfies two conflicting objectives:

1. **Target Alignment (The Edit):** For the specific subject key m_*, the output should map to a new target vector v_*. This v_* is optimized specifically to maximize the probability of the target token t^*:

$$\widehat{W} m_* = v_* \tag{25}$$

2. **Knowledge Preservation:** For all other keys m drawn from the general distribution of text (characterized by covariance $C = \mathbb{E}[mm^T]$), the output should remain unchanged:

$$\min_{\widehat{W}} \mathbb{E}_{m \sim C}[\|\widehat{W} m - W_0 m\|^2] \tag{26}$$

Defining $\Delta W = \widehat{W} - W_0$, the problem simplifies to minimizing the norm of the change weighted by the covariance C, subject to the linear constraint:

$$\begin{aligned} \min_{\Delta W} \quad & \text{Tr}(\Delta W C \Delta W^T) \\ \text{s.t.} \quad & \Delta W m_* = v_* - W_0 m_* \end{aligned} \tag{27}$$

B.2 Analytical Solution

Using Lagrange multipliers (similar to the standard derivation in (Meng et al., 2022a)), the optimal closed-form solution is given by:

$$\Delta W = \frac{v_* - W_0 m_*}{m_*^T C^{-1} m_*} (C^{-1} m_*)^T \tag{28}$$

This confirms that the update is a rank-one matrix, defined by the outer product of the *residual vector* $(v_* - W_0 m_*)$ and the *whitened key direction* $(C^{-1} m_*)^T$.

B.3 Connection to Residual Steering and Target Dependency

We now link this result back to the residual steering abstraction in Section 3.2. Substituting the derived update ΔW (Eq. 28) into the steering equation (Eq. 9), the effective steering vector δ_l injected into the residual stream for the subject m_* is:

$$\begin{aligned} \delta_l &= \Delta W m_* \\ &= \left(\frac{v_* - W_0 m_*}{m_*^T C^{-1} m_*} (C^{-1} m_*)^T \right) m_* \\ &= (v_* - W_0 m_*) \frac{m_*^T C^{-1} m_*}{m_*^T C^{-1} m_*} \\ &= v_* - W_0 m_* \end{aligned} \tag{29}$$

This derivation explicitly reveals the dependency discussed in the main text:

- The steering vector $\delta_l = v_* - u_{original}$ is exactly the difference required to shift the representation from the old knowledge to the new target vector v_*.
- Since v_* is optimized to maximize $P(t^*|x)$, the steering vector δ_l is inherently constructed to align with the unembedding direction w_{t^*} (i.e., $\delta_l \approx \alpha w_{t^*}$).

This mathematically validates our claim in Section 3.2: ROME's weight update is equivalent to hard-coding a steering vector that is functionally dependent on the oracle label t^*.

C Localization and Identification of Factual Interference

This section describes the procedure for identifying factual interference within the model's layers. While traditional causal analysis focuses on facilitatory sites, our study prioritizes the extraction of disruptive layers.

C.1 Causal Tracing and the ROME Reference Site

The ROME framework employs **Causal Tracing** to evaluate the contribution of each layer to a specific factual prediction. By calculating the **Average Indirect Effect (AIE)**, the method quantifies how the probability of a target token t^* recovers when a layer's activation is restored to its "clean" state during a corrupted inference run.

In standard practice, ROME identifies a specific layer with the maximum positive AIE to serve as the singular editing site. We denote this traditional reference index as l_{ROME}:

$$l_{\text{ROME}} = \arg\max_{l \in \{1,\dots,L\}} \text{AIE}_l. \tag{30}$$

C.2 Defining the Interference Set as l^*

Our research shifts the focus toward layers that exhibit a negative causal impact. We define the set of **Interference Layers** as l^*. This set comprises all layers where the restoration of clean activations leads to a further suppression of the target probability:

$$l^* = \{l \mid \text{AIE}_l = P_{\text{restored}}(t^*) - P_{\text{corr}}(t^*) < 0\}. \tag{31}$$

A negative AIE suggests that these layers host factual conflicts or inhibitory mechanisms relative to the target fact.

C.3 Detailed Algorithm

The following algorithm formalizes the extraction of the interference set l^* alongside the traditional ROME index l_{ROME}.

Algorithm 1 Extraction of Interference Layer Set l^*

Require: Model $\mathcal{M}$, prompt x with subject s, target t^*, noise scale σ.
Ensure: l^* and l_{ROME}.

$P_{\text{clean}}(t^*), \{h_l\}_{l=1}^{L} \leftarrow \mathcal{M}(x)$ {Baseline Run}
$x_{\text{corr}} \leftarrow \text{Embed}(x) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2 I)$ {Corrupted Run}
$P_{\text{corr}}(t^*) \leftarrow \mathcal{M}(x_{\text{corr}})$
Initialize empty set $l^* = \emptyset$
for each layer $l \in \{1, \dots, L\}$ **do**
  $P_{\text{restored}}(t^*) \leftarrow \mathcal{M}(x_{\text{corr}} \mid \text{do}(h_l^{(s)} \leftarrow h_l))$
  $\text{AIE}_l \leftarrow P_{\text{restored}}(t^*) - P_{\text{corr}}(t^*)$
  if $\text{AIE}_l < 0$ **then**
    $l^* \leftarrow l^* \cup \{l\}$
  end if
end for
$l_{\text{ROME}} \leftarrow \arg\max_l(\text{AIE}_l)$
return l^*, l_{ROME}

D Dataset Descriptions

This section provides a comprehensive overview of the datasets utilized in our experiments, organized by their primary evaluation tasks.

Data Selection and Scaling. Following the data processing methodology in AdaCAD (Wang et al. (2025)) and considering computational resource constraints, we conduct our evaluation on representative subsets of the original benchmarks. Specifically, for QA and reasoning tasks, we evaluate on: NQ (3,481 samples), HotpotQA (4,690 samples), NQ-Swap (4,000 samples), TabMWP (1,000 samples), TriviaQA (2,000 samples), and SQuAD (3,000 samples). For summarization tasks, we utilize XSum (600 samples), CNN-DailyMail (600 samples), and TofuEval (300 samples).

D.1 Question Answering and Reasoning

Natural Questions (NQ) (Kwiatkowski et al., 2019) consists of real search queries from Google. It requires models to find answers within Wikipedia articles, evaluating real-world information retrieval and extraction.

HotpotQA (Yang et al., 2018) focuses on multi-hop reasoning, where the model must aggregate information across multiple documents to arrive at the final answer.

TriviaQA (Joshi et al., 2017) includes a large collection of questions authored by trivia enthusiasts. It provides a challenging test for long-form reading comprehension and open-domain QA.

SQuAD (Rajpurkar et al., 2016) is a benchmark where the answer to every question is a span

of text from the provided reading passage, testing basic reading comprehension.

TabMWP (Lu et al., 2022) provides mathematical word problems based on tabular data, necessitating both linguistic understanding and numerical reasoning.

NQ-Swap (Longpre et al., 2021) is a robustness benchmark that substitutes ground-truth entities with plausible alternatives. It is used to assess how models resolve conflicts between their internal parameters and the provided external context.

D.2 Summarization and Faithfulness

CNN-DailyMail (DM) (Hermann et al., 2015) contains news stories and multi-line summaries. It is widely used to evaluate both extractive and abstractive summarization capabilities.

XSum (Narayan et al., 2018) provides BBC articles paired with single-sentence summaries. The task is highly abstractive, demanding significant compression and rephrasing.

TofuEval (Tang et al., 2024) is designed to measure the factual consistency and faithfulness of models, particularly in scenarios involving fictitious information or knowledge unlearning.

E Details of Decoding-time Correction Methods

Following the discussion in the main text, this section provides the formal definitions for the decoding-time strategies used to mitigate knowledge conflicts. We denote the context-aware distribution as $P_{ctx} = P(y_t|y_{<t}, x, c)$ and the context-free (null) distribution as $P_{null} = P(y_t|y_{<t}, x)$.

1. CAD (Shi et al., 2024) Context-Aware Decoding (CAD) applies a contrastive objective to the logit space to amplify the influence of the external context c while suppressing parametric priors. The corrected distribution is formulated as:

$$P_{CAD} \propto \exp\left((1+\alpha)\log P_{ctx} - \alpha \log P_{null}\right) \tag{32}$$

where $\alpha \geq 0$ is a hyperparameter that determines the strength of the contextual shift.

2. AdaCAD (Wang et al., 2025) AdaCAD improves upon the static nature of CAD by introducing an adaptive weight α_t, allowing the model to dynamically regulate the importance of the context at each decoding step t:

$$P_{AdaCAD} \propto \exp\left(\log P_{ctx} + \alpha_t \log \frac{P_{ctx}}{P_{null}}\right) \tag{33}$$

By scaling the log-ratio of the two distributions, AdaCAD selectively follows the context only when the discrepancy between P_{ctx} and P_{null} indicates a potential knowledge conflict.

3. COIE (Yuan et al., 2024) The Contextual Information-Entropy (COIE) framework utilizes an entropy-based constraint to discern and resolve conflicts. It measures the uncertainty and information gain through the cross-entropy between the context-aware and prior distributions:

$$\mathcal{H}(P_{ctx}, P_{null}) = -\sum_{y \in \mathcal{V}} P_{ctx}(y) \log P_{null}(y) \tag{34}$$

This entropy constraint serves as a diagnostic tool to determine the reliability of the retrieved context, ensuring the model adaptively balances its internal memory with external evidence based on the information-theoretic properties of the current token.

F Implementation Details

For the hyperparameter configurations of the decoding methods used in our experiments, we follow the settings established by prior research to ensure fair comparison. The specific details are as follows:

- **CAD:** Following Shi et al. (2024), we set the scaling factor $\alpha = 1$ for all Question Answering (QA) datasets to emphasize the context. For summarization datasets, α is set to 0.5 to balance the prior knowledge and the source document.
- **COIECD:** In accordance with Yuan et al. (2024), the hyperparameter λ is fixed at 0.25 across all tasks. The entropy constraint parameter α follows the same task-specific settings as CAD: $\alpha = 1$ for QA datasets and $\alpha = 0.5$ for summarization datasets.
- **ADACAD:** Unlike the fixed settings above, ADACAD(Wang et al. (2025)) employs a dynamic adjustment mechanism for the parameter α. The value of α is computed instance-wisely, varying according to the quantified degree of knowledge

conflict detected between the model's parametric memory and the provided context.

For our proposed method, we maintain $\alpha = 1$ across all datasets, while the values of k and λ are tailored to the task requirements. Specifically, for QA tasks, we set $k = 10$, with $\lambda = 0.8$ for reasoning-intensive datasets (e.g., HotpotQA) and $\lambda = 1.0$ for standard QA datasets. For summarization tasks, we adopt $k = 5$ and $\lambda = 0.5$ to strike a balance between context faithfulness and linguistic fluency.

G α Rectification

To further elucidate the mechanism of hidden state intervention, we conduct a sensitivity analysis of the scaling factor α using the Natural Questions (NQ) dataset. This parameter defines a tunable spectrum of intervention intensity: (i) when $0 < \alpha < 1$, the framework performs *partial suppression*, attenuating but not fully neutralizing the conflicting parametric bias; (ii) at $\alpha = 1$, the operation strictly *neutralizes* the suppression, aligning with the pure disinhibition paradigm discussed in Sec. 4.2; and (iii) for $\alpha > 1$, the intervention transcends mere removal of inhibition, actively *promoting* the evidence for the target token $\tilde{t}^*$ by flipping the sign of the corrective signal.Experimental results on the NQ dataset, illustrated in Figure 8, reveal a concave relationship between model performance and the scaling factor. We observe that the performance peaks at $\alpha = 1$, confirming that strict neutralization of parametric bias offers the most balanced intervention. While partial suppression ($0 < \alpha < 1$) shows steady improvement over the baseline ($\alpha = 0$), aggressive promotion ($\alpha = 2$) leads to a performance degradation. This decline suggests that excessive amplification of the corrective signal may introduce over-correction artifacts, potentially distorting the original semantic space of the hidden states.Notably, as α continues to increase to an extreme scale, the intervention will degenerate into a form of direct decoding guided solely by the $\tilde{t}^*$.

H More Case Studies about Direct Decoding and Hidden State Rectification

The qualitative examples in Fig. 9 illustrate the efficacy and robustness of our proposed rectification mechanism.Unlike naive constrained decoding methods that force the model to output a specific token, our approach operates within the pure "disinhibition" paradigm discussed in Sec. 4.2.we do not strictly override the model's generative process. Instead, we steer the underlying representations toward the correct semantic manifold. This distinction is crucial for maintaining model robustness: even if the external knowledge or the target token selection contains slight noise, the latent space retains its internal linguistic coherence.

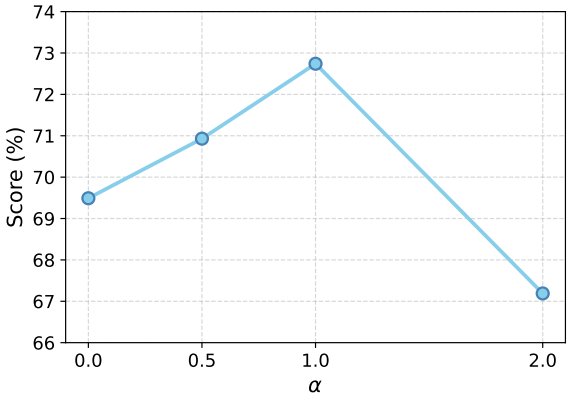

Figure 8: Sensitivity analysis of the scaling factor α on the Natural Questions (NQ) dataset. The peak at $\alpha = 1$ indicates that exact neutralization of conflicting signals achieves optimal performance.

Context: ⟨Table⟩ ⟨Tr⟩ ⟨Th⟩ Contestant ⟨/Th⟩ ... ⟨Td⟩ Amber Brauner ⟨/Td⟩ ⟨Td⟩ 37 ⟨/Td⟩ ... ⟨Td⟩ Winner March 31 ⟨/Td⟩ *(truncated)*
Q: winner of worst cooks in america season 5?
Gold: Amber Brauner
Direct Decoding: Lance Green
Ours (Rect.): Amber Brauner

Context: ⟨P⟩ The Super Bowl LI Halftime show took place on February 5, 2017... headlined by Lady Gaga, who performed a medley of her songs... *(truncated)*
Q: who performed the halftime show at super bowl 51?
Gold: Lady Gaga
Direct Decoding: liba Gaga performed the...
Ours (Rect.): Lady Gaga performed the...

Figure 9: Qualitative examples illustrating the efficacy of our proposed rectification mechanism in both table-based (top) and text-based (bottom) QA tasks.