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

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) can fail under knowledge conflicts, where a model's parametric prior competes with retrieved evidence and leads to unfaithful outputs. Existing mitigation methods typically operate on the final decoding distribution or rely on weight editing with access to ground-truth targets. Through layer-wise analysis, we identify a **parametric suppression** pattern: in many conflict cases, evidence-supported answers become salient in intermediate representations but are weakened by later layer outputs. To address this, we propose CoRect (**Co**ntext-Aware Logit Contrast for Hidden State **Rect**ification), a training-free inference-time method. By contrasting contextualized and context-free forward passes, CoRect first selects an evidence-supported token direction without using gold labels. It then rectifies hidden states in layers whose outputs suppress this direction. Across Question Answering (QA) and summarization benchmarks, CoRect improves average faithfulness and reduces hallucinations compared with strong decoding baselines.

1 Introduction

Large Language Models (LLMs) have demonstrated strong capabilities on knowledge-intensive tasks (Brown et al., 2020; Touvron et al., 2023). However, because much of their factual knowledge is encoded in static pre-training corpora, LLMs may hallucinate or produce outdated statements when queried about evolving facts.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 (Lewis et al., 2020; Guu et al., 2020). Despite this promise, RAG remains vulnerable to knowledge conflicts, where the retrieved evidence contradicts the model's parametric prior (Longpre et al., 2021; Chen et al., 2022). In such settings, LLMs may under-utilize the provided context and instead favor memorized but incorrect knowledge. As illustrated in Figure 1, when queried about the start date of the right to buy scheme, an LLM can prefer the parametric answer 1979 even when the context specifies 1980.

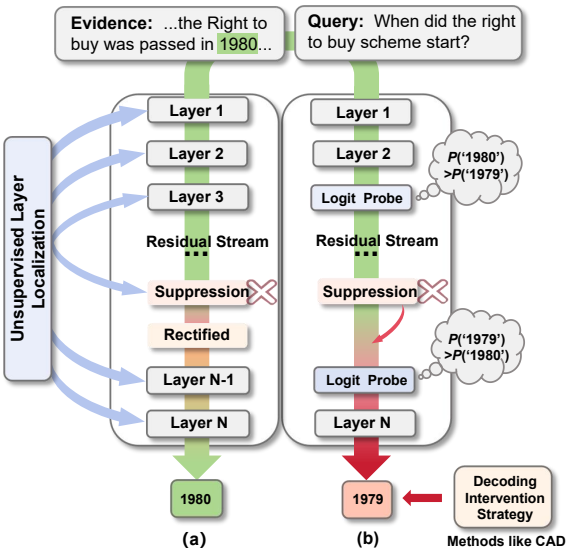

Figure 1: Comparison of intervention strategies. (a) Our method localizes and rectifies parametric suppression layers within the model's internal residual stream. (b) Baseline methods treat the model as a black box by intervening only at the output stage.

Recent work has proposed decoding-time corrections to address this failure mode. Prominent approaches adjust the final token distribution by subtracting logits attributed to parametric priors (Shi et al., 2024; Wang et al., 2025), or dynamically balance context and memory using entropy- and confidence-aware criteria (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. As a result, they do not directly address how intermediate representations evolve under conflict. Moreover, strong output-level manipulation can disturb useful linguistic priors, sometimes leading to semantic instability or

quality degradation.

In this work, we shift from output-level correction to a more internal view of how knowledge conflicts unfold inside the model. Using a layer-wise analysis of the residual stream via Logit Lens (nostalgebraist, 2020), we observe a recurring **parametric suppression** pattern: under conflict, the retrieved answer often becomes salient in intermediate layers, but later layers reduce its rank and the final prediction reverts to the parametric answer. Motivated by mechanistic interpretability and model-editing studies, we focus on Feed-Forward Network (FFN) layers, which have been associated with factual key-value memories (Geva et al., 2021; Dai et al., 2022). Our analysis suggests that some FFN outputs contribute negatively to evidence-consistent representations and can bias generation toward the parametric prior.

One possible response is to directly modify FFN parameters. ROME-style editing methods update weights in selected FFN layers to rewrite factual associations (Meng et al., 2022a,b). However, such approaches assume access to the desired target answer, which is unrealistic for open-ended RAG. At inference time, the system typically does not know the gold answer in advance. This limitation motivates a different goal: resolving conflicts without target labels during inference.

We therefore propose a training-free framework that mitigates knowledge conflicts through hidden-state intervention. CoRect first uses context-aware logit contrast and an attention-based evidence filter to select a reliable token direction without gold labels. It then identifies FFN outputs that negatively project onto this direction and applies a minimal rectification to attenuate their suppressive contribution. Our contributions are summarized as follows:

- **Mechanistic analysis of parametric suppression.** We identify a recurring pattern in knowledge-conflict settings: evidence-supported answers become salient in intermediate layers but are suppressed before final decoding. This motivates intervening in the residual stream rather than only modifying the output distribution.
- **Target-agnostic inference-time rectification.** We introduce CoRect, a hidden-state intervention that selects answer-relevant token directions and suppressive FFN layers without access to target answers. The localized layers align with ROME-style causal tracing at over **60%** F1, and FFN-only interventions confirm their causal effect on final logits.
- **Empirical mitigation across diverse tasks.** Across question answering and summarization benchmarks,spanning factual, multi-hop, and mathematical reasoning, as well as short and long-form generation,CoRect consistently improves performance and factual alignment over strong decoding-time baselines.

2 Related Work

We review work on RAG knowledge conflicts from three perspectives: decoding-time intervention, model editing, and activation intervention.

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 issue, strategies such as CAD (Shi et al., 2024), AdaCAD (Wang et al., 2025), COIECD (Yuan et al., 2024), and CoCoA (Khandelwal et al., 2025) adjust the final token distribution, typically by contrasting context-augmented logits against a context-free baseline or by adaptively weighting conflict signals. These methods are attractive because they require no weight updates, but they intervene after the internal computation has already produced a final representation. Layer-contrast decoding methods such as DoLa (Chuang et al., 2024) also exploit intermediate logits to improve factuality, but they are not explicitly designed to arbitrate between retrieved evidence and parametric memory under RAG conflicts.

Mechanistic Localization and Editing. Mechanistic studies suggest that FFNs can behave like key-value memories for factual associations (Geva et al., 2021; Dai et al., 2022). Methods such as ROME (Meng et al., 2022a) and MEMIT (Meng et al., 2022b) leverage this insight to edit FFN weights. However, these approaches require the target answer to compute updates, making them difficult to apply in open-ended RAG where the correct answer is unknown. In addition, permanent parameter modifications are less suited to transient, query-specific conflicts.

Inference-Time Activation Intervention. Beyond decoding, recent methods resolve conflicts via external systems or model tuning. For instance, TruthfulRAG (Liu et al., 2026) uses pipeline-level consolidation and knowledge graphs. At the parameter level, CLEAR (Gao et al., 2025) and TCR (Ye

et al., 2026) employ attention fine-tuning and soft prompts, while activation engineering methods such as ITI (Li et al., 2023) and RepE (Zou et al., 2023) inject static vectors derived from trained probes. Although effective, these methods require external modules, additional training, or labeled supervision. In contrast, our framework is **training-free and instance-specific**. We first select evidence-supported token directions at inference time and then rectify hidden states 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 (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 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 the 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 $\Delta W^{(l)}$:

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

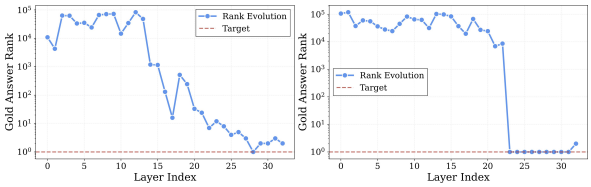

Figure 2: **Rank evolution analysis.** (Left) Middle layer flip pattern. (Right) Last layer flip pattern.

ROME constructs $\Delta W^{(l)}$ to align the induced δ_l with the target direction w_{t^*}; Appendix B provides the derivation, and Appendix B.3 connects this rank-one update back to residual steering. This highlights an important limitation for RAG: existing weight-editing approaches rely on an *oracle target* t^* to determine the update direction. This dependency makes them difficult to use 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 even when relevant evidence is present, we investigate internal representation dynamics using the Logit Lens (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 inspect the model's token preference at different depths. We observe that the gold answer token often receives a high probability in intermediate layers but is suppressed in the final output. We term this pattern **Parametric Suppression**.

Qualitative Analysis. As illustrated in Figure 2, we analyze the rank evolution of the gold answer for a representative query. In intermediate layers, the model's internal states favor the correct answer. As information propagates to later layers, however, the answer rank drops substantially and the model ultimately outputs an incorrect token. This discrepancy suggests that the retrieved evidence can be represented internally but later weakened during the forward pass.

To estimate the prevalence of this pattern, we conduct a systematic evaluation on 500 sampled cases from the Natural Questions (NQ) training set that involve knowledge conflicts. We track 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 cases into two patterns: Middle Flip (154 cases): the gold answer surfaces in intermediate layers but is lost in later stages. Last Layer Flip (128 cases): the gold answer remains dominant until the final projection layer. These patterns suggest that the model can represent the evidence-supported answer in many cases, even when the final prediction is incorrect. Although some rank fluctuations may reflect legitimate reasoning dynamics, the fact that over 56% of sampled error cases contain the correct answer internally motivates an intervention that preserves evidence-supported information rather than injecting new knowledge.

4.2 Transition: From Supervised Overwriting to Autonomous Rectification

The foregoing analysis highlights a mismatch between conventional editing and open-ended RAG. Conventional editors operate as *supervised overwriting*: an externally specified t^* is used to construct a parameter patch that pushes the model toward a designated output. In contrast, our approach differs along two axes, namely *how the target direction is obtained* and *how the intervention is applied*. We introduce a two-stage procedure: (1) **Token Selection** We infer a reliable token direction $\tilde{t}^*$ during the generation of the current token. Because generation is autoregressive, this step-token intervention can extend to multi-token answers and abstractive generation (see Appendix D for detailed analysis). (2) **Hidden State Rectification** Instead of forcing generation toward $\tilde{t}^*$, we apply a minimal intervention to the FFN output u_l that attenuates components inhibiting $\tilde{t}^*$. By removing suppressive components, we aim to preserve the model's native generative dynamics while mitigating hallucinations. The overall architecture of our method is shown in Figure 3.

4.3 Stage 1: Trustworthy Token Selection

Simple logit subtraction can amplify noise alongside evidence-supported tokens. To select a robust target token $\tilde{t}^*$ at the current step, we combine Contextual Mutual Information with attention-based evidence.

Contextual Mutual Information Estimation We perform two forward passes: one with the full input prompt and one with a null context containing only the question tokens, so that the null pass cap-

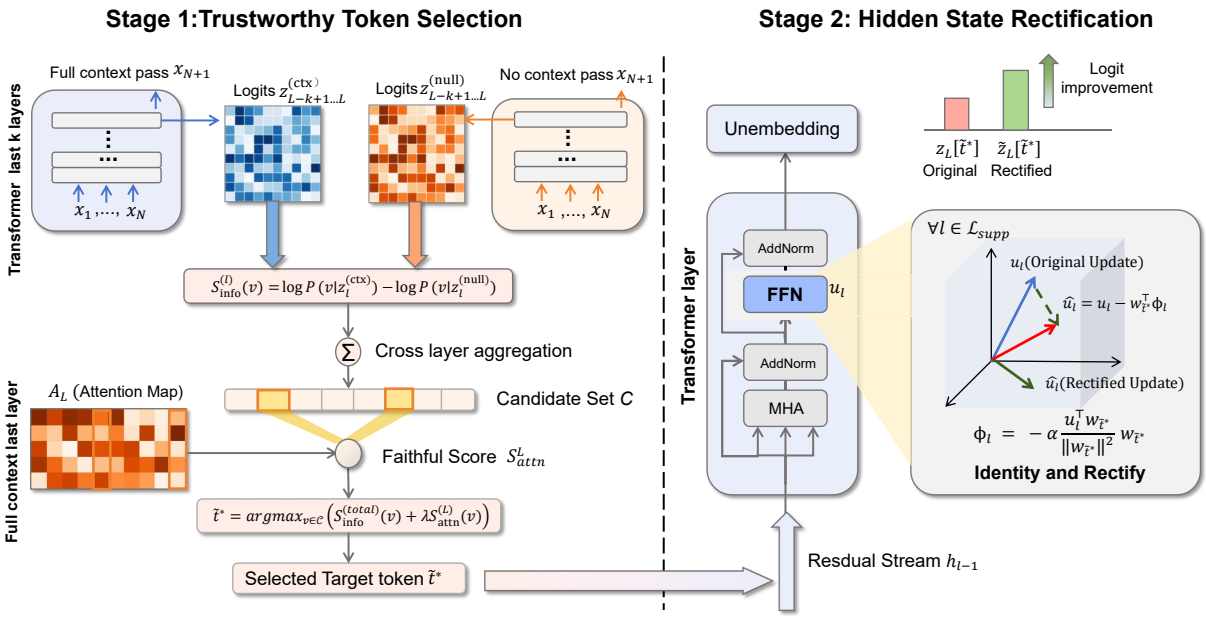

Figure 3: The overall architecture of our proposed model during the generation of the current token.

tures prior preferences independent of the retrieved evidence. 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 $S_{\text{info}}^{(l)}(v)$ which captures the boost in likelihood for token v provided by the context:

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

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:

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

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

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

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 $S_{\text{info}}^{(\text{total})}$. 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 token matching 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 token v appears. We then quantify the evidence for v by aggregating the attention the model allocates to these positions:

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

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

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

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 FFN

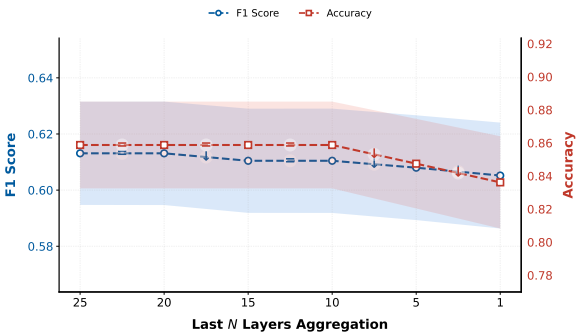

Figure 4: Localization analysis. The F1 score (blue) measures the alignment between our identified layer l and the ROME-based causal tracing layer $l^\star$, while the red curve reflects the corresponding downstream QA accuracy.

output has a negative direct contribution to the target direction:

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

To mitigate this negative contribution, we construct an additive patch ϕ_l that cancels 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 intervention controlled by α. Specifically, setting $\alpha = 1$ neutralizes the suppressive projection without adding an extra positive component. Conversely, $\alpha > 1$ actively *promotes* $\tilde{t}^*$ by turning the contribution positive. Under the linear residual-channel approximation used above, the net change in the final logit is positive:

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

We conduct causal interventions to compare our identified layer l with the ROME causal tracing layer $l^\star$ (see Appendix C). This analysis evaluates whether our localization criterion selects layers that are related to factual interference. Appendix C.4 further reports aggregate causal evidence that the rectified FFN outputs directly increase evidence-supported target-token scores. We also isolate the respective contributions of the Attention and FFN modules (see Appendix G for details). As illustrated in Figure 4, the trend in QA accuracy closely follows the F1 score. This correlation suggests that better alignment with causally traced layers is associated with stronger downstream performance.

5 Experiments

We evaluate our proposed method on both Question Answering (QA) and Summarization tasks, covering different levels of generation complexity. For QA, we assess performance on fact-based questions as well as tasks requiring multi-hop and numerical reasoning. For Summarization, we evaluate both short- and long-form generation. This evaluation tests whether hidden-state rectification improves contextual faithfulness beyond extractive QA. Detailed formalizations of the decoding-time correction baselines and the hyperparameter configurations for all methods are provided in Appendix I and Appendix J, respectively.

5.1 Experimental Setup

Datasets and Metrics. To ensure a broad evaluation, we test CoRect across several task families. For **QA and Reasoning**, we use 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 the high-conflict benchmark NQ-Swap (Longpre et al., 2021). For **Summarization**, we include XSum (Narayan et al., 2018), CNN-DM (Hermann et al., 2015), and TofuEval (Tang et al., 2024). We report Exact Match (EM) for QA, along with ROUGE-L (Lin, 2004), BERTScore-P (Zhang et al., 2019), and AlignScore (Zha et al., 2023) for summarization quality and faithfulness. Detailed dataset statistics and metric formalizations are provided in Appendix K.

5.2 Main Results

Question Answering. Table 1 summarizes performance across QA benchmarks. On the high-conflict NQ-Swap benchmark, CoRect yields strong gains, indicating that hidden-state rectification is effective when context directly contradicts parametric priors. CoRect also maintains competitive performance on natural datasets that mix conflict and non-conflict queries; Error-introduction rate after intervention remains low at 0.79%, substantially lower than CAD's 4.06%, suggesting that CoRect preserves the model's originally correct predictions while mitigating conflict-induced errors (Appendix F). We additionally compare against DoLa, a layer-contrast factuality decoding baseline, in Appendix E. Across backbones, CoRect achieves the best average EM in our evaluation.

Model	Method	Exact Match (EM)						Avg
		NQ	NQ-Swap	HotpotQA	TriviaQA	TabMWP	SQuAD	
Qwen2.5-32B-Instruct	Greedy	67.08	60.56	47.34	83.87	83.01	90.46	72.05
	CAD	67.58	73.01	45.29	79.22	78.79	89.96	72.31
	COIECD	69.88	68.94	48.96	84.53	82.69	91.97	74.50
	AdaCAD	69.40	70.72	48.24	84.62	83.55	**92.97**	74.92
	CoCoA	**72.32**	72.42	49.05	79.25	83.00	90.53	74.43
	CoRect	71.72	**74.49**	**50.07**	**85.09**	**85.71**	92.47	**76.59**
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
	CoCoA	69.63	78.12	39.25	81.95	52.30	84.83	67.68
	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
	CoCoA	46.37	56.94	27.15	71.85	38.00	72.47	52.13
	CoRect	**56.31**	**70.37**	**36.06**	**78.60**	**38.50**	**74.33**	**59.03**
LLaMA2-70B-chat	Greedy	64.50	58.89	34.78	81.00	49.82	78.53	61.25
	CAD	59.86	79.44	36.48	74.40	42.24	79.03	61.91
	COIECD	68.63	64.94	35.85	72.55	50.90	74.00	61.15
	AdaCAD	68.55	69.61	38.61	79.55	50.78	77.03	64.02
	CoCoA	68.79	73.56	34.55	76.50	46.60	80.53	63.42
	CoRect	**68.97**	**80.27**	**42.79**	**81.60**	**56.56**	**81.37**	**68.59**

Table 1: Comparison of different methods across multiple models on NQ, NQ-Swap, HotpotQA, TriviaQA, TabMWP and SQuAD. For the Greedy baseline, we use a constrained prompt template that instructs the models to rely on the provided context.

Summarization and Alignment. We evaluate CoRect on summarization benchmarks in Table 2. CoRect improves factual alignment on CNN/DailyMail, XSum, and TofuEval while preserving summary quality, suggesting that the method extends beyond extractive QA.

5.3 Ablation Studies

Q1: How Effective is the Token Selection Strategy? We compare our token selection strategy against baseline alternatives. As shown in Figure 5(a), when the number of intervened layers is $k = 1$, our token selection strategy already exceeds the baseline method. The same trend on both **NQ** and **XSum** indicates that the context-aware information score and attention filter provide a useful target direction.

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 apply the Attention Filter on both NQ and XSum. As shown in Figure 5(c)(d), the attention weight is important for performance. On **NQ**, performance gradually improves as the attention weight increases. On **XSum**, performance peaks at $\lambda = 0.5$. This difference is consistent with the nature of summarization, where the model must preserve broad semantic coverage rather than copy a single span. A moderate attention weight therefore balances context grounding and summary fluency.

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
CoCoA	20.86	85.55	85.03	17.89	86.71	82.28	66.43	68.86 / 64.00
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 improves factual alignment (AlignScore) while maintaining summary quality.

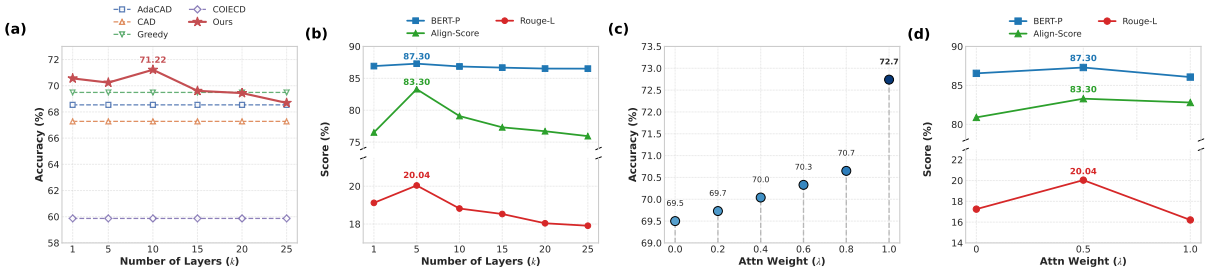

Figure 5: **Hyperparameter sensitivity and performance analysis. (a)** Accuracy with varying number of layers K on NQ. **(b)** Effect of layer depth on generation scores on XSUM. **(c)** Impact of attention weight (λ) on model accuracy on NQ. **(d)** Impact of attention weight (λ) on generation scores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

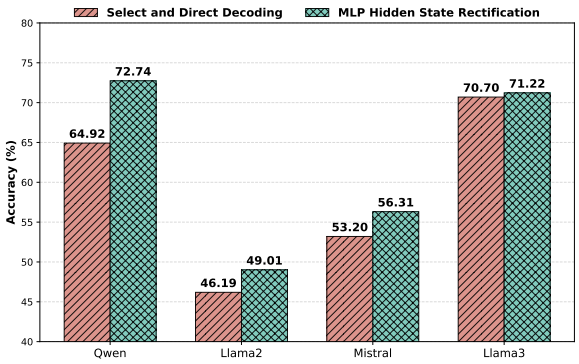

Figure 6: Accuracy comparison of different decoding strategies. Hidden State Rectification outperforms Direct Decoding across the evaluated backbones.

Q4: Is Hidden State Rectification Effective? To evaluate the hidden state rectification component, we compare it with directly decoding the target token selected in Stage 1. As shown in Figure 6, hidden-state rectification outperforms the direct decoding baseline. The selected token $\tilde{t}^*$ provides a useful direction, but it is not always the optimal final token. Direct decoding can therefore force semantically awkward outputs. In contrast, our rectification mechanism treats the selected token as a guiding direction for neutralizing suppressive components, allowing the model to preserve its generative behavior while surfacing context-supported information. We further analyze the rectification strength α in Appendix M. Figure 7 shows a representative case. Additional success and failure cases are discussed in Appendix N and Appendix L.

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 (CoRect): Cliff Richard sang the song...

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

6 Conclusion

In this work, we identify Parametric Suppression, a recurring pattern in which evidence-supported answers become salient in intermediate layers but are weakened before final decoding under knowledge conflicts. To address this issue, we propose CoRect, a target-agnostic hidden-state rectification method that selects evidence-supported token directions, localizes suppressive FFN outputs, and neutralizes their negative contribution. Evaluations across QA and summarization show that internal rectification can improve RAG faithfulness beyond output-level calibration. CoRect provides a practical, interpretable, and training-free framework for mitigating context-memory conflicts at inference.

7 Limitations

While CoRect mitigates knowledge conflicts, it introduces additional computational overhead compared with standard decoding. The framework uses multiple forward passes and stage-wise hidden-state rectification, requiring access to intermediate FFN states. In practice, cache-aware decoding, cached steering directions, and lightweight forward hooks reduce redundant computation and avoid retaining full intermediate states throughout generation. Appendix H reports the measured overhead. CoRect is therefore most suitable when factual faithfulness is prioritized over minimal decoding latency.

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.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. DoLa: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

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.

Linfeng Gao, Baolong Bi, Zheng Yuan, Le Wang, Zerui Chen, Zhimin Wei, Shenghua Liu, Qinggang Zhang, and Jinsong Su. 2025. Probing latent knowledge conflict for faithful retrieval-augmented generation. *arXiv preprint arXiv:2510.12460*.

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.

Anant Khandelwal, Manish Gupta, and Puneet Agrawal. 2025. CoCoA: Confidence- and context-aware adaptive decoding for resolving knowledge conflicts in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6835–6855, Suzhou, China. Association for Computational Linguistics.

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.

Shuyi Liu, Yu-Ming Shang, and Xi Zhang. 2026. Truthfulrag: Resolving factual-level conflicts in retrieval-augmented generation with knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 32168–32176.

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, Mexico City, Mexico. Association for Computational Linguistics.

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, Mexico City, Mexico. Association for Computational Linguistics.

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, Albuquerque, New Mexico. Association for Computational Linguistics.

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.

Hua Ye, Siyuan Chen, Ziqi Zhong, Canran Xiao, Haoliang Zhang, Yuhan Wu, and Fei Shen. 2026. Seeing through the conflict: Transparent knowledge conflict handling in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34423–34431.

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, Bangkok, Thailand. Association for Computational Linguistics.

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 derivation supports the claim in Section 3.2: ROME's weight update can be viewed as injecting 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 model layers. While standard causal analysis focuses on facilitatory sites, our analysis also considers layers whose restoration reduces the target probability.

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}:

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

C.2 Defining the Interference Set as l^*

We define the set of **Interference Layers** as l^*. This set contains layers where restoring clean activations further suppresses 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_clean(t*), {h_l}_{l=1}^L ← M(x) {Baseline Run}
x_corr ← Embed(x) + ε, ε ~ N(0, σ²I) {Corrupted Run}
P_corr(t*) ← M(x_corr)
Initialize empty set l* = ∅
for each layer l ∈ {1, ..., L} do
    P_restored(t*) ← M(x_corr | do(h_l^(s) ← h_l))
    AIE_l ← P_restored(t*) − P_corr(t*)
    if AIE_l < 0 then
        l* ← l* ∪ {l}
    end if
end for
l_ROME ← arg max_l(AIE_l)
return l*, l_ROME
```

C.4 Aggregate Causal Evidence for FFN Rectification

To complement the layer-localization procedure above, we conduct an aggregate causal intervention analysis on 1,000 NQ examples using LLaMA3-8B-Instruct, yielding 27,000 layer-level effect

rows. Table 3 summarizes the effects of method-processed FFN outputs on evidence-supported target-token scores.

Causal Metric	Avg. Effect	Positive Rate
Necessity drop (log-prob.)	1.09	99.4%
Sufficiency gain (log-prob.)	2.21	85.0%
Average causal effect (log-prob.)	1.65	87.1%
Average causal effect (margin)	3.37	90.4%
Restore raw FFN hurts	–	99.5%
Top-1 rescue rate	–	23.2%

Table 3: Aggregate causal intervention results for method-processed FFN outputs on 1,000 NQ examples using LLaMA3-8B-Instruct.

The necessity result shows that removing the processed FFN source reliably decreases the target-token score, while the sufficiency result shows that injecting it usually increases the score. The positive average causal effects in both log-probability and margin indicate that the rectified FFN outputs contribute directly to the evidence-supported direction. In addition, restoring the raw FFN output after our method hurts the target score in 99.5% of cases, suggesting that CoRect's processed FFN states are causally different from the original suppressive states. The 23.2% top-1 rescue rate further shows that the processed FFN source can independently recover the target token in a substantial subset of examples.

D Detailed Analysis on Abstractive Generation and Multi-step Reasoning

This section provides a detailed analysis of CoRect's mechanics and empirical effectiveness in scenarios involving abstractive generation, multi-token entities, and complex multi-step reasoning, particularly when verbatim evidence is absent.

First, in abstractive and non-verbatim generation settings, the attention filter in our target selection mainly serves as a safeguard for context faithfulness. The candidate tokens identified in Step 1 through logit contrast are semantically plausible for the current generation step. However, when target answers do not appear verbatim in the context, these semantically related candidates may also be absent from the context. For example, in yes/no questions from HotpotQA, the candidate tokens often do not appear in the evidence. In such cases, the attention filter acts as a conservative grounding constraint, while the contrastive logit signal contributes more to candidate selection. This behavior helps CoRect remain applicable when evidence tokens are not explicitly copied.

Furthermore, for multi-token entities and compositional answers, CoRect operates at the autoregressive step-token level. Because multi-token answers are generated sequentially, intervention at each decoding step can adjust the generation trajectory and reduce the risk of early error propagation.

Case 1: Arithmetic Hallucination

[Context]: black jacket | $27.84; pair of jeans | $30.30; green sweatshirt | $11.26 . . .

[Query]: *How much money does Jon need to buy a green sweatshirt and a pair of jeans?*

Baseline (Incorrect): The green sweatshirt is $11.26 and the jeans are $30.80. Therefore, $11.26 + $30.80 = $42.06.

CoRect (Correct): $30.30 + $11.26 = $41.56

Analysis: The baseline correctly identified the items but **hallucinated the price of the jeans** ($30.80 instead of $30.30). **CoRect anchors the value and executes the arithmetic.**

Case 2: Multi-step Addition

[Context]: Table Title: Trips to the grocery store . . . Frequency for 2 trips: 10; 3 trips: 18; 4 trips: 12; 5 trips: 16

[Query]: *How many people went to the grocery store at least 2 times?*

Baseline (Incorrect): Based on the table, 14 people went 2 times and 12 people went 3 times. 14 + 12 = 26. The correct answer is 26.

CoRect (Correct): raw count 10 + 18 + 12 + 16 = 56

Analysis: CoRect anchors the intermediate tokens to the correct frequencies for ≥ 2 trips, enabling the correct multi-step addition.

Table 4: Case studies illustrating CoRect's behavior on multi-step reasoning tasks. By rectifying intermediate "pivot tokens," CoRect reduces cascading reasoning failures.

To evaluate these capabilities beyond simple extractive QA, we highlight the method's performance on multi-step reasoning datasets, specifically **HotpotQA** (multi-hop deduction) and **TabMWP** (tabular reasoning and arithmetic). In multi-step reasoning, logical chains often depend on critical "pivot tokens." Hallucinating a single intermediate variable can cause downstream failures. By rectifying hidden states at these junctures sequentially, CoRect anchors intermediate variables to the context, supporting more accurate subsequent steps. Table 4 provides case studies illustrating this process.

E Comparison with Layer-Contrast Decoding

To further evaluate whether generic layer-contrast factuality decoding is sufficient for resolving RAG conflicts, we compare CoRect with DoLa (Chuang et al., 2024) on NQ, NQ-Swap, and TabMWP. We focus on LLaMA3-8B-Instruct and Mistral-7B, where layer-wise contrastive decoding is computationally feasible. We treat DoLa as a diagnostic baseline rather than a full main-table baseline because it is designed as a general factuality decoding method and does not explicitly use retrieved evidence or context-memory conflict signals, whereas our main baselines directly target RAG conflict resolution. We therefore report a representative comparison on two open backbones and three datasets that cover standard QA (NQ), high-conflict QA (NQ-Swap), and tabular numerical reasoning (TabMWP).

Model	Method	NQ	NQ-Swap	TabMWP
Mistral-7B	Greedy	46.34	44.51	25.20
	DoLa	54.50	38.18	26.30
	CoRect	**56.31**	**70.37**	**38.50**
LLaMA3-8B-Instruct	Greedy	68.72	60.67	47.40
	DoLa	70.93	61.87	50.80
	CoRect	**71.22**	**79.32**	**52.60**

Table 5: Comparison with DoLa on selected QA and reasoning benchmarks. Results are Exact Match (EM).

As shown in Table 5, DoLa improves over greedy decoding on standard NQ and TabMWP in most settings, confirming the usefulness of layer-wise contrast for factual generation. However, its gains are limited on the high-conflict NQ-Swap benchmark and it can underperform greedy decoding on Mistral-7B. This suggests that generic layer-contrast decoding does not consistently resolve context-memory conflicts. In contrast, CoRect explicitly selects evidence-supported token directions and rectifies suppressive FFN outputs, leading to stronger performance on both conflict-heavy and reasoning-oriented datasets.

F Detailed Analysis on Non-Conflict Queries

In the broader literature, knowledge conflicts are generally categorized into three types: Intra-Memory Conflict (internal inconsistencies), Inter-Context Conflict (contradictions between documents), and Context-Memory Conflict (external context versus parametric knowledge). CoRect is primarily positioned within Context-Memory Conflict, aiming to improve faithfulness when parametric priors contradict retrieved evidence.

However, a practical intervention should also avoid degrading performance when the retrieved context is non-conflicting. In our evaluation, apart from NQ-Swap (which is explicitly constructed to be 100% conflicting), the other datasets naturally contain a mixture of conflicting and non-conflicting samples. Therefore, the main results already reflect CoRect's behavior on mixed data.

To explicitly quantify the risk of performance degradation on non-conflicting data, we conduct a fine-grained breakdown analysis on the Natural Questions (NQ) dataset. We categorize prediction outcomes relative to a standard greedy decoding baseline into four groups, allowing us to compare correction capability with error introduction.

As shown in Table 6, CoRect's correction capacity (Wrong → Right) is comparable to CAD, while its Right → Wrong error introduction rate is 0.79%, lower than CAD's 4.06%. CoRect also preserves 67.93% of initially correct answers. These results suggest that CoRect mitigates knowledge conflicts with limited degradation on standard, non-conflicting queries.

Metric	Definition	CAD	CoRect
Correction Capacity	Wrong → Right	3.51%	3.29%
Error Introduction	Right → Wrong	4.06%	**0.79%**
Preservation	Right → Right	64.66%	**67.93%**
Unfixed Errors	Wrong → Wrong	27.77%	27.99%

Table 6: Breakdown of prediction changes on the NQ dataset relative to the greedy decoding baseline. CoRect reduces the error introduction rate (Right → Wrong) compared to CAD, indicating stronger robustness on non-conflicting queries.

G Component Ablation: Attention vs. FFN Layers

To examine the mechanism of parametric suppression, we conduct a component-level ablation study comparing interventions applied after the Multi-Head Attention (MHA) module and after the Feed-Forward Network (FFN).

Rectifying hidden states immediately after the MHA can shift the representation toward the target token and yields marginal Exact Match (EM) improvements. Mechanistic interpretability work suggests a useful functional distinction: MHA often routes contextual information, whereas FFNs

Error Category	Query	Gold Answer	Model Generation	Analysis of Misdirection
1. Constraint / Modifier Omission	what is the length of a cricket bat?	no more than 4.25 inches (108 mm)	4.25 inches	Stage 1 assigns high confidence to the core numerical entity (4.25 inches) but misses the crucial constraint modifier (no more than), resulting in an incomplete but highly related answer.
2. Entity Formatting Variations	how many books are in the diary of a whimpy kid series plus one do it yourself book plus?	twelve Wimpy Kid books plus one do it yourself book and two movie diaries.	twelve books plus one do it yourself book plus two movie diaries.	Stage 1 selects the correct entity, but minor morphological variations and increased verbosity cause a mismatch with exact-match metrics.
3. Attention Shift / Distraction	when added additional verses to where have?	1955	where have all the flowers gone was written in May 1960 by Joe Hickerson...	The model frequently disproportionately attends to the initial token (BOS), sub-word suffixes of fragmented entities, punctuation marks, or tokens that align with parametric priors.

Table 7: Error analysis on suboptimal target token ($\tilde{t}^*$) selection. The table categorizes the primary failure modes when Stage 1 provides an imperfect guiding vector.

can encode factual associations. Specifically, Geva et al. (2021) show that FFN value vectors encode semantically clustered concepts. In addition, causal tracing analysis by Meng et al. (2022a) finds that restoring mid-layer FFN outputs is especially important for factual predictions.

To empirically evaluate this distinction, we use LLaMA3-8B-Instruct on the Natural Questions (NQ) and HotpotQA datasets. We compare the baseline (Greedy Decoding) against applying our rectification method at two intervention points: (1) after the MHA module and (2) after the FFN module (CoRect).

Dataset	Baseline (Greedy)	Intervention After MHA	Intervention After FFN (Ours)
NQ	68.72	69.97	**71.22**
HotpotQA (Multi-hop)	39.47	40.17	**41.15**

Table 8: Ablation on intervention location using LLaMA3-8B-Instruct. The results show Exact Match (EM) scores. FFN-level intervention outperforms MHA-level intervention in this setting.

As shown in Table 8, post-MHA intervention provides a slight gain, while FFN-level intervention yields higher accuracy. This ablation supports our focus on FFN outputs as a major site for the parametric suppression pattern we observe.

H Detailed Inference Latency Analysis

To evaluate the computational overhead introduced by CoRect, we conduct a wall-clock inference latency analysis.

Method	Avg. Latency (s/sample)	Relative to Greedy
Greedy Decoding	0.7195	1.00×
CAD	1.4060	1.95×
COIECD	1.1260	1.56×
AdaCAD	1.2134	1.69×
CoCoA	1.2161	1.69×
DoLa	1.5402	2.14×
CoRect (Ours)	1.3939	1.94×

Table 9: Average inference latency comparison on the Natural Questions (NQ) dataset using LLaMA3-8B-Instruct.

Cache-aware Implementation We first run the context-aware and context-free passes to select the target token and compute the steering directions, then cache these directions for subsequent decoding steps. During generation, we use the model's KV cache so that after the initial full forward pass each step only processes the newly generated token. The rectification itself is implemented with lightweight forward hooks on the attention/MLP modules; the hook modifies only the current-token

output by removing its negative projection onto the selected direction. For the attention-based evidence filter, we avoid enabling full `output_attentions` over all layers: instead, we hook only the last attention layer used by the filter and move the captured attention tensor to CPU. We also compute Logit-Lens scores only for the selected current position in the last few layers. These optimizations do not remove CoRect's extra cost relative to CAD, but they prevent the implementation from storing all residual streams or recomputing full prefixes throughout decoding.

Discussion on CoRect Efficiency As reported in Table 9, CoRect's latency is comparable to standard contrastive decoding methods, indicating that the additional hidden-state operations remain practical for settings where factuality is important.

I 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:

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

Specifically, α_t is dynamically inferred based on the degree of knowledge conflict, which is measured by the Jensen-Shannon (JS) divergence between the contextual distribution P_{ctx} and the prior distribution P_{null}. By scaling the log-ratio of the two distributions, AdaCAD selectively follows the context only when the JS divergence indicates a potential knowledge conflict.

3. COIECD (Yuan et al., 2024) The Contextual Information-Entropy Constraint Decoding (COIECD) framework discerns knowledge conflicts by measuring changes in distribution entropy at the token level. It evaluates the uncertainty of the context-aware distribution using Shannon entropy:

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

Unlike static contrastive methods, COIECD utilizes this entropy constraint as a diagnostic threshold. It classifies tokens into conflicting and non-conflicting states based on whether the normalized entropy surpasses a tuned threshold. It then employs a dual-mode decoding strategy: applying tailored contrastive adjustments to resolve conflicts when detected, while reverting to standard generation in the absence of conflicts to prevent unnecessary performance degradation.

4. CoCoA (Khandelwal et al., 2025) CoCoA is a confidence- and context-aware adaptive decoding method for knowledge conflicts. It combines three token-level signals: the entropy gap between context-aware and context-free distributions, the peakedness of the context-aware distribution, and a generalized Jensen-Shannon divergence between the two distributions. These signals are used to decide when and how strongly to apply a context-oriented decoding adjustment.

5. DoLa (Chuang et al., 2024) DoLa is a layer-contrast decoding method that improves factuality by contrasting logits projected from mature and premature layers. Unlike context-aware decoding methods, DoLa does not compare context-augmented and context-free distributions; it instead relies on internal layer-wise disagreement to adjust the next-token distribution.

J Implementation Details

For the hyperparameter configurations of the decoding methods used in our experiments, we follow the settings established by prior work 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 per instance, varying according to the quantified degree of knowledge conflict detected between the model's parametric memory and the provided context.
- **CoCoA:** Following Khandelwal et al. (2025), CoCoA computes a token-level adaptive gate from the Rényi divergence between the context-free and context-aware distributions, the entropy gap, and the contextual peakedness signal. We use the default settings reported in the original paper for all datasets: Rényi order $\alpha = 0.5$, contextual peakedness weight $z = 5.0$, entropy-gap weight $\gamma = 1.0$, and numerical stability constant $\delta = 10^{-8}$. The final distribution is obtained by power interpolation between the context-free and context-aware distributions using the adaptive gate λ_t, and decoding proceeds greedily from the adjusted distribution.

For our proposed method, we maintain $\alpha = 1$ and M=10 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.

K Dataset Descriptions

This section provides an overview of the datasets used in our experiments, organized by their primary evaluation tasks. Following the data processing methodology in AdaCAD (Wang et al., 2025) and considering computational resource constraints, we conduct our evaluation on the original benchmarks, as detailed below.

K.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. We evaluate on 3,481 samples.

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. For our experiments, we utilize 4,690 samples.

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. We construct our evaluation using 2,000 instances.

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. We conduct our evaluation on 3,000 samples.

TabMWP (Lu et al., 2022) provides mathematical word problems based on tabular data, necessitating both linguistic understanding and numerical reasoning. We use 1,000 samples for this task.

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. Our evaluation uses 4,000 samples from this benchmark.

K.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. We utilize 600 samples.

XSum (Narayan et al., 2018) provides BBC articles paired with single-sentence summaries. The task is highly abstractive, demanding significant compression and rephrasing. We evaluate on 600 instances.

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. We utilize 300 samples for our experiments.

L Error Analysis of Suboptimal Target Token Selection

While our method demonstrates strong performance, the quality of the target token $\tilde{t}^*$ selected in Stage 1 affects the trajectory of Stage 2. A suboptimal $\tilde{t}^*$ can introduce biases. Our error analysis shows that many such misdirections are fine-grained semantic deviations or formatting issues rather than unrelated hallucinations.

To examine this issue, we conduct a detailed error analysis on failure cases. We identify three primary categories where a suboptimal $\tilde{t}^*$ misdirects Stage 2, as detailed in Table 7.

As the examples illustrate, even when Stage 1 provides an imperfect $\tilde{t}^*$, Stage 2 rectification often keeps the generation close to the provided context or a relevant semantic neighbor, maintaining a degree of controlled generation despite the initial suboptimal selection.

M α Rectification

To further analyze 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 *neutralizes* the suppression, aligning with the disinhibition paradigm discussed in Sec. 4.2; and (iii) for $\alpha > 1$, the intervention *promotes* 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, show a concave relationship between model performance and the scaling factor. Performance peaks at $\alpha = 1$, suggesting that neutralizing parametric suppression provides a balanced intervention. Partial suppression ($0 < \alpha < 1$) improves over the baseline ($\alpha = 0$), while stronger promotion ($\alpha = 2$) degrades performance. This decline suggests that excessive amplification of the corrective signal may introduce over-correction artifacts and distort the hidden-state geometry. At extreme values, the intervention approaches a form of direct decoding guided primarily by $\tilde{t}^*$.

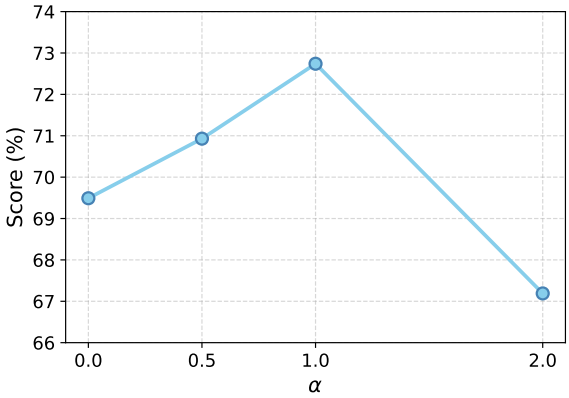

Figure 8: Sensitivity analysis of the scaling factor α on the Natural Questions (NQ) dataset. The peak at $\alpha = 1$ suggests that neutralizing conflicting signals provides the best balance in this setting.

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 (CoRect): 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 (CoRect): Lady Gaga performed the...

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

N More Case Studies about Direct Decoding and Hidden State Rectification

The qualitative examples in Fig. 9 illustrate the behavior of our rectification mechanism. Unlike constrained decoding methods that force the model to output a specific token, our approach follows the disinhibition paradigm discussed in Sec. 4.2: it attenuates suppressive components rather than strictly overriding the model's generative process. This distinction helps preserve linguistic coherence even when the selected target token is imperfect.