Context-Fidelity Boosting: Enhancing Faithful Generation through Watermark-Inspired Decoding

Weixu Zhang[1,2*], **Fanghua Ye**[1*], **Qiang Gao**[1,3], **Jian Li**[1†], **Haolun Wu**[2], **Yuxing Tian**[4], **Sijing Duan**[5], **Nan Du**[1], **Xiaolong Li**[1]
[1]Hunyuan AI Digital Human, Tencent [2]McGill University & MILA
[3]Wuhan University [4]University of Montreal [5]Tsinghua University
{fanghua.ye.21, lijianjack}@gmail.com

Abstract

Large language models (LLMs) often produce content that contradicts or overlooks information provided in the input context, a phenomenon known as *faithfulness hallucination*. In this paper, we propose **Context-Fidelity Boosting (CFB)**, a lightweight and general decoding-time framework that reduces such hallucinations by increasing the generation probability of source-supported tokens. Motivated by logit-shaping principles from watermarking techniques, CFB applies additive token-level logit adjustments based on whether a token is supported by the input context and how strongly it is supported. Specifically, we develop three boosting strategies: *static boosting*, which applies a fixed bias to source-supported tokens; *context-aware boosting*, which scales this bias using the divergence between context-aware and context-free next-token distributions; and *token-aware boosting*, which further redistributes the adaptive bias according to local relevance estimated from source-position attention and source-scoped semantic similarity. CFB requires no retraining or architectural changes, making it compatible with a wide range of LLMs. Experiments on summarization and question answering tasks across multiple open-source LLMs show that CFB consistently improves faithfulness metrics with minimal generation overhead. Our implementation is fully open-sourced.[1]

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language tasks. In many practical settings, however, models are expected to generate outputs that faithfully follow user-provided context, such as in retrieval-augmented generation (RAG), summarization (Laban et al., 2024), question answering (Chen et al., 2025), and role-playing (Huang et al., 2024). When external evidence conflicts with a model's internal parametric memory, the generated content may contradict, ignore, or distort the provided context (Mallen et al., 2023; Liu et al., 2024c), leading to **faithfulness hallucination**, outputs that are fluent and plausible but inconsistent with the input.

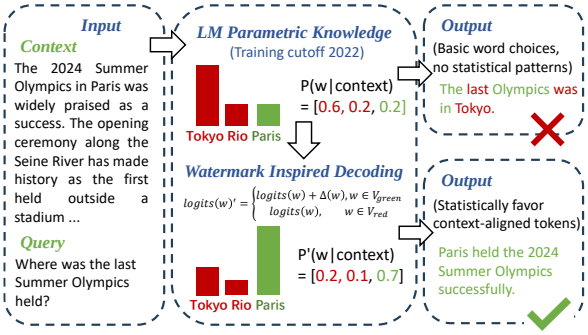


Figure 1: Illustration of context-faithful decoding: Traditional decoding relies on parametric knowledge (favoring "Tokyo"), while our logit-shaping approach dynamically adjusts token probabilities to better align with the given context about "Paris 2024".

Faithfulness hallucination is distinct from the more commonly studied *factuality hallucination*, which concerns incorrect or fabricated facts regardless of context. In high-stakes domains such as healthcare (Zhu et al., 2024), legal (Cui et al., 2024), and finance (Lee et al., 2025), it is essential for model outputs to remain faithful to the given input even when the model's internal knowledge offers plausible but contextually irrelevant or conflicting alternatives.

Existing methods for mitigating faithfulness hallucination generally fall into three categories: (1) training-time approaches that require fine-tuning or architectural changes (Hu et al., 2024), (2) prompting techniques that rely on hand-crafted inputs and may behave inconsistently across tasks or mod-

[1] Work done during an internship at Tencent Hunyuan.
[2] * Equal Contribution.
[3] † Corresponding Author.
[1] https://github.com/weixuzhang/CFB

Findings of the Association for Computational Linguistics: ACL 2026, pages 42748–42759

els (Zhang et al., 2024), and (3) decoding-time interventions that modify generation behavior at inference time (Shi et al., 2024; Wang et al., 2024). Among these, decoding-time methods are especially attractive because they are efficient, model-agnostic, and easy to deploy. However, many existing approaches still face a difficult trade-off between enforcing context fidelity and preserving fluency, or they depend on carefully tuned contrastive objectives and heuristic control rules.

In this work, we introduce **Context-Fidelity Boosting (CFB)**, a decoding-time framework that improves context alignment by selectively adjusting token probabilities during generation. CFB is inspired by logit-shaping mechanisms originally developed for text watermarking (Kirchenbauer et al., 2024; Liu et al., 2024a; Liu and Bu, 2024). While watermarking methods bias generation toward designated token sets in order to embed detectable signals, our goal is different: we use a similar logit-shaping principle to gently favor tokens supported by the input context, thereby reducing faithfulness hallucination without degrading fluency.

CFB operates through three levels of control. *Static boosting* applies a fixed bias to source-supported tokens. *Context-aware boosting* scales this bias using the divergence between context-aware and context-free next-token distributions, allowing the method to adapt to the degree of contextual influence. *Token-aware boosting* further redistributes the adaptive bias across source-supported tokens according to token-level relevance, combining source-position attention with source-scoped semantic similarity to provide finer-grained control during decoding. Importantly, CFB requires no retraining or architectural modification and introduces only lightweight overhead during inference. This makes it a practical framework for improving faithfulness in real-world deployments.

Our key contributions are as follows:

- We propose CFB, a lightweight and model-agnostic decoding framework that improves contextual faithfulness while preserving output quality, making it especially suitable for high-stakes context-grounded generation.
- We develop a three-level boosting mechanism with static, context-aware, and token-aware variants, enabling finer-grained and more flexible control over context-sensitive decoding through additive logit shaping.
- We demonstrate the effectiveness of CFB across multiple model scales and tasks, including summarization and question answering, showing consistent gains in faithfulness metrics with minimal decoding overhead.

2 Related Work

2.1 Faithfulness Hallucinations in LLMs

Despite their impressive capabilities, LLMs frequently generate hallucinated content (Hase et al., 2024; Chuang et al., 2024; Ming et al., 2024). Recent studies distinguish two major forms of hallucination. *Factuality hallucination* (Yang et al., 2024) arises when model outputs diverge from verifiable real-world facts, such as incorrect historical dates or fabricated attributions. In contrast, *faithfulness hallucination* (Wu et al., 2024; Qiu et al., 2024) occurs when generated content contradicts, ignores, or fabricates information relative to the provided input context, such as unsupported details in a summary. The latter is particularly problematic in context-grounded settings including summarization, retrieval-augmented generation, and question answering, where external evidence may conflict with a model's parametric knowledge acquired during pretraining. To evaluate faithfulness, prior work has proposed a range of metrics, including semantic similarity, entailment measures, and fact-checking frameworks (Niu et al., 2024; Hong et al., 2024).

2.2 Existing Mitigation Methods

Prior work has explored hallucination mitigation at different stages of the LLM pipeline (Huang et al., 2023). Training-time approaches introduce architectural or objective modifications, such as enhanced attention or knowledge-aware training, but often require substantial computation and exhibit limited cross-domain generalization (Tonmoy et al., 2024). Prompting-based methods, including chain-of-thought reasoning (Wei et al., 2023) and self-consistency, provide model-agnostic alternatives but show variable effectiveness across models and tasks (Hou et al., 2024). Decoding-time interventions directly modify generation behavior at inference time, for example through constrained decoding, contrastive decoding, or adaptive reweighting of token probabilities, yet frequently face a trade-off between enforcing faithfulness and preserving fluency (Gema et al., 2024).

Among these approaches, decoding-time methods are particularly attractive because they do not

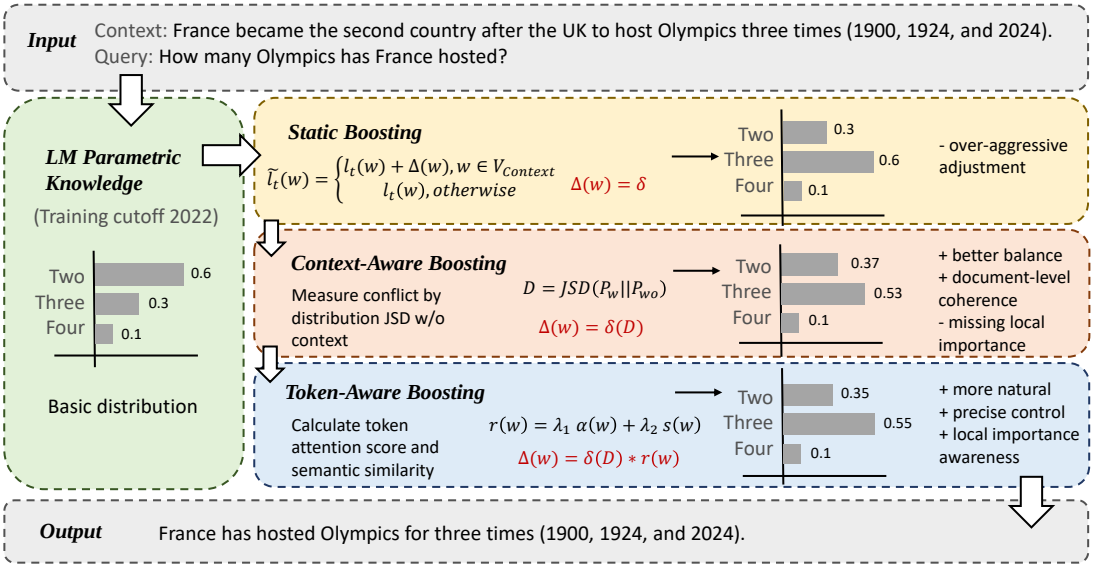


Figure 2: Overview of the proposed **Context-Fidelity Boosting (CFB)** framework. CFB applies additive logit shaping through three strategies of increasing adaptivity: (1) *Static Boosting*, which uniformly increases logits of source-supported tokens by a fixed value; (2) *Context-Aware Boosting*, which scales the boost using the divergence between context-aware and context-free next-token distributions; and (3) *Token-Aware Boosting*, which further redistributes the adaptive boost according to token-level relevance estimated from attention over source positions and source-scoped semantic similarity.

require retraining and can be applied to a broad range of open-weight LLMs. Our method belongs to this family, but differs from prior approaches in two important ways. First, rather than contrasting full distributions or imposing hard decoding constraints, CFB uses lightweight additive logit shaping to favor tokens supported by the external context. Second, CFB provides multiple levels of control, ranging from a fixed context bias to sample-level adaptive scaling and token-aware redistribution, enabling a simple and interpretable trade-off between contextual fidelity and generation quality.

Related cache-based and memory-augmented decoding methods improve coherence by boosting tokens from recent generation history, primarily targeting perplexity reduction or long-range dependency modeling. In contrast, our approach operates purely at decoding time and explicitly boosts tokens grounded in the external input context, aiming to mitigate faithfulness hallucination in context-grounded generation rather than optimize general language modeling objectives.

2.3 Watermarking in LLMs

Recent work on text watermarking has demonstrated that subtle logit shaping can effectively steer model outputs while preserving generation quality. These methods typically partition the vocabulary into "green" and "red" token sets and adjust token probabilities to embed detectable statistical patterns (Liu et al., 2024b). Subsequent advances have proposed soft or adaptive watermarking schemes that dynamically adjust token probabilities based on context (Kirchenbauer et al., 2024), semantic invariance constraints (Liu et al., 2024a), and theoretical analyses of the trade-off between watermark strength and naturalness (Golowich and Moitra, 2024). While watermarking aims to embed identifiable signals for downstream detection, our work repurposes similar logit-shaping mechanisms to improve context faithfulness. Instead of favoring a predefined statistical token subset, CFB uses additive logit shaping to bias generation toward tokens supported by the input context, with adaptive scaling and token-aware control. In this sense, CFB transfers the controllability insight of watermarking to the problem of reducing faithfulness hallucination in context-grounded generation.

3 Methodology

We introduce **Context-Fidelity Boosting (CFB)**, a decoding-time framework designed to reduce faithfulness hallucination by adaptively adjusting token probabilities based on their support from the input context. Motivated by recent logit-shaping techniques used in text watermarking, CFB selectively promotes source-supported tokens during generation to better align outputs with the provided evi-

Algorithm 1: Context-Fidelity Boosting (CFB)

Input: Context passage C, query Q, language model M
Parameters: δ, $\delta_{\min}$, $\delta_{\max}$, λ_1, λ_2
Output: Generated output with boosted context fidelity

$S \leftarrow \text{ResolveSourceSpan}(C, Q)$ // source-supported span
$T_S \leftarrow \text{UniqueTokens}(S)$, $\text{pos}(w) \leftarrow \text{Occurrences}(w, S)$, $s(w) \leftarrow \text{SourceScopedSemanticSimilarity}(w, S)$
$D \leftarrow \text{JSD}(M(C+Q), M(Q))$, $\delta_D \leftarrow \delta_{\min} + (\delta_{\max} - \delta_{\min}) \cdot D$
$output\_ids \leftarrow \text{Tokenize}(C+Q)$
while not terminated **do**
  $l_t \leftarrow M(output\_ids)[-1]$
  if mode is "static" **then** $\tilde{l}_t(w) \leftarrow l_t(w) + \delta,\ \forall w \in T_S$
  else if mode is "context-aware" **then** $\tilde{l}_t(w) \leftarrow l_t(w) + \delta_D,\ \forall w \in T_S$
  else if mode is "token-aware" **then**
    $a_t(p) \leftarrow$ GetAttentionScores over source positions in S
    $\alpha_t(w) \leftarrow \text{Agg}(\{a_t(p) : p \in \text{pos}(w)\})$, $r_t(w) \leftarrow \lambda_1 \alpha_t(w) + \lambda_2 s(w)$
    $\hat{r}_t(w) \leftarrow \text{Normalize}(r_t(w))$ over $w \in T_S$, $\tilde{l}_t(w) \leftarrow l_t(w) + \delta_D \hat{r}_t(w)$
  $P^* \leftarrow \text{Softmax}(\tilde{l}_t)$, $next\_token \leftarrow \text{Sample}(P^*)$
  $output\_ids \leftarrow [output\_ids; next\_token]$
return $\text{Decode}(output\_ids)$

Table 1: Pseudocode for Context-Fidelity Boosting (CFB). The token-aware variant first computes a sample-level adaptive boost and then redistributes it across source-supported tokens using attention-based and semantic relevance signals.

dence, as illustrated in Figure 2.

3.1 Problem Formulation

Given a context passage C and a query Q, our goal is to enhance contextual fidelity by increasing the probability of generating tokens supported by the source content in C. Let $P(y_t \mid y_{<t}, C, Q)$ denote the generation probability at time step t, and let $V_S \subseteq V$ denote the set of vocabulary tokens appearing in the source span extracted from the input context. We reshape the logits as follows:

$$\tilde{l}_t(w) = \begin{cases} l_t(w) + \Delta_t(w), & \text{if } w \in V_S, \\ l_t(w), & \text{otherwise.} \end{cases} \quad (1)$$

Here, $l_t(w)$ is the original logit for token w, and $\Delta_t(w)$ is a boosting factor determined by the support that token receives from the input context.

3.2 Context-Fidelity Boosting Framework

We propose a three-level logit-shaping strategy, progressing from fixed boosting to sample-level adaptive scaling and token-aware redistribution.

3.2.1 Static Boosting

The simplest strategy assigns a fixed boost δ to all tokens in the source-supported vocabulary:

$$\Delta_t(w) = \delta. \quad (2)$$

While effective in encouraging context preference, this strategy does not account for variation across inputs or for differences in the relevance and importance of individual source-supported tokens.

3.2.2 Context-Aware Boosting

To dynamically adjust the boost according to the influence of the context, we compute the Jensen-Shannon divergence between context-aware and context-free next-token distributions:

$$D = \text{JSD}(P_{\text{w}} \| P_{\text{wo}}), \quad (3)$$

where P_{w} and P_{wo} denote the predicted next-token distributions with and without context, respectively. $\text{JSD}(\cdot)$ is bounded in $[0, 1]$ under base-2 logarithms.

We use D to scale the boost adaptively:

$$\Delta_t(w) = \delta(D) = \delta_{\min} + (\delta_{\max} - \delta_{\min}) \cdot D, \quad (4)$$

where $\delta_{\min}$ and $\delta_{\max}$ define the boosting range. Intuitively, larger divergence indicates that the context substantially changes the model's next-token preference, and therefore warrants a stronger context-fidelity bias.

3.2.3 Token-Aware Boosting

The most fine-grained strategy retains the sample-level adaptive boost and further redistributes it across source-supported tokens according to token-level relevance. Specifically, at decoding step t, we first compute attention scores over positions in the extracted source span. For each token $w \in V_S$, we aggregate the attention mass over all of its occurrences in the source:

$$\alpha_t(w) = \text{Agg}\left(\{a_t(p) : p \in \mathcal{P}(w, C)\}\right), \quad (5)$$

where $\mathcal{P}(w, C)$ denotes the set of source positions containing token w, and $\mathrm{Agg}(\cdot)$ is an aggregation operator. In our implementation, we use summation to accumulate attention over repeated occurrences.

We further compute a source-scoped semantic relevance score by averaging the cosine similarity between the embedding of token w and the embeddings of tokens in the extracted source span:

$$s(w) = \frac{1}{|S|} \sum_{c \in S} \mathrm{cosine}(e_w, e_c), \tag{6}$$

where S denotes the extracted source span, and e_w and e_c are token embeddings. The token relevance score is then defined as

$$r_t(w) = \lambda_1 \alpha_t(w) + \lambda_2 s(w), \tag{7}$$

with $\lambda_1 + \lambda_2 = 1$.

To maintain a comparable overall boost scale across eligible tokens, we normalize the relevance scores:

$$\hat{r}_t(w) = \frac{r_t(w)}{\frac{1}{|V_S|} \sum_{u \in V_S} r_t(u)}. \tag{8}$$

The final token-aware boost is then

$$\Delta_t(w) = \delta(D) \cdot \hat{r}_t(w). \tag{9}$$

This design enables finer-grained control than context-aware boosting by allocating larger boosts to source-supported tokens that are estimated to be more relevant under the current decoding state. In all experiments, we set $\lambda_1 = 0.6$ and $\lambda_2 = 0.4$.

3.3 Implementation Details

Table 1 outlines the full CFB decoding algorithm. In practice, we first resolve the source span from the input prompt and restrict boosting to tokens supported by this span, which reduces interference from prompt scaffolding and instruction text. For token-aware boosting, source-scoped semantic relevance is precomputed once per example, while attention over source positions is recomputed at each decoding step to capture the current decoding state. Token-level relevance is then obtained by aggregating attention over token occurrences and normalizing the resulting scores before applying the final additive logit bias. This design keeps the computation lightweight while preserving fine-grained control during generation.

4 Experiments

4.1 Experiment Setup

Models We evaluate our method on several state-of-the-art LLMs, including Llama2-13B-chat-hf, Llama3-8B-Instruct, and Mistral-7B-Instruct.

Datasets We consider two types of tasks.

- **Summarization:** We use CNN-DM (See et al., 2017) and XSum (Narayan et al., 2018) to evaluate the model's ability to generate faithful and context-grounded summaries. For XSum, we randomly sample 500 examples for evaluation. For this task, we measure ROUGE-L (Lin, 2004) for summary quality, factKB (Feng et al., 2023) for knowledge consistency, and BERT-P (Zhang et al., 2020) for semantic preservation.

- **Question Answering:** We use NQ-SWAP (Longpre et al., 2021) and NQ-Synth (Wang et al., 2024) to evaluate the model's ability to leverage context information. NQ-SWAP contains synthetic knowledge conflicts, while NQ-Synth consists of examples where context aligns with the model's parametric knowledge. For this task, we also report accuracy.

Baselines We compare our method against several strong decoding-time baselines: Context-Aware Decoding (CAD) (Shi et al., 2024), which uses a fixed hyperparameter to control adjustment of output probabilities; Adaptive Context-Aware Decoding (ADACAD) (Wang et al., 2024), which dynamically infers adjustment based on Jensen-Shannon divergence; and Contextual Information-Entropy Constraint Decoding (COIECD) (Yuan et al., 2024), which employs distinct strategies for conflicting and non-conflicting tokens. For consistent comparison, we use top-p sampling across all methods under a zero-shot setting, with hyperparameters following their original papers.

4.2 Results

Overall Performance Overall, CFB remains competitive with strong decoding-time baselines across both summarization and QA tasks. A clear pattern is that CFB is particularly effective when the task benefits from reinforcing source-supported content, such as summarization and the complementary-knowledge setting in NQ-Synth. In contrast, on NQ-Swap, where context explicitly

Model	Method	CNN/DM			XSum		
		ROUGE-L	FactKB	BERT-P	ROUGE-L	FactKB	BERT-P
Mistral-7B	CAD (Shi et al., 2024)	33.19	96.39	**89.65**	**16.57**	39.22	89.93
	AdaCAD (Wang et al., 2024)	25.71	85.67	86.20	14.46	29.19	86.42
	COIECD (Yuan et al., 2024)	22.65	75.08	84.84	11.93	27.09	84.27
	Static CFB (ours)	34.22	95.37	89.48	14.66	54.25	89.55
	Context-aware CFB (ours)	34.05	95.93	89.51	14.56	**56.02**	**89.64**
	Token-aware CFB (ours)	**34.52**	**96.87**	89.44	14.75	48.69	88.85
Llama2-13B	CAD (Shi et al., 2024)	35.63	97.26	89.38	13.96	26.91	88.86
	AdaCAD (Wang et al., 2024)	24.10	89.97	85.60	10.74	38.83	83.68
	COIECD (Yuan et al., 2024)	19.37	80.19	83.57	9.49	9.51	84.16
	Static CFB (ours)	37.40	**98.85**	89.61	13.77	54.38	89.53
	Context-aware CFB (ours)	**37.52**	98.69	89.62	14.62	**55.02**	89.49
	Token-aware CFB (ours)	36.16	97.24	**89.83**	**15.25**	37.91	**89.57**
Llama3-8B	CAD (Shi et al., 2024)	35.92	94.57	89.07	12.92	45.77	87.05
	AdaCAD (Wang et al., 2024)	21.80	87.67	84.60	8.69	42.81	82.07
	COIECD (Yuan et al., 2024)	19.11	83.46	83.96	10.59	51.90	83.80
	Static CFB (ours)	**36.79**	95.15	89.63	12.46	66.33	**88.69**
	Context-aware CFB (ours)	36.78	**97.23**	**89.85**	12.59	**66.85**	88.67
	Token-aware CFB (ours)	35.81	94.31	89.38	**13.23**	55.29	88.45

Table 2: Main summarization results on CNN/DM and XSum. We report ROUGE-L, FactKB, and BERT-P. Best results for each model are shown in **bold**.

Model	Method	NQ-Synth				NQ-Swap			
		ROUGE-L	FactKB	BERT-P	Acc	ROUGE-L	FactKB	BERT-P	Acc
Mistral-7B	CAD (Shi et al., 2024)	26.64	57.92	87.07	48.20	35.11	49.46	74.56	50.06
	AdaCAD (Wang et al., 2024)	7.71	**67.46**	86.78	48.30	**36.41**	**67.83**	87.53	**73.79**
	COIECD (Yuan et al., 2024)	12.20	48.41	85.65	20.70	5.01	27.07	83.60	3.15
	Static CFB (ours)	24.72	54.74	87.73	56.60	16.29	51.91	88.36	36.08
	Context-aware CFB (ours)	24.67	54.74	87.73	56.50	16.26	51.94	88.37	36.01
	Token-aware CFB (ours)	**28.77**	50.21	**88.18**	**60.10**	17.36	48.46	**88.50**	35.73
Llama2-13B	CAD (Shi et al., 2024)	29.34	56.45	86.78	47.90	**23.22**	35.59	84.95	**44.91**
	AdaCAD (Wang et al., 2024)	12.68	**65.20**	86.52	39.70	20.11	34.60	82.25	74.21
	COIECD (Yuan et al., 2024)	14.67	43.13	85.02	20.60	2.52	24.24	74.13	1.50
	Static CFB (ours)	34.43	48.77	83.70	56.00	12.66	55.27	88.32	26.03
	Context-aware CFB (ours)	34.43	48.95	83.71	56.00	12.64	**55.35**	**88.33**	26.03
	Token-aware CFB (ours)	**39.05**	49.68	**87.64**	**64.00**	13.10	54.19	81.20	11.13
Llama3-8B	CAD (Shi et al., 2024)	28.19	32.26	86.50	66.80	**26.40**	33.66	87.05	58.49
	AdaCAD (Wang et al., 2024)	6.26	**51.36**	86.50	48.40	12.52	39.14	85.82	**86.50**
	COIECD (Yuan et al., 2024)	15.24	21.97	84.54	32.10	5.73	26.15	84.49	6.33
	Static CFB (ours)	29.87	44.53	**88.34**	73.10	14.14	**48.95**	88.48	34.88
	Context-aware CFB (ours)	29.87	44.50	88.33	73.10	14.16	48.93	**88.49**	34.91
	Token-aware CFB (ours)	**32.90**	45.94	88.13	**73.40**	14.54	40.92	87.99	32.43

Table 3: Main QA-style generation results on NQ-Synth and NQ-Swap. We report ROUGE-L, FactKB, BERT-P, and accuracy (%). Best available values within each model–dataset block are shown in **bold**. Best results for each model are shown in **bold**.

conflicts with parametric knowledge, stronger contrastive suppression methods such as ADACAD often remain more effective.

Summarization Performance For summarization tasks, as shown in Table 2, CFB consistently improves over strong baselines on CNN/DM across all three models. On Mistral-7B, Token-aware CFB achieves the best ROUGE-L (34.52) and FactKB (96.87), while CAD remains slightly better on BERT-P (89.65). On Llama2-13B, Context-aware CFB achieves the best ROUGE-L (37.52), Static CFB attains the highest FactKB (98.85), and Token-aware CFB obtains the best BERT-P (89.83). On Llama3-8B, Static CFB gives the best ROUGE-L (36.79), whereas Context-aware CFB achieves the strongest factual consistency and semantic preservation, with FactKB 97.23 and BERT-P 89.85.

On XSum, the pattern is more model-dependent.

Method	Human Ratings			LLM Evaluation		
	Faith.	Flu.	Info.	Consist.	Hall.	Contra.
CAD	3.82	4.15	3.76	0.83	1.24	0.12
ADACAD	4.03	4.21	3.89	0.87	0.95	0.09
Token-aware CFB (Ours)	**4.31**	4.18	**4.12**	**0.91**	**0.67**	**0.05**

Table 4: Human and LLM-based evaluation results. Faith., Flu., and Info. denote faithfulness, fluency, and informativeness, respectively. Consist. denotes consistency, Hall. denotes the average number of hallucinations per output, and Contra. denotes contradiction rate. Human ratings are on a 1–5 scale, where higher is better.

For Mistral-7B, CAD still achieves the best ROUGE-L (16.57), but Context-aware CFB substantially improves factual consistency and semantic preservation, reaching the best FactKB (56.02) and BERT-P (89.64). For Llama2-13B, all three CFB variants outperform baselines by large margins, with Token-aware CFB achieving the best ROUGE-L (15.25), Context-aware CFB the best FactKB (55.02), and Token-aware CFB the best BERT-P (89.57). For Llama3-8B, Token-aware CFB achieves the best ROUGE-L (13.23), while Context-aware CFB achieves the best FactKB (66.85) and Static CFB the best BERT-P (88.69). Overall, these results suggest that on summarization, CFB reliably strengthens faithfulness-related metrics, while the best variant depends on the model and the desired trade-off between lexical overlap, factual consistency, and semantic preservation.

Question Answering Performance In QA tasks, shown in Table 3, we observe a clear difference between NQ-Synth and NQ-Swap. On NQ-Synth, where the provided context is complementary to the model's parametric knowledge, CFB consistently improves generation quality and answer accuracy. Token-aware CFB achieves the best accuracy for all three models: 60.10 on Mistral-7B, 64.00 on Llama2-13B, and 73.40 on Llama3-8B. It also delivers the best ROUGE-L for all three models, indicating that token-level redistribution is particularly helpful when the model can benefit from reinforcing context-supported evidence.

On NQ-Swap, however, the trend differs. ADACAD performs best on accuracy for all three models, achieving 73.79 on Mistral-7B, 74.21 on Llama2-13B, and 86.50 on Llama3-8B. In comparison, CFB variants generally lag behind on this conflict-heavy setting, although they remain competitive on some faithfulness-related metrics. For example, Context-aware CFB achieves the best FactKB (55.35) and BERT-P (88.33) on Llama2-13B, and Static/Context-aware CFB achieve the strongest FactKB and BERT-P among CFB variants on Llama3-8B. These results suggest that CFB is especially effective when context should be amplified, whereas methods explicitly designed to suppress parametric priors remain stronger when context and internal knowledge are in direct conflict.

Model-Specific Analysis CFB's behavior also varies across model architectures. On CNN/DM, all three models benefit substantially from CFB, but the strongest variant differs: Token-aware CFB works best for Mistral-7B, Context-aware CFB is strongest on Llama2-13B, and Static or Context-aware CFB are most effective on Llama3-8B depending on the metric. On XSum, CFB provides especially large factuality gains over baselines for Llama2-13B and Llama3-8B, while Mistral-7B remains more competitive with CAD on ROUGE-L. In QA, Llama3-8B shows the largest gains from CFB on NQ-Synth, reaching 73.40 accuracy with Token-aware CFB, but also exhibits the sharpest gap to ADACAD on NQ-Swap. Taken together, these results suggest that CFB is robust for source-grounded generation, especially in summarization and complementary-knowledge QA, while its effectiveness under explicit knowledge conflict depends more strongly on model architecture and the choice of decoding strategy.

4.3 Human Evaluation

To complement automatic metrics, we further conduct human and LLM-based evaluation to assess generation quality from a qualitative perspective. We randomly sample 100 examples each from CNN-DM and NQ-Swap, and compare outputs from CAD, ADACAD, and Token-aware CFB. The results are reported in Table 4.

Evaluation Protocol Three expert annotators independently rate each output on three dimensions: *faithfulness*, *fluency*, and *informativeness*, using a 1–5 scale. For *faithfulness*, a score of 1 indicates that the output contains major contradictions or unsupported content, while a score of 5 indicates that it is fully consistent with the input context. For *fluency*, a score of 1 indicates severe grammatical or coherence issues, while a score of 5 indicates natural and fluent language. For *informativeness*, a score of 1 indicates that the output is largely incomplete or irrelevant, while a score of 5 indicates that it is highly relevant and sufficiently informative.

Models	Base Model	CAD	AdaCAD	COIECD	Static CFB	Context-aware CFB	Token-aware CFB
FLOPS	3.40e+12	4.92e+07	1.15e+08	1.31e+08	8.19e+07	9.83e+07	2.86e+08

Table 5: Estimated FLOPS per decoding step. "Base Model" reports standard transformer decoding cost, while other entries denote additional method-specific overhead.

Method Variant	ROUGE-L	FactKB	BERT-P
Token-aware CFB	**35.81**	**94.31**	**89.38**
- w/o attention	35.60	93.74	88.48
- w/o semantic	4.45	66.84	67.68
- w/o JSD	35.24	93.60	88.43

Table 6: Ablation study of Token-aware CFB on Llama3-8B on CNN-DM. We report ROUGE-L, FactKB, and BERT-P.

Human Evaluation Results As shown in Table 4, Token-aware CFB achieves the best human rating on faithfulness (4.31) and informativeness (4.12), outperforming both CAD and ADACAD. Its fluency score (4.18) remains comparable to the baselines, indicating that stronger contextual grounding does not noticeably harm language quality. These results suggest that CFB improves factual alignment and content coverage while preserving readability.

LLM-based Evaluation We further use GPT-4o as an automatic judge with a structured rubric to evaluate consistency, hallucination, and contradiction. Token-aware CFB again performs best, achieving the highest consistency (0.91), the lowest hallucination count (0.67), and the lowest contradiction rate (0.05). This trend is consistent with the human judgments and provides additional evidence that CFB improves contextual reliability.

As shown in Table 4, our method achieves the highest scores across most metrics, with particularly strong performance in faithfulness (4.31/5.0) and informativeness (4.12/5.0). While fluency scores remain comparable across methods, the significant reductions in hallucination (0.67 average instances) and contradiction rates (5%) demonstrate the effectiveness of our constrained factual boosting approach.

4.4 Computational Efficiency

We compute the estimated FLOPS per decoding step for each method by breaking down their component operations (e.g., attention, similarity, logit shaping). Calculations assume a standard Llama-like setup: batch size 1, sequence length 128, hidden size 4096, 32 layers, and context length 512.

As shown in Table 5, all CFB variants are efficient relative to the base model. In particular, the Static and Context-aware CFB variants require less than 0.003% of the base model's FLOPS, while Token-aware CFB, though more compute-intensive, remains lightweight and offers finer control. Compared to baselines like CAD, AdaCAD, and COIECD, our variants achieve favorable trade-offs between complexity and fidelity control. We further report empirical runtime per decoding step in Table 8 in Appendix A. The results confirm that CFB decoding is practical in real-world settings, with Static and Context-aware variants offering strong efficiency-performance balance.

4.5 Ablation Study and Parameter Analysis

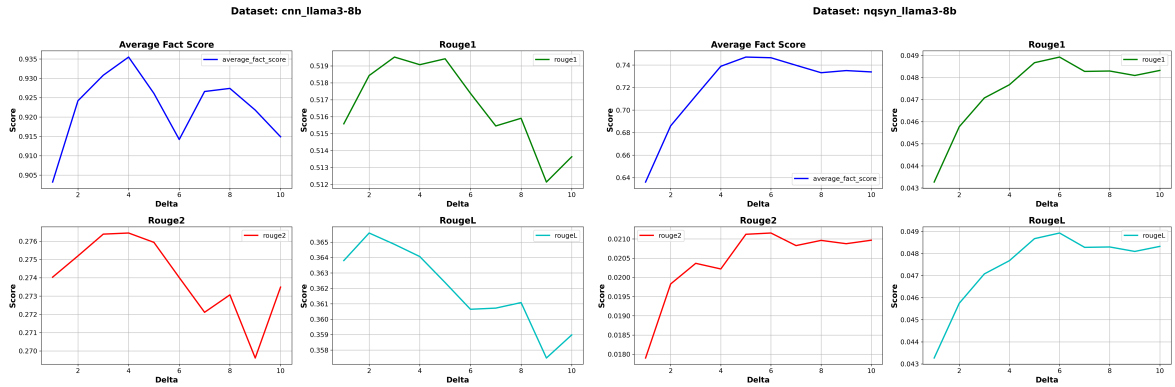
To understand the impact of individual components in Token-aware CFB, we conduct ablation studies and parameter sensitivity analysis using Llama3-8B on CNN-DM.

Component Ablation We ablate the main components of Token-aware CFB in Table 6. The full model achieves the best performance on all three metrics, with 35.81 ROUGE-L, 94.31 FactKB, and 89.38 BERT-P. Removing attention causes only a modest drop, suggesting that the attention signal provides useful but limited additional benefit in the current formulation. Removing JSD also degrades all metrics, confirming that sample-level adaptive scaling remains important even when token-level reweighting is used. In contrast, removing semantic similarity leads to a severe collapse across all metrics, indicating that the semantic component plays a crucial stabilizing role in the token-aware variant.

Parameter Analysis We further analyze the effect of the boost value δ in Figure 3. For CNN-DM, moderate boost values yield the best trade-off between faithfulness and generation quality, while overly large boosts lead to degraded performance. On NQ-Synth, performance is more stable across a wider range of boost values, suggesting that stronger context reinforcement is more tolerable when the input context complements the model's parametric knowledge. We also sweep the importance weights λ_1 and λ_2 under the constraint

Component	Content
Context	The Great Wall of China's actual length was determined in 2012 to be 21,196 kilometers (13,171 miles), much longer than the previously believed 8,850 kilometers. This finding came after a detailed archaeological survey using advanced mapping techniques, which also revealed that the wall contained sections from more diverse historical periods than originally thought...
Question	What is the actual length of the Great Wall of China according to the 2012 survey, and how does it compare to previous estimates?
Gold Answer	According to the 2012 survey, the Great Wall is 21,196 kilometers long, which is much longer than the previous estimate of 8,850 kilometers.
Standard (Greedy)	The Great Wall of China is approximately 8,850 kilometers long.
CAD	The 2012 survey found the Great Wall to be 21,196 kilometers, longer than previous estimates.
COIECD	The Great Wall is about 13,171 miles (8,850 kilometers) long based on traditional measurements.
CFB (Ours)	According to the 2012 survey, the Great Wall is 21,196 kilometers long, much longer than the previously believed 8,850 kilometers.

Table 7: Qualitative comparison showing how different methods handle knowledge conflicts.



(a) CNN-DM (b) NQ-Synth

Figure 3: Impact of boost values (δ) on fact scores and ROUGE metrics using Llama3-8B. We show the average fact score (top-left), ROUGE-1 (top-right), ROUGE-2 (bottom-left), and ROUGE-L (bottom-right) scores.

$\lambda_1 + \lambda_2 = 1$, and find that $\lambda_1 = 0.6$ and $\lambda_2 = 0.4$ give the best overall performance, which we use as the default setting for Token-aware CFB.

4.6 Case Studies

Case 1: High Knowledge Conflict As shown in Table 7, when the provided context conflicts with common parametric knowledge about the Great Wall's length (21,196 vs. 8,850 kilometers), greedy decoding and COIECD fall back to the widely known but context-inconsistent estimate of 8,850 kilometers. CAD partially resolves the conflict by mentioning the updated measurement. In contrast, CFB produces the most faithful response by correctly stating the 2012 survey result and explicitly contrasting it with the earlier estimate.

Case 2: Complementary Context When the input context provides additional evidence that is consistent with the model's prior knowledge, CFB can incorporate this information while maintaining coherence. In such cases, the method not only preserves the main answer but also better reflects supporting contextual details, leading to more complete and context-grounded responses.

Case 3: Low Knowledge Conflict When the conflict between context and parametric knowledge is weak, CFB behaves more conservatively and tends to preserve fluent generation while still favoring source-supported content. This behavior is consistent with the goal of improving contextual faithfulness without introducing unnecessary distortion in low-conflict settings.

5 Conclusion

We present Context-Fidelity Boosting (CFB), a decoding-time framework for improving contextual faithfulness in language model outputs. Across summarization and question answering tasks, CFB shows that simple additive logit shaping can effectively encourage context-grounded generation without retraining. Our results further suggest that different CFB variants offer complementary trade-offs between robustness and fine-grained control.

Limitations

CFB has several limitations. First, it requires access to model internals such as logits and, for the token-aware variant, signals like attention or token embeddings, which makes it difficult to apply in black-box API settings. Second, although CFB is lightweight compared with retraining-based methods, it still introduces additional decoding overhead from divergence computation and token-level relevance scoring. Third, the token-aware variant depends on the quality of local relevance estimation and is not consistently superior in high-conflict settings. Future work could explore black-box approximations, more robust token-level relevance modeling, and further reductions in decoding cost.

References

Zhongwu Chen, Chengjin Xu, Dingmin Wang, Zhen Huang, Yong Dou, and Jian Guo. 2025. Rulerag: Rule-guided retrieval-augmented generation with language models for question answering. *Preprint*, arXiv:2410.22353.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. 2024. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 1419–1436. Association for Computational Linguistics.

Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. 2024. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model. *Preprint*, arXiv:2306.16092.

Shangbin Feng, Vidhisha Balachandran, Yuyang Bai, and Yulia Tsvetkov. 2023. Factkb: Generalizable factuality evaluation using language models enhanced with factual knowledge. *Preprint*, arXiv:2305.08281.

Aryo Pradipta Gema, Chen Jin, Ahmed Abdulaal, Tom Diethe, Philip Teare, Beatrice Alex, Pasquale Minervini, and Amrutha Saseendran. 2024. Decore: Decoding by contrasting retrieval heads to mitigate hallucinations. *CoRR*, abs/2410.18860.

Noah Golowich and Ankur Moitra. 2024. Edit distance robust watermarks for language models. *Preprint*, arXiv:2406.02633.

Peter Hase, Thomas Hofweber, Xiang Zhou, Elias Stengel-Eskin, and Mohit Bansal. 2024. Fundamental problems with model editing: How should rational belief revision work in llms? *CoRR*, abs/2406.19354.

Giwon Hong, Aryo Pradipta Gema, Rohit Saxena, Xiaotang Du, Ping Nie, Yu Zhao, Laura Perez-Beltrachini, Max Ryabinin, Xuanli He, Clémentine Fourrier, and Pasquale Minervini. 2024. The hallucinations leaderboard - an open effort to measure hallucinations in large language models. *CoRR*, abs/2404.05904.

Bairu Hou, Yang Zhang, Jacob Andreas, and Shiyu Chang. 2024. A probabilistic framework for LLM hallucination detection via belief tree propagation. *CoRR*, abs/2406.06950.

Minda Hu, Bowei He, Yufei Wang, Liangyou Li, Chen Ma, and Irwin King. 2024. Mitigating large language model hallucination with faithful finetuning. *CoRR*, abs/2406.11267.

Le Huang, Hengzhi Lan, Zijun Sun, Chuan Shi, and Ting Bai. 2024. Emotional rag: Enhancing role-playing agents through emotional retrieval. *Preprint*, arXiv:2410.23041.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *CoRR*, abs/2311.05232.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2024. A watermark for large language models. *Preprint*, arXiv:2301.10226.

Philippe Laban, Alexander R. Fabbri, Caiming Xiong, and Chien-Sheng Wu. 2024. Summary of a haystack: A challenge to long-context llms and rag systems. *Preprint*, arXiv:2407.01370.

Jean Lee, Nicholas Stevens, and Soyeon Caren Han. 2025. Large language models in finance (finllms). *Neural Computing and Applications*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2024a. A semantic invariant robust watermark for large language models. *Preprint*, arXiv:2310.06356.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2024b. A semantic invariant robust watermark for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yantao Liu, Zijun Yao, Xin Lv, Yuchen Fan, Shulin Cao, Jifan Yu, Lei Hou, and Juanzi Li. 2024c. Untangle the KNOT: Interweaving conflicting knowledge and reasoning skills in large language models. In *Proceedings of the 2024 Joint International Conference*

on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 17186–17204, Torino, Italia. ELRA and ICCL.

Yepeng Liu and Yuheng Bu. 2024. Adaptive text watermark for large language models. *Preprint*, arXiv:2401.13927.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Yifei Ming, Senthil Purushwalkam, Shrey Pandit, Zixuan Ke, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. 2024. Faitheval: Can your language model stay faithful to context, even if "the moon is made of marshmallows". *CoRR*, abs/2410.03727.

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, Brussels, Belgium. Association for Computational Linguistics.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10862–10878. Association for Computational Linguistics.

Zexuan Qiu, Zijing Ou, Bin Wu, Jingjing Li, Aiwei Liu, and Irwin King. 2024. Entropy-based decoding for retrieval-augmented large language models. *CoRR*, abs/2406.17519.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

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: Short Papers, NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 783–791. Association for Computational Linguistics.

S. M. Towhidul Islam Tonmoy, S. M. Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *CoRR*, abs/2401.01313.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2024. Adacad: Adaptively decoding to balance conflicts between contextual and parametric knowledge. *CoRR*, abs/2409.07394.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Kevin Wu, Eric Wu, and James Y. Zou. 2024. Clasheval: Quantifying the tug-of-war between an llm's internal prior and external evidence. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Dingkang Yang, Dongling Xiao, Jinjie Wei, Mingcheng Li, Zhaoyu Chen, Ke Li, and Lihua Zhang. 2024. Improving factuality in large language models via decoding-time hallucinatory and truthful comparators. *CoRR*, abs/2408.12325.

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, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 3903–3922. Association for Computational Linguistics.

Qingru Zhang, Xiaodong Yu, Chandan Singh, Xiaodong Liu, Liyuan Liu, Jianfeng Gao, Tuo Zhao, Dan Roth, and Hao Cheng. 2024. Model tells itself where to attend: Faithfulness meets automatic attention steering. *CoRR*, abs/2409.10790.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Yinghao Zhu, Changyu Ren, Shiyun Xie, Shukai Liu, Hangyuan Ji, Zixiang Wang, Tao Sun, Long He, Zhoujun Li, Xi Zhu, and Chengwei Pan. 2024. Realm: Rag-driven enhancement of multimodal electronic health records analysis via large language models. *Preprint*, arXiv:2402.07016.

A Additional Efficiency Results

We report empirical runtime per sample (in seconds) for each decoding method across three benchmark datasets. Measurements were taken on a single GPU with batch size 1.

Method	CNN-DM	XSum	NQ-Synth
COIECD	3.22	2.56	0.45
Static CFB	**1.38**	**1.28**	**0.15**
Context-aware CFB	2.00	1.86	0.20
Token-aware CFB	10.39	9.35	0.51

Table 8: Average runtime per decoding sample (in seconds) across datasets.

As shown in Table 8, Static CFB is the most efficient variant and is consistently faster than COIECD across all three datasets. Context-aware CFB introduces only a modest additional cost over Static CFB, reflecting the overhead of divergence computation. Token-aware CFB is substantially slower due to step-wise attention-based scoring and token-level relevance computation, but remains practical for small-batch evaluation. These results are consistent with our design goal of providing a spectrum of methods with different trade-offs between efficiency and fine-grained control.

B Supplementary Results on Reasoning Tasks

To explore the boundary of CFB's applicability in reasoning-intensive scenarios, we conduct preliminary evaluations on multi-hop and reading comprehension QA datasets, namely HotpotQA and TriviaQA. These tasks require multi-step inference and compositional reasoning, which are not explicitly modeled by CFB.

Method	HotpotQA Acc.	TriviaQA Acc.
Baseline (CAD)	**39.5**	40.2
Static CFB	34.7	69.1
Context-aware CFB	36.8	70.4
Token-aware CFB	37.1	**71.5**

Table 9: Accuracy on reasoning tasks using Llama3-8B. While not directly designed for multi-hop reasoning, CFB improves context grounding and yields strong results on TriviaQA.

Table 9 shows that CFB is not uniformly beneficial on all reasoning-heavy tasks. On HotpotQA, CAD remains the strongest method, suggesting that simple context boosting alone is insufficient for tasks requiring stronger multi-step reasoning. In contrast, all CFB variants substantially improve over CAD on TriviaQA, with Token-aware CFB achieving the best accuracy. This pattern suggests that CFB is more effective when improved context grounding directly supports answer generation, but is less reliable when success depends primarily on complex reasoning rather than context alignment alone.