

FRUGALPROMPT: Reducing Contextual Overhead in Large Language Models via Token Attribution

Syed Rifat Raiyan^{1,*}, Md Farhan Ishmam^{2,*},
Abdullah Al Imran³, Mohammad Ali Moni⁴

¹Islamic University of Technology, Bangladesh ²University of Utah, USA

³University of Liverpool, UK ⁴University of Queensland, Australia

rifatraiyan@iut-dhaka.edu, farhan.ishmam@utah.edu *Equal Contribution

🌟 Starscream-11813/Frugal-ICL

🌟 frugalprompt.github.io

Abstract

Human communication heavily relies on laconism and inferential pragmatics, allowing listeners to successfully reconstruct rich meaning from sparse, telegraphic speech. In contrast, large language models (LLMs) owe much of their stellar performance to expansive input contexts, yet such verbosity inflates monetary costs, carbon footprint, and inference-time latency. This overhead manifests from the redundant low-utility tokens present in typical prompts, as only a fraction of tokens typically carries the majority of the semantic weight. Inspired by the aforementioned cognitive psycholinguistic processes, we address this inefficiency by introducing FRUGALPROMPT, a novel prompt compression framework for LLMs, which retains only the most semantically significant tokens. Leveraging two state-of-the-art token attribution methods, GlobEnc and DecompX, we assign saliency scores to every token in an input sequence, rank them to retain the top- $k\%$ tokens, and obtain a sparse *frugalized* prompt. We establish the theoretical stability of our approach and provide strong empirical results across a suite of four NLP tasks to study the trade-off between the portion of retained tokens and performance. Experimental findings across retention settings reveal asymmetric performance patterns that suggest potential task contamination effects. We posit that our work contributes to a more nuanced understanding of LLM behavior in performance-efficiency trade-offs and delineates the boundary between tasks tolerant of contextual sparsity and those requiring exhaustive context.

“Why waste time say lot word when few word do trick?”

—Kevin Malone, *The Office* (Season 8, Episode 2)

1 Introduction

As large language models (LLMs) are now becoming general-purpose intelligence systems, they are supporting workflows that go beyond their initial

zero-shot text generation: from coding systems capable of using tools to multi-step agentic systems solving multimodal problems (Yang et al., 2024; Schmidgall et al., 2025). Yet the same empirical scaling laws that fuel these capabilities into trillion-parameter models (Fedus et al., 2022; Hudson et al., 2023) also intensify a basic cognitive and computational tension. Human language is richly *redundant*¹, while modern LLM systems being trained and conditioned on human text are constrained by context length (Wu et al., 2025), latency (Wan et al., 2024), and energy costs (Stojkovic et al., 2024) due to the verbosity of human language.

A core observation of studying linguistics from the lens of information theory is that natural language is not an information-minimal code (Shannon, 1951). Human utterances inherently contain statistical redundancy, *e.g.*, text is repeated for clarification, and function words serve grammatical rather than semantic roles. Many tokens, therefore, contribute little incremental information once the local context is known. Psycholinguistic evidence similarly suggests that comprehenders can often recover intended meaning from “telegraphic” input, a property exploited in simplified registers and second-language communication (Gibson et al., 2013; Klein and Perdue, 1997). These findings suggest a cognitively motivated route to efficiency in NLP, and we ask: To what extent do LLMs require full linguistic explicitness, and to what extent can they reconstruct meaning from pragmatically sufficient cues?

We aim to answer these questions with a theoretically grounded work motivated by information-theoretic approaches in linguistics. Using SOTA token attribution techniques, we compute saliency scores for the prompt and selectively retain the semantically rich tokens while filtering redundant

¹We define redundancy as any verbiage that extends beyond the most succinct form of communication.

ones. This yields a training-free compressor that (i) better preserves meaning-critical content, (ii) systematically removes low-value discourse material and function words, and (iii) avoids the cost of an auxiliary large model. Importantly, our approach requires only a lightweight pre-trained encoder, where the saliency scores are generated by state-of-the-art token attribution methods (Modarressi et al., 2022, 2023) applied to a single 110M-parameter BERT model (Devlin et al., 2019), incurring substantially lower parameter overhead than the auxiliary language models used by prior compressors (Jiang et al., 2023a; Chuang et al., 2024).

Our filtering criteria are grounded in two linguistic principles underlying textual redundancy. First, there is an intrinsic redundancy to any form of text produced by humans (Wit and Gillette, 1999). For instance, in the text, “*What are your plans for the weekend, by the way?*”, the phrase “*by the way*” contributes no additional meaning to an LLM. Second, the English language has low-importance tokens that add little to no semantic weight, generally in the form of articles and auxiliary verbs. For instance, most of the meaning is conveyed if the statement “*I have received a parcel.*”, is simplified to “*I received parcel.*”, with the latter mimicking simplified English, often used by non-native speakers (Shubert et al., 1995). Our filtering methods eliminate both forms of redundancy. Our key contributions are as follows:

- We introduce a novel, training-free prompt frugalization strategy for LLMs that controllably filters low-importance tokens based on saliency scores from pre-trained encoders.
- Our method is theoretically grounded with empirical results showing strong retention–performance trade-offs. Across four NLP tasks, a 20% prompt reduction preserves performance for most models with negligible parameter overhead.
- We reveal behavioral insights into how scaling performance relates to inference cost and potential task contamination in benchmarks.

2 Related Work

2.1 Redundancy in Language

Language, by nature, is not an information-minimal code (Shannon, 1951) and introduces several forms of redundancy. Through the lens of information theory (Shannon, 1948), redundancy of a language can

be defined as the difference between the maximum possible entropy of the language and its actual entropy. Theories in inferential pragmatics (Richards, 1990) and relevance-driven communication argue that interlocutors reconstruct implicit content by focusing on semantically informative cues rather than processing every token with equal weight. Such laconism is not an anomaly but a fundamental property of human communicative efficiency (Levshina and Moran, 2021).

2.2 LLM Efficiency

The current literature on LLM efficiency (Wan et al., 2024) stems from earlier works on efficient transformers (Tay et al., 2022). Inference-time efficiency can be achieved via model compression methods, such as quantization (Dettmers et al., 2023b), parameter pruning (Ma et al., 2023), low-rank decomposition (Li et al., 2023a), and knowledge distillation (Timiryasov and Tastet, 2023). Training efficiency can be achieved during pre-training (Liu et al., 2023; Yao et al., 2024) and fine-tuning, either through parameter-efficient (Hu et al., 2022) or memory-efficient (Dettmers et al., 2023a) approaches. Finally, architectural changes also improve efficiency, *e.g.*, efficient attention (Child et al., 2019; Ainslie et al., 2023) and state-space models (Gu and Dao, 2024).

Distinct from these methods, our work adopts a data-centric approach, where the input prompt is optimized rather than the model’s underlying weights or architecture (Chang et al., 2024). In essence, we perform prompt compression, which minimizes input redundancy while retaining task-specific information (Li et al., 2023b; Jiang et al., 2023a,b). Contrasting previous approaches that often use a paraphrasing model (Chuang et al., 2024), which can lead to instability, ours takes a theoretically grounded approach by filtering the least salient tail-end tokens. Among token-level compressors, Selective Context (Li et al., 2023b) prunes lexical units with low self-information under a causal language model, while LLMingua-2 (Jiang et al., 2023b) distills GPT-4 compression annotations into a trained token classifier.

2.3 Token Attribution

The self-attention weights in transformers have been primarily explored as a proxy for token importance (Htut et al., 2019; Clark et al., 2019), yet they fail to align with the true attribution of tokens (Jain and Wallace, 2019). While including vector

norms can partially mitigate this issue (Kobayashi et al., 2020), it overlooks the intricacies of the non-attention components in a transformer. Instead, we rely on subsequent token attribution methods (see Appendix A): GlobEnc (Modarressi et al., 2022), which quantifies global token attribution by aggregating layer-wise analyses across the entire encoder using attention rollout and vector norms, and DecompX (Modarressi et al., 2023), which propagates locally decomposed token representations globally to incorporate all model components, including non-linear feed-forward networks. These methods were selected for their superior ability to capture comprehensive semantic contributions beyond raw attention weights, enabling identification of salient tokens with minimal computational overhead.

3 Methodology

Figure 1 portrays an overview of our proposed pipeline. We obtain the saliency scores using the token attribution methods: GlobEnc (Modarressi et al., 2022) and DecompX (Modarressi et al., 2023), (details in § A), and employ frugalization via token filtration to generate a condensed yet semantically representative version of an input sample. This technique identifies and retains only the most salient tokens, thereby reducing input dimensionality while preserving the core meaning for the given task. Inputs longer than the encoder’s 512-token window are scored via sentence-level chunking, as detailed in Appendix C.

3.1 Ranking via Saliency Scores

We define a task-specific scoring function $\varphi_\tau \in \{\text{GlobEnc}, \text{DecompX}\}$ for a task τ , which assigns a saliency score to each token in a given text, *i.e.*,

$$\varphi_\tau : \mathcal{T} \rightarrow \mathbb{R}^n,$$

where $\mathcal{T}$ denotes the space of the token sequences and n is the length of the sequence. For the input token sequence, $T = \langle t_1, t_2, \dots, t_n \rangle$, the scoring function produces a saliency vector,

$$\mathbf{s} = \varphi_\tau(T) = \langle s_1, s_2, \dots, s_n \rangle, \quad s_i \in \mathbb{R},$$

where each score s_i reflects the relative contribution of token t_i to the model’s output for the specified task. Tokens are then ranked in monotone decreasing order of their attribution scores to obtain a permutation, π , formed of the token indices, such that,

$$s_{\pi(1)} \geq s_{\pi(2)} \geq \dots \geq s_{\pi(n)}.$$

3.2 Top- k Filtering

To construct the *frugalized representation*, we retain only the top $p = \lceil \frac{k}{100} \cdot n \rceil$ tokens, where k is the retention percentage. The selected indices form the sequence,

$$S_k = \langle \pi(1), \pi(2), \dots, \pi(p) \rangle.$$

We reindex S_k in monotone increasing order to produce $S_{k\uparrow}$. The frugalized sequence F_k is produced by restoring the selected tokens to their original order, *i.e.*,

$$F_k = \langle t_i | i \in S_{k\uparrow} \rangle$$

ensuring that the temporal coherence of the sequence is preserved, while uninformative tokens are discarded.

We elucidate our frugalization approach with an example. For the input sequence $T = \langle \text{“The”, “movie”, “was”, “good”, “,”, “and”, “I”, “liked”, “it”, “very”, “much”} \rangle$, suppose the attribution scores for sentiment analysis prioritize content and sentiment words (*e.g.*, “movie”: 0.95, “good”: 0.90, “much”: 0.85, “liked”: 0.80). Upon retaining the top $k = 35\%$ ($p = \lceil 0.35 \cdot 11 \rceil = 4$) and reordering them according to their original relative order, we get the reduced sequence

$$F_{35} = \langle \text{“movie”, “good”, “liked”, “much”} \rangle$$

where the original order is preserved to maintain syntactic coherence. The frugalized output “*movie good liked much*” is thus created, discarding the less salient tokens (*e.g.*, “The”, “was”, etc.) while retaining semantic essence.

3.3 Theoretical Bounds and Relationships

We now formalize when frugalization is safe. Our main result (Corollary 1) shows that the loss incurred by retaining only the top- $k\%$ salient tokens is bounded by two terms: (i) a linear term in the saliencies of the *deleted* tokens, which is minimized by our top- k rule, and (ii) a second-order interaction term that grows quadratically with the number of deletions. This decomposition explains why top- k filtering works and aggressive deletion eventually breaks down. The bounds formalize the relationship between dropping $(100 - k)\%$ of tokens and the estimated performance.

Deletion Effect. For a filtered or deleted set of indices $D \subseteq [n] = \{1, \dots, n\}$, we define the retained subsequence of tokens $T \setminus D := T_{[n] \setminus D}$. For task τ and model parameters θ , let $\ell_\tau(T, y; \theta) \in \mathbb{R}^+$ be the task loss or evaluation score. We define

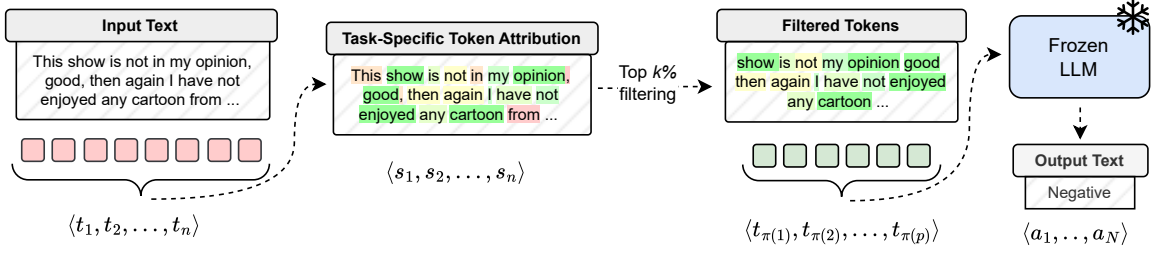


Figure 1: Overview of the FRUGALPROMPT pipeline. The input prompt with tokens $\langle t_1, t_2, \dots, t_n \rangle$ is passed to our task-specific token attribution module to generate the saliency scores for each token $\langle s_1, s_2, \dots, s_n \rangle$. The tokens are ordered based on their saliency in a permutation π and the top $p = \lceil \frac{k}{100} \cdot n \rceil$ tokens are retained while maintaining the order of the text to form the reduced text $\langle t_{\pi(1)}, t_{\pi(2)}, \dots, t_{\pi(p)} \rangle$. The reduced text is inferred by a frozen LLM to generate the answer tokens $\langle a_1, a_2, \dots, a_N \rangle$.

the deletion-effect set function:

$$f(D) := \underbrace{\ell_\tau(T \setminus D, y; \theta)}_{\text{Loss after Deletion}} - \underbrace{\ell_\tau(T, y; \theta)}_{\text{Original Loss}}. \quad (1)$$

Trivially, $f(\emptyset) = 0$. We emphasize that all results are inference-time, *i.e.*, the model parameters θ are fixed. We treat ℓ_τ as a loss throughout (lower is better), so that $f(D) > 0$ indicates deletion hurts performance; for metrics where higher is better (*e.g.*, accuracy, ROUGE), ℓ_τ should be read as the negative of that metric.

Discrete Marginals and Interactions. To bound $f(D)$ when $|D|$ is large, we decompose it into per-token marginals (Eq. 2) and pairwise interactions (Eq. 3), and the assumptions below then control each piece separately. For a set of deleted tokens $A \subseteq [n]$ and a new token $i \notin A$, we define the marginal deletion:

$$\Delta_i f(A) := f(A \cup \{i\}) - f(A). \quad (2)$$

For tokens $i \neq j$ with $i, j \notin A$, we define the second-order discrete difference, *i.e.*, the pairwise interaction as

$$\begin{aligned} \Delta_{ij} f(A) &:= \Delta_i \Delta_j f(A) = f(A \cup \{i, j\}) \\ &\quad - f(A \cup \{i\}) - f(A \cup \{j\}) + f(A). \end{aligned} \quad (3)$$

Assumption 1 (Saliency upper-bounds singleton deletion harm). *There exists a task constant, $C_\tau \geq 0$, such that,*

$$f(\{i\}) \leq C_\tau s_i, \quad \forall i \in [n]. \quad (4)$$

Assumption 1 is a faithfulness condition: it asserts that the attribution score s_i is not merely a heuristic ranking but a calibrated upper bound (up to the task-dependent constant C_τ) on the harm of removing token i in isolation. This is precisely the property that GlobEnc and DecompX are designed to provide, and is empirically supported by the faithfulness evaluations in Modarressi et al. (2022, 2023). We treat C_τ as an empirical constant that

absorbs any departure from the ideal faithfulness for task τ .

Assumption 2 (Bounded pairwise interaction). *There exists a task constant, $\gamma_\tau \geq 0$, such that,*

$$\Delta_{ij} f(A) \leq \gamma_\tau, \quad \forall A \subseteq [n], \forall i, j \notin A. \quad (5)$$

Assumption 2 is a submodular-flavored condition standard in coverage and feature-selection theory: it prevents the joint harm of deleting two tokens from exceeding the sum of their individual harms by more than a constant γ_τ . We impose only the one-sided bound because negative interactions are benign (and indeed expected when deleted tokens are mutually redundant): two redundant tokens deleted together may hurt less than the sum of their singleton deletions. The constant γ_τ thus captures the worst-case *compositional* fragility of task τ to multi-token deletion.

Theorem 1 (Deletion bound via saliency and interaction). *Under Assumptions 1 and 2, for any deletion set D with $|D| = q$,*

$$f(D) \leq C_\tau \sum_{i \in D} s_i + \frac{\gamma_\tau}{2} q(q-1). \quad (6)$$

The proof of Theorem 1 is provided in Appendix §B. Theorem 1 establishes the stability of token deletion by upper-bounding the deletion effect.

Retention Relationship. Following §3.2, we define the deleted set $D_k := \{\pi(p+1), \dots, \pi(n)\}$ and the frugalized sequence $F_k := T \setminus D_k$.

Corollary 1 (Performance vs. $k\%$ retention). *Under Assumptions 1 and 2, with $q = n - p$,*

$$\begin{aligned} \ell_\tau(F_k, y; \theta) - \ell_\tau(T, y; \theta) &\leq C_\tau \sum_{j=p+1}^n s_{\pi(j)} \\ &\quad + \frac{\gamma_\tau}{2} (n-p)(n-p-1). \end{aligned} \quad (7)$$

Corollary 1 follows directly by substituting D in Theorem 1 and expanding $f(D_k)$ via Eq. 1. The

Model	Prompt Reduction Method	$k\%$	Tasks									
			CLS		SUM						QA	RSN
			Acc	F1	BLEU	R-1	R-2	R-L	BERT	METEOR	Acc	pass@1
Llama-3 8B	N/A	100	0.949	0.949	0.020	0.232	0.073	0.193	0.872	0.335	0.813	0.786
	GlobEnc	80	0.942	0.942	0.017	0.226	0.069	0.189	0.871	0.330	0.798	0.500
		60	0.942	0.942	0.015	0.220	0.064	0.181	0.870	0.319	0.754	0.243
		50	0.921	0.921	0.013	0.213	0.057	0.175	0.868	0.303	0.716	0.146
	DecompX	80	0.936	0.936	0.015	0.218	0.061	0.179	0.869	0.311	0.747	0.652
		60	0.900	0.900	0.014	0.206	0.055	0.169	0.866	0.286	0.681	0.425
		50	0.868	0.868	0.012	0.198	0.048	0.161	0.864	0.270	0.670	0.265
	LLMLingua2	80	0.920	0.919	0.014	0.225	0.068	0.185	0.864	0.323	0.790	0.740
		60	0.910	0.909	0.012	0.215	0.056	0.178	0.862	0.302	0.710	0.450
		50	0.641	0.641	0.008	0.209	0.052	0.171	0.860	0.293	0.680	0.280
	SelectiveContext	80	0.919	0.918	0.016	0.218	0.066	0.186	0.863	0.327	0.740	0.540
		60	0.901	0.899	0.007	0.197	0.046	0.155	0.859	0.271	0.720	0.270
		50	0.638	0.638	0.006	0.196	0.043	0.153	0.857	0.264	0.720	0.150
Llama-3 70B	N/A	100	0.953	0.953	0.020	0.235	0.073	0.196	0.876	0.333	0.874	0.919
	GlobEnc	80	0.948	0.948	0.018	0.235	0.071	0.195	0.875	0.331	0.862	0.669
		60	0.943	0.943	0.017	0.231	0.068	0.192	0.874	0.329	0.838	0.362
		50	0.938	0.938	0.016	0.228	0.065	0.189	0.873	0.316	0.816	0.231
	DecompX	80	0.949	0.949	0.017	0.226	0.065	0.187	0.870	0.316	0.839	0.818
		60	0.891	0.891	0.015	0.215	0.057	0.176	0.868	0.292	0.797	0.587
		50	0.839	0.837	0.014	0.209	0.054	0.171	0.866	0.281	0.770	0.409
	LLMLingua2	80	0.938	0.937	0.015	0.231	0.069	0.195	0.867	0.329	0.860	0.860
		60	0.889	0.890	0.017	0.229	0.067	0.191	0.866	0.293	0.835	0.590
		50	0.644	0.643	0.012	0.225	0.057	0.185	0.865	0.265	0.797	0.440
	SelectiveContext	80	0.920	0.919	0.015	0.230	0.068	0.193	0.868	0.329	0.855	0.810
		60	0.887	0.887	0.009	0.214	0.058	0.173	0.867	0.292	0.815	0.440
		50	0.634	0.634	0.010	0.203	0.045	0.163	0.866	0.268	0.740	0.220

Table 1: Impact of prompt-compression on Llama-3 base models (8B and 70B) across tasks. Base and best prompt-reduced performance for each model are highlighted in Gray and **bold** Light Blue cells, respectively.

bound also generalizes to the dataset level. Given that $(T, y) \sim \mathcal{D}_\tau$ denotes the test distribution for task τ , we define the risk as the expected loss,

$$R_\tau(\theta) := \mathbb{E}_{(T, y) \sim \mathcal{D}_\tau} [\ell_\tau(T, y; \theta)], \quad (8)$$

and taking expectations on both sides of Eq. 7 bounds the increase in risk under frugalization by the expected value of its right-hand side.

Interpretation. Because we delete the lowest-saliency tokens, $\sum_{j=p+1}^n s_{\pi(j)}$ is *minimized* over all subsets of size $n - p$. Top- k filtering is the optimal one-shot deletion strategy with respect to the bound in Eq. 7. The bound also separates two failure modes. The linear term $C_\tau \sum_{j=p+1}^n s_{\pi(j)}$ captures *information loss*, which is controllable by the ranking and small when the saliency distribution is concentrated on a few tokens. The quadratic term $\frac{\gamma_\tau}{2}(n - p)(n - p - 1)$ captures *compositional breakdown*, which is uncontrollable by the ranking and dominant when $n - p$ is large. This decomposition predicts the experimental pattern observed in §4. Tasks with sparse saliency (classification, question answering) tolerate aggressive deletion, while tasks with dense token dependencies (mathematical reasoning)

are interaction-limited and degrade quadratically as k shrinks.

4 Results and Analysis

Having established the performance bounds and their relationship with the retention $k\%$ in §3.3, we now provide empirical evidence in this section, with additional experimental details reported in Appendix §D.

4.1 Experimental Setup

We set the temperature to 0 for all inference runs, *i.e.*, greedy decoding, ensuring deterministic and reproducible generations. For the frugalized setup, we experiment by retaining the top $k \in \{50, 60, 80\}\%$ of the input tokens. We also include the baseline prompt, using 100% of the tokens to set a proper frame of reference for the performance delta across all models. This setup enables us to investigate whether low-attribution filler tokens can be excluded with minimal degradation in task performance, thereby reducing API costs and carbon footprint. Additional experimental details on the tasks (§D.1), datasets (§D.2), models

Model	Prompt Reduction Method	k%	Tasks									
			CLS		SUM						QA	RSN
			Acc	F1	BLEU	R-1	R-2	R-L	BERT	METEOR	Acc	pass@1
GPT-3.5	N/A	100	0.949	0.949	0.039	0.282	0.093	0.237	0.889	0.359	0.779	0.772
	GlobEnc	80	0.945	0.945	0.017	0.176	0.056	0.146	0.874	0.291	0.753	0.498
		60	0.925	0.925	0.015	0.172	0.052	0.141	0.874	0.285	0.716	0.264
		50	0.918	0.918	0.014	0.170	0.049	0.140	0.873	0.278	0.705	0.158
	DecompX	80	0.942	0.942	0.036	0.268	0.084	0.225	0.888	0.337	0.704	0.660
		60	0.724	0.704	0.031	0.253	0.073	0.210	0.885	0.311	0.648	0.419
		50	0.642	0.595	0.027	0.241	0.065	0.200	0.882	0.291	0.619	0.288
	LLMLingua2	80	0.933	0.932	0.035	0.265	0.092	0.221	0.886	0.325	0.750	0.700
		60	0.910	0.909	0.032	0.265	0.084	0.220	0.883	0.312	0.740	0.460
		50	0.710	0.708	0.029	0.261	0.079	0.218	0.880	0.323	0.700	0.300
	SelectiveContext	80	0.920	0.919	0.031	0.277	0.099	0.234	0.884	0.366	0.751	0.650
		60	0.910	0.908	0.025	0.244	0.072	0.199	0.881	0.302	0.720	0.360
50		0.681	0.680	0.022	0.228	0.063	0.180	0.879	0.283	0.650	0.200	
Gemini 2.0 Flash Thinking	N/A	100	0.952	0.952	0.034	0.262	0.081	0.219	0.885	0.345	0.880	0.956
	GlobEnc	80	0.947	0.947	0.031	0.252	0.081	0.212	0.882	0.344	0.879	0.704
		60	0.934	0.934	0.029	0.247	0.077	0.208	0.881	0.335	0.846	0.423
		50	0.920	0.919	0.026	0.239	0.071	0.199	0.879	0.322	0.827	0.277
	DecompX	80	0.855	0.852	0.031	0.251	0.075	0.209	0.883	0.328	0.856	0.856
		60	0.713	0.690	0.028	0.236	0.068	0.194	0.878	0.307	0.795	0.665
		50	0.627	0.571	0.024	0.225	0.059	0.185	0.876	0.283	0.774	0.463
	LLMLingua2	80	0.940	0.939	0.030	0.248	0.079	0.242	0.882	0.390	0.790	0.910
		60	0.920	0.919	0.029	0.235	0.076	0.230	0.877	0.367	0.840	0.640
		50	0.901	0.899	0.023	0.224	0.072	0.214	0.873	0.326	0.810	0.460
	SelectiveContext	80	0.950	0.949	0.043	0.270	0.096	0.225	0.880	0.366	0.790	0.800
		60	0.902	0.901	0.032	0.247	0.079	0.206	0.875	0.318	0.800	0.400
50		0.897	0.897	0.025	0.235	0.064	0.192	0.872	0.299	0.750	0.200	
o3-mini	N/A	100	0.957	0.957	0.023	0.221	0.065	0.182	0.860	0.297	0.845	0.961
	GlobEnc	80	0.956	0.956	0.020	0.216	0.060	0.176	0.859	0.290	0.826	0.724
		60	0.941	0.941	0.019	0.212	0.059	0.173	0.858	0.282	0.802	0.462
		50	0.932	0.932	0.018	0.204	0.055	0.166	0.857	0.272	0.785	0.332
	DecompX	80	0.842	0.839	0.020	0.208	0.056	0.170	0.858	0.273	0.787	0.850
		60	0.727	0.707	0.018	0.195	0.049	0.159	0.854	0.253	0.724	0.679
		50	0.641	0.593	0.017	0.187	0.045	0.152	0.853	0.236	0.686	0.533
	LLMLingua2	80	0.970	0.969	0.019	0.213	0.057	0.168	0.858	0.274	0.820	0.920
		60	0.930	0.928	0.017	0.204	0.052	0.158	0.856	0.270	0.820	0.700
		50	0.910	0.908	0.016	0.203	0.047	0.155	0.853	0.262	0.790	0.530
	SelectiveContext	80	0.930	0.929	0.019	0.232	0.067	0.186	0.856	0.312	0.810	0.830
		60	0.910	0.908	0.017	0.203	0.052	0.157	0.854	0.269	0.770	0.460
50		0.671	0.670	0.016	0.186	0.043	0.145	0.852	0.238	0.710	0.280	

Table 2: Impact of prompt-compression on proprietary models (GPT-3.5, Gemini 2.0 Flash Thinking, and o3-mini) across tasks. Base and best prompt-reduced performance for each model are highlighted in Gray and **bold** Light Blue cells, respectively.

(§D.3), and evaluation metrics (§D.4) have been reported in the Appendix §D.

4.2 Performance across Tasks

Tables 1 and 2 reveal that performance degradation becomes more pronounced as the retention threshold decreases, particularly below 60% token retention, with distinct patterns emerging across tasks. For instance, at 50% retention, sentiment classification accuracy drops by 1.5–3.2 points with GlobEnc but by 8–33 points with DecompX, commonsense QA accuracy drops by 5–16 points,

and summarization metrics such as ROUGE-1 decline by 3–15% relative to the baseline (with GPT-3.5 under GlobEnc as an outlier at roughly 40%), as seen in Figures 2 and 3. The performance decline is noticeably smaller for higher values of k , e.g., GlobEnc using 80% tokens achieves close to the baseline performance in classification and QA tasks.

Mathematical reasoning shows the steepest fall, with pass@1 scores dropping by 43–69 points at 50% retention (falling to as low as 14.6% for Llama-3 8B), reflecting the loss of critical numer-

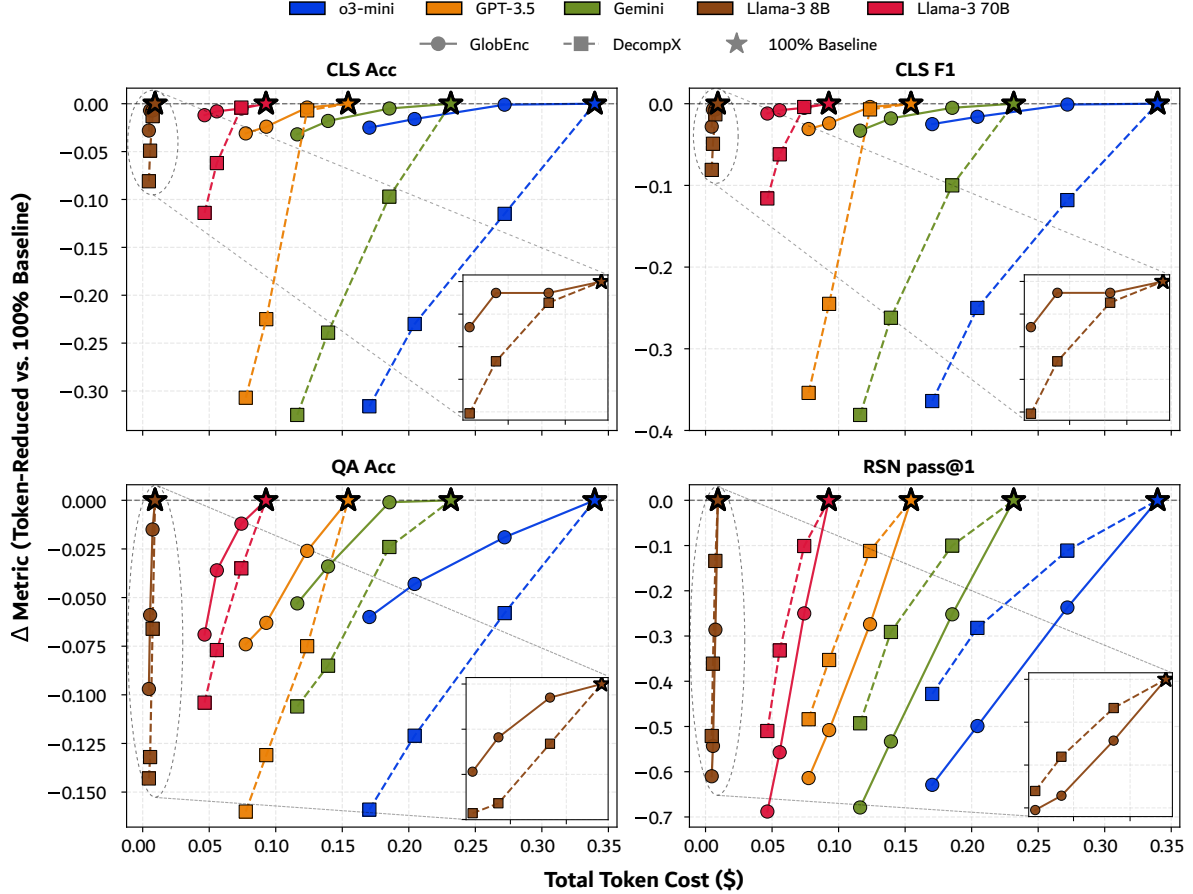


Figure 2: Performance difference between the token-reduced prompts and the 100%-token baseline, plotted against the total token cost, for the metrics of three tasks: text classification (CLS), question answering (QA), and reasoning (RSN). Insets zoom into the low-cost Llama-3 8B region.

ical and logical connectors. We hypothesize that this degradation is more pronounced for reasoning tasks because they rely on the sequential integrity of tokens and numerical data. Low $k\%$ thresholds exacerbate this by discarding essential context, amplifying errors in tasks requiring unbroken symbolic chains, while simpler classification and summarization tasks generally tolerate greater omission by inferring from salient cues.

4.3 Choice of Attribution Method

Following Tables 1 and 2 and Figure 2, for classification and QA tasks, GlobEnc outperforms DecompX in nearly all settings, with the only exception of reasoning. DecompX’s subword-level attribution, which includes linear activations, may help retain more important tokens for reasoning but might overwhelm models on other tasks. For summarization (Fig. 3), GlobEnc usually outperforms DecompX, except for the GPT-3.5 model. While DecompX is reported to produce *better* attribution

scores (Modarressi et al., 2023), most models find the token sequence filtered by attention-based aggregation, such as GlobEnc, to be more useful for retaining performance across most tasks.

4.4 Comparison with Baselines

Tables 1 and 2 also report two prompt compression baselines, LLMLingua-2 (Jiang et al., 2023b) and Selective Context (Li et al., 2023b), at matched retention ratios. For classification and QA, GlobEnc is the most robust method under aggressive compression: at 50% retention, the sentiment accuracy of both baselines falls to 0.63–0.71 on the Llama-3 models and GPT-3.5, whereas GlobEnc retains 0.92–0.94 across all five models, and GlobEnc yields the best QA accuracy in 11 of the 15 model-retention settings. Conversely, LLMLingua-2 is the strongest method for mathematical reasoning at 80% retention on all five models, and Selective Context attains the best summarization ROUGE-1 at 80% retention on the proprietary models,

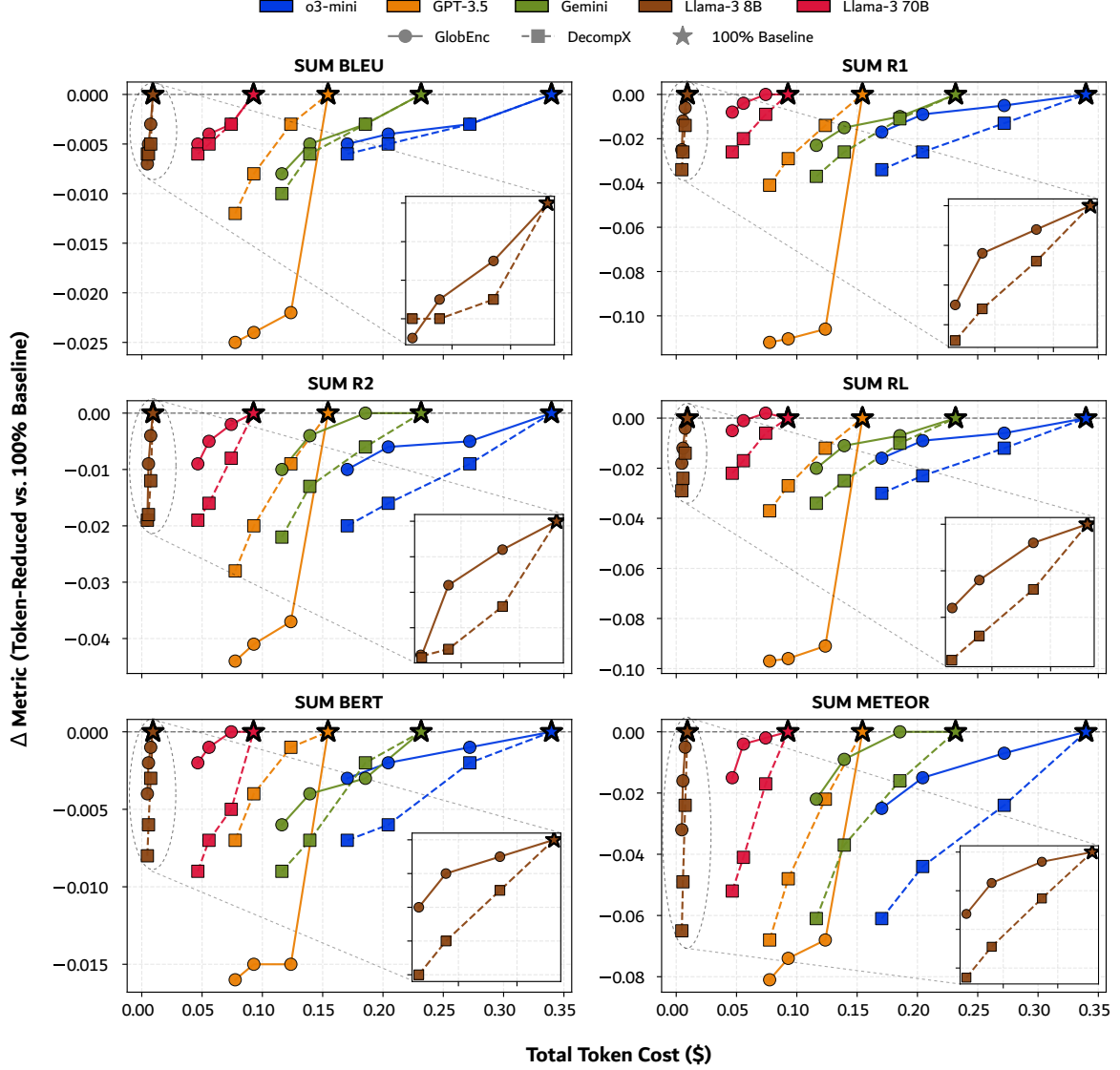


Figure 3: Performance difference between the token-reduced prompts and the 100%-token baseline, plotted against the total token cost, across the six summarization metrics. Insets zoom into the low-cost Llama-3 8B region.

while GlobEnc remains the best summarizer on the Llama-3 models. Overall, training-free saliency filtering is most advantageous for tasks whose saliency mass is concentrated in a few content tokens, whereas sequentially dependent tasks favor compressors that preserve token continuity.

4.5 Performance vs. Cost

The cost per million tokens for each model was taken from the respective provider’s website, at the time of writing, and is reported in Table 4. Note that the price is subject to change. The cost of the open-source Llama models was estimated using the lowest available rate on OpenRouter².

Figures 2 and 3 show the performance difference between the reduced set of tokens and 100%

tokens for all the models. For sentiment analysis and question answering, we observe that using a reduced set of tokens generally maintains performance and shows a gentler decline as we decrease the cost. However, reducing tokens sharply degrades performance on reasoning tasks, making token reduction unsuitable. The results are mixed for summarization tasks. While most models have a softer performance decline across the metrics, models such as GPT-3.5 using GlobEnc experience rapid performance degradation.

4.6 Performance across Models

Figures 2 and 3 show that for smaller models, token attribution offers negligible cost savings relative to the sharper performance degradation it entails.

Attribution Type	Token Attribution Method	$k\%$	Tasks									
			CLS		SUM					QA		RSN
			o3-mini		GPT-3.5					Gemini-2.0 FT		o3-mini
			Acc	F1	BLEU	R-1	R-2	R-L	BERT	METEOR	Acc	pass@1
N/A		100	0.955	0.955	0.039	0.282	0.093	0.237	0.889	0.359	0.880	0.961
Random	GlobEnc	80	0.951	0.951	0.030	0.261	0.078	0.217	0.886	0.329	0.850	0.373
		60	0.893	0.893	0.022	0.240	0.064	0.197	0.883	0.293	0.761	0.103
		50	0.886	0.886	0.020	0.223	0.053	0.182	0.880	0.267	0.713	0.058
	DecompX	80	0.942	0.942	0.033	0.259	0.080	0.218	0.887	0.328	0.837	0.381
		60	0.901	0.901	0.024	0.237	0.064	0.196	0.882	0.289	0.779	0.114
		50	0.893	0.893	0.022	0.223	0.055	0.182	0.879	0.262	0.734	0.011
Bottom	GlobEnc	80	0.914	0.914	0.026	0.234	0.067	0.195	0.883	0.289	0.764	0.040
		60	0.760	0.759	0.012	0.139	0.030	0.117	0.865	0.158	0.672	0.028
		50	0.626	0.616	0.007	0.084	0.012	0.072	0.850	0.091	0.625	0.014
	DecompX	80	0.857	0.854	0.031	0.247	0.073	0.205	0.884	0.302	0.809	0.045
		60	0.755	0.740	0.022	0.208	0.053	0.170	0.876	0.249	0.714	0.021
		50	0.696	0.666	0.020	0.192	0.047	0.156	0.873	0.225	0.670	0.014

Table 3: Impact of retaining random and bottom $k\%$ tokens using the two variants of FRUGALPROMPT with the best performing models for each task: text classification (CLS), text summarization (SUM), question answering (QA), and mathematical reasoning (RSN). Base and best performance for each attribution type are highlighted in Gray and **bold** Light Blue cells, respectively.

Model	Input (\$)	Output (\$)
Llama-3 8B	0.03	0.06
Llama-3 70B	0.30	0.40
GPT-3.5	0.50	1.50
Gemini-2.0 FT	0.10	0.40
o3-mini	1.10	4.40

Table 4: Cost per 1 million input and output tokens for each model. The Gemini 2.0 Flash Thinking (FT) model has output cost fluctuations due to the number of thinking tokens produced.

Consequently, token reduction is more practical for larger and costlier models. Legacy models such as GPT-3.5 often display abrupt drops in performance, *e.g.*, in summarization using GlobEnc, necessitating greater caution when reducing tokens.

4.7 Random & Bottom k tokens

We select random $k\%$ and the bottom $k\%$ tokens (based on attribution scores), while preserving the text order, to establish a baseline for our token reduction method. As shown in Table 3, classification, summarization, and question answering exhibit surprisingly strong retention in performance, particularly at 80% retention, where even the bottom- $k\%$ tokens retain 91.4% classification accuracy and 76–81% QA accuracy. This strongly suggests potential task contamination (Li and Flanagan, 2024), since the ground-truth answers for the test sets are readily available online. Reasoning tasks experience a far sharper decline (pass@1 be-

low 0.4 even with random 80% tokens and below 0.05 with bottom- $k\%$ tokens), indicating a genuine reliance on contextual information to solve the task.

5 Conclusion

The surging cost of LLMs has often made them inaccessible to people in under-resourced environments, while increasing carbon footprint and inference-time latency. Our work aims to bridge this gap by introducing prompt compression at the token-attribution level, thereby reducing the cost of LLM inference and making it accessible to a wider range of users. We also identify potential instances of task contamination to consider when evaluating models. We wish to explore adaptive frugalization strategies that dynamically adjust token retention based on task complexity and model confidence. Our results suggest that retention is better treated as a task-conditioned judgment than a global setting. We wish to adaptively tune token retention using task complexity and model confidence, targeting long-context and agentic regimes where verbosity exacts the highest toll. Ultimately, the question of how much language a model requires is inseparable from the question of how much meaning it can supply unaided: every token we elide is a wager on what the listener already knows, and in learning to compress our prompts, we are learning to articulate what we believe our machines can understand without being told.

Limitations

Attribution as a proxy for causal necessity. Our pipeline presumes that the salience an encoder assigns to a token is a faithful surrogate for the harm incurred by deleting that token from the prompt of a far larger autoregressive model. Assumption 1 states this premise explicitly, but it relates two distinct systems: the scores are produced by a 110M-parameter BERT-class encoder, whereas the deletion effects of interest are realized in Llama-3, GPT-3.5, Gemini 2.0 Flash Thinking, and o3-mini. We neither verify this cross-model correspondence empirically nor estimate the task constants C_τ and γ_τ . Theorem 1 and Corollary 1 should accordingly be read as a *conditional* guarantee, *i.e.*, an account of why top- k selection is the optimal one-shot deletion rule *when* faithfulness and bounded interaction hold, rather than as a certificate of stability for the models we evaluate. Wherever salience and causal necessity diverge, the ranking will discard tokens the target model genuinely requires, and the bound will remain too loose to predict the retention threshold at which compression ceases to be safe.

Reliance on task-aligned encoders. The scoring function φ_τ is task-specific by construction. FRUGALPROMPT is therefore training-free with respect to the *inference* model, in that no gradient update is applied to the LLM whose context we compress, but it is not assumption-free with respect to the *compressor*, since it presupposes an encoder whose objective is aligned with the downstream task. To the extent that such alignment is procured through task-specific supervision, our comparison against compressors that operate without that supervision should be interpreted with this asymmetry in view. The requirement also narrows applicability in open-ended, multi-task, and instruction-following regimes, where no single task label exists on which to condition attribution.

An impoverished account of redundancy. Our linguistic motivation treats function words and discourse expressions as carriers of negligible semantic weight, yet this conflates propositional content with syntactic and pragmatic function. Determiners encode definiteness, auxiliaries encode tense, aspect, and modality, negation particles invert truth conditions, and discourse markers such as “*by the way*” regulate information structure and speaker stance while contributing little propositionally. Retention in our framework is governed by a single

scalar per token, an ordering that admits no principled distinction among these levels of linguistic organization, and we do not measure separately what is lost when pragmatic scaffolding, rather than lexical content, is removed.

The contamination interpretation is suggestive, not conclusive. The performance retained under random- $k\%$ and bottom- $k\%$ selection (Table 3) is consistent with task contamination but does not isolate it. Ordinary robustness to elided context, label priors, answer-choice artifacts, residual lexical cues, and world knowledge acquired during pretraining are all plausible competing explanations, and our design cannot adjudicate among them. A decisive test would require contamination-controlled evidence, such as newly constructed or privately held test sets, or systematically perturbed variants of the same items, none of which we provide. The same ambiguity qualifies our principal findings in the opposite direction: insofar as these benchmarks are contaminated, the retention-performance curves partly reflect memorization rather than successful semantic preservation, which weakens any inference about how much meaning survives frugalization.

Scope of the empirical evaluation. Our benchmarks are standard but modest in length, and they are not the settings in which contextual overhead is most burdensome: protracted multi-turn dialogue, retrieval-augmented contexts, and agentic pipelines bearing extensive system prompts and tool specifications. The sentence-level chunking described in Appendix C admits inputs of arbitrary length, but scores each sentence in isolation, leaving cross-sentential and document-level dependencies invisible to the attribution encoder. All experiments are conducted in English, on 1000 samples per dataset, under deterministic decoding at a single temperature, so multilingual behavior and sensitivity to the decoding configuration remain uncharacterized.

Partial accounting of efficiency. Our cost analysis is denominated in provider input-token prices and thus captures only one component of end-to-end expenditure. It excludes the compute and wall-clock latency of the attribution pass, the cost of output tokens (appreciable for reasoning models that emit long deliberative traces), and the mismatch between the encoder’s tokenizer and that of the target model, which entails that retained subwords do not correspond one-to-one to billed LLM tokens. Our

appeals to inference latency and carbon footprint are therefore plausible extrapolations from token savings rather than directly measured quantities. As noted in §4, the savings are already slender for smaller models relative to the degradation incurred, and a complete end-to-end audit may well reveal a net loss in that regime.

Safety and instruction integrity. Because the ranking is indifferent to illocutionary force, compression may excise negation, quantifiers, numerical constraints, or the instruction-defining and safety-critical clauses that determine what the model is being asked to do. Our reasoning results already demonstrate the fragility of sequential and symbolic dependencies under deletion; the analogous hazard for instruction-following and safety-conditioned prompts, including negation-sensitive inputs, multi-instruction prompts, and adversarial injections, is untested here. Pending such an evaluation, we advise against deploying FRUGAL-PROMPT on prompts whose correctness or safety rests upon the integrity of every constraint.

Ethical Considerations

As identified in our work, performance retention in certain tasks might stem from the model’s exposure to benchmark data during pretraining. This raises concerns about the validity of LLM evaluation and the need for contamination-free datasets. Prompt compression can remove safety-critical tokens and should therefore be used with caution.

Acknowledgements

We convey our heartfelt gratitude to the anonymous reviewers for their constructive criticisms and insightful feedback which were conducive to the improvement of the research work outlined in this paper. We also appreciate the Systems and Software Lab (SSL) of the Islamic University of Technology (IUT) and The University of Queensland for the generous provision of computing resources during the course of this project. Syed Rifat Raiyan, in particular, wants to thank his parents, Syed Sirajul Islam and Kazi Shahana Begum, for everything.

References

Samira Abnar and Willem Zuidema. 2020. [Quantifying attention flow in transformers](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online. Association for Computational Linguistics.

Hadeer Ahmed, Issa Traore, and Sherif Saad. 2017. Detection of online fake news using n-gram analysis and machine learning techniques. In *International conference on intelligent, secure, and dependable systems in distributed and cloud environments*, pages 127–138. Springer.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. [GQA: Training generalized multi-query transformer models from multi-head checkpoints](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.

Kaiyan Chang, Songcheng Xu, Chenglong Wang, Yingfeng Luo, Tong Xiao, and Jingbo Zhu. 2024. [Efficient prompting methods for large language models: A survey](#). *Preprint*, arXiv:2404.01077.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. [Generating long sequences with sparse transformers](#). *Preprint*, arXiv:1904.10509.

Yu-Neng Chuang, Tianwei Xing, Chia-Yuan Chang, Zirui Liu, Xun Chen, and Xia Hu. 2024. [Learning to compress prompt in natural language formats](#). 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 7756–7767, Mexico City, Mexico. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. [What Does BERT Look at? An Analysis of BERT’s Attention](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023a. [QLoRA: Efficient finetuning of quantized LLMs](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefer, and Dan Alistarh. 2023b. [SpQR: A sparse-quantized representation for near-lossless LLM weight compression](#). *Preprint*, arXiv:2306.03078.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of*

- the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.
- Edward Gibson, Leon Bergen, and Steven T Piantadosi. 2013. Rational integration of noisy evidence and prior semantic expectations in sentence interpretation. *Proceedings of the National Academy of Sciences*, 110(20):8051–8056.
- Google. 2024. Introducing Gemini 2.0: Our new AI model for the agentic era. <https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/>. Accessed: 2026-09-10.
- Albert Gu and Tri Dao. 2024. [Mamba: Linear-time sequence modeling with selective state spaces](#). In *First Conference on Language Modeling*.
- Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R Bowman. 2019. Do attention heads in BERT track syntactic dependencies? *arXiv preprint arXiv:1911.12246*.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. [Cosmos QA: Machine reading comprehension with contextual commonsense reasoning](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.
- Nathaniel C Hudson, J Gregory Pauloski, Matt Baughman, Alok Kamatar, Mansi Sakarvadia, Logan Ward, Ryan Chard, André Bauer, Maksim Levental, Wenyi Wang, and 1 others. 2023. Trillion parameter ai serving infrastructure for scientific discovery: A survey and vision. In *Proceedings of the IEEE/ACM 10th International Conference on Big Data Computing, Applications and Technologies*, pages 1–10.
- Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. *arXiv preprint arXiv:1902.10186*.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023a. [LlmLingua: Compressing prompts for accelerated inference of large language models](#). In *Conference on Empirical Methods in Natural Language Processing*.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023b. [LLMLingua: Compressing prompts for accelerated inference of large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.
- Wolfgang Klein and Clive Perdue. 1997. The basic variety (or: Couldn’t natural languages be much simpler?). *Second language research*, 13(4):301–347.
- Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. [Attention is not only a weight: Analyzing transformers with vector norms](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.
- Natalia Levshina and Steven Moran. 2021. Efficiency in human languages: Corpus evidence for universal principles. *Linguistics Vanguard*, 7(s1):20200081.
- Changmao Li and Jeffrey Flanigan. 2024. Task contamination: Language models may not be few-shot anymore. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18471–18480.
- Yixiao Li, Yifan Yu, Qingru Zhang, Chen Liang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2023a. LoSparse: Structured compression of large language models based on low-rank and sparse approximation. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org.
- Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023b. [Compressing context to enhance inference efficiency of large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353, Singapore. Association for Computational Linguistics.
- Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. 2023. Sophia: A scalable stochastic second-order optimizer for language model pre-training. *arXiv preprint arXiv:2305.14342*.
- Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*.
- Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. [Learning word vectors for sentiment analysis](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.
- Meta AI. 2024. Introducing Meta Llama 3: The most capable openly available LLM to date. <https://ai>.

- meta.com/blog/meta-llama-3/. Accessed: 2026-09-10.
- Ali Modarressi, Mohsen Fayyaz, Ehsan Aghazadeh, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2023. [DecompX: Explaining transformers decisions by propagating token decomposition](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2649–2664, Toronto, Canada. Association for Computational Linguistics.
- Ali Modarressi, Mohsen Fayyaz, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2022. [GlobEnc: Quantifying global token attribution by incorporating the whole encoder layer in transformers](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 258–271, Seattle, United States. Association for Computational Linguistics.
- Christine Richards. 1990. Inferential pragmatics and the literary text. *Journal of Pragmatics*, 14(4):571–588.
- Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. 2025. Agent laboratory: Using LLM agents as research assistants. *arXiv preprint arXiv:2501.04227*.
- Claude E Shannon. 1951. Prediction and entropy of printed English. *Bell system technical journal*, 30(1):50–64.
- Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.
- Serena K Shubert, Jan H Spyridakis, Heather K Holmback, and Mary B Coney. 1995. The comprehensibility of simplified English in procedures. *Journal of technical writing and communication*, 25(4):347–369.
- Jovan Stojkovic, Esha Choukse, Chaojie Zhang, Inigo Goiri, and Josep Torrellas. 2024. Towards greener llms: Bringing energy-efficiency to the forefront of llm inference. *arXiv preprint arXiv:2403.20306*.
- Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. [Efficient transformers: A survey](#). *ACM Comput. Surv.*, 55(6).
- Inar Timiryasov and Jean-Loup Tastet. 2023. [Baby Llama: Knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty](#). In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289, Singapore. Association for Computational Linguistics.
- Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, and Mi Zhang. 2024. [Efficient large language models: A survey](#). *Transactions on Machine Learning Research*.
- EC Wit and Marie Gillette. 1999. What is linguistic redundancy. *University of Chicago*.
- Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2025. Inference scaling laws: An empirical analysis of compute-optimal inference for llm problem-solving. In *The Thirteenth International Conference on Learning Representations*.
- John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. SWE-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652.
- Yiqun Yao, Zheng Zhang, Jing Li, and Yequan Wang. 2024. [Masked structural growth for 2x faster language model pre-training](#). In *The Twelfth International Conference on Learning Representations*.

A Preliminaries: Token Attribution

To identify which tokens in an input text sequence carry significant influence on the language model’s decision-making process for a given task, we compute the token attribution scores using task-specific BERT-class encoders (Devlin et al., 2019). We obtain the saliency scores from these models by adopting two state-of-the-art (SoTA) token attribution methods: the global token attribution analysis method (GlobEnc) (Modarressi et al., 2022) and the globally propagated locally decomposed attribution analysis method (DecompX) (Modarressi et al., 2023), which are discussed in this section.

A.1 GlobEnc

GlobEnc aggregates the layerwise analysis method $\mathcal{N}_{\text{ENC}} := (\|\tilde{x}_{i \leftarrow j}\|) \in \mathbb{R}^{n \times n}$ from input token j to output token i , across all the encoder layers using attention rollout (Abnar and Zuidema, 2020). As elucidated in Kobayashi et al. (2020), $\tilde{x}_{i \leftarrow j}$ approximates the attributed vector contribution from input token j to output token i , computed as the weighted and transformed value vector $\alpha_{ij} f(x_j)$, where f is the affine transformation to value space. The vector is adjusted for residual connections and layer normalization to capture the true *mixed* influence beyond attention weights. For each encoder layer,

the output is computed as:

$$\tilde{z}_{i \leftarrow j} = g_{z_i^+} \left(\sum_{h=1}^H \alpha_{ij}^h f^h(x_j) + \mathbb{1}[i=j]x_i \right) \quad (9)$$

$$\tilde{x}_{i \leftarrow j} \approx g_{z_i^+}(\tilde{z}_{i \leftarrow j}) = \frac{\tilde{z}_{i \leftarrow j} - \text{mean}(\tilde{z}_{i \leftarrow j})}{\text{std-dev}(\tilde{z}_{i \leftarrow j})} \odot \gamma \quad (10)$$

such that $\mathcal{N}_{\text{RESLN}} := (\|\tilde{z}_{i \leftarrow j}\|) \in \mathbb{R}^{n \times n}$ is the attribution analysis to measure the influence of an encoder layer's input token j on its corresponding output token i , considering the residual connections and layer normalizations in the attention block. Here, α_{ij}^h denotes the raw attention weight from the i th token to the j th token in the attention head $h \in \{1, \dots, H\}$, collectively forming the set of attention weights $\mathbf{A}^h = \{\alpha_{ij}^h \mid 1 \leq i, j \leq n\}$ for the h th attention head and $f^h(x)$ is the transformation function as defined in Kobayashi et al. (2020). For layer l , the attention rollout with respect to the input is computed as per the following recursive function:

$$\tilde{\mathbf{A}}_l = \begin{cases} \hat{\mathbf{A}}_l \tilde{\mathbf{A}}_{l-1} & ; \text{ if } l > 1 \\ \hat{\mathbf{A}}_l & ; \text{ if } l = 1 \end{cases} \quad (11)$$

$$\hat{\mathbf{A}}_l = 0.5 \bar{\mathbf{A}}_l + 0.5 \mathbf{I} \quad (12)$$

where $\bar{\mathbf{A}}_l$ is the mean raw attention grid across all the heads relevant to the layer l .

A.2 DecompX

While GlobEnc aggregates layer-wise analyses with a modified attention rollout and approximates FFNs via residuals and scalar norms, it risks information loss. DecompX improves this by propagating the locally decomposed vectors throughout the layers to construct a global decomposition, including FFNs with linear activation approximations, and extending attributions to prediction-based values. Formally, the locally decomposed vector at layer ℓ is expressed as:

$$\tilde{z}_i^\ell = \text{LN} \left(\sum_{k=1}^N \left[\mathbf{x}_{i \leftarrow k}^\ell + \mathbf{z}_{i \leftarrow k}^\ell \right] \right), \quad (13)$$

where the normalized contribution is given by:

$$\begin{aligned} \text{LN} \left(\mathbf{z}_i^{+\ell} \right) &= \sum_{k=1}^N \underbrace{g_{z_i^{+\ell}} \left(\mathbf{z}_{i \leftarrow k}^{+\ell} \right)}_{\tilde{\mathbf{z}}_{i \leftarrow k}^\ell} + \beta \\ g_{z_i^{+\ell}} \left(\mathbf{z}_{i \leftarrow k}^{+\ell} \right) &:= \frac{\mathbf{z}_{i \leftarrow k}^{+\ell} - m \left(\mathbf{z}_{i \leftarrow k}^{+\ell} \right)}{s \left(\mathbf{z}_{i \leftarrow k}^{+\ell} \right)} \odot \gamma \end{aligned} \quad (14)$$

DecompX explicitly integrates all encoder layer components, notably the non-linear feed-forward networks, to enhance representational fidelity. The FFN operation is defined as:

$$\begin{aligned} \mathbf{z}_{\text{FFN}}^\ell &= \text{FFN}(\tilde{\mathbf{z}}_i^\ell) \\ &= f_{\text{act}} \left(\underbrace{\tilde{\mathbf{z}}_i^\ell \mathbf{W}_{\text{FFN}}^1 + \mathbf{b}_{\text{FFN}}^1}_{\zeta_i^\ell} \right) \mathbf{W}_{\text{FFN}}^2 + \mathbf{b}_{\text{FFN}}^2 \end{aligned} \quad (15)$$

$$\begin{aligned} z_{\text{FFN},i}^\ell &= f_{\text{act}}^{(\zeta_i^\ell)} \left(\sum_{k=1}^N \zeta_{i \leftarrow k}^\ell \right) W_{\text{FFN}}^2 + b_{\text{FFN}}^2 \\ &= \sum_{k=1}^N \underbrace{\theta(\zeta_i^\ell) \odot \zeta_{i \leftarrow k}^\ell + b_{\text{FFN}}^2}_{z_{\text{FFN},i \leftarrow k}^\ell} \end{aligned} \quad (16)$$

A propagation mechanism is implemented for decomposed token representations across multiple layers, preserving their distinct identities and preventing representation conflation:

$$\begin{aligned} x_i^{\ell+1} &= \text{LN} \left(\sum_{k=1}^N \underbrace{\left[\tilde{z}_{i \leftarrow k}^\ell + z_{\text{FFN},i \leftarrow k}^\ell \right]}_{z_{\text{FFN}+,i \leftarrow k}^\ell} \right) \\ &= \sum_{k=1}^N \underbrace{g_{z_{\text{FFN}+,i}^\ell} \left(z_{\text{FFN}+,i \leftarrow k}^\ell \right) + \beta}_{x_{i \leftarrow k}^{\ell+1}} \end{aligned} \quad (17)$$

The final classification mechanism uses these decomposed vectors to quantitatively assess the precise directional impact (positive or negative) of each input token on specific target classes. The established formulations within this section permit the derivation of $x_{i \leftarrow k}^{\ell+1}$ from its antecedent, $x_{i \leftarrow k}^\ell$. Through the iterative application of this procedure across the entirety of the layered architecture, the final value, $x_{i \leftarrow k}^{L+1}$, is consequently ascertained. For the task-specific classification head,

$$y_c = \sum_{k=1}^N y_{c \leftarrow k} \quad (18)$$

where the term $y_{c \leftarrow k}$ quantifies the specific contribution of the k th token towards the overall prediction score.

B Proof

The proof proceeds in three steps: (i) order the deletions arbitrarily and telescope $f(D)$ into a sum of marginals; (ii) reduce each marginal $\Delta_{d_r} f(A_{r-1})$

to a singleton marginal $\Delta_{d_r} f(\emptyset)$ by peeling tokens off A_{r-1} one at a time, accumulating at most $\gamma_\tau(r-1)$ slack via Assumption 2; and (iii) bound each singleton marginal by $C_\tau s_{d_r}$ via Assumption 1. The bound is symmetric in D , so the choice of ordering $d_1, \dots, d_q$ is immaterial.

Proof. We order $D = \{d_1, \dots, d_q\}$ and let $A_r := \{d_1, \dots, d_r\}$ be the set of tokens deleted after r steps and $A_0 = \emptyset$. For each r and $u \in A_{r-1}$,

$$\begin{aligned} \Delta_{d_r, u} f(A_{r-1} \setminus \{u\}) &\leq \gamma_\tau & [\text{Eq. 5}] \\ \implies \Delta_{d_r} \Delta_u f(A_{r-1} \setminus \{u\}) &\leq \gamma_\tau & [\text{Eq. 3}] \\ \implies \Delta_{d_r} f(A_{r-1}) - \Delta_{d_r} f(A_{r-1} \setminus \{u\}) &\leq \gamma_\tau \\ \implies \Delta_{d_r} f(A_{r-1}) &\leq \Delta_{d_r} f(A_{r-1} \setminus \{u\}) + \gamma_\tau. \end{aligned}$$

Iterating over all $u \in A_{r-1}$ gives

$$\begin{aligned} \Delta_{d_r} f(A_{r-1}) &\leq \Delta_{d_r} f(\emptyset) + \gamma_\tau(r-1) \\ &= f(\{d_r\}) - f(\emptyset) + \gamma_\tau(r-1) \\ &= f(\{d_r\}) + \gamma_\tau(r-1) \\ &\leq C_\tau s_{d_r} + \gamma_\tau(r-1). \end{aligned} \quad [\text{Eq. 4}]$$

Formulating the telescopic sum:

$$\begin{aligned} f(D) = f(A_q) &= \sum_{r=1}^q (f(A_r) - f(A_{r-1})) \\ &= \sum_{r=1}^q \Delta_{d_r} f(A_{r-1}) & [\text{Eq. 2}] \\ &\leq \sum_{r=1}^q C_\tau s_{d_r} + \gamma_\tau \sum_{r=1}^q (r-1). \end{aligned}$$

$$\text{Hence, } f(D) \leq C_\tau \sum_{i \in D} s_i + \frac{\gamma_\tau}{2} q(q-1). \quad \square$$

C On Handling Long Inputs

The token-attribution backbones used by FRUGALPROMPT are BERT/RobERTa-style encoders, whose self-attention cost and learned positional embeddings impose a fixed maximum input length (512 subword tokens in the models used here). Rather than feeding the entire prompt or document to the attribution encoder at once, we use a sentence-level chunking strategy. Given an input document $x = (w_1, \dots, w_n)$, we first segment it into sentences $x = \{s_1, \dots, s_m\}$ using a sentence tokenizer. Each sentence s_i is then tokenized independently with the corresponding BERT/RobERTa tokenizer and passed through the attribution model as a separate encoder input. This ensures that the

encoder never receives the full long document as a single sequence.

For DecompX, each sentence-level input is encoded with tokenizer-level truncation enabled, so any pathological sentence that exceeds the encoder limit is clipped to the model’s maximum sequence length before attribution is computed. DecompX then decomposes the classifier output over the subword tokens of that sentence. Let $z_{i,j}$ denote the j -th subword token in sentence s_i and let $a_{i,j}^{\text{Dec}}$ be its DecompX attribution score. The scores are min-max normalized within the sentence-level forward pass:

$$\tilde{a}_{i,j}^{\text{Dec}} = \frac{a_{i,j}^{\text{Dec}} - \min_k a_{i,k}^{\text{Dec}}}{\max_k a_{i,k}^{\text{Dec}} - \min_k a_{i,k}^{\text{Dec}} + \epsilon},$$

where ϵ is a small constant for numerical stability.

For GlobEnc, each sentence is likewise processed independently. The model returns layer-wise attention/norm information for the sentence-level input, and GlobEnc computes the rollout flow from the classification token to the input subwords. If R_i is the rollout matrix for sentence s_i , the token score is computed from the classification-token row of the flipped layer-wise rollout and averaged across layers:

$$\tilde{a}_{i,j}^{\text{Glob}} = \frac{1}{L} \sum_{\ell=1}^L \frac{R_{i, [\text{CLS}], j}^{(\ell)}}{\max_k R_{i, [\text{CLS}], k}^{(\ell)} + \epsilon}.$$

After sentence-level attribution, the token-score pairs from all sentences are concatenated back in their original document order. The frugalized prompt is formed by ranking tokens globally by their attribution scores, retaining the top $k\%$ of tokens, and restoring the retained tokens to their original order. Optionally, a sentence-importance weight λ_i (e.g., LexRank) can be applied before aggregation, yielding $\hat{a}_{i,j} = \lambda_i \tilde{a}_{i,j}$, although the core long-context handling remains the same. This procedure avoids document-level truncation while preserving local syntactic context for attribution and permitting arbitrary-length prompts to be scored by fixed-window BERT/RobERTa encoders.

D Experimental Details

D.1 Tasks

We evaluate the efficacy of the token reduction by experimenting across four NLP tasks: sentiment analysis (text classification), summarization

(text generation), commonsense question answering, and mathematical reasoning. These tasks span both discriminative and generative paradigms, enabling evaluation of the LLM’s linguistic and cognitive demands under diverse conditions.

D.2 Dataset

For sentiment analysis, we use the IMDb movie review dataset (Maas et al., 2011), a binary sentiment analysis corpus comprising movie reviews labeled positive or negative. We use Argilla News to evaluate abstractive summarization on news articles (Ahmed et al., 2017). For commonsense QA, we use CosmosQA (Huang et al., 2019), which primarily focuses on commonsensical and causal reasoning within short narrative passages. The questions stipulated in CosmosQA are easy for humans to solve, but usually challenging for LLMs. We use GSM8k (Cobbe et al., 2021) for mathematical reasoning, which contains grade-school math word problems (MWPs) to assess basic arithmetic reasoning and multi-step logical deduction. For evaluating our method, we take 1000 random samples from each dataset.

D.3 Models

We perform inference using a set of contemporary frontier models, varying in architecture and scale, to examine the generalizability of our method. The models are: Llama-3 8B, Llama-3 70B (Meta AI, 2024), GPT-3.5 (?), Gemini 2.0 Flash Thinking (Google, 2024), and o3-mini (?). The closed-source models are accessed via their respective APIs, and the open-source models are accessed via OpenRouter², using standardized prompts to minimize formatting-induced variability.

D.4 Evaluation Metrics

We employ task-specific metrics to evaluate performance across both full and frugalized setups. For sentiment classification, we report accuracy and F1 score. Summarization performance is assessed using BLEU, ROUGE-1, ROUGE-2, ROUGE-L, BERTScore, and METEOR to capture both superficial and semantic quality of the responses. For commonsense question answering, we use accuracy as the dataset follows a multiple-choice format. Mathematical reasoning is evaluated using pass@1, which measures the proportion of problems correctly solved on the first attempt.

²<https://openrouter.ai/>