# Short-Context Dominance: How Much Local Context Natural Language Actually Needs?

**Vala Vakilian**[1] **Zimeng Wang**[1] **Ankit Singh Rawat**[2] **Christos Thrampoulidis**[1]

[1]University of British Columbia [2]Google DeepMind

## Abstract

We investigate the short-context dominance hypothesis: that for most sequences, a short contiguous prefix of the most recent tokens suffices to predict their next tokens. Using large language models as statistical oracles, we measure the minimum context length (MCL) needed to reproduce accurate full-context predictions across datasets with sequences of varying lengths. For sequences with 1–7k tokens from long-context documents, we consistently find that 75–80% require only the last 96 tokens at most. Given the dominance of short-context tokens, we then ask whether it is possible to detect challenging long-context sequences for which a short local prefix does not suffice for prediction. We introduce a practical proxy to MCL, called Distributionally Aware MCL (DaMCL), that does not require knowledge of the actual next-token and is compatible with sampling strategies beyond greedy decoding. Our experiments validate that simple thresholding of the metric defining DaMCL achieves high performance in detecting long vs. short context sequences. Finally, to counter the bias that short-context dominance induces in LLM output distributions, we develop an intuitive decoding algorithm that leverages our detector to identify and boost tokens that are long-range-relevant. Across Q&A tasks and model architectures, we confirm that mitigating the bias improves performance.

## 1 Introduction

When prompted to continue a piece of text, how much preceding context do humans rely on? Do they focus on recent words and local coherence, or plan with a broader, narrative-wide perspective? For example, when writing a story, do they recall events from earlier chapters or rely mostly on recent developments? While it is difficult to study these questions rigorously in humans, large language models (LLMs) offer a tractable experimental analogue. This leads us to ask: *How far back must a model look to predict the next token accurately?* Although modern transformer-based LLMs can attend to thousands of tokens [1, 2], it remains unclear and unquantified how often that capacity is used at inference time, and how often predictions depend on distant information or primarily on local spans.

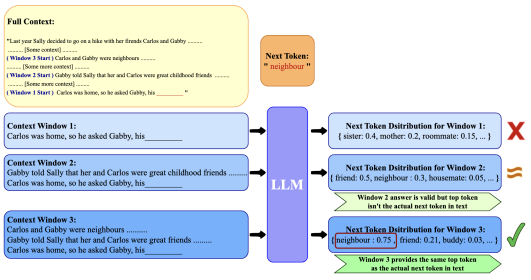



Figure 1: **Short-context dominance hypothesis.** In this example, with context 1 the model fails to predict the ground-truth "neighbour," Context 2 produces a semantically valid alternative "friend," and when using Context 3, the model correctly predicts "neighbour." Our work **(1)** systematically validates the hypothesis, **(2)** develops methods to detect when longer context is truly needed, and **(3)** leverages these insights to improve model decoding by correcting for short-context bias.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Correspondence: vaalaa@student.ubc.ca Code: github.com/vala-vakilian/ShortContextDominance

[2]Google DeepMind, New York, USA; Email: ankitsrawat@google.com (Participated in an advisory capacity only.)

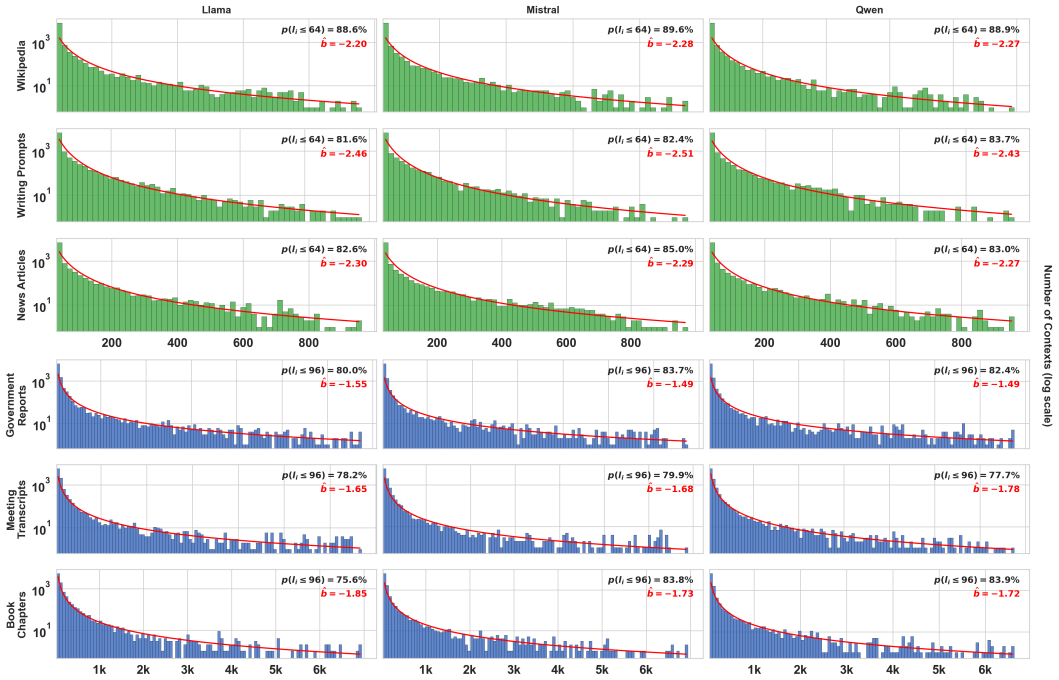



Figure 2: **Distribution of MCL:** Minimum context window needed to confidently predict the next token across sampled sequences, six datasets, and three LLMs. $\hat{b}$ denotes the slope of the log-log fit. Blue/green distinguishes long/short documents($\geq$ 6k/$\geq$ 1k tokens)

We posit the *short-context dominance hypothesis*: for the majority of natural-language sequences, the information required to accurately predict a valid next token is contained within a *short*, *local*, contiguous prefix, i.e., the last $k \ll n$ tokens, where $n$ is the full-context length of the sequence. We study this hypothesis in three steps: we *measure* how much context individual sequences require, *detect* the minority of sequences that genuinely depend on distant information, and *correct* for the bias this imbalance induces in LLMs.

**Measuring natural-language local context dependency.** To test our hypothesis, we introduce **Minimal Context Length (MCL)**, which quantifies how much local context suffices for a language model, used as oracle, to confidently and correctly predict the ground-truth next token of a sequence. We sample sequences of varying lengths from various sources and filter for cases where an LLM correctly predicts the actual next token in the corpus, aiming to mitigate confounding effects of LLM limitations. To quantify MCL, we iteratively increase the prefix length of each sequence until the model confidently outputs the ground-truth next token. We find that 75-80% of sequences, irrespective of total length, rely on their 32-96 last tokens, supporting our hypothesis. See Fig. 2.

**Practical long-context detection.** Since relying on a short prefix is appropriate for most sequences, any correction of short-context bias must be targeted, which first requires detecting the minority of sequences that need long context. We develop a practical and novel variant of MCL, the **Distributionally Aware Minimum Context Length (DaMCL)**, that does *not require ground-truth knowledge, confidence or correctness* and instead uses *the next token distributions* for detection, making it compatible with sampling strategies beyond greedy decoding. We validate that DaMCL remains consistent with the short-context dominance hypothesis, although to a lesser extent. Importantly, we show that it enables accurate classification of sequences as *short-context* (where a 32-64 token prefix suffices) or *long-context* (requiring longer context). Since our detection operates without ground-truth information, it is also applicable during inference unlike previous methods for context length analysis which rely on perplexity and ground truth probability [3–6].

**Post-hoc short-context dominance correction.** Having a long-context detector, we finally test a hypothesis that short-context dominance induces a corresponding bias in LLMs themselves: Since sequences predominantly require only short context, models are implicitly trained on distributions heavily skewed toward local dependencies, potentially biasing them toward common completions

predictable from short local context and undermining the model's performance on inherently long-context sequences [7–9]. We validate such a short-context bias by applying TaBoo (**Ta**rgeted **Boo**sting), an intuitive decoding algorithm that counteracts the skew: for sequences detected as long-context, TaBoo identifies and boosts tokens that are long-context-relevant. On Q&A and summarization datasets with inherent long-context dependencies, TaBoo consistently outperforms vanilla nucleus sampling and is competitive with relevant logit-adjustment methods.

## 2 Related Work

We summarize the most closely related works below (defer detailed discussion to Appx. B).

**Long-context utilization analysis.** Prior work has shown that language models often underutilize available context: extending the input prefix or preserving distant tokens does not always yield corresponding improvements in prediction or downstream performance [5, 3, 10]. These studies typically characterize context use through aggregate quantities, such as average perplexity changes under truncation or perturbation of local and distant tokens. Such metrics are useful for measuring average sensitivity to context, but they do not directly assign a minimal context requirement to each individual sequence/next-token pair. We illustrate this distinction in Appx. B and Fig. 8. These observations have motivated algorithmic interventions at both the system and data levels to encourage models to use long-range information more effectively [11–15]. More closely related recent work detects or weights tokens according to their long-range relevance for long-context training or context extension [4, 6, 16]; however, their definition of a long context focuses on sequences whose dependencies extend beyond 8K tokens. Our work is complementary but asks a different question. Rather than comparing fixed short- and long-context conditions, we estimate the minimal suffix length sufficient to recover the relevant full-context behavior for each sequence. Moreover, while token-relevance methods typically rely on the probability assigned to the observed ground-truth token, our distributional formulation compares the full next-token distributions induced by short and full contexts. This removes the need for ground-truth next tokens and makes the resulting detector applicable during inference.

**Contrastive decoding.** Several inference-time methods address hallucination in long-context generation through contrastive approaches [17–19]. Building on [20, 21], works such as [8, 9] and [22] reweight token probabilities by contrasting next token distributions in the absence of relevant text. CAD [9] and AdaCAD [23] are most closely related to our TaBoo algorithm, but differ fundamentally in both motivation and implementation. Focusing on hallucination, for Q/A or summarization tasks, they contrast next token distributions in the absence of relevant documents to identify non-contextual biases. Our algorithm stems from the short-context dominance hypothesis and attempts to combat locality bias. TaBoo detects long context sequences by contrasting local to full context distributions. Once such sequences are detected, *it applies targeted adjustments to their long-range-relevant tokens only*. We provide additional discussion of contrastive methods in the appendix. Beyond distinctions specific to TaBoo, we systematically study minimal context length requirements in natural language and our methods could apply beyond inference-time, for example to identify long-context relevant sequence during pre-training or fine-tuning.

**n-gram models.** Recent work has revisited n-gram models as useful complements to neural language models [24–26]. These results show that surface-level histories remain highly predictive for next-token. In particular, Infini-gram demonstrates that n-gram methods can scale beyond small fixed orders by retrieving arbitrarily long suffix matches from large corpora. Our work asks a complementary question. Whereas n-gram methods predict from exact suffix recurrence and backoff using an external corpus, MCL and DaMCL measure how much of the current input context is needed to recover a modern LLM's own full-context prediction on each individual sequence. This distinction separates surface recurrence from context dependence: a token may be predictable from a recurring suffix without implying that earlier parts of the current context are irrelevant, and conversely a model's full-context prediction may depend on information that is not captured by exact suffix overlap. Thus, n-gram results provide indirect evidence for the usefulness of local context, while our framework directly quantifies when an LLM's full-context behavior is already determined by a short suffix and when longer-range context changes that behavior.

## 3 Least Context for Prediction

In this section, we answer the following question: *For a given randomly sampled context and next-token, what is the minimum sub-context needed to predict the actual next token correctly?*

### 3.1 Minimal Context Length

We isolate sequences $\mathbf{s}$ for which the LLM, using greedy decoding, **correctly** and **confidently** predicts the actual next token $t$ in the corpus. By focusing on these high-quality predictions, we approximate using the LLM as a statistical oracle to study context dependency. We define MCL as follows.

**Definition 1.** *The **Minimal Context Length** (MCL) of sequence* $\mathbf{s}$ *given its next token* $t$ *is the length* $\ell$ *of the shortest prefix* $\boldsymbol{s}_{[-\ell:]}$ *so that the model output given the prefix is correct and confident:*

$$\mathsf{MCL}\left(\mathbf{s}|t\right) := \arg\min_{l \in |\mathbf{s}|} \left\{ l \;\middle|\; \mathsf{Top}_1(\mathbf{s}_{[-l:]}) = t, \;\; \Delta\mathsf{Conf}(\mathbf{s}_{[-l:]}) \geq \delta \right\}. \tag{1}$$

Here, $\mathbf{s}_{[-\ell:]}$ denotes the prefix of the last $\ell$ tokens of sequence $\mathbf{s}$ of length $|\mathbf{s}|$; $\mathsf{Top}_1(\cdot)$ returns the token with highest model output probability given the input sequence; and $\Delta\mathsf{Conf}(\cdot)$ returns the gap in probability between the top-ranked and second-best token, which must exceed confidence threshold $\delta \in [0, 1]$. For concreteness in our experiments, we set $\delta = 0.2$ and provide ablations in Appx. D.1. By definition, a larger MCL implies that the model requires information from earlier in the context to predict the next token correctly, while smaller MCL indicates greater reliance on local context. Note that since MCL searches over contiguous prefixes, it measures the distance back to the earliest token needed for the prediction, not the number of tokens the model actually relies on: an MCL of 128 means the required information lies within the last 128 tokens, not that all 128 are used. Depending on confidence, MCL is applicable to 40-60% of context/next-tokens. This is a natural limitation when relying on ground truth tokens (similar to [4]), one which we address in Sec.4 and further discuss in Appx. C.

### 3.2 Experimental Setup

**Datasets.** We experiment on the following natural-language documents. Short documents: Reddit *Writing Prompts* [27], CNN/DailyMail *News Articles* [28, 29] and *Wikipedia* articles from WikiText-103 [30]. Long documents: U.S. *Government Reports* from GovReport [31], *Meeting Transcripts* from QMSum [32] and story *Book Chapters* BookSum [33]. Much of these datasets appear in the LongBench [34] benchmarks and are believed to have inherent long-context qualities which are the target of our studies. We further experiment with documents of different languages using [35] and datasets specialized for math and medical domains (Appx. D.2).

**LLM Oracles.** We evaluate three open-weight models: LLaMA-3-8B [36], Mistral-7B-Instruct (v0.3) [37] and Qwen3-8B [38]. We assume that these models are sufficiently capable to exhibit reliable performance on next-token prediction and question answering tasks. Recall also that we isolate sequences where the models confidently predict the next token. All experiments are performed on a V100 Nvidia GPU with 32GB of memory.

**Choice of sequences.** We form sequences $\boldsymbol{s}$ by parsing documents from the datasets. For short documents we sample 100 unique documents of length at least 1k and only keep their first 1k tokens. For long documents we sample documents of lengths $n \in [6, 7]$k, in order to focus our analysis on long-context inputs. We sample 100 sequences $\boldsymbol{s}$ and their ground-truth next-token $t$ from each document. We filter for sequences with correct and confident predictions. We also make sure to avoid biases toward either shorter or longer sequences. Concretely, for our 1k token length windows ($0 - 1$k for regular documents and $6 - 7$k for long context data), we ensure to sample the same number of sequences from sizes $[32, 100], [100, 200], \cdots, [900, 1000]$. Overall, this yields 10k unique sequences (100 sequences × 100 documents) and their respective next tokens for each dataset.

**MCL Algorithm.** To determine the MCL as per Defn. 1, we evaluate a model's predictions using increasing prefix sizes $l \in \{32, 48, 64, \ldots, |\mathbf{s}|\}$, starting from 32 tokens and incrementing by 16. For the longer documents, we start from 32 and increment by 64. Starting from 32 tokens is to differentiate from any n-gram analysis as the highest achievable n-gram approximation model is currently limited to 32 tokens [25]. For each window size, we evaluate the model's output distribution and stop once it confidently predicts the ground-truth next token. In practice, for the sub context window, we cut the sequence to provide only the last $l$ tokens to generate the next-token probabilities.

### 3.3 Results and Discussion

Fig. 2 shows that the distribution of $\mathsf{MCL}\,(\mathbf{s}|t)$ as defined in Equation 1 is highly skewed (note the histogram y-axis in log scale), indicating that the model requires only the last $32 - 96$ tokens for the majority of sequences ($\sim$ 80–90%) to confidently predict the next token ($\mathsf{MCL}\,(\mathbf{s}|t) \leq 32$ or 96). This observation formally confirms that **for majority of individual queries, the LLM only needs highly localized information from the context to correctly and confidently predict the token**.

To quantify short-context reliance at finer granularity, we examine the power-law exponent $\hat{b}$ by fitting $y = a \cdot x^{-b}$ in log-log space. For shorter documents, we observe values of $\hat{b} \in [-2.5, -2]$, and for longer documents, $\hat{b}$ falls in the range $[-2, -1.5]$. Both ranges indicate strong power-law decay, demonstrating short-context sufficiency even in inherently long-context datasets such as *Government Reports*, *Meeting Transcripts*, and *Book Chapters*. We further validate these findings across linguistic and domain variations through experiments detailed in Appx. D.2 and provide additional results regarding the implications of a token's position in word (Appx. D.3) and its part-of-speech label (Appx. D.4). **Short-context dominance persists across all of our experimental setups**. It also holds for larger models (Qwen3-14B, Mistral-Small-24B; Appx. H) and on synthetic novel documents that cannot have been memorized (Appx. I).

Validating the short-context dominance hypothesis carries implications for both pretraining and evaluation: Since most predictions require only highly localized information, then standard training objectives and perplexity-based evaluations are inherently skewed toward short-range dependencies, potentially obscuring true progress on long-context reasoning. This observation on individual contexts aligns with previous findings that suggest average perplexity does not improve noticeably beyond a certain prefix size [5, 3] . It also reinforces recent critiques [4] that token-level perplexity is an insufficient metric for long-context evaluation—even on datasets with genuine long context dependencies, such as *Government Reports*.

## 4 Distributional Awareness

MCL evaluates whether the ground-truth next token from the dataset can be predicted with a shorter prefix. Thus, **1)** it is similar to [4], which is limited to cases where the model performs well on a target context/next token and more importantly **2)** *neglects how language prediction and language itself permits multiple possible valid next tokens*. A language model may assign higher probability to plausible alternatives that differ from the ground-truth token in the dataset. Focusing on probability of ground truth token [6, 16, 4] and average perplexity [5, 3] neglects this important property of language and language modeling. Finally note that Definition 1 is limited to greedy decoding, while popular natural language generation methods often rely on sampling strategies that draw from multiple probable tokens [39–41]. These considerations (see also Appx. E.1) motivate us to introduce here a more flexible formulation for context length analysis.

### 4.1 Distribution Aware MCL

Consider decoding strategy $\phi$ that modifies the model's raw probability distribution. For example, nucleus sampling [39] which selects a subset of tokens whose cumulative probability mass reaches threshold $p$ (e.g. $p = 0.9$), then produces a renormalized distribution $\mathbf{p}_\phi(\mathbf{s})$ with support on this subset and zero probability to tokens otherwise. Such strategies are often motivated in generation as they prevent monotonous text (compared to a greedy setup) while controlling possible noise by clipping irrelevant tokens. More generally, our framework accommodates any decoding strategy $\phi$ that produces a probability distribution $\mathbf{p}_\phi(\mathbf{s})$. We introduce DaMCL to find the smallest prefix $\mathbf{s}_{[-\ell:]}$ such that $\mathbf{p}_\phi(\mathbf{s}_{[-\ell:]})$ is sufficiently similar to $\mathbf{p}_\phi(\mathbf{s})$.

**Definition 2.** *The **Distribution-aware Minimal Context Length** (DaMCL) of a sequence* $\mathbf{s}$*, given decoding strategy* $\phi$*, a probability-similarity metric* $\mathcal{M}$*, and a similarity threshold* $\epsilon$*, is defined as the length of the shortest prefix for which the decoding-based next-token distribution given the prefix* $\mathbf{p}_\phi(\mathbf{s}_{[-\ell:]})$ *lies within an* $\epsilon$*-neighborhood of the full-context distribution* $\mathbf{p}_\phi(\mathbf{s})$*. Formally:*

$$\mathsf{DaMCL}_\phi^{\mathcal{M}}\,(\mathbf{s}, \epsilon) := \arg\min_{l \in |\mathbf{s}|} \left\{ l \;\middle|\; \mathcal{M}(\mathbf{p}_\phi(\mathbf{s}_{[-l:]})\,;\, \mathbf{p}_\phi(\mathbf{s})) \leq \epsilon \right\} .$$

While this definition accommodates any distance metric $\mathcal{M}$, we specifically use the Jensen-Shannon Distance (JSD) throughout our experiments. For distributions $\mathbf{p}_1, \mathbf{p}_2$ in the $|\mathcal{V}|$-dimensional simplex,

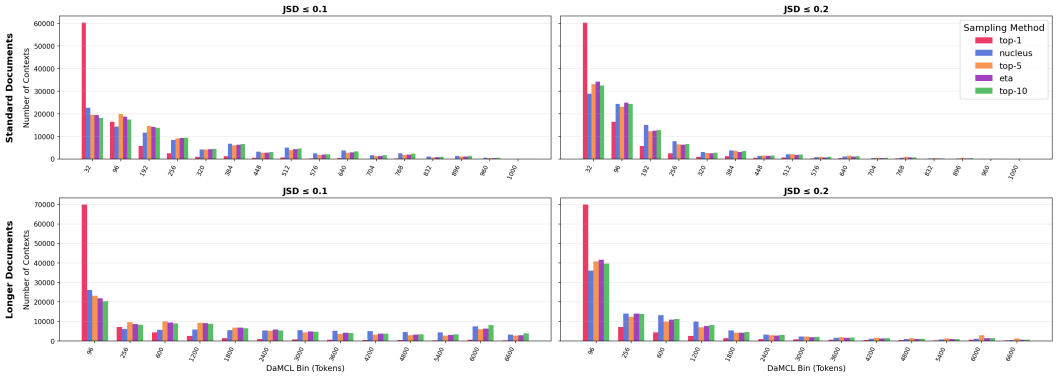



Figure 3: **Distribution of DaMCL:** DaMCL measurements for various sampling strategies and relative thresholds. Results are shown separately for short- (top row) and long- (bottom row) documents to highlight potential differences in behavior. While the overall trend resembles the heavy-tailed decaying pattern observed in standard MCL (see Fig. 2), the choice of threshold influences the outcome. Each subplot reflects results aggregated over all model–dataset combinations, as no major deviations were observed across different configurations under identical hyperparameters.

JSD is defined as $\text{JSD}(\mathbf{p}_1 \,;\, \mathbf{p}_2) := \sqrt{\frac{1}{2}\text{KL}(\mathbf{p}_1\|\mathbf{q}) + \frac{1}{2}\text{KL}(\mathbf{p}_2\|\mathbf{q})}$, where $\mathbf{q} := \frac{1}{2}(\mathbf{p}_1 + \mathbf{p}_2)$, and $\text{KL}(\mathbf{p}_1\|\mathbf{p}_2) := \sum_{t\in\mathcal{V}}[\mathbf{p}_1]_t \log \frac{[\mathbf{p}_1]_t}{[\mathbf{p}_2]_t}$ is the Kullback-Leibler divergence. We choose JSD because it is a proper distance metric satisfying the triangle inequality and is widely employed in knowledge distillation [42] and language model evaluation [43]. *See Fig. 17 in appendix* which illustrates how JSD distribution shift can be used to identify sequences with long context relevance. Compared to MCL, the above definition of DaMCL **1)** relies on measuring the difference between distributions as opposed to a single token, **2)** does not require information regarding the ground-truth next token of a given sequence $\mathbf{s}$ and **3)** can be defined on all context/next-tokens pairs.

### 4.2 Experimental Setup

**DaMCL evaluation.** We evaluate DaMCL over several decoding strategies: Top-$K$ sampling ($K$=1 for greedy and $K$=5, 10 similar to [44, 27]) , nucleus sampling with $p$=0.9 [39], and truncation ($\eta-$) sampling with $\epsilon$=0.0009 [41]. We evaluate two threshold values JSD $\leq 0.1$ and JSD $\leq 0.2$ to study the effect of varying similarity criteria. Lower thresholds correspond to stricter conditions for accepting a local prefix's next token distribution as equivalent to that of the full context. We provide additional analysis with other distance metrics (TVD, 1-F1, KL) and thresholds in Appx. E.3. For standard vs longer documents, the length of sampled contexts is similar to Sec. 3.3. For standard documents we increment the windows by 64 tokens while for the longer docs, we include windows of size 96, 256 and multiples of 300.

### 4.3 Results and Discussion

To begin with, Fig. 3 for JSD $\leq 0.2$ reveals a similar bias toward short context as with MCL. While top-1 shows a similar behavior when compared to MCL, for the distributional cases (top-5, top-10, top$p$ and $\eta$-sampling), the drop in DaMCL is not as dramatic as observed when using MCL. As these methods spread probability mass across a wider support set, the resulting distributions become smoother and more diffuse, making convergence under tight JSD thresholds more difficult. Additionally, except for top-1 (Greedy) which is very similar to MCL, all decoding strategies more or less show a similar trend in the DaMCL bin distributions, thus illustrating how the phenomenon seems to be agnostic to our target sam-

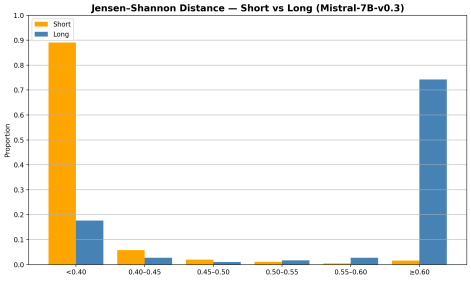



Figure 4: Distribution of LSDS ($\mathbf{s}$) for short-/long-context sequences for Mistral-7B-v0.3 on controlled validation setup.

pling strategies. The decay in the DaMCL window trends are less extreme compared to MCL, but this is an artifact of noise when comparing next token distributions (with vocabulary sizes as high as 128K) rather than focusing on the single ground truth next token ( also note that later bins have larger increments, done to reduce computation) . When enforcing stricter similarity requirements ($\text{JSD} \leq 0.1$) the DaMCL decay count for longer context bins are less severe. Setting the threshold value would depend on how strictly we want to enforce distribution similarity. Given the aforementioned noise even after applying decoding methods, we believe a less-strict threshold $\epsilon = 0.2$ is more reasonable. We provide further analysis of DaMCL in Appx but for the remainder of the paper, we stick to nucleus sampling and JSD.

# 5 Long-Context Sequence Detection

We have seen that for the majority of sequences a valid next-token can be generated with access to only a short local prefix. Specifically, in Sec. 4, to avoid the need for knowing the actual next-token of a sequence, we introduced the idea of quantifying whether a prefix is a good proxy of the full-context by evaluating the JSD of the respective model output probabilities. Here, we build on these insights to develop a long-context detector that can distinguish between *short-context sequences*, where a short prefix of fixed length suffices, and *long-context sequences*, which require longer context information.

## 5.1 Distributional-aware long-context sequence detection

To develop our long-context detector, we first define the following metric. This metric, while derived from DaMCL, has a simpler form with focus on only differentiating between the short vs longer context in a binary setup.

**Definition 3.** *The **Long-Short Distribution Shift (LSDS)** of sequence* $\mathbf{s}$ *is the JSD between the next-token distributions obtained with decoding strategy* $\phi$ *when given a short prefix of length* 32 *versus the full context. Formally,* $\text{LSDS}(\mathbf{s}) = \text{JSD}(\mathbf{p}_\phi(\mathbf{s}_{[-32:]}), \mathbf{p}_\phi(\mathbf{s}))$ .

For concreteness, unless otherwise stated, we fix $\phi$ to nucleus sampling with parameter $p = 0.9$ [39]. (Ablation results on different metrics, $p$ or short-context lengths are provided in Appx. F.2 and Appx. F.3.) Finally, we fix the prefix length to 32 based on our findings in Sec. 3.

**Controlled validation.** To demonstrate that LSDS can effectively detect long-context sequences, we conduct a controlled needle-in-a-haystack experiment adapted from [45]. We create short and long queries, such that only in the former the answer appears in the final 32 tokens; see Appx. F.1.1 for details and example illustration. Figure 4 shows the resulting LSDS distributions for Mistral-7B-v0.3. We observe a clear separation between categories despite no ground-truth next-token information: short-context sequences concentrate at low LSDS values ($\leq 0.4$), while long-context sequences dominate high LSDS values ($\geq 0.7$). Similar patterns hold for LLaMA and Qwen Appx. F.1.1).

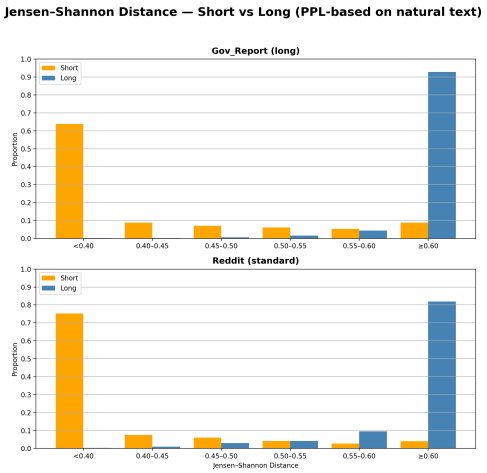



Figure 5: Distribution of $\text{LSDS}(\mathbf{s})$ for short- and long-contexts on GovReport (long doc) and Reddit Writing Prompts (short doc) on Mistral-7B-v0.3. Illustrates a clear distinction between long and short contexts.

**Long-context detector.** Based on these findings, we define our long-context detector as a simple thresholding classifier: $\text{LSDS}(\mathbf{s}) \gtrless_{\text{short}}^{\text{long}} \tau$ where $\tau$ is a threshold determined from validation data or prefixed. Next, we analyze the behavior of the detector on natural text where no benchmark datasets exist with labeled contexts and their minimal length. Thus, we will be using other metrics with knowledge of actual next token, such as MCL, to establish ground truth labels for context length.

### 5.2 Evaluation on Natural Text

In natural text, determining the true context dependency requires an oracle with access to the actual next token. We employ two such oracles to establish ground-truth short-context vs long-context labels for sequences. (1) *MCL Oracle:* For $\mathbf{s}$ with next-token $t$, we classify $\mathbf{s}$ as long-context iff $\mathsf{MCL}(\mathbf{s}|t) > 32$ (see Defn. 1). (2) *LSD Oracle:* Following [4], we classify $\mathbf{s}$ as long-context iff $\mathrm{LSD}(\mathbf{s}|t) > 2$ & $\mathrm{LCL}(\mathbf{s}|t) \geq -1$ (details in Appx. F.1.4 ). Both oracles require knowledge of ground-truth next token and can be evaluated on a subset of contexts with which the model is confident on. We show that LSDS-based classification agrees well with these oracles despite not knowing the ground-truth.

**Results.** Figure 5 demonstrates strong agreement between LSDS and the LSD Oracle across *Government Reports* and *Reddit Writing Prompts* datasets. Using $\tau = 0.6$, we find consistently $80 - 90\%$ of oracle-labeled long-context sequences being classified by LSDS as long-context, while fewer than $5 - 10\%$ of short-context sequences are mislabeled. This pattern holds consistently across LLaMA-3-8B/Mistral-7B-v0.3/Qwen-2-7B and both oracles (Appx. F.1), verifying LSDS as a reliable proxy for short vs long context detection.

**Threshold robustness.** Through extensive ablations referred to Appx. F.1, we find that threshold choices are robust across models and datasets. While a fixed threshold works well generally, task-specific tuning can optimize precision-recall trade-offs. Quantitative results across models and datasets are summarized in Table 6.

**Ablation on short-context length.** We test prefixes-lengths $\{8, 16, 32, 64\}$ on Mistral-7B-v0.3 (GovReport). Consistent with intuition, too short prefixes $(8, 16)$ lead to larger overlap between short/long distributions, while our chosen values $(32, 64)$ show clearer separation with fewer false positives. We also show good performance particularly with respect to consistency among datasets when adaptively setting the context length to $0.1|\mathbf{s}|$ . See Appx. F.3 and Fig. 29.

**Computational overhead.** LSDS requires one extra short-context forward pass plus a JSD computation beyond normal generation. With a sliding-window cache for the short context, TaBoo adds only 3–5% time per token with FlashAttention (up to ≈22% with SDPA) when generating from a 5K-token context, across models from 0.6B to 32B parameters (Appx. F.4).

## 6 Long-Context Token Boosting

Fewer exposures to long-context sequences may create a potential bias toward completions that follow from local context, overshadowing long-range dependencies. *Can we identify tokens in the vocabulary that are more relevant to full-context information rather than local context?*

If so, we could favor generating such tokens when we detect a sequence requires long-context reasoning (which, as shown in the previous section, we can reliably identify). Here, we show this is possible and evaluate our method on text-based Q/A and summarization tasks, which are standard benchmark for assessing long-context understanding [10, 1, 34].

### 6.1 Identifying Long-Context-Relevant Tokens

First, we show how to identify tokens that are more relevant to full-context information than to local context. Consistent with our theme, the insight is that tokens requiring long-range dependencies should exhibit larger probability increases when given access to full versus short context.

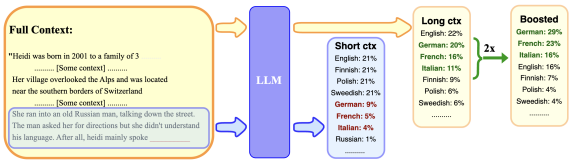



Figure 6: An example illustrating how a shift in the model's next token probabilities when given the long vs short context could help us recognizing the more relevant tokens and boosting their probabilities.

**Definition 4.** *The Long-Short Probability Shift (LSPS) of a vocabulary token $t$ given sequence $\mathbf{s}$ is defined as the change in the assigned probability moving from short to full context under decoding $\phi$:*

$$\mathsf{LSPS}(t|\mathbf{s}) = \left[\mathbf{p}_\phi(\mathbf{s})\right]_t - \left[\mathbf{p}_\phi(\mathbf{s}_{[-32:]})\right]_t .$$

Table 1: QA performance across methods. Main numbers are average scores; numbers in parentheses are best scores. Bold marks the best average within each model block and dataset/metric; underlining marks the best corresponding best-score value.

| Model | Method | NarrativeQA | | | HotpotQA | | | MultifieldQA-en | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | F1(↑) | BLEU(↑) | ROUGE-L(↑) | F1(↑) | BLEU(↑) | ROUGE-L(↑) | F1(↑) | BLEU(↑) | ROUGE-L(↑) |
| Llama-3.1-8B | Vanilla | 26.3 (50.0) | 5.5 (12.9) | 35.5 (59.7) | 26.3 (50.5) | 8.4 (17.0) | 38.1 (63.1) | 26.0 (44.2) | 9.5 (21.4) | 33.2 (54.1) |
| | CAD | 34.1 (55.1) | 6.9 (13.5) | **45.5** (62.1) | 22.2 (42.6) | 6.9 (14.2) | 39.4 (63.1) | 26.5 (45.7) | 10.0 (21.9) | 33.8 (55.0) |
| | AdaCAD | 33.9 (54.5) | 6.8 (13.4) | 45.3 (62.6) | 21.7 (40.6) | 6.9 (13.1) | 39.3 (60.5) | 26.7 (46.0) | 10.5 (23.4) | 34.5 (55.5) |
| | TaBoo | **35.4** (55.2) | **8.1** (15.8) | 45.1 (64.3) | **28.9** (50.4) | **9.5** (17.8) | **42.1** (63.6) | **27.7** (46.8) | **11.1** (24.4) | **35.6** (57.3) |
| Mistral-7B-v0.3 | Vanilla | 26.4 (49.9) | 5.6 (12.8) | 35.1 (58.8) | 32.0 (55.8) | 10.2 (19.6) | 41.9 (65.5) | 23.3 (40.2) | 8.2 (17.9) | 31.0 (51.1) |
| | CAD | 33.8 (52.3) | 6.9 (12.9) | 41.9 (60.5) | 33.8 (56.4) | 11.0 (19.4) | 40.8 (62.7) | 24.1 (41.0) | 8.3 (20.5) | 31.9 (51.3) |
| | AdaCAD | 33.1 (52.9) | 6.7 (13.2) | 42.3 (60.6) | 33.5 (53.8) | 10.7 (18.7) | 41.2 (61.8) | 25.0 (41.5) | 9.4 (20.6) | 32.9 (51.9) |
| | TaBoo | **36.0** (54.5) | **7.9** (14.9) | **44.8** (63.3) | **35.7** (56.8) | **11.5** (19.8) | **46.5** (65.8) | **26.6** (44.6) | **9.7** (20.7) | **34.8** (54.7) |
| Qwen3-8B-Base | Vanilla | 43.0 (55.8) | 11.2 (16.4) | 53.4 (65.0) | 52.8 (67.5) | 18.5 (24.9) | 59.1 (74.2) | 35.6 (42.2) | 18.3 (24.7) | 42.9 (51.2) |
| | CAD | 41.5 (52.5) | 10.8 (14.9) | 53.5 (62.0) | 44.2 (56.9) | 15.8 (20.7) | 53.5 (66.6) | 35.9 (44.4) | 18.8 (25.9) | 43.4 (52.2) |
| | AdaCAD | 41.2 (52.4) | 10.5 (14.5) | 53.6 (61.8) | 45.1 (56.8) | 15.7 (20.7) | 53.7 (65.5) | 36.7 (46.4) | 19.0 (27.4) | 44.8 (59.6) |
| | TaBoo | **45.1** (54.8) | **11.9** (15.9) | **56.0** (64.3) | **55.4** (65.8) | **19.5** (24.0) | **63.4** (73.6) | **37.5** (47.9) | **19.3** (28.3) | **48.0** (60.4) |

Probability vector $\mathbf{p}$, $[\mathbf{p}]_t$ denotes the $t$-th entry. As in the previous section, we fix the prefix length to 32 and, unless otherwise stated, for $\phi$, we use nucleus sampling ($p = 0.9$). Our hypothesis is that tokens genuinely requiring long-context information will show positive LSPS values, as their relevance becomes apparent only with access to the complete context. Conversely, tokens predictable from local context should show small or (even) negative shifts.

**Validation on NarrativeQA.** We validate this hypothesis using *NarrativeQA* [46]. We focus on QA pairs where answers are 1–2 words (typically character names or locations), filtering out cases resolvable without the story context. Given a context with the story, question, and partial answer, we classify the ground-truth next token as an *Answer Token* and all other vocabulary tokens as *Non-Answer Tokens*. Fig. 7 shows LSPS distributions for both categories using LLaMA-3-8B.

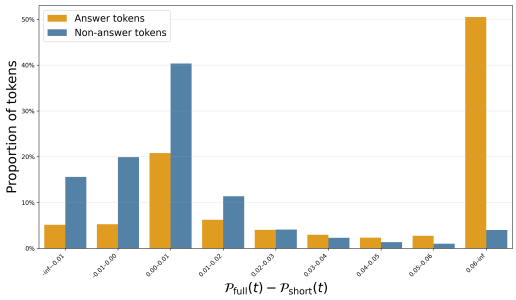



Figure 7: Example shows how a shift in the long- vs. short-context next-token distribution can signal most long-context-relevant tokens.

Observe the clear separation: Answer tokens exhibit significantly higher LSPS values, confirming that long-context relevant tokens can be identified through their probability shifts. Concretely, using threshold $\epsilon \in [0.05, 0.07]$, we capture over 50% of answer tokens while maintaining low false positive rates ($< 5\%$) on non-answer tokens. *Our objective is capturing long-context-relevant next tokens from the distribution while minimizing misclassifications of irrelevant ones*, which this approach achieves effectively. In Appx. G, we compare our method to a modified version that uses log-probability ratio, common in prior work [9, 8, 22, 4], demonstrating the stability of our approach.

### 6.2 Targeted Long-Context Token Boosting

Knowing how to (1) detect long-context sequences using LSDS and (2) identify long-context relevant tokens via LSPS, we now combine these tools to improve Q&A performance. The core insight is to promote those identified long-context relevant tokens, downplaying biases from the short context. Simultaneously, this can help downplaying tokens that are assigned high probability as an artifact of noise [7], potential word/token imbalances in training data [47, 48] and hallucination [49, 50].

**TaBoo (Targeted) Boosting.** Our algorithm TaBoo modifies the probability distribution as per the above intuition. Algorithm 1 details the complete procedure: (1) Detect long-context sequences using LSDS with threshold $\gamma$, (2) Identify long-context relevant tokens where LSPS $\geq \epsilon$, and (3) boost their probabilities by factor $\lambda$ before renormalization and nucleus sampling. See Algorithm 1 in Appx. G.

**Experimental setup.** We evaluate on NarrativeQA [46], HotpotQA [51], and MultiFieldQA [34] from the LonBench dataset. Focusing on stories with $\geq 1000$ tokens to emphasize long-context dependencies, we sample 1,000 examples from NarrativeQA and HotpotQA while using the full 150 examples from MultiFieldQA. We set $\epsilon = 0.05, \lambda = 4$ (no hyperparameter optimization) and based on analysis done in Appx. G.3, set the boosting threshold $\tau = 0.15$ as the results seem consistent around this range. We test on LLaMA-3-8B, Mistral-7B-v0.3, and Qwen-3-8B. For each

question-text pair, we generate 5 answers using nucleus sampling and report both average and best-of-5 F1/BLEU/ROUGE scores. While our intent is not targeted at hallucination, we still compare our TaBoo approach against two baselines: (1) vanilla nucleus sampling, and (2) Context Aware Decoding (CAD) [9] and with AdaCAD [23] as they have similar nature in focusing on token distribution. Both methods shift the probabilities of every next token distributions, while our targeted boosting only applies to long-context relevant tokens and sequences. For CAD we set the reweighting factor $\alpha = 0.5$ as described best by the authors while AdaCAD changes the $\alpha$ adaptively.

**Results.** Table 1 shows that on average, TaBoo consistently outperforms vanilla nucleus sampling across all models and datasets and is competitive with CAD and AdaCAD. These gains also hold for larger models (Appx. H). We provide additional Q/A Tab. 14 and summarization Tab. 13 analysis where longer generations are tested. The improvements generalizes across architectures and datasets. Best-per-example scores are also competitive, while highly variable among the cases. Additional results in Appx. G.1, G.5 present ablations on hyperparameter selection ($\gamma, \epsilon, \lambda$). Additionally, in Appx. F.4 we show that the computational overhead of LSDS (and thus TaBoo) is minimal for long sequences. While decoding algorithms are not our main focus, we believe our results can motivate further analysis to better understand and manage the implications of short context dominance.

**Reasoning tasks.** We also explore TaBoo's implications for reasoning models. Their answer stage is a natural test case for short-context bias: the final answer should depend on the entire reasoning trace, yet, as we observe for reasoning traces in Fig. 12, the model may lean heavily on its most recent tokens.

We use Qwen3-8B in thinking mode on AIME 2024/2025, MATH-500 (Level 5) [52, 53], and LiveCodeBench (Easy/Medium) [54], with 50 problems per set. For each problem, the model first generates a single reasoning trace (up to 8K tokens); problems whose trace does not terminate are excluded. Conditioned on this fixed trace, we then sample $k$=8 answers (up to 1K tokens each) with either vanilla sampling or TaBoo, so both methods share the same reasoning. TaBoo uses the last 32 reasoning tokens as the short context and, since the answer follows a long trace, a stricter detection threshold with a stronger boost ($\epsilon$=0.12, $\lambda$=8, $\tau$=0.20). As shown in Table 2, TaBoo improves accuracy on math, most notably on AIME (+13 points Mean@8, +16 Pass@8), while gains on code are marginal. These preliminary results suggest that short-context bias also affects how reasoning models use their own reasoning, motivating future work on targeted long-context corrections for reasoning.

Table 2: TaBoo on reasoning with Qwen3-8B. Mean@8 / Pass@8 (%).

| Task | Vanilla | TaBoo |
|---|---|---|
| MATH-500 (L5) | 71.7 / 94.1 | **76.8 / 100** |
| AIME | 42.0 / 56.0 | **55.0 / 72.0** |
| LCB (Easy) | 64.7 / 65.5 | **65.3** / 65.5 |
| LCB (Medium) | 53.0 / 57.1 | **53.2 / 57.3** |

## 7 Outlook

Our work provides a systematic framework for understanding context dependency in language models, with implications for inference efficiency in tasks like QA and summarization through principled post-hoc decoding. Specifically, our ability to detect individual long-context sequences without ground-truth tokens opens opportunities for context-aware generation methods, but also for more targeted evaluation of long-context capabilities and improved training approaches — for instance, by using the LSDS detector to reweight long-context relevant sequences during pretraining or finetuning. Beyond immediate applications, our findings reveal the short-context dominance hypothesis as an inherent property of individual natural language sequences, with potential broader implications for understanding how language models process and generate text. While datasets with ground-truth labels for minimal context length at our level of granularity are not readily available (this and additional limitations described in Appx. C), we hope our results motivate further research into the understanding of context utilization by modern LLMs.

## Acknowledgements and Disclosures

This work was supported by a Work Learn International Undergraduate Research Award (WLIURA), a Canada Graduate Scholarships–Doctoral (CGS-D) award, NSERC Discovery Grant No. 2021-03677, Alliance Grant ALLRP 581098-22 and a gift from Google.org. Computational resources were also provided by the UBC Advanced Research Computing (ARC) Sockeye cluster. We thank

Sadegh Mahdavi for valuable discussions during the early stages of this project. Finally, we thank the anonymous reviewers for their useful suggestions that helped improve this work.

**Competing interests.** The authors declare no competing interests.

## References

[1] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.

[2] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics.

[3] Simeng Sun, Kalpesh Krishna, Andrew Mattarella-Micke, and Mohit Iyyer. Do long-range language models actually use long-range context? In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 807–822, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[4] Lizhe Fang, Yifei Wang, Zhaoyang Liu, Chenheng Zhang, Stefanie Jegelka, Jinyang Gao, Bolin Ding, and Yisen Wang. What is wrong with perplexity for long-context language modeling?, 2025.

[5] Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. Sharp nearby, fuzzy far away: How neural language models use context. 2018.

[6] Falko Helm, Nico Daheim, and Iryna Gurevych. Token weighting for long-range language modeling, 2025.

[7] Pratyusha Sharma, Jordan T. Ash, and Dipendra Misra. The truth is in there: Improving reasoning in language models with layer-selective rank reduction. 2023.

[8] Nikolay Malkin, Zhen Wang, and Nebojsa Jojic. Coherence boosting: When your pretrained language model is not paying enough attention. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8214–8236, Dublin, Ireland, May 2022. Association for Computational Linguistics.

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

[10] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

[11] Zhenyu Zhang, Runjin Chen, Shiwei Liu, Zhewei Yao, Olatunji Ruwase, Beidi Chen, Xiaoxia Wu, and Zhangyang Wang. Found in the middle: How language models use long contexts better via plug-and-play positional encoding, 2024.

[12] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, et al. Improving language models by retrieving from trillions of tokens. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022,*

*Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR, 2022.

[13] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: few-shot learning with retrieval augmented language models. 24(1), January 2023.

[14] Jianghao Chen, Junhong Wu, Yangyifan Xu, and Jiajun Zhang. LADM: Long-context training data selection with attention-based dependency measurement for LLMs, 2025.

[15] Yung-Sung Chuang, Benjamin Cohen-Wang, Shannon Zejiang Shen, Zhaofeng Wu, Hu Xu, Xi Victoria Lin, James Glass, Shang-Wen Li, and Wen tau Yih. Selfcite: Self-supervised alignment for context attribution in large language models, 2025.

[16] Haoran Deng, Yingyu Lin, Zhenghao Lin, Xiao Liu, Yizhou Sun, Yi-An Ma, and Yeyun Gong. Beyond length: Quantifying long-range information for long-context llm pretraining data, 2025.

[17] Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada, July 2023. Association for Computational Linguistics.

[18] Zheng Zhao, Emilio Monti, Jens Lehmann, and Haytham Assem. Enhancing contextual understanding in large language models through contrastive decoding. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4225–4237, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[19] Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. Dexperts: Decoding-time controlled text generation with experts and anti-experts. 2021.

[20] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California, June 2016. Association for Computational Linguistics.

[21] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[22] Liam van der Poel, Ryan Cotterell, and Clara Meister. Mutual information alleviates hallucinations in abstractive summarization. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5956–5965, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[23] Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Adacad: Adaptively decoding to balance conflicts between contextual and parametric knowledge, 2025.

[24] Huayang Li, Deng Cai, Jin Xu, and Taro Watanabe. $n$-gram is back: Residual learning of neural text generation with $n$-gram language model, 2022.

[25] Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens, 2025.

[26] Timothy Nguyen. Understanding transformers via n-gram statistics, 2024.

[27] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[28] Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA, 2015. MIT Press.

[29] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In Stefan Riezler and Yoav Goldberg, editors, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, August 2016. Association for Computational Linguistics.

[30] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *ArXiv*, abs/1609.07843, 2016.

[31] Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. Efficient attentions for long document summarization. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online, June 2021. Association for Computational Linguistics.

[32] Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. QMSum: A new benchmark for query-based multi-domain meeting summarization. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online, June 2021. Association for Computational Linguistics.

[33] Wojciech Kryscinski, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. BOOKSUM: A collection of datasets for long-form narrative summarization. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6536–6558, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[34] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[35] Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online, April 2021. Association for Computational Linguistics.

[36] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, et al. The llama 3 herd of models, 2024.

[37] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[38] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025.

[39] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[40] Sourya Basu, Govardana Sachithanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. Mirostat: a neural text decoding algorithm that directly controls perplexity. In *International Conference on Learning Representations*, 2021.

[41] John Hewitt, Christopher D. Manning, and Percy Liang. Truncation sampling as language model desmoothing, 2022.

[42] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models, 2025.

[43] Haozhe Ji, Pei Ke, Zhipeng Hu, Rongsheng Zhang, and Minlie Huang. Tailoring language generation models under total variation distance, 2023.

[44] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[45] G. Kamradt. Llmtest_needleinahaystack: A needle-in-a-haystack retrieval test for long-context llms. https://github.com/gkamradt/LLMTest_NeedleInAHaystack, 2023.

[46] Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

[47] Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot numerical reasoning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 840–854, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[48] Nora Kassner, Benno Krojer, and Hinrich Schütze. Are pretrained language models symbolic reasoners over knowledge? In Raquel Fernández and Tal Linzen, editors, *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 552–564, Online, November 2020. Association for Computational Linguistics.

[49] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. 55(12), March 2023.

[50] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA, 2021. Association for Computing Machinery.

[51] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[52] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021.

[53] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023.

[54] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code, 2024.

[55] Stanley F. Chen and Joshua T. Goodman. An empirical study of smoothing techniques for language modeling. 1996.

[56] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, March 2003.

[57] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[58] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[60] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, et al. Qwen2 technical report, 2024.

[61] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, et al. Gemma: Open models based on gemini research and technology, 2024.

[62] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C), February 2024.

[63] Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation, 2022.

[64] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation, 2023.

[65] Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[66] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[67] Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. QuALITY: Question answering with long input texts, yes! In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States, July 2022. Association for Computational Linguistics.

[68] Wenhong Zhu, Hongkun Hao, Zhiwei He, Yiming Ai, and Rui Wang. Improving open-ended text generation via adaptive decoding. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[69] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025.

[70] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[71] Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. Openwebmath: An open dataset of high-quality mathematical web text, 2023.

[72] Microsoft. LCC_Python dataset. https://huggingface.co/datasets/microsoft/LCC_python, 2024. Accessed: 2025-02-15.

[73] Yuxuan Zhou, Margret Keuper, and Mario Fritz. Balancing diversity and risk in LLM sampling: How to select your method and parameter for open-ended text generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26352–26365, Vienna, Austria, July 2025. Association for Computational Linguistics.

[74] Chen Jia. Adversarial moment-matching distillation of large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc.

[75] Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph E. Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. How long can open-source LLMs truly promise on context length? LMSYS Org Blog, https://lmsys.org/blog/2023-06-29-longchat, June 2023.

[76] W. J. Youden. Index for rating diagnostic tests. *Cancer*, 3(1):32–35, 1950.

[77] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics.

[78] Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. L-eval: Instituting standardized evaluation for long context language models, 2023.

[79] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, page 70–79. Association for Computational Linguistics, 2019.

[80] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. 2018.

## A Summary of paper's notations

Table 3 provides a summary of the notation used in this paper.

Table 3: Notation used throughout the paper.

| Symbol | Description |
|---|---|
| $a$ / $\mathbf{a}$ | Scalar / vector (boldface) |
| $[i]$ | $i$-th entry of vector |
| $[i:j]$ | Index set $\{i, i+1, \ldots, j\}$ |
| $[-l:]$ | Suffix of  of length $l$ |
| $\mathcal{V}$ | Vocabulary of tokens, $V = \lvert\mathcal{V}\rvert$ |
| $\mathbf{s} = [t_1, \ldots, t_n]$ | Tokenized document of length $n$ |
| $\mathbf{s}_{[i]}$ | Prefix of $\mathbf{s}$ of length $i$ |
| $\Delta^n$ | Probability simplex in $\mathbb{R}^n$ |
| $\mathsf{Top}_k(\cdot)$ | Indices of $k$ largest entries of a vector; |
| $\pi_\theta(\mathbf{s})$ | LM distribution over $\mathcal{V}$ given context $\mathbf{s}$ |
| $\Delta\mathsf{Conf}(\mathbf{s})$ | Confidence = gap between top-1 and top-2 probabilities |
| $\mathrm{LSDS}(s)$ | Long–short distribution shift (JSD of short vs full context) |
| $\mathrm{LSPS}(t\|s)$ | Long–short probability shift for token $t$ in sequence $s$ |

## B Discussion and Related Work

**Pursuit of Long Context:** Capturing dependencies that extend beyond a few tokens has been a long-standing difficulty in language modeling. Early statistical and neural models either truncated context to short n-grams or attempted to maintain memory through recurrence, but both approaches faced limitations with sparsity or vanishing gradients [55–58]. The Transformer architecture [59] marked a turning point, with self-attention providing a scalable mechanism for integrating information from across the entire input. Contemporary open-access models such as LLaMA 3 [36], Mistral [37], Qwen2 [60], and Gemma [61] support context lengths from 8K to 128K tokens, large enough to encode entire novels in a single pass. To make such extensions feasible, architectural innovations like rotary position encodings (RoPE) [62], attention linear biases (ALiBi) [63], and position interpolation [64] enable models to extrapolate beyond their training horizon, while retrieval-augmented designs [12, 13, 65] surface or cache relevant information as an alternative to enlarging the raw attention window. Yet, greater architectural capacity does not imply that models make effective use of long-range context at inference time—the focus of our analysis.

**Context Utilization:** Despite larger context windows, studies show that LMs often do not utilize long-range information. [5] characterize predictions as "sharp nearby, fuzzy far away," with sensitivity concentrated in the most recent span, while [3] demonstrate that only a small fraction of tokens benefit from context beyond the first few thousand tokens. Such works, while developed on earlier language models, point towards the idea that majority of next token predictions rely on local information. Their method of measurement is mainly focused on average perplexity. Both studies consider the overall perplexity over documents in the absence of longer prefixes. While describing average behavior, metrics based on the probability of next token do not necessarily provide enough information about individual contexts. For example, if we consider the setup in [5], while done on an older LSTM model, the threshold of $\tau = 0.05$ as is used to justify the argument that beyond 50 token prefix, the perplexity on documents do not improve noticeably. In Fig. 8, similar to our setup in Fig. 2, we use the perplexity metric $\| \left[\mathbf{p}_\phi(\mathbf{s})\right]_t - \left[\mathbf{p}_\phi(\mathbf{s}_{[-w:]})\right]_t \| \leq 0.05$ as the indicator to determine the minimal context length for an individual context. Note that the y-axis is in log scale and as we can see, the results are drastically different and do not reflect the same short context dominance described by MCL or DaMCL. **Our contributions with our metrics help allow for recognition of individual sequence with long context relevant with or without the knowledge of ground truth next token.**

Such works emphasize the inherent recency bias in language models. Complementing these results, [10] document a strong positional bias—models attend to evidence at the edges of long prompts while neglecting the middle, a challenge further analyzed by [11], who trace the effect to rotary positional encodings and propose positional modifications to resolve it. To mitigate these limitations, retrieval-

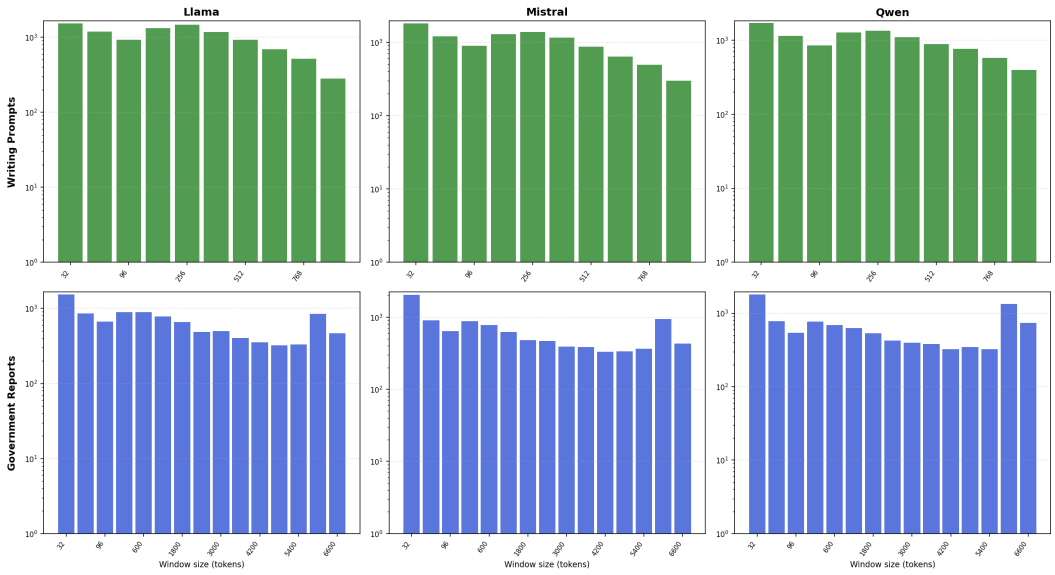



Figure 8: **Perplexity as metric:** Similar to setup in Fig. 2 and Fig. 3, but we use $|| \left[\mathbf{p}_\phi(\mathbf{s})\right]_t - \left[\mathbf{p}_\phi(\mathbf{s}_{[-w:]})\right]_t ||| \leq 0.05$ to find the smallest required window (or minimal context) for sequences in document. Evidently, this metric does not illustrate the same short context dominance on an individual level per sequences.

based methods surface relevant passages at inference time [12, 13], while recent work emphasizes the role of training data quality in enabling long-context utilization. In particular, [14] introduce an attention-based dependency measurement framework (LADM) to identify long documents with strong internal dependencies, showing that selecting such high-quality data for continual pretraining substantially improves long-context performance. Beyond interventions at the system and data level, attribution-based methods such as [15] directly test the necessity and sufficiency of context spans, offering a finer-grained perspective on how LLMs actually use long inputs. Similarly, we study the utilization of context from a minimal required sub-context viewpoint, showing that majority of next token queries utilize very local information. Additionally, we provide methodology to identify the minimal required sub context, based on ground truth next token or next token distribution matching, rather than only distinguishing and separating long-short context queries.

**Contrastive Decoding:** Beyond architectural and data-level interventions, several ad hoc inference-time methods aim to improve token generation. Early work encouraged generation that depends more strongly on the given context rather than defaulting to frequent or generic outputs. [20] propose a mutual-information–based objective for dialogue generation to discourage generic responses, while [21] address this issue in multiple-choice QA by normalizing candidate likelihoods against unconditional probabilities, thereby encouraging context-dependent answers. [8] introduced coherence boosting, a general inference-time method that reweights the next-token distribution to favor predictions supported by the full long context by explicitly contrasting it against the distribution induced by a shortened context. [22] proposed an entropy-aware decoding strategy for summarization which revolves around promoting tokens that appear more often in the source text whenever next-token-entropy (model uncertainty) is high, thereby discouraging hallucinations and reducing the tendency to select high-frequency tokens based on the model's own biases (promoting high-frequency tokens in language which do not have a strong correlation to the source material). Similar to [8] but operating on a logit level, [9] introduce Context-Aware Decoding (CAD), an inference-time method designed to reduce hallucinations by explicitly amplifying the difference between a model's output probabilities with and without the provided context. Further, [23] introduces AdaCAD which allows self-tuning the $\alpha$ exponent of CAD through comparing the distributional shift between, as a measure to combat hallucination per context more selectively. Much of this line of research grows out of the broader family of contrastive decoding methods [17–19] which are designed for ad hoc modification to the next token distribution to improve language generation. For our case, instead of focusing on performance we first provide an objective study of short context bias of language and use our findings

to design intuitive and explainable algorithms to identify long contexts, relevant tokens and how to combine this knowledge to help improve generation.

**n-gram and its short context relevance:** Recent work has revisited n-gram models as complements or interpretive tools for neural language models. [24] showed that residual learning with a small n-gram LM can regularize neural text generation and reduce hallucination. More recently, [25] introduced Infini-gram, which scales n-gram models to trillions of tokens and supports unbounded context length using suffix arrays, enabling both strong next-token prediction and new diagnostic analyses of neural LMs. In parallel, [26] argue for understanding Transformers through the lens of n-grams, showing that simple n-gram rulesets can approximate a majority of model predictions (e.g., covering 68–79% of top-1 predictions across benchmarks). Given that n-grams are inherently short-range models—even Infini-gram typically captures at most 32-token dependencies—the fact that they achieve performance comparable to Transformers on standard text suggests a structural bias in language toward short-context sufficiency, which allows such models to achieve moderate success despite their limited horizon. Our observations regarding the overwhelming prevalence of inherently short contexts in natural language can help explain the reason behind the success of such methods which rely on local information using n-grams for generation.

**Long Context Evaluation:** Much of the evaluation of a model's contextual understanding has focused on tasks such as question answering, retrieval, and needle-in-the-haystack probing, evaluated on datasets such as NarrativeQA [46], TriviaQA [66], QuALITY [67], and LongBench [34]. While these benchmarks test a model's ability to extract specific information from distant context, they differ from standard language modeling and tend to be highly task-specific. A recent study by [4] proposes a method for identifying tokens with long-context dependencies and encourages metrics that distinguish such tokens. Following a similar pattern, [6] uses a similar strategy to promote long context tokens during pretraining. [16] on the other hand, uses a similar context length analysis to analyze documents with more long context reliance and promotes long context training by emphasizing on such documents during pretraining. There are a number of notable difference between such works and our analysis. Both [4, 6] focus on the context length extension of models, thus their comparison of short vs long context consists of a binary classification, **identifying tokens with long-prefix reliance beyond 4000-8000 (compared to our focus on smaller windows of 32, 64, 128, ... tokens).** We adopt a more fine-grained perspective: treating the language model as a probabilistic oracle and estimating the minimal context required for each next-token prediction in natural text. Additionally, we provide methods for detection of long-context and tokens with long-context relevance **without requiring the actual ground truth next token, making our methods applicable during inference**. While such works focus more on the long context retrieval capabilities of models and its practical implications, we analyze the short context bias of language itself, illustrating the short-context dominance of human language itself and provide one example of how to combat such biases using TaBoo.

**Decoding Strategies:** Given our assumption that language models serve as strong proxies for language understanding, it is important to account for the decoding strategy used during inference. A growing body of research has shown that different sampling methods—such as greedy decoding, top-$k$ sampling [44, 27], nucleus ($p$) sampling [39], and adaptive techniques [41, 40, 68]—can substantially influence output diversity, factuality, and calibration. Greedy decoding in particular has been shown to produce degenerate or overly deterministic outputs, while adaptive and dynamic approaches aim to adjust sampling entropy and generate a high-quality, contextually valid subset of tokens [39, 68, 40]. When treating the language model as a statistical oracle for analyzing context usage, it is essential to consider how decoding strategy influences conclusions about effective context length. This perspective may help improve the practical utility of methods such as [4] and [6], which focus primarily on a single sample from the next-token distribution to classify tokens by their context length requirements. Accordingly, we provide a dedicated analysis of how decoding strategies impact context dependence in next-token prediction and its relation to short-long context detection.

# C Limitations

While our work provides a systematic framework for understanding and leveraging context dependency in language models, several limitations are worth noting.

**LLMs as statistical oracles.** Throughout our analysis, we treat pretrained language models as proxies for the true natural language distribution, using their next-token predictions to quantify context dependency via MCL and DaMCL. However, LLM predictions reflect both the underlying structure of language and artifacts of their training data, architecture, and optimization procedure with many potential biases and structural failures. As a result, our measurements of minimal context length capture the context dependency of the model's learned distribution rather than that of natural language itself. Separating these two effects is not straightforward and we leave this to future work.

**Lack of ground-truth labels for minimal context length.** A fundamental difficulty in our work is that there is no ground-truth label for how much context a given token truly requires. MCL is model-dependent and only defined for tokens the model predicts correctly and confidently, covering 40–60% of context/next-token pairs (Sec. 3). DaMCL removes the ground-truth dependency but relies on distributional similarity thresholds, which introduce their own approximation. In practice, both metrics serve as useful proxies rather than definitive measures, and our findings should be interpreted with this in mind. Ideally, a human annotated dataset which would describe long context dependence on individual sequences is required to better study the short context dominance.

**Fixed short-context prefix length.** We fix the short-context prefix to 32 tokens throughout, motivated by prior work and our ablations (Appx. F.1). This choice may not be universally optimal — the appropriate prefix length likely depends on the task, domain, and tokenization scheme, and could vary considerably across settings. We conduct a limited set of experiments with an adaptive prefix length of $0.1|\mathbf{s}|$, where local context is approx 10% of the total length and we find that the short-context dominance phenomenon persists.

**Evaluation scope.** Our downstream evaluation of TaBoo focuses on question answering and summarization tasks, where reference-based metrics such as F1, BLEU, and ROUGE-L provide automatic, scalable evaluation. Ideally, one would also assess the impact of short-context bias correction on open-ended generation tasks, where long-range coherence and narrative consistency are particularly relevant. This would also allow for a better separation between hallucination contrastive methods such as CAD [9]. However, evaluating generation quality at this level requires either human annotation or LLM-based judgment, both of which introduce their own biases and scalability constraints. We leave a more comprehensive evaluation of TaBoo on generation tasks to future work.

# D Additional Results on MCL

## D.1 Ablation Results: Model Size and Confidence threshold $\delta$

We perform ablation experiments on MCL results regarding model size and the impact of confidence threshold $\delta$ in Fig. 9 and Fig. 10 respectively. Observations further confirm that the short-context dominance hypothesis appears irrespective of experimental setup.

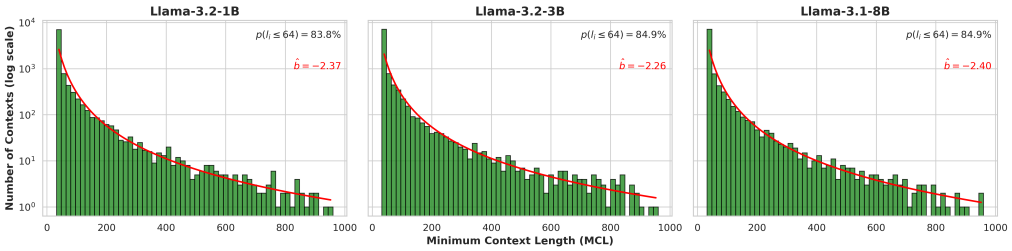



Figure 9: **Impact of model size:** Similar setup to Fig. 2 but Llama models withe different sizes. We can see that the distribution of MCL behavior does not change across different models sizes of 1,3 and 7 billion parameter sizes.

## D.2 Language and Domain Results

We conduct two sets of experiments to confirm the robustness of the short-context dominance with respect toe different languages and domains. First, we examine the language effect by running inference on Wikipedia article translations in Arabic, Chinese, French, German, Korean, Russian, and Thai, using the Qwen2-7B model [60]. These texts are sampled from the Wikimedia dataset [35],

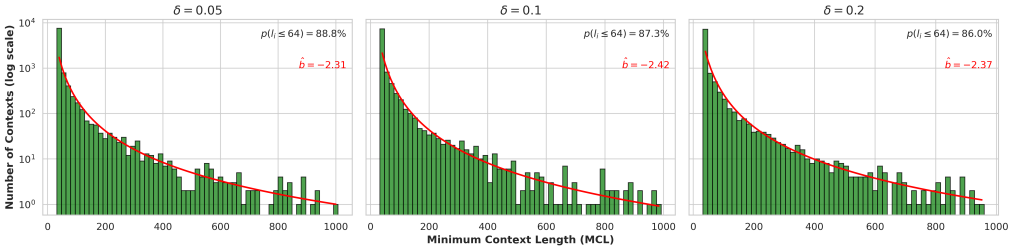



Figure 10: **Impact of confidence threshold $\delta$:** Similar setup to Fig. 2 with Llama-3 8B running MCL experiments with different values of $\delta \in \{0.05, 0.1, 0.2\}$. Results suggest that the exponential decay persists irrespective of the confidence threshold.

Table 4: Estimated power-law exponents ($\hat{b}$) for MCL distributions on the Wikimedia dataset using Qwen.

| | English | Arabic | French | German | Chinese | Russian | Korean | Thai |
|---|---|---|---|---|---|---|---|---|
| $\hat{b}$ | $-2.27$ | $-2.27$ | $-2.30$ | $-2.21$ | $-2.27$ | $-2.26$ | $-2.25$ | $-2.25$ |

and follow the same selection and truncation procedure as our English WikiText documents. Results in Table 4 and Fig. 11 confirm that the reliance on short context is preserved across languages, with similarly highly skewed token count distributions and consistent heavy-tailed fits. Additionally, we perform a small test analyzing the Chain-of-Thought (CoT) generated text by the S1 model [69], analyzing the MCL using Qwen to validate the short-context dominance. Noticeably, looking at the exponential decay in Fig. 12, we see that the CoT contexts exhibit a strong short-context dominance.

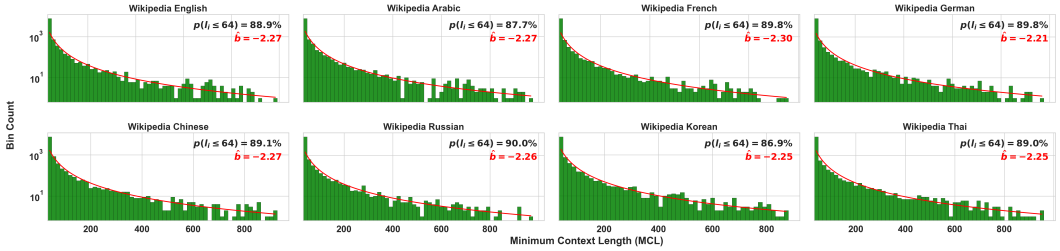



Figure 11: **Distribution of MCL on Different Languages:** Similar setup to Fig. 2 but performed over a set of articles from Wikipedia and their respective translations into different languages. We see that the same trend appears irrespective of target language. This analysis was conducted on Qwen-3-8B.

Second, to evaluate domain knowledge effects, we test on three specialized datasets: CCDV PubMed Summarization [70] for biomedical abstracts, Open Web Math [71]for mathematical content, and LCC_Python [72] for code documents, across all 3 models: LLaMA-3-8B [36], Mistral-7B-Instruct (v0.3) [37] and Qwen3-8B [38]. More specifically, the contexts selected from the math and medical datasets are within 1024 tokens, while the context length varies from 6000 to 7000 tokens on the code dataset. As shown in Fig. 13, the short-context dominance pattern remains intact across all three domains and models. This suggests models' reliance on local context persist even when dealing with knowledge-intensive content.

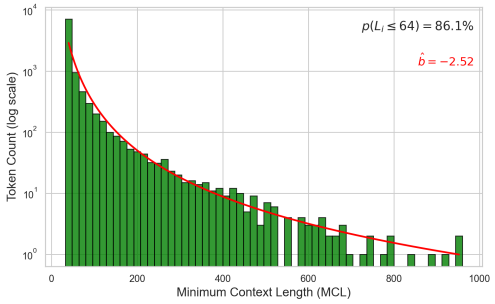



Figure 12: MCL distribution for Qwen evaluated on context extracted from the generated CoT text generated by the S1 model. We observe that the short context dominance persists and is slightly stronger on this reasoning model generated dataset.

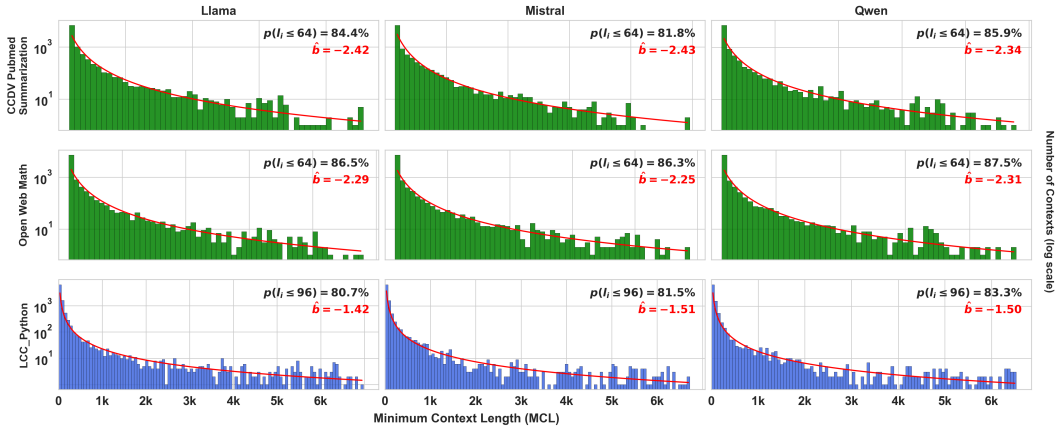



Figure 13: **Distribution of MCL on Math and Medical Datasets:** This figure shows the distribution of MCL values over three specialized domains: medical (CCDV PubMed Summarization), mathematical (Open Web Math) and , code (LCC_Python) with a similar setup as Fig. 2. The MCL distribution consistently follows the same trend across all domains and all three models. This suggests that the observed phenomenon is robust to changes in domain knowledge.

### D.3 MCL Results on word/subword tokens

While we demonstrated the short-context dominance of language in the setting of LLMs, one natural question arises: *Could this phenomenon be an artifact of subword tokenization?* For example, a word like `establishments` may be tokenized into `est`, `_ablish`, and `_ments`. Intuitively, the model might require more context to correctly predict the first subword compared to the later ones. This raises the question: does the model require less context to guess a non-starting subword (e.g., `_ments`) than a starting one?

To investigate this, we partition tokens into three inline categories: (1) **full-word tokens**, where a token corresponds to an entire word; (2) **subword-starting tokens**, which occur at the beginning of a word; and (3) **subword non-starting tokens**, which appear in the middle or end of a word. We then compute MCL separately for these categories, using Mistral-7B-Instruct on a news dataset and Qwen2-7B on writing prompts. As shown in Fig. 15, we observe no meaningful differences in the MCL distributions across the three tokenization categories. If tokenization breaks were driving short-context dominance, we would expect noticeably different decay patterns for full-word versus subword tokens. Instead, the curves remain highly similar.

### D.4 MCL Results on Part-of-Speech

Similar to our analysis on word–versus–subword token behavior, we investigate whether there is any relationship between a token's Part-of-Speech (POS) category and its tendency to exhibit short-context domination. In particular, we ask whether certain types of words—such as nouns or adjectives—are inherently more or less likely to rely on long-range context.

Because POS categories are defined at the word level rather than the token level, we first identify each word in the sequence and assign it to one of four categories: noun, verb, adjective, or adverb. All other words and tokens (e.g., determiners, punctuation, and special tokens) are excluded from our analysis. Each subword token then inherits the POS label of the word it belongs to. We follow a sampling strategy similar to our earlier MCL experiments while keeping the number of tokens per category roughly balanced.

We evaluate two setups: Mistral-7B-Instruct-v0.2 on the CNN News dataset and Qwen2-7B on the Reddit Writing Prompts dataset. Results are presented in Fig. 15. We find that short-context dominance persists across all POS categories. However, compared to nouns and verbs, tokens corresponding to adjectives and adverbs show a slightly greater tendency toward long-context dependence.

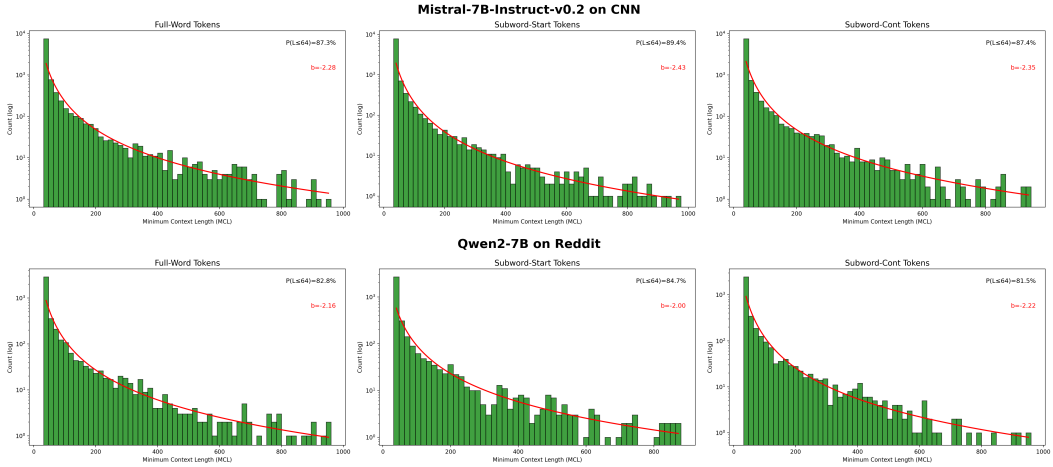



Figure 14: **Distribution of MCL on sub-word tokens:** This figure illustrates the distribution of MCL values for two models and two datasets, separated according to whether the next token is a 1) *full word token*, 2) *sub-word token starting a word* and *sub-word non-starting token* . The experiment illustrates no major difference between the tokens MCL distributions irrespective of its full-word/sub-word.

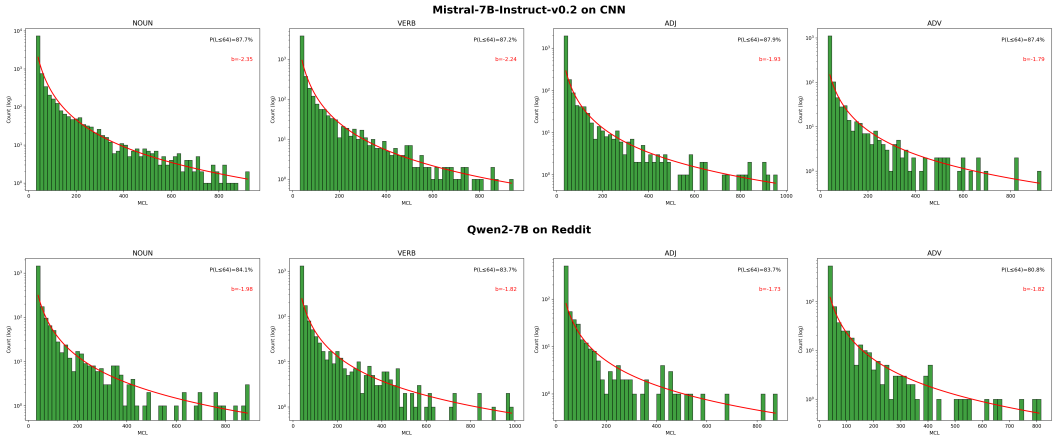



Figure 15: **Distribution of MCL on tokens based on Part-of-speech labeling:** This figure shows the distribution of MCL values for two models and two datasets, separated according to whether the next token belongs to a word in one of the four POS categories: noun, verb, adjective, or adverb. The results reveal that short-context dominance persists across all categories. However, the long-tail behavior suggests that tokens associated with adjectives and adverbs tend to exhibit greater long-context dependence compared to nouns and verbs.

# E DaMCL

## E.1 Motivations on DaMCL

In Sec. 3, we posed the question of determining the minimum subcontext prefix needed to predict the next token in a given dataset. A key limitation of this formulation is that it is constrained by the specific realization of the natural language distribution underlying that dataset.

Put simply, given a context, there are often multiple valid next tokens—valid in terms of the underlying (but unknown) distribution of natural language. While we cannot access this true distribution, we have treated pretrained LLMs as statistical oracles. However, in defining MCL in Definition 1, we constrain these oracles by evaluating them against only the actual next token from the dataset. Furthermore, we rely solely on greedy decoding, which outputs a single token, thereby underutilizing the model's full predictive distribution as a language oracle.

We summarize the issues as follows:

1. Even if the oracle's top-1 prediction does not match the next token in the source text, i.e., $\text{Top}_1(\mathbf{s}_{[-l:]}) \neq t$, this does not invalidate the model's output or imply a lack of contextual understanding. As shown in Fig. 16, the model assigns high probability to several plausible continuations, even if the dataset token is not ranked first. This suggests that relying solely on the dataset token may mislead any context-length detection method.
2. Using the Top-1 token from the sampling distribution is not always a reliable way to evaluate next-token prediction, as greedy decoding often results in low-quality or repetitive outputs [39]. More recent sampling strategies instead aim to identify a set of valid next tokens [68, 73], shifting the focus away from single-token probabilities toward broader support coverage.

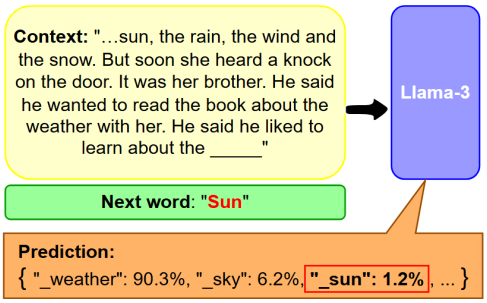



Figure 16: An example illustrating the limitations of relying on greedy sampling and the actual next token: the Top-1 prediction is a valid response but does not match the ground-truth token. Metrics like MCL may overlook such cases, misidentifying the minimal required context.

These issues motivate the need for a broader definition of MCL—one that 1) **relies on the model's own next-token distribution rather than the actual next token**, and 2) **accounts for the sampling strategy used during inference**. The goal of DaMCL is to mitigate these limitations and offer a more faithful metric for contextual understanding.

### E.2 Illustrative Example: Short- vs. Long-Context Tokens in Natural Text

To strengthen intuition for the DaMCL metric, we visualize a short passage from the Cosmopedia dataset and annotate each token based on whether its next-token prediction requires long-range context. For each token position, we compute $\text{LSDS}(\mathbf{s}) = \text{JSD}\big(p_{\eta=0.9}(\mathbf{s}_{[-32:]}),\, p_{\eta=0.9}(\mathbf{s})\big)$ and use a coloring gradient to illustrate the JSD values. Fig. 17 provide an illustration of how JSD operates for detection the minimal context length.

Contexts which are followed by proper nouns such as *"Sammy"* and *"Harmonyville"*, if not repeated in local passage (last 32 tokens), appear with a higher JSD values. Contexts followed by fillers such as *"if, as, of, and"* are often shorter context in nature. Additionally, tokens such as *"footprint"* is given a lighter JSD, as it the word *"carbon"* narrowing the possibility of next tokens. Finally note that a the long contextness not always obvious based on the following token. For example *"these changes"* or *"little friends"* is not a unique to the following token, but rather concepts that are only grasped once the model is provided a longer prefix, making the JSD distribution show a larger shift in possible next token distributions. Additionally, note that a large majority of the subsequences would be considered local if we use a $\tau = 0.2 - 0.6$ as threshold, further illustrating the short context dominance on natural language.

### E.3 Additional Metrics for DaMCL

In addition the main JSD metric used in Sec. 4, we perform the same experiments with a number of other common metrics used to represent similarity between sets and distributions. For the given distributions $\mathbf{p}_1, \mathbf{p}_2 \in \Delta^{\mathcal{V}}$ in the $|\mathcal{V}|$-dimensional simlex we use the following standard distributional similarity metrics:

$$\textbf{Total Variation Distance:} \quad \text{TVD}(\mathbf{p}_1, \mathbf{p}_2) := \frac{1}{2}\|\mathbf{p}_1 - \mathbf{p}_2\|_1\,,$$

$$\textbf{Kullback-Leibler Divergence:} \quad \text{KL}(\mathbf{p}_1 \| \mathbf{p}_2) := \sum_{v \in \mathcal{V}} \mathbf{p}_1(v) \log \frac{\mathbf{p}_1(v)}{\mathbf{p}_2(v)}\,.$$

These metrics are widely employed in applications such as knowledge distillation and in assessing performance degradation of large language models under various conditions [42, 43, 74].




Figure 17: Annotation of a passage from Cosmopedia where next tokens are colored based on the JSD values comparing the short (last 32 tokens) vs the full context distributions for the prefix sequence. First 32 tokens are grayed out. Warmer colored tokens imply that the next token distribution shifts substantially between the local and full context — indicating greater long-range dependency — while colder colors imply that the next token distribution does not change noticeably between the local and full context. We can see that names of characters and locations, certain traits and concepts (for example "carbon" ) have higher dependence on longer context.

Furthermore, we consider metrics which rely on the inclusions of tokens in the support set rather than the probability distributions. To this end, we define define the Recall, Precision and F1 metric from set $P$ to set $Q$ as:

$$\mathsf{Recall}\left(\mathbf{p}_1 \mid \mathbf{p}_2\right) := \frac{|\mathbf{p}_1 \cap \mathbf{p}_2|}{|\mathbf{p}_2|} \in [0,1] \qquad \mathsf{Prec}\left(\mathbf{p}_1 \mid \mathbf{p}_2\right) := \frac{|\mathbf{p}_1 \cap \mathbf{p}_2|}{|P|} \in [0,1]\,,$$

$$\mathsf{F1}\left(\mathbf{p}_1 \mid \mathbf{p}_2\right) := \frac{2 \times \mathsf{Recall}\left(\mathbf{p}_1 \mid \mathbf{p}_2\right) \times \mathsf{Prec}\left(\mathbf{p}_1 \mid \mathbf{p}_2\right)}{\mathsf{Recall}\left(\mathbf{p}_1 \mid \mathbf{p}_2\right) + \mathsf{Prec}\left(\mathbf{p}_1 \mid \mathbf{p}_2\right)} \in [0,1]\,.$$

**Recall** measures the proportion of elements in set $\mathbf{p}_2$ that are also present in set $\mathbf{p}_1$, i.e., how much of $\mathbf{p}_2$ is recovered by $\mathbf{p}_1$. **Precision**, on the other hand, quantifies the proportion of elements in $\mathbf{p}_1$ that are relevant—those that also belong to $\mathbf{p}_2$. A Recall of 1 implies $\mathbf{p}_2 \subseteq \mathbf{p}_1$, meaning all elements of $\mathbf{p}_2$ are captured by $\mathbf{p}_1$. Conversely, a Precision of 1 implies $\mathbf{p}_1 \subseteq \mathbf{p}_2$, indicating that $\mathbf{p}_1$ contains no extraneous elements outside of $\mathbf{p}_1$. The **F1** score is defined as the harmonic mean of Recall and Precision, providing a balanced measure that accounts for both. These definitions are standard for analysis of set similarity and coverage. We specifically calculate $1 - \mathsf{F1}$ in order to match the preference for smaller values, similar to the JSD, TVD and KL metrics.

In comparison to JSD, we observe that both TVD and KL exhibit similar bimodal distribution trends under stricter threshold values. Moreover, neither metric shows substantially different behavior across sampling strategies. In contrast, the $1 - \mathsf{F1}$ metric, while following the same general trend, displays several notable deviations. Specifically, the distributions tend to be more heavily biased toward requiring longer subcontexts to meet threshold requirements. This effect is particularly pronounced in smaller documents when using adaptive sampling, where DaMCL values skew more strongly toward full-context reliance under stricter thresholds.

Table 5: **MCL vs. DaMCL results for LLaMA-3.** GovReport is shown at 96 tokens, CNN and Wiki at 64 tokens.

| Dataset | MCL (%) | DaMCL (%) |
|---|---|---|
| GovReport ($\leq 96$) | 75.19 | 36.36 |
| News Articles ($\leq 64$) | 80.52 | 45.43 |
| Wikipedia ($\leq 64$) | 78.15 | 31.65 |

Our focus on JSD is motivated by several desirable properties that make it especially well-suited for DaMCL. Notably, JSD is a proper distance metric and satisfies the triangle inequality. Furthermore, as a smoothed and symmetric variant of KL divergence, JSD is generally more robust to noise and less sensitive to zero-probability events—properties that are particularly beneficial when working with

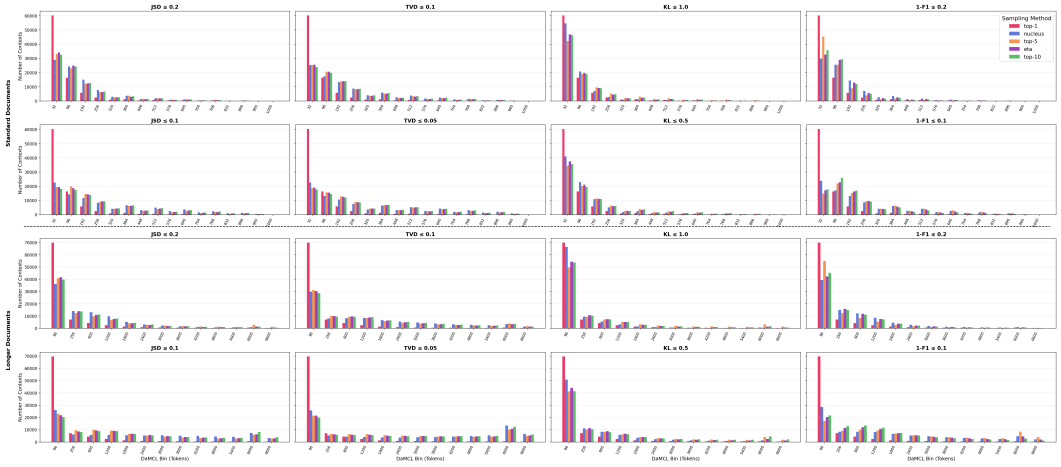



Figure 18: **Distribution of DaMCL:** The distributionally-aware MCL is defined using a range of metrics applied with relative thresholds. Results are presented separately for standard documents (top row) and long-context documents (bottom row) to highlight potential differences in behavior. While the overall trend resembles the exponentially decaying pattern observed in standard MCL (see Fig. 2), the choice of metric and threshold clearly influences the outcome. Each subplot reflects results aggregated over all model–dataset combinations, as only minor deviations were observed across different configurations under identical hyperparameters.

LLM next-token distributions. Nonetheless, we acknowledge that further exploration of alternative metrics may reveal additional insights or complementary advantages. We leave this to future work.

In order to further analyze the behavioral change of DaMCL compared to MCL, note the results provided in Table 5 indicate that compared to MCL, a smaller portion of context queries can be resolved with the short sub-contexts. Additionally, an important question to consider is whether longer contexts compared to shorter ones in the same document have a bias toward lower DaMCL values. To answer such this, we provide Fig. 19 which provides the DaMCL bins per context length for contexts in wikipedia articles. We observe that for longer contexts, the model rarely requires the full length to achieve a close next-token distribution in terms of JSD. In contrast, for shorter contexts ($|\mathbf{s}| \leq 200$), the model more frequently depends on the full context. For a threshold of $0.2$ (Figure a), only a small fraction of longer contexts lead the model to utilize the entire sequence. Under a stricter threshold of $0.1$ (Figure b), the distribution becomes more uniform. This suggests that the bias towards smaller MCL/DaMCL bins is something that exists in a document exists irrespective of the full length of the context itself.

# F Long-Context Sequence Detection

## F.1 Additional Setups and Oracles

Here, we provide further experiments with various setups and oracles that complement the results in Section 5. These additional studies reinforce the effectiveness of JSD as a robust detector of long-context dependence.

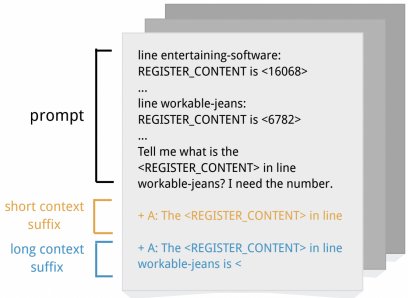
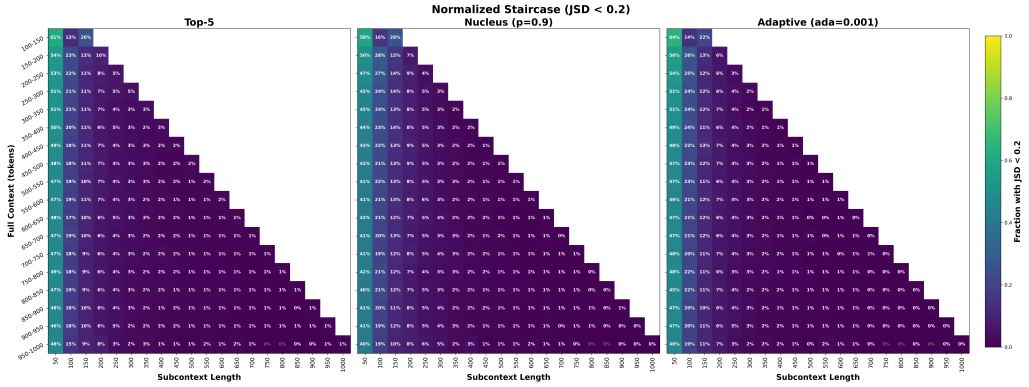


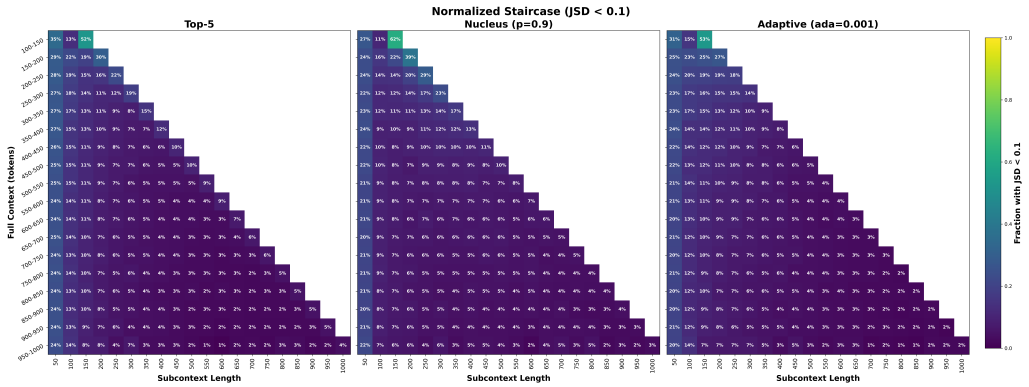

Figure 20: An example of the controlled validation experiment setup on the LongEval benchmark.

### F.1.1 Controlled Validation Experiment on LongEval Task

Using the LongEval benchmark [75], we construct prompts containing multiple register lines, each with a *line_id* and *<REGISTER_CONTENT>* value, where the model must identify specific content. We create two types of queries with known ground-truth labels: *short-context* queries where the answer (a *line_id*) appears in the last 32 tokens, and *long-context* queries where the answer (a *<REGISTER_CONTENT>*) requires information from the full context. We conduct the experiments on three models: LLaMA-3-8B [36], Mistral-7B-Instruct (v0.2) [37], and Qwen2-7B [60]. As shown in Figure 22, across all models, we observe the same distributional trend, confirming the robustness of the results.




(a) JSD $\leq 0.2$




(b) JSD $\leq 0.1$

Figure 19: **DaMCL heatmaps with fixed sub-contexts.** Each row represents contexts of a certain size and each column represents sub-contexts tested for different JSD thresholds. Subplots show results for (a) JSD $\leq 0.2$ and (b) JSD $\leq 0.1$.

### F.1.2 Needle-in-a-Haystack Experiment on General Text

We next adapt a needle-in-a-haystack (NIAH) style setup to general text. Following the classic needle-in-a-haystack test by [45], given a natural context, we insert a needle statement with a randomly generated 6-digit number: *"The magic number is xxxxxx"* at a specific position in the text. At the end of the document, we append a query prompt: *"The magic number mentioned in the provided text is "*, expecting the model to output the correct number from the needle. If the needle statement is placed within the final 32 tokens, the answer is recoverable from the local suffix, and we label the case as a short context. Otherwise, when the needle is inserted far away, answering requires long-range recall, and we label it as a long context. We only retain cases where the model outputs the correct number, ensuring the prediction truly relies on the inserted statement rather than hallucination.

Figure 21 presents the JSD distributions under this setup, showing that long-context tokens yield consistently higher JSD values compared to short-context tokens, validating JSD as a detector in this natural-text scenario.

### F.1.3 Additional Long-short context detection on LongEval

In addition to the setup described above, which for the most part follows a similar structure to that of [4], we consider a different setup where instead of using the *line_id* as the token representing a short context token, we use *<REGISTER_CONTENT>* in a recent line (less than 32 tokens away). Compared to above, the *long-context* queries remain the same, but for the *short-context* queries we use a *<REGISTER_CONTENT>* from a recent line. This allows us to have a more direct comparison between long and short context as the nature of both tokens are the same (same style of registered

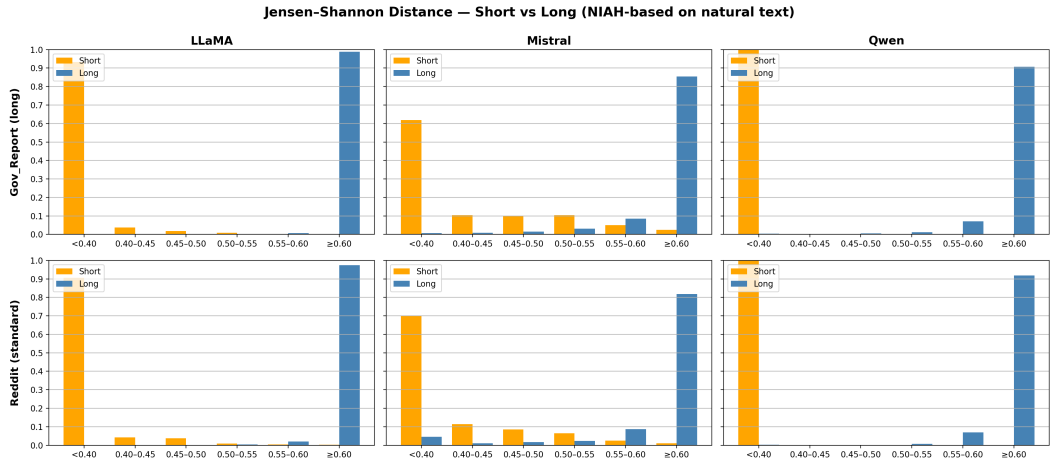



Figure 21: **NIAH-based experiment on General text result:** LSDS distribution across three models (LLaMA, Mistral, Qwen) on long-context dataset (GovReport) and standard-length context (Reddit), each with 10000 samples. Each subplot shows the proportion of tokens falling into fixed JSD bins.

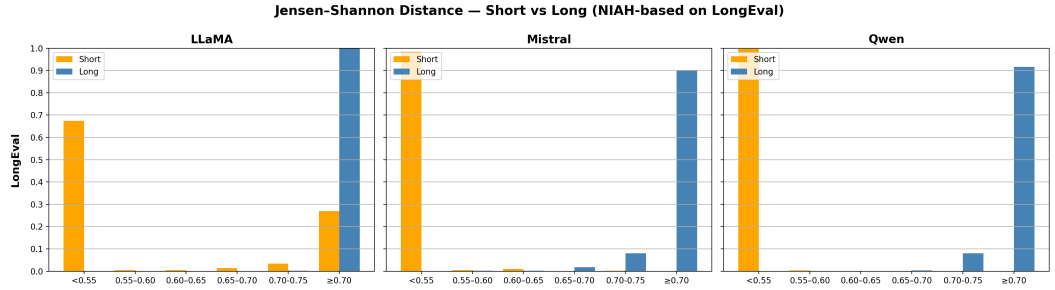



Figure 22: **Controlled validation experiment on LongEval task result:** LSDS distribution across three models (LLaMA, Mistral, Qwen) on the LongEval synthetic benchmark. Each subplot shows the proportion of tokens falling into fixed JSD bins.

content as opposed to comparing to line-id) while the positions of the reference line in the local or long context position changes their LSDS behavior.

the only other difference compared to our previous setup is how here, we consider the last 64 tokens as local context, only to ensure that a full line from a local context will always fit inside the short context window. Results are provided in Fig. 23 and illustrate how this setup complies with our previous observations.

### F.1.4 Prior-Work Oracle (LSD/LCL) on General Text

As described in Section 5, we evaluate a *prior-work oracle* detector that follows the log-probability intervention from [4]. The oracle Long-Short Difference (LSD) and Long-Context Likelihood (LCL) are defined as follows:
For sequence $\mathbf{s}$ with corpus next-token $t$, and a language model $P_\theta$,

$$\mathrm{LSD}_\theta(\mathbf{s}|t) = \log\left[\mathbf{p}(\mathbf{s})\right]_t \; - \; \log\left[\mathbf{p}(\mathbf{s}_{[-32:]})\right]_t \tag{2}$$

and

$$\mathrm{LCL}_\theta(\mathbf{s}|t) = \log\left[\mathbf{p}(\mathbf{s})\right]_t \tag{3}$$

Figure 24 shows the per-model distributions of LSDS for LLaMA-3-8B, Mistral-7B-Instruct, and Qwen2-7B on the same data used in the main body. The per-model trends mirror the pooled result: using a threshold of $\tau = 0.6$, we find consistently that $80 - 90\%$ of oracle-labeled long-context sequences achieve $\mathsf{LSDS}\,(\mathbf{s}) \geq 0.6$, while fewer than $5 - 10\%$ of short-context sequences exceed this threshold.

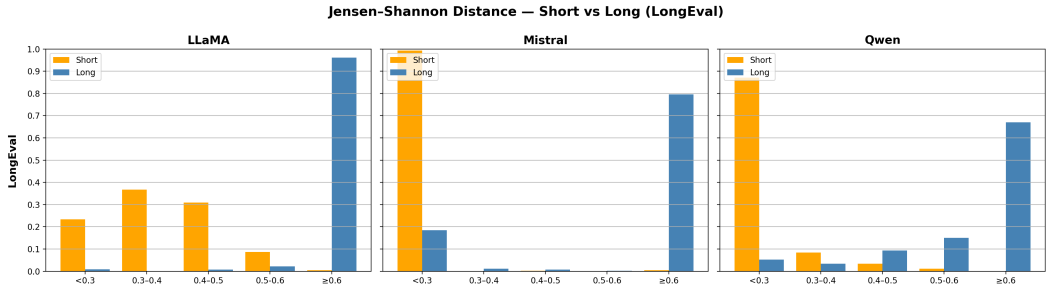



Figure 23: **Controlled validation experiment on LongEval task result with local lines as short-context**: similar results to that presented in 22 except here we use the *<REGISTER_CONTENT>* from local lines as the local context token instead of *line_id*.




Figure 24: **Prior-work oracle (LSD/LCL) on general text result:** LSDS distribution across three models (LLaMA, Mistral, Qwen) on long-context dataset (GovReport) and standard-length context (Reddit), each with 10000 samples. Each subplot shows the proportion of tokens falling into fixed JSD bins.

### F.1.5 MCL Oracle on General Text

Additionally, we evaluate our *MCL Oracle* on general text. Following the definition and computation procedure in Section 3, we compute the MCL of each given context. Tokens with small MCL values ($\leq 32$) are classified as short-context, while those requiring larger suffixes ($> 32$) are classified as long-context. A limitation of this approach is that the MCL is not valid for every token, as MCL only exists for tokens which the model predicts correctly and confidently. Thus we can only perform our analysis on context/token pairs which the model performs well on.

We then compare the JSD distributions of tokens across the two categories. Figure 25 shows that the detected trend is still preserved.

### F.1.6 Quantifying LSDS separation

In addition to the clear visual separation observed in the LSDS distributions for short vs. long contexts, we also report the ROC area under the curve (AUC), which provides a single quantitative measure of binary separability. The ROC curve captures the trade-off between the true-positive and false-positive rates when detecting long-context usage across a range of LSDS thresholds $\epsilon$. A higher AUC corresponds to stronger separability between the LSDS values under long vs. short contexts, thereby validating LSDS as a reliable signal for distinguishing context usage.

Figure 27 (a) shows the resulting AUC values on the GovReport dataset using LLaMA with LSD+LCL as the oracle. We additionally report results on the LongEval task (b,c), where short/long-contexts are formed as examples in Figure 26. Similar trends were observed across other datasets and models but are omitted here for brevity.

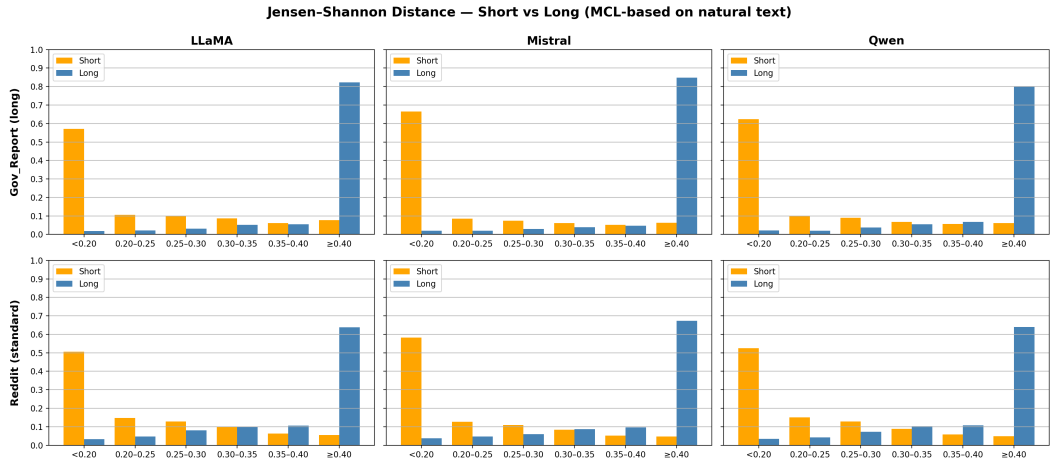



Figure 25: **MCL oracle on general text result:** LSDS distribution across three models (LLaMA, Mistral, Qwen) on long-context dataset (GovReport) and standard-length context (Reddit), each with 10000 samples. Each subplot shows the proportion of tokens falling into fixed JSD bins.

To further quantify detection performance, we select the optimal LSDS threshold $\tau$ by sweeping 500 candidate values over the observed JSD score range and choosing the one that maximises F1 on a stratified 70% validation split, using LSD+LCL as the ground-truth oracle. We evaluate this across three models (LLaMA3, Mistral-v0.3, Qwen3) and two datasets (Reddit, GovReport), with the remaining 30% held out for testing. Table 6 reports the best $\tau$ and corresponding validation metrics. The consistently high accuracy and ROC-AUC across all model–dataset combinations confirm that JSD is a reliable detection metric, and that a simple threshold suffices for accurate long-context classification.

Short Context:
line entertaining-software: REGISTER_CONTENT is <16068>
ine ruthless-closet: REGISTER_CONTENT is <35504>
...
line workable-jeans: REGISTER_CONTENT is <6782>
A: The <REGISTER_CONTENT> in line workable-jeans is <

Long Context:
line entertaining-software: REGISTER_CONTENT is <16068>
ine ruthless-closet: REGISTER_CONTENT is <35504>
...
line workable-jeans: REGISTER_CONTENT is <6782>
...
...
...
A: The <REGISTER_CONTENT> in line workable-jeans is <

Figure 26: An example of another synthetic setup on the LongEval benchmark.

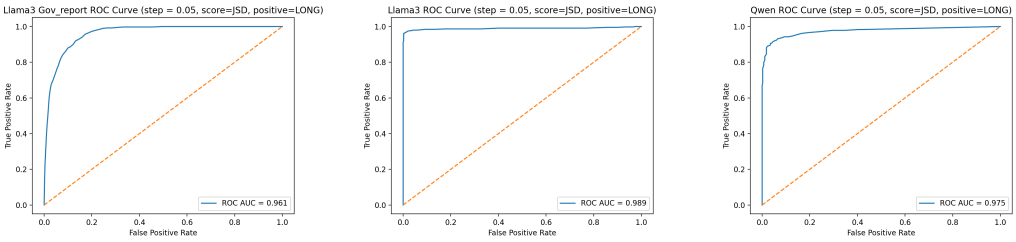
Table 6: Best threshold $\tau$ and validation metrics for LSDS-based long-context detection across models and datasets. Ground truth labels are provided by the LSD+LCL oracle.

| Model | Dataset | Best $\tau$ | Accuracy | F1 | Precision | Recall | ROC-AUC |
|---|---|---|---|---|---|---|---|
| LLaMA3 | Reddit | 0.6925 | 0.9476 | 0.6580 | 0.6513 | 0.6648 | 0.9606 |
| LLaMA3 | GovReport | 0.6925 | 0.9476 | 0.6580 | 0.6513 | 0.6648 | 0.9606 |
| Mistral-v0.3 | Reddit | 0.6807 | 0.9719 | 0.5670 | 0.5331 | 0.6056 | 0.9751 |
| Mistral-v0.3 | GovReport | 0.6925 | 0.9574 | 0.7296 | 0.7065 | 0.7542 | 0.9733 |
| Qwen3 | Reddit | 0.6908 | 0.9740 | 0.5826 | 0.5270 | 0.6513 | 0.9714 |
| Qwen3 | GovReport | 0.6758 | 0.9473 | 0.7060 | 0.6602 | 0.7586 | 0.9688 |

### F.2 Ablation on Nucleus Sampling Parameter ($p$)

In this section, we examine how varying the nucleus sampling parameter $p$ influence our LSDS and the resulting JSD threshold. We report results for 4 settings: no nucleus sampling, $p = 0.95$, $p = 0.90$, and $p = 0.80$. We conduct the experiment with Mistral-7B-Instruct (v0.2) [37] on 5000 contexts randomly selected from Government Report [31], with full context length ranging from 100–1000 tokens. The ground truth oracle we use here is *Prior Work Oracle* as introduced in Sec. 5.

Across all settings, we observe that the separation between short- and long-context distribution is preserved (see Figure 28). Regardless of $p$, short-context examples consistently concentrate in low




(a) LLaMA 3 on Gov Reports

(b) LLaMA 3 on LongEval

(c) Qwen 3 on LongEval

Figure 27: **AUC for LSDS:** ROC area under curve for LSDS accuracy for capturing the long contexts. For **(a)** we use the real data with LCL+LSD as oracle and for **(b,c)** we use the LongEval dataset.

JSD bins ($< 0.40$), while long-context examples dominate the high JSD bins ($\geq 0.60$). However, the optimal detection thresholds vary slightly depending on $p$ value.

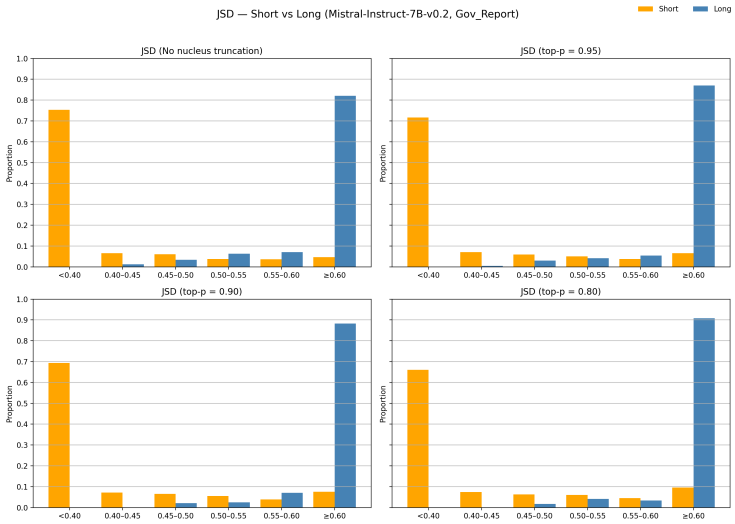



Figure 28: LSDS distribution with Mistral-7B-Instruct-v0.2 on GovReport(5000 samples) with no nucleus sampling, $p = 0.95$, $p = 0.90$, and $p = 0.80$.

To quantitatively assess performance, we compute the maximized Youden's $J$ statistic [76]:

$$J = \max_{\theta} \{\text{TPR}(\theta) - \text{FPR}(\theta)\}$$

where TPR is the true positive rate and FPR is the false positive rate, and $\theta$ is the decision threshold. Unlike accuracy, which can be biased by imbalanced class distributions, $J$ simultaneously accounts for both sensitivity (true positive rate) and specificity (true negative rate). In our case, detecting long- versus short-context tokens requires balancing correct identification of long-context samples with avoidance of false detections. Thus Youden's $J$ provides an appropriate criterion for threshold selection. Table 7 summarizes the best thresholds and corresponding performance metrics.

The setting $p = 0.90$ achieves the highest Youden's $J$ (0.840), indicating the most balanced trade-off between sensitivity (TPR) and specificity ($1 - \text{FPR}$). While $p = 0.80$ achieves the highest accuracy (91.9%), its $J$ value is lower, reflecting weaker overall discriminative balance. Therefore, we adopt $p = 0.90$ as our standard nucleus sampling configuration in the main experiments. This choice is consistent with prior work recommending $p$ in the 0.9–0.95 range for stable yet diverse generation quality [39].

Table 7: Ablation study on different nucleus sampling parameters ($p$). Reported are the optimal threshold, maximized Youden's $J$, true positive rate (TPR), false positive rate (FPR), precision, recall, and accuracy.

| Case | Threshold | $J$ | TPR | FPR | Precision | Recall | Accuracy |
|---|---|---|---|---|---|---|---|
| No nucleus | 0.49 | 0.834 | 0.967 | 0.133 | 0.268 | 0.967 | 87.2% |
| top-$p$=0.95 | 0.53 | 0.831 | 0.950 | 0.119 | 0.286 | 0.950 | 88.4% |
| top-$p$=0.90 | 0.55 | **0.840** | 0.954 | 0.114 | 0.296 | 0.954 | 88.9% |
| top-$p$=0.80 | 0.62 | 0.816 | 0.895 | 0.079 | 0.361 | 0.895 | **91.9%** |

### F.3 Ablation on Short-Context Length

We investigate the effect of varying the short-context prefix length $\ell$ on LSDS and JSD threshold. Experiments are run with Mistral-7B-v0.3 on Government Report, using 5000 randomly selected samples, with full context length ranging from 100–1000 tokens. The ground truth oracle we use here is *Prior Work Oracle* as introduced in Sec. 5, and the short prefix length in Equation 2 is adjusted accordingly when computing LSD.

**Different fixed short-prefix length.** Figure 30 shows results for $\ell = 8, 16, 32, 64$. We observe that the overall distributional separation between short and long contexts is preserved across all choices. However, very small short-context prefixes (e.g., $\ell = 8$) produce more imbalanced distributions: nearly 20% of short-context tokens fall above the detection threshold. This indicates that while separation is robust, extremely small short prefixes may introduce higher false-positive rates. In contrast, $\ell = 32, 64$ yield clearer separation, with false negative rates less than 5%, suggesting they are preferable for stable detection. Although using longer prefixes incurs slightly higher computational cost, the improved robustness makes the tradeoff worthwhile—especially in applications where false positives are costly.

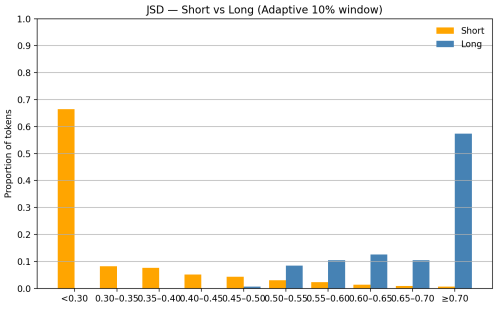



Figure 29: LSDS distribution with Mistral-7B-v0.3 on GovReport(5000 samples) with short prefix length $\ell = 0.1|s|$, consistent separation of the distributions is preserved.

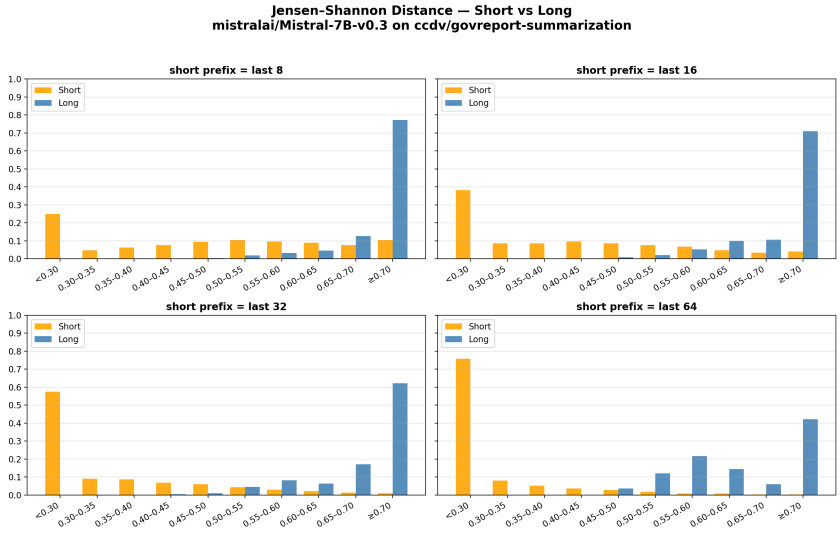



Figure 30: LSDS distribution with Mistral-7B-v0.3 on GovReport(5000 samples) with short prefix length $\ell = 8, 16, 32, 64$.

**Adaptive Short-Context Length.** We also evaluate an adaptive strategy where the short-context length is chosen as a proportion of the full sequence length, $\ell = 0.1|s|$. Figure 29 demonstrates consistent separation trends under this setup, while adapting naturally to dataset-specific context lengths. Such adaptive schemes may provide better cross-dataset generalization, although their computational implications require further study. We leave exploration of such direction for future work.

### F.4 Computational overhead of LSDS detection and TaBoo

We measure the cost of TaBoo for multi-token generation. TaBoo runs the full and short contexts in the same batch, using a standard growing KV-cache for the full context and a fixed-size sliding-window cache for the short context (always the last 32 tokens, with the oldest token evicted at each step). We generate 100 tokens from a 5K-token GovReport context, average over 20 runs on an H100 GPU, and report the overhead of TaBoo relative to vanilla decoding on the same context.

Table 8: Overhead of TaBoo relative to vanilla decoding (100-token generation from a 5K-token context; average of 20 runs on an H100). Time/token, GFLOPs, and peak memory are measured with the SDPA attention kernel; the last column reports time/token with FlashAttention. "–": not measured.

| | | SDPA | | | FlashAttention |
|---|---|---|---|---|---|
| Model | Params | Time/token | GFLOPs | Peak mem. | Time/token |
| Qwen3-0.6B | 0.6B | 7.76% | 1.8% | 68.4% | 5.41% |
| Qwen3-1.7B | 1.7B | 9.28% | 1.9% | 41.0% | 4.23% |
| Ministral-3B | 3.8B | 12.63% | 2.0% | 22.8% | – |
| Qwen3-4B | 4.0B | 10.87% | 2.0% | 33.0% | 3.84% |
| Mistral-7B | 7.2B | 15.76% | 2.0% | 9.3% | 4.61% |
| Ministral-8B | 8.0B | 15.83% | 2.0% | 13.4% | 4.32% |
| DeepSeek-R1-8B | 8.0B | 18.73% | 2.0% | 13.5% | 4.25% |
| Qwen3-8B | 8.2B | 12.82% | 2.0% | 15.1% | – |
| Mistral-Nemo | 12.2B | 19.42% | 1.9% | 9.9% | 3.74% |
| Qwen3-14B | 14.8B | 20.08% | 2.1% | 9.2% | 2.97% |
| Mistral-Small-24B | 23.6B | 22.36% | 2.1% | 5.4% | 3.58% |

As shown in Table 8, with the SDPA kernel the throughput overhead of TaBoo is at most about 22%. It is largest for the bigger models but does not grow linearly with model size. Roughly 4–6% of this overhead comes from the JSD computation and token boosting; the rest comes from the additional short-context forward pass, which is inherent to any contrastive decoding method. The SDPA kernel also requires extra padding for the short context. With FlashAttention, which avoids this padding, the time-per-token overhead drops to 2.7–5.4% across all models. The GFLOPs overhead stays around 2%, and the relative peak-memory overhead decreases with model size, from 68% for Qwen3-0.6B to 5% for Mistral-Small-24B.

## G Long-Context Token Detection and Boosting

### G.1 Probability Shift as Selection Metric

For our selection of long context relevant tokens, we use the change in the probability values as our determining metric. While we decide on using the probability difference $\mathsf{LSPS}\,(t|\mathbf{s}) = \mathbf{p}_{p=0.9}(\mathbf{s})_t - \mathbf{p}_{p=0.9}(\mathbf{s}_{[-32:]})_t$ as the determining factor, alternative choice could be the use of probability ratios rather than the difference. In other words we can define **Long-Short Probability Ratio**:

$$\mathsf{LSPR}\,(t|\mathbf{s}) = \log\left(\frac{\mathbf{p}_{p=0.9}(\mathbf{s})_t}{\mathbf{p}_{p=0.9}(\mathbf{s}_{[-32:]})_t}\right)$$

A comparison between this form of metric and our probability difference **LSPS** becomes important since works such as [4, 6] use the log probability ratio to detect long-short context/next-tokens. Additionally, in [8] and [9] the authors use the log ratio for probability and logits with in the form

Table 9: Analysis of ground truth $\hat{t}$ token being selected for probability improvement. A comparison between LSPD (diff) and LSPR (log ratio).

(a) LSPD (diff case)

| $\epsilon$ | Best | Bad | Worst | Neutral |
|---|---|---|---|---|
| 0.04 | 66.8% | 10.0% | 14.2% | 9.1% |
| 0.06 | 65.5% | 7.8% | 12.8% | 13.9% |
| 0.08 | 64.3% | 6.4% | 11.1% | 18.2% |
| 0.10 | 62.9% | 5.4% | 9.5% | 22.1% |
| 0.12 | 61.4% | 4.4% | 8.6% | 25.7% |
| 0.14 | 59.8% | 3.6% | 7.7% | 28.9% |
| 0.16 | 58.1% | 3.0% | 6.8% | 32.1% |
| 0.18 | 56.5% | 2.4% | 6.1% | 35.0% |
| 0.20 | 55.0% | 2.0% | 5.6% | 37.5% |
| 0.22 | 53.3% | 1.5% | 5.1% | 40.1% |

(b) LSPR (log ratio case)

| $\epsilon$ | Best | Bad | Worst | Neutral |
|---|---|---|---|---|
| 0.5 | 60.3% | 10.7% | 20.2% | 8.8% |
| 1.0 | 57.7% | 7.2% | 22.1% | 13.0% |
| 1.5 | 56.2% | 5.2% | 23.1% | 15.4% |
| 2.0 | 54.8% | 4.3% | 23.9% | 17.0% |
| 2.5 | 53.2% | 3.7% | 24.8% | 18.3% |
| 3.0 | 51.7% | 3.1% | 25.8% | 19.5% |
| 3.5 | 49.7% | 2.6% | 27.5% | 20.3% |
| 4.0 | 46.8% | 2.0% | 29.4% | 21.8% |
| 4.5 | 43.2% | 1.6% | 31.5% | 23.7% |
| 5.0 | 38.5% | 1.4% | 34.2% | 26.0% |

of $\mathbf{p}(\mathbf{s})_t^{1.5}\mathbf{p}_{[-32:]}(\mathbf{s})_t^{0.5}(t)$ to improve the final probability of long-context tokens. Implicitly, such methods select and increase the probability of tokens who's probability ratio increases rather than the difference itself.

In order to compare the choice of metric, we compare the use of **LSPS** and **LSPR** when boosting the final word/token of on the validation set of LAMBADA [77] dataset. For each context, we have a ground truth next token, which we call the target token $\hat{t}$. All other tokens we call incorrect tokens. We analyze the choice of which token's are selected for probability increase rather than the improvement itself, therefore we don't include multiplicative factors like $\lambda$. For each case of boosting, we consider the following scenarios:

- Scenario 1 (Best): $\hat{t} \in \mathcal{B}$ and $\mathbf{p}_{p=0.9}(\mathbf{s})_{\hat{t}} \geq \mathbf{p}_{p=0.9}(\mathbf{s})_t$ for any other $t \in \mathcal{B}$. Under such a scenario, even if some non answer tokens are selected for probability increase, if we grow $\lambda$ eventually greedy sampling would select select the answer token.
- Scenario 2 (Bad): $\hat{t} \in \mathcal{B}$ however, there exists at least one token $t \in \mathcal{B}$ with a higher probability than the answer token $\mathbf{p}_{p=0.9}(\mathbf{s})_{\hat{t}} \geq \mathbf{p}_{p=0.9}(\mathbf{s})_t \; \forall \; t \in \mathcal{B}$. Under such a scenario, while a higher $\lambda$ value increases the probability of $\hat{t}$ and thus improving perplexity, but there isn't a possibility of the ground token to be selected by a greedy sampler.
- Scenario 3 (Worst): $\hat{t} \notin \mathcal{B} \neq \varnothing$. In this case we didn't detect the ground truth token but rather chose a set of incorrect tokens. Under such circumstances, the model next token perplexity would suffer.
- Scenario 4 (Neutral): $\mathcal{B} = \varnothing$ and so no change is made to the probability distribution.

We use nucleus sampling with $p = 0.9$ and set a minimum probability of $10^{-6}$ as the minimum, in order to prevent numerical instability for the log probability calculations. We will select a range of $\epsilon$ values as the selection threshold with respect to the metrics themselves. We report the ratio of each of above's scenarios from the 5153 examples in the LAMBADA validation dataset.

We observe that using log probability ratio is more likely to lead to a wrong token selection for the boosted set without including the target token $\hat{t}$. This case is particularly dangerous as it negatively impacts a metric such as perplexity. Additionally, results suggest that the selection performance highly sensitive to the choice of $\epsilon$.

Such observations can be explained by the sensitivity of log ratio. Particularly for tokens with near zero probability, small perturbation when moving from short to long context probability can be registered under large change in probability ratio changes. Consider using $\log_{10}$ for this explanation only (experiments are done with natural log). If a token's probability goes from $\mathbf{p}_{p=0.9}(\mathbf{s}_{[-32:]})_t = 10^{-6} \rightarrow \mathbf{p}_{p=0.9}(\mathbf{s})_t = 10^{-2}$, the log probability ratio is set to $4$. Similarly if the token's probability goes from $10^{-3} \rightarrow 10$ we have a similar probability ratio change. However, for the first scenario the full context token probability is near zero, while for the second case the token has a $10\%$ probability. If we go with a the log ratio, both these tokens are treated equally.

On the other hand, for the probability difference case, we have implicitly set a minimal lower bound on the $\mathbf{p}_{p=0.9}(\mathbf{s})_t \geq \epsilon$ by only selecting tokens where $|\mathbf{p}_{p=0.9}(\mathbf{s})_t - \mathbf{p}_{p=0.9}(\mathbf{s}_{[-32:]})_t| \geq \epsilon$. This allows us to circumvent tokens who have very low probability where the log ratio behavior could be noisy and misleading. We believe this is a potential reason for why the long context token selection algorithm in [4] uses $\log [\mathbf{p}(\mathbf{s})]_t \geq 10^{-2}$ ( **Long-Context Likelihood (LCL)** ) in addition to their log probability ratio to classify tokens.

While we find probability difference to be a safer and more effective option for our studies, further analysis could help discover new methods for ad-hoc detection of long-context relevant token by comparing short-long context distributions. We believe this can serve a direction for future research.

### G.2 Taboo Algorithm

Algorithm 1 presents our TaBoo method.

**Algorithm 1** Long Context Token Boosting with JSD and Nucleus Constraint

**Input:** Short- and full-context distributions $\mathbf{p}_\phi(\mathbf{s}_{[-32:]})$, $\mathbf{p}_\phi(\mathbf{s})$; thresholds $\epsilon$, $\tau$; boost factor $\lambda$; decoding $\phi$: nucleus sampling with parameter $p$; support set $\mathcal{V}^{\text{nuc}}$ of $\mathbf{p}_\phi(\mathbf{s})$
**Output:** TaBoo distribution $\widetilde{\mathbf{p}}_\phi(\mathbf{s})$

Initialize (to the raw probability distribution): $\widetilde{\mathbf{p}}(\mathbf{s}) \leftarrow \mathbf{p}_\phi(\mathbf{s})$

**Step 1:** Identify whether sequence is long-context:
**if** $\mathsf{LSDS}(\mathbf{s}) \leq \tau$ **then**
  **return** $\mathbf{p}_\phi(\mathbf{s})$  // Return unmodified nucleus distribution
**end if**

**Step 2:** Identify set of long-context-relevant tokens:
$\mathcal{B} \leftarrow \{t \in \mathcal{V}^{\text{nuc}} \mid \mathsf{LSPS}(t|\mathbf{s}) > \epsilon\}$

**Step 3:** Targeted Boosting:
**for** each token $t \in \mathcal{B}$ **do**
  $[\widetilde{\mathbf{p}}(\mathbf{s})]_t \leftarrow \lambda \cdot [\mathbf{p}_\phi(\mathbf{s})]_t$
**end for**

Re-normalize and apply nucleus decoding: $\widetilde{\mathbf{p}}(\mathbf{s}) \overset{\text{nuc}}{\leftarrow} \widetilde{\mathbf{p}}(\mathbf{s}) / \operatorname{sum}(\widetilde{\mathbf{p}}(\mathbf{s}))$ for all $t \in \mathcal{V}$

**return** $\widetilde{\mathbf{p}}(\mathbf{s})$

### G.3 Ablation Study: JSD Detection Threshold in TaBoo

To justify the choice of JSD detection threshold $\tau$ in the TaBoo algorithm, we run an ablation on a 500-example subset of NarrativeQA, comparing three values $\tau \in \{0.1, 0.3, 0.6\}$ against Vanilla nucleus sampling and CAD baselines. Results are reported in Table 11.

Table 10: Similar setup to Tab. 9 but this time without any nucleus sampling.

(a) LSPD (diff case)

| $\epsilon$ | Best | Bad | Worst | Neutral |
|---|---|---|---|---|
| 0.04 | 66.7% | 9.3% | 13.8% | 9.1% |
| 0.06 | 65.4% | 7.4% | 12.0% | 15.2% |
| 0.08 | 63.9% | 5.9% | 10.2% | 20.0% |
| 0.10 | 62.0% | 4.9% | 8.7% | 24.0% |
| 0.12 | 60.3% | 3.9% | 7.7% | 28.1% |
| 0.14 | 58.6% | 3.2% | 6.6% | 31.7% |
| 0.16 | 56.9% | 2.6% | 5.9% | 34.7% |
| 0.18 | 55.2% | 2.1% | 5.2% | 37.6% |
| 0.20 | 53.6% | 1.6% | 4.8% | 40.9% |
| 0.22 | 51.9% | 1.2% | 4.3% | 42.6% |

(b) LSPR (log ratio case)

| $\epsilon$ | Best | Bad | Worst | Neutral |
|---|---|---|---|---|
| 0.5 | 60.6% | 11.2% | 28.1% | 0.1% |
| 1.0 | 58.5% | 7.5% | 34.6% | 0.3% |
| 1.5 | 57.0% | 5.4% | 38.0% | 0.4% |
| 2.0 | 55.6% | 4.3% | 40.4% | 0.5% |
| 2.5 | 54.0% | 3.6% | 42.7% | 0.6% |
| 3.0 | 52.4% | 3.0% | 45.0% | 0.7% |
| 3.5 | 50.3% | 2.4% | 47.9% | 0.7% |
| 4.0 | 47.3% | 1.7% | 51.6% | 0.8% |
| 4.5 | 43.8% | 1.3% | 56.4% | 0.9% |
| 5.0 | 39.0% | 1.0% | 61.7% | 0.9% |

Table 11: TaBoo performance on NarrativeQA (500 examples) under different JSD detection thresholds $\tau$. All other hyperparameters are fixed ($\lambda = 4$, $\epsilon = 0.05$).

| Method | Avg F1 | Best F1 | Avg BLEU | Best BLEU | Avg R-L | Best R-L |
|---|---|---|---|---|---|---|
| Vanilla | 0.3603 | 0.5445 | 0.0996 | 0.1688 | 0.4719 | 0.6860 |
| CAD | 0.3410 | 0.4772 | 0.0874 | 0.1399 | 0.4582 | 0.6088 |
| TaBoo $\tau$=0.1 | 0.4051 | 0.5298 | 0.1110 | 0.1627 | 0.5300 | 0.6661 |
| TaBoo $\tau$=0.3 | 0.3919 | 0.5386 | 0.1043 | 0.1606 | 0.5135 | 0.6697 |
| TaBoo $\tau$=0.6 | 0.3921 | 0.5262 | 0.1039 | 0.1602 | 0.5123 | 0.6692 |

Across all three values of $\tau$, TaBoo consistently outperforms both Vanilla and CAD on average F1 and ROUGE-L, confirming that the boosting mechanism is beneficial regardless of the exact threshold. The differences between threshold values are small — Avg F1 ranges from 0.39 to 0.41 and Avg ROUGE-L from 0.51 to 0.53 — suggesting the method is robust to this choice. We set $\tau = 0.12$ in our main experiments as it yields a relatively stronger average performance while remaining conservative enough to avoid over-triggering boosting on short-context sequences.

### G.4 Analysis on additional datasets/models

In Table . 12 we provde standard error results for the experiments in Table . 1. Additionally, we results for Q/A and summarization in Table . 14 and Table . 13. For Q/A we use subsets from L-Eval [78], namely *NaturalQuestions*, *MultiDocQA* and *ScientificQA*. To ensure that the QA tasks genuinely require contextual information (i.e., cannot be answered in a closed-book setting), we first run each model on the question alone and remove any example for which the model achieves an **F1** score of $\geq 0.3$. As shown in Tab. 14, both models find these Q/A tasks challenging due to their longer contexts and higher difficulty. For summairzation, we consider CNN/DailyMail [28], SamSum [79] and XSUM [80]. For each case, we sample 100 summarization examples from the target dataset, at 5 generations per sample. Finally note that for both summarization and generation, we allow up to 2 times as many token generation as the gold answer. This is to prevent the model from generating too many tokens and holding a standard on the context length. For AdaCAD, a $\lambda = 0.3$ warmup period is suggested by the original authors for longer generations, which we only apply during longer summarizations. Overall the results indicate that TaBoo remains competitive with CAD and AdaCAD and outperforms Vanilla for Q/A. For longer generations on summarizations, CAD and TaBoo provide generally similar performance with AdaCAD providing similar results.

Table 12: Standard errors (SE) for Table 1. Values shown as Average (±SE). NarrativeQA and HotpotQA: $n$=1000 examples. MultifieldQA: $n$=150 examples.

| Model | Method | NarrativeQA | | | HotpotQA | | | MultifieldQA-en | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | F1(↑) | BLEU(↑) | ROUGE-L(↑) | F1(↑) | BLEU(↑) | ROUGE-L(↑) | F1(↑) | BLEU(↑) | ROUGE-L(↑) |
| Llama-3.1-8B | Vanilla | 26.3 (±0.99) | 5.5 (±0.35) | 35.5 (±1.18) | 26.3 (±1.18) | 8.4 (±0.58) | 38.1 (±1.40) | 26.0 (±1.50) | 9.5 (±1.13) | 33.2 (±1.57) |
| | CAD | 34.1 (±1.32) | 6.9 (±0.42) | 45.5 (±1.51) | 22.2 (±1.07) | 6.9 (±0.50) | 39.4 (±1.47) | 26.5 (±1.63) | 10.0 (±1.04) | 33.8 (±1.82) |
| | AdaCAD | 33.9 (±1.31) | 6.8 (±0.40) | 45.3 (±1.50) | 21.7 (±1.11) | 6.9 (±0.52) | 39.3 (±1.55) | 26.7 (±1.66) | 10.5 (±1.18) | 34.5 (±1.79) |
| | TaBoo | 35.4 (±1.29) | 8.1 (±0.48) | 45.1 (±1.39) | 28.9 (±1.26) | 9.5 (±0.63) | 42.1 (±1.48) | 27.7 (±1.57) | 11.1 (±1.22) | 35.6 (±1.66) |
| Mistral-7B-v0.3 | Vanilla | 26.4 (±1.06) | 5.6 (±0.40) | 35.1 (±1.26) | 32.0 (±1.33) | 10.2 (±0.61) | 41.9 (±1.47) | 23.3 (±1.46) | 8.2 (±1.00) | 31.0 (±1.66) |
| | CAD | 33.8 (±1.38) | 6.9 (±0.43) | 41.9 (±1.46) | 33.8 (±1.48) | 11.0 (±0.69) | 40.8 (±1.54) | 24.1 (±1.48) | 8.3 (±1.02) | 31.9 (±1.71) |
| | AdaCAD | 33.1 (±1.30) | 6.7 (±0.41) | 42.3 (±1.44) | 33.5 (±1.51) | 10.7 (±0.68) | 41.2 (±1.58) | 25.0 (±1.60) | 9.4 (±1.11) | 32.9 (±1.77) |
| | TaBoo | 36.0 (±1.35) | 7.9 (±0.48) | 44.8 (±1.42) | 35.7 (±1.44) | 11.5 (±0.67) | 46.5 (±1.57) | 26.6 (±1.54) | 9.7 (±1.10) | 34.8 (±1.72) |
| Qwen3-8B-Base | Vanilla | 43.0 (±1.40) | 11.2 (±0.62) | 53.4 (±1.45) | 52.8 (±1.73) | 18.5 (±0.95) | 59.1 (±1.64) | 35.6 (±1.99) | 18.3 (±1.97) | 42.9 (±2.00) |
| | CAD | 41.5 (±1.40) | 10.8 (±0.63) | 53.5 (±1.50) | 44.2 (±1.71) | 15.8 (±0.95) | 53.5 (±1.68) | 35.9 (±2.07) | 18.8 (±1.85) | 43.4 (±2.19) |
| | AdaCAD | 41.2 (±1.39) | 10.5 (±0.62) | 53.6 (±1.51) | 45.1 (±1.73) | 15.7 (±0.90) | 53.7 (±1.69) | 36.7 (±2.16) | 19.0 (±1.96) | 44.8 (±2.29) |
| | TaBoo | 45.1 (±1.48) | 11.9 (±0.67) | 56.0 (±1.50) | 55.4 (±1.80) | 19.5 (±1.00) | 63.4 (±1.68) | 37.5 (±2.00) | 19.3 (±1.88) | 48.0 (±2.16) |

### G.5 Impact of Decoding

Another question we ask is the impact of using decoding methods when implementing the token selection during Taboo. In order to test this out, we provide a similar table to Tab . 9 but this time without any nucleus sampling and using the raw probability distributions.

Results are provided in Tab . 10. Comparing results when using LSPS, we don't observe a major change, one case being better than the other depending on the value of $\epsilon$. For LSPR on the other hand, we can see that not doing any nucleus sampling leads to much more **Worst** case scenarios

Table 13: Summarization performance across methods and datasets. Main numbers are average scores; numbers in parentheses are best scores. Bold marks the best average within each model block and dataset/metric; underlining marks the best corresponding best-score value.

| Model | Method | CNN/DailyMail | | | SAMSum | | | XSUM | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | F1(↑) | BLEU(↑) | ROUGE-L(↑) | F1(↑) | BLEU(↑) | ROUGE-L(↑) | F1(↑) | BLEU(↑) | ROUGE-L(↑) |
| Mistral-7B | Vanilla | 20.3 (25.5) | 3.7 (6.1) | 19.5 (24.5) | 17.9 (29.6) | 2.0 (6.1) | 18.1 (29.7) | 16.6 (24.7) | 1.6 (3.9) | 15.4 (22.4) |
| | CAD | 24.1 (32.4) | 5.6 (10.5) | 21.7 (30.8) | **26.0** (38.5) | **4.0** (10.7) | **26.4** (39.8) | 17.2 (27.2) | **1.9** (4.7) | 15.6 (25.3) |
| | AdaCAD | **24.4** (30.9) | 5.9 (10.3) | **22.3** (28.4) | 25.4 (38.2) | 3.9 (10.2) | 25.9 (39.3) | 17.0 (25.8) | 1.8 (4.1) | 15.8 (23.8) |
| | TaBoo | **24.4** (31.4) | **6.2** (10.2) | 22.2 (29.1) | 25.6 (39.0) | 3.9 (10.8) | 24.8 (37.4) | **17.4** (26.1) | 1.8 (3.0) | **15.9** (23.5) |
| Qwen3-8B | Vanilla | 21.8 (27.2) | 3.2 (5.9) | 20.7 (26.4) | 23.8 (32.6) | 3.1 (7.1) | 24.7 (34.0) | 18.0 (23.7) | 1.6 (2.9) | **17.2** (23.1) |
| | CAD | **22.8** (28.0) | **3.5** (6.2) | **20.9** (26.1) | **25.2** (33.9) | **3.9** (8.4) | **26.0** (35.7) | 18.1 (23.6) | 1.6 (2.6) | 16.9 (22.1) |
| | AdaCAD | 22.6 (27.7) | 3.4 (6.2) | 20.8 (25.6) | 24.5 (32.8) | 3.6 (8.0) | 25.8 (35.0) | 18.1 (23.5) | 1.5 (2.4) | 16.8 (21.7) |
| | TaBoo | 22.7 (28.1) | 3.2 (5.9) | 20.8 (25.9) | 25.1 (33.9) | 3.6 (7.9) | **26.0** (35.1) | **18.4** (24.0) | **1.7** (2.7) | **17.2** (23.4) |
| Llama-3.1-8B | Vanilla | 20.4 (30.3) | 3.7 (9.8) | 18.9 (29.1) | 21.3 (36.2) | 3.0 (10.1) | 21.5 (37.3) | 17.2 (27.5) | 1.6 (4.4) | 15.5 (24.9) |
| | CAD | 24.6 (32.5) | 5.8 (11.5) | 22.6 (31.7) | **29.1** (44.6) | 4.9 (14.8) | **30.1** (44.9) | 17.6 (26.7) | 1.6 (3.9) | 15.3 (23.1) |
| | AdaCAD | 24.4 (32.7) | 5.7 (11.7) | 22.5 (31.7) | 28.6 (41.7) | 4.7 (12.4) | 29.2 (43.3) | 17.7 (26.5) | 1.5 (3.6) | 15.5 (24.2) |
| | TaBoo | **25.1** (32.9) | **5.9** (12.1) | **23.2** (32.2) | 28.9 (42.1) | **5.1** (14.0) | 30.0 (43.0) | **17.8** (27.2) | **1.7** (4.4) | **15.8** (24.5) |

Table 14: F1, BLEU, and ROUGE-L (Average; standard error in parentheses) on additional datasets. Bold = best Average within each dataset block.

| Model | Method | NaturalQuestions | | | MultiDocQA | | | ScientificQA | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | F1(↑) | BLEU(↑) | ROUGE-L(↑) | F1(↑) | BLEU(↑) | ROUGE-L(↑) | F1(↑) | BLEU(↑) | ROUGE-L(↑) |
| Qwen3-8B | Vanilla | 37.06 (±3.70) | 11.79 (±1.64) | 41.94 (±3.90) | **21.13** (±1.16) | **6.33** (±0.84) | **23.32** (±1.35) | 22.85 (±1.53) | 7.48 (±0.90) | 30.94 (±1.78) |
| | CAD | 34.19 (±3.62) | 11.17 (±1.68) | 37.67 (±3.76) | 18.69 (±1.14) | 5.17 (±0.72) | 20.79 (±1.33) | 22.52 (±1.58) | 7.08 (±0.86) | 29.55 (±1.86) |
| | AdaCAD | 31.05 (±3.47) | 9.00 (±1.50) | 34.84 (±3.63) | 19.49 (±1.26) | 5.80 (±0.84) | 21.37 (±1.41) | **24.24** (±1.63) | **8.18** (±1.02) | 31.43 (±1.86) |
| | TaBoo | **39.19** (±3.85) | **12.61** (±1.72) | **43.93** (±3.99) | 20.89 (±1.18) | 6.20 (±0.79) | 23.18 (±1.33) | 24.22 (±1.66) | 8.10 (±1.01) | **32.44** (±1.94) |
| Mistral-7B | Vanilla | 19.90 (±1.96) | 4.96 (±0.62) | 23.19 (±2.09) | 11.97 (±0.64) | 3.08 (±0.50) | 12.76 (±0.76) | 14.85 (±0.91) | 3.92 (±0.47) | 20.33 (±1.14) |
| | CAD | 11.66 (±1.55) | 3.17 (±0.52) | 14.16 (±1.75) | 8.42 (±0.76) | 3.05 (±0.61) | 8.98 (±0.85) | 17.56 (±1.20) | 5.23 (±0.69) | 23.56 (±1.50) |
| | AdaCAD | 19.32 (±2.13) | 4.95 (±0.67) | 22.32 (±2.27) | 9.55 (±0.59) | 2.71 (±0.42) | 9.89 (±0.65) | 17.11 (±1.26) | 5.25 (±0.73) | 23.53 (±1.48) |
| | TaBoo | **28.06** (±2.83) | **7.93** (±1.11) | **31.36** (±3.02) | **12.45** (±0.91) | **4.37** (±0.80) | **13.40** (±1.04) | **21.35** (±1.28) | **6.18** (±0.76) | **27.70** (±1.59) |
| Llama-3.1-8B | Vanilla | 16.03 (±1.77) | 4.77 (±0.81) | 20.03 (±2.11) | 14.54 (±0.90) | 5.09 (±0.77) | 15.74 (±1.05) | 14.26 (±0.86) | 3.92 (±0.51) | 21.07 (±1.05) |
| | CAD | 13.48 (±1.77) | 4.05 (±0.72) | 19.23 (±2.40) | 9.91 (±0.79) | 2.66 (±0.63) | 9.56 (±0.89) | 15.19 (±1.02) | 4.80 (±0.61) | 21.39 (±1.24) |
| | AdaCAD | 17.47 (±2.13) | 4.86 (±0.80) | 22.47 (±2.49) | 10.22 (±0.71) | 2.82 (±0.53) | 10.21 (±0.79) | 16.60 (±1.08) | 5.57 (±0.70) | 23.71 (±1.26) |
| | TaBoo | **24.52** (±2.69) | **7.10** (±1.09) | **27.95** (±2.87) | **15.46** (±1.19) | **6.81** (±1.03) | **17.02** (±1.39) | **19.63** (±1.22) | **6.96** (±0.87) | **27.37** (±1.48) |

where we don't boost the answer token but rather a irrelevant one. This observation both points towards the robustness of LSPS w.r.t the next token probability distribution (with or without decoding). Additionally it points our the potential problem with using the log ratio of probabilities.

### G.6 Illustrative Example of Long-context Token Boosting on NarrativeQA

Figure 31 shows some examples from the NARRATIVEQA dataset to illustrate how our boosting method (Taboo) operates.

The boosted output generated by Taboo matches the ground truth, with tokens highlighted by context origin: orange tokens correspond to short-context (names from the question), while blue tokens correspond to long-context (the reasoning required from the story). The figure also shows which tokens were boosted, showing that the algorithm consistently promotes the correct words needed to form the right answer. This example illustrates how Taboo leverages long-context signals to reinforce accurate completions.

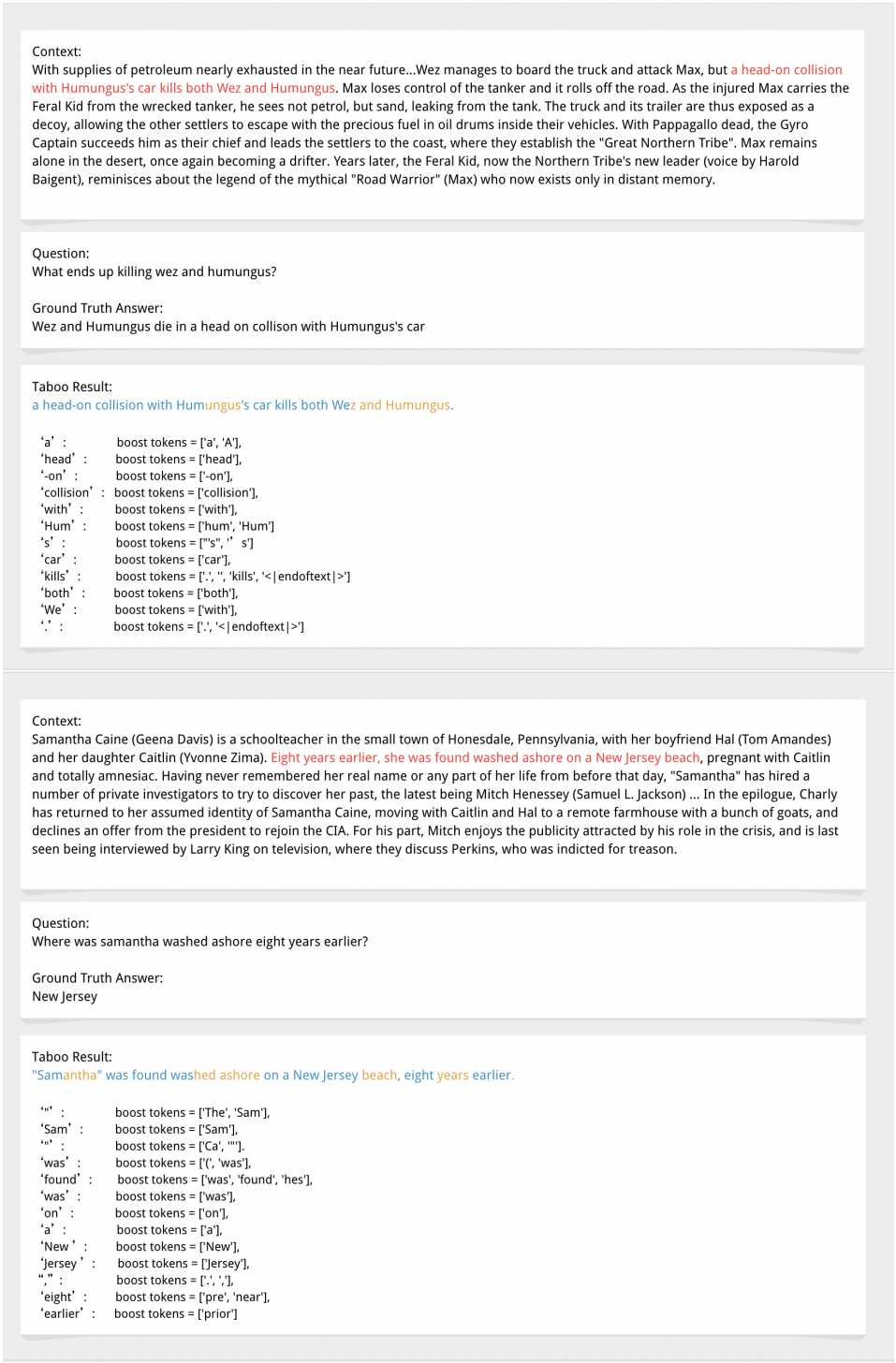



Figure 31: Example from NarrativeQA with Taboo. Orange tokens are short-context (names from the question), blue tokens are long-context (story-based reasoning). Boosted tokens align with the correct answer, showing that Taboo consistently promotes the right completions.

## H Results on Larger Models

To test whether our findings hold at larger scale, we repeat our analysis with Qwen3-14B and Mistral-Small-24B, roughly 2× and 3× the size of the models in the main text, covering both the measurements (MCL and DaMCL, Secs. 3–4) and the intervention (TaBoo, Sec. 6).

**MCL.** Following the setup of Sec. 3, we measure MCL on a long-document dataset (GovReport) and a standard-length one (Writing Prompts). As shown in Fig. 32, the larger models exhibit the same short-context dominance as the smaller ones: 79–83% of sequences have an MCL of at most 64 tokens on Writing Prompts and at most 96 tokens on GovReport, with heavy-tailed distributions and log-log slopes similar to those in Fig. 2.

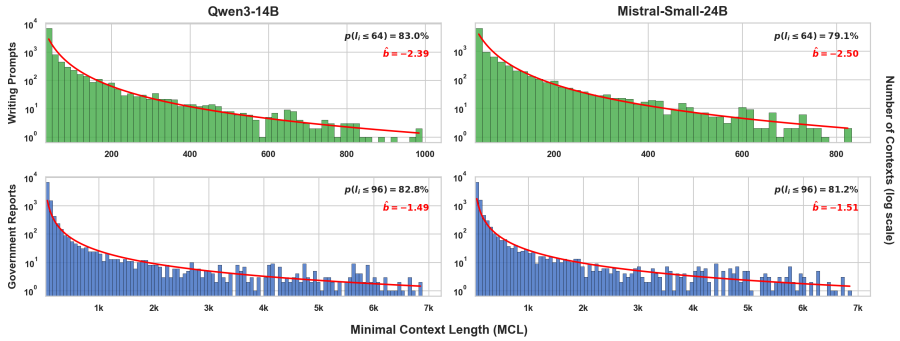



Figure 32: **MCL distribution for larger models.** Qwen3-14B (left) and Mistral-Small-24B (right) on Writing Prompts (top) and GovReport (bottom). $\hat{b}$ denotes the slope of the log-log fit.

**DaMCL.** As with the smaller models, DaMCL is less concentrated than MCL, but the same heavy-tailed decay holds across sampling methods (Fig. 33): the smallest window contains the largest share of sequences, which decreases steadily with window size.

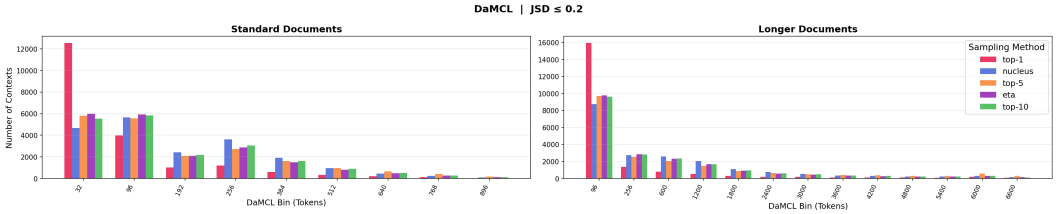



Figure 33: **DaMCL distribution across sampling methods**, pooled over Qwen3-14B and Mistral-Small-24B, on standard (left) and long (right) documents (JSD $\leq 0.2$).

**TaBoo.** Table 15 shows that TaBoo still improves over vanilla nucleus sampling at this scale. The datasets differ between the two models, so the results should be read per model. TaBoo improves on all settings except MultiDocQA with Qwen3-14B, where the two methods are essentially tied. TaBoo also remains competitive with logit-adjustment baselines: on ScientificQA with Qwen3-14B, it reaches an F1 of 28.35, compared to 25.50 for CAD and 23.97 for AdaCAD. Overall, we observe no notable difference in gains between smaller and larger models.

Table 15: TaBoo vs. vanilla nucleus sampling on larger models (mean ± standard error). Bold marks the better method per metric.

**Qwen3-14B**

| | F1 | | ROUGE-L | |
|---|---|---|---|---|
| Dataset | Vanilla | TaBoo | Vanilla | TaBoo |
| CNN/DM | 24.92±0.65 | **26.63**±0.71 | 23.33±0.66 | **25.06**±0.73 |
| SAMSum | 26.45±0.75 | **27.52**±0.81 | 27.84±0.76 | **28.94**±0.84 |
| MultiDocQA | **21.43**±1.19 | 21.31±1.27 | 24.35±1.40 | **24.36**±1.49 |
| ScientificQA | 24.03±1.47 | **28.35**±1.82 | 35.85±1.82 | **41.30**±2.17 |

**Mistral-Small-24B**

| | F1 | | ROUGE-L | |
|---|---|---|---|---|
| Dataset | Vanilla | TaBoo | Vanilla | TaBoo |
| NQ | 28.72±2.35 | **35.84**±3.09 | 31.27±2.47 | **37.83**±3.19 |
| MultiFieldQA | 18.61±1.40 | **21.80**±1.81 | 30.35±1.83 | **35.84**±2.26 |
| SAMSum | 22.25±0.71 | **27.92**±0.76 | 22.77±0.65 | **28.99**±0.75 |

# I Controlling for Memorization: Synthetic Novel Documents

A possible confound in our analysis is pretraining memorization: if a model has seen a document during pretraining, it may predict the next token from a short prefix by recalling the document rather than by using local context. To control for this, we construct a strictly novel synthetic corpus. We use Llama-3.1-70B [36] to generate documents of regular length (∼1K tokens) and long length (∼7K tokens). Each document is an article or story built strictly around invented content: two fictional person names, one fictional place, one fictional organization, one fabricated numeric fact, and one specific year. The model is instructed to keep all contextual information internal to the document, and we remove any references to real-world entities. We then measure MCL and DaMCL with Mistral-7B and Qwen3-8B, following the same setup as in the main text.

**MCL.** As shown in Fig. 34, short-context dominance persists on these novel documents: 75–79% of sequences have an MCL of at most 64 tokens on the regular documents, and 69–70% have an MCL of at most 96 tokens on the long documents, with the same heavy-tailed decay as in our main results.

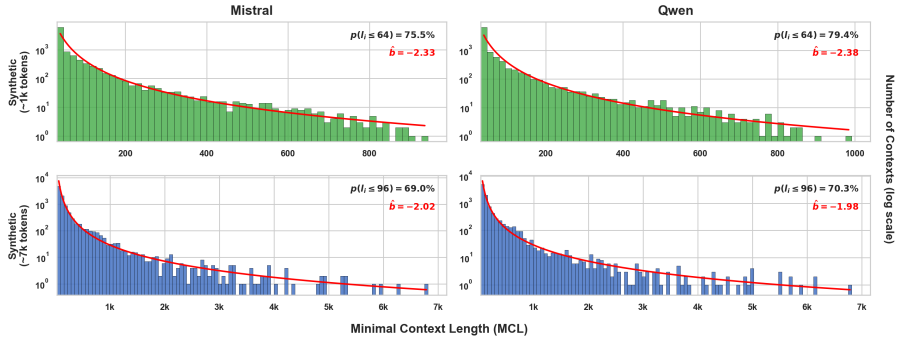



Figure 34: **MCL distribution on synthetic novel documents.** Mistral-7B (left) and Qwen3-8B (right) on regular (∼1K tokens, top) and long (∼7K tokens, bottom) documents. $\hat{b}$ denotes the slope of the log-log fit.

**DaMCL.** The same pattern holds for DaMCL across sampling methods (Fig. 35): the smallest window contains the largest share of sequences, which decreases steadily with window size.

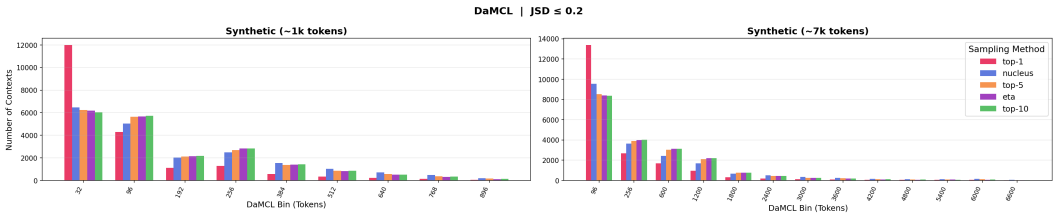



Figure 35: **DaMCL distribution across sampling methods on synthetic novel documents**, pooled over Mistral-7B and Qwen3-8B, for regular (∼1K tokens, left) and long (∼7K tokens, right) documents (JSD $\leq 0.2$).

**Discussion.** The effect is slightly weaker than in our main results, especially for long documents. We see a few plausible explanations: our prompting style explicitly constrains the model to maintain contextual relevance throughout the document, the generated text lacks the boilerplate language common in natural text, and pretraining memorization effects are eliminated. Overall, these results indicate that short-context dominance is not an artifact of memorization.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: [N/A].

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have provided a section in the appendix describing the limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The work doesn't provide strong theoretical assumptions.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided ample information for all experimental setups in addition to example codes used to generate the experimental results to the best of our ability.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All models and datasets are open source.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided all required detail to the best of our ability.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For experimental results, specially on TaBoo, we provide information regarding standard errors of the experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All information is provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Done to the best of our ability.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: We helping in better understanding and improving language models, we don't believe this work has direct societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The work does not include any high risk models or data.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: No notable licensing is required and all models/datasets are open source.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Done to the best of our ability.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No human crowd sourcing was done.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human crowd sourcing was done.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were used to help with editing and as assistant for developing code for experimental results.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.