UT-ACA: Uncertainty-Triggered Adaptive Context Allocation for Long-Context Inference

Anonymous ACL submission

Abstract

Long-context inference remains challenging for large language models due to attention dilution and out-of-distribution degradation. Context selection mitigates this limitation by attending to a subset of key-value cache entries, yet most methods allocate a fixed context budget throughout decoding despite highly non-uniform token-level contextual demands. To address this issue, we propose Uncertainty-Triggered Adaptive Context Allocation (UT-ACA), an inference-time framework that dynamically adjusts the context window based on token-wise uncertainty. UT-ACA learns an uncertainty detector that combines semantic embeddings with logit-based confidence while accounting for uncertainty accumulation across decoding steps. When insufficient evidence is indicated, UT-ACA selectively rolls back, expands the context window, and regenerates the token with additional support. Experiments show that UT-ACA substantially reduces average context usage while preserving generation quality in long-context settings.

1 Introduction

Large language models (LLMs) are increasingly expected to operate in long-context settings, such as long-document question answering, multiple document summarization, and evidence-grounded reasoning over extensive inputs (Zhang et al., 2025b; Wu et al., 2025c; Li et al., 2024a). In these scenarios, effective long-context handling is pivotal for robust generation, requiring reliable evidence retrieval and globally consistent generation. However, as context length increases, redundant or weakly relevant tokens dilute attention and hinder precise evidence retrieval. Furthermore, long contexts present Out-Of-Distribution (OOD) dependencies and positional patterns diverge from pre-training data, resulting in degraded calibration and unstable reasoning during length extrapolation.

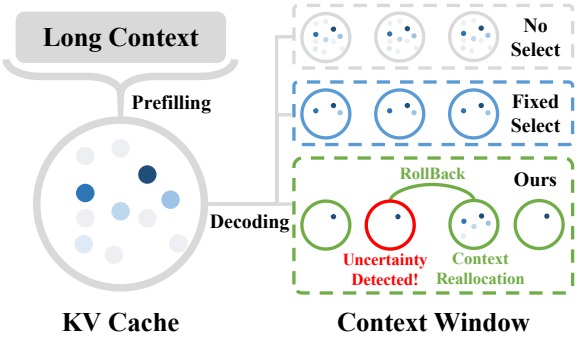

Figure 1: Context management techniques: (1) No Select: complete context. (2) Fixed Select: fixed-size context window. (3) Ours: adaptive context window.

Context selection alleviates long-context issues by attending only to relevant Key-Value (KV) entries, thereby reducing attention dilution and computational cost. However, most existing methods (Wu et al., 2025b; Hao et al., 2025; Tang et al., 2024) rely on a fixed context budget throughout decoding, implicitly assuming that generation difficulty is uniform across tokens. In practice, token-level difficulty varies substantially: many tokens are determined by local context, whereas others require long-range evidence dispersed across the prompt. Consequently, fixed-size context windows are often inefficient and can be insufficient for difficult generation steps, motivating adaptive, token-wise context allocation during decoding.

A natural implication is that context requirements are determined at the token level and adjusted dynamically during decoding. Token-wise generation difficulty can be inferred from uncertainty signals available at inference time, which enables adaptive context control. Following this idea, decoding starts with a small context window and monitors uncertainty at each generation step. When high uncertainty or hallucination tendency is detected, the process reverts to the previous state, expands the context window, and regen-

erates the token with additional evidence. In this formulation, uncertainty estimation serves as an active decision signal for selective context enlargement rather than a passive diagnostic measure.

Accordingly, we propose an inference-time framework, called Uncertainty-Triggered Adaptive Context Allocation (UT-ACA), that dynamically adjusts the context window based on token-wise uncertainty. As shown in Figure 1, UT-ACA enables selective rollback and regeneration only when insufficient contextual evidence is detected, improving both efficiency and robustness. The framework estimates uncertainty using the margin between top two logits as a lightweight confidence signal, avoiding the overhead of multi-sample decoding (Khairi et al., 2025; Li et al., 2025a; Kuhn et al., 2023). We construct a synthetic dataset to train and evaluate our uncertainty detector, using hidden states and output logits of LLMs as supervision signals. The detector adopts a dual-encoder architecture to fuse semantic and confidence features, and incorporates temporal modeling (Zhao et al., 2025b; Qiu et al., 2024) to capture uncertainty accumulation during decoding. Experimental results show that UT-ACA generalizes reliably across settings, enabling effective context allocation that improves long-context generation performance while reducing computational cost.

2 Preliminary

LLM is typically pretrained on a corpus $C = \{s_1, s_2, \ldots, s_n\}$, where each sequence s_i consists of tokens (DeepSeek-AI and others, 2025; Dubey et al., 2024; OpenAI, 2024). Given a token sequence $s_i = (x_1, x_2, \ldots, x_T)$, an autoregressive language model parameterized by θ defines the joint probability

$$p_\theta(x_{1:T}) = \prod_{t=1}^{T} p_\theta(x_t \mid x_{<t}), \tag{1}$$

where $x_{<t} = (x_1, \ldots, x_{t-1})$ and $p_\theta(x_t \mid x_{<t})$ denotes the next-token prediction distribution. The model is trained by maximizing the log-likelihood of the training data, equivalently minimizing the negative log-likelihood

$$\mathcal{L}(\theta) = -\mathbb{E}_{x_{1:T}\sim\mathcal{D}}\left[\sum_{t=1}^{T} \log p_\theta(x_t \mid x_{<t})\right], \tag{2}$$

where $\mathcal{D}$ denotes the data distribution.

During inference, the model generates tokens autoregressively and relies on self-attention and positional encoding to incorporate contextual information. Since these components are optimized under the sequence-length distribution observed during pretraining, the long-context capability of a pretrained LLM is closely tied to the maximum length and distribution of its training data. When a model trained primarily on short contexts is deployed on much longer sequences, it experiences a distribution shift in both long-range dependency patterns and positional representations (Ding et al., 2024; Peng et al., 2024), which can induce OOD behaviors and degrade decoding reliability.

Inference with KV Cache and Context Window. At inference time, LLMs maintain a KV cache that stores intermediate representations of past tokens at each attention layer (Hooper et al., 2024; Liu et al., 2024). At decoding step t, the next-token distribution $p_\theta(x_t \mid x_{<t})$ is computed by attending to a selected KV cache $\mathcal{C}_t \subseteq \{(K_i, V_i)\}_{i=1}^{t-1}$. We refer to $\mathcal{C}_t$ as the *context window* at step t. Importantly, the context window denotes the subset of KV entries participating in attention computation, rather than the maximum supported sequence length of the model. Formally, self-attention at step t is computed as

$$\mathrm{Attn}(Q_t, \mathcal{C}_t) = \mathrm{softmax}\left(\frac{Q_t K_{\mathcal{C}_t}^\top}{\sqrt{d}}\right) V_{\mathcal{C}_t}, \tag{3}$$

where Q_t is the query at step t, and d represents the dimension of Q_t. $K_{\mathcal{C}_t}, V_{\mathcal{C}_t}$ denote the keys and values in the selected context window. In long-context settings, attending to the full KV cache is computationally expensive and often unnecessary (An et al., 2024), motivating adaptive strategies that dynamically adjust the size of $\mathcal{C}_t$.

Rollback and Regeneration. Autoregressive generation of LLMs is susceptible to error accumulation (Huang et al., 2024), as uncertainty or incorrect predictions at earlier steps can propagate through the KV cache and affect subsequent decoding. We denote the decoding state at step t as $\xi_t = (x_{<t}, \mathcal{C}_t)$. When a token is generated under insufficient contextual evidence, its associated KV cache update may degrade later predictions. We therefore consider a rollback-and-regenerate operation at inference time, which restores the decoding state from ξ_t to ξ_{t-1} by discarding the current token x_t and its corresponding KV cache entries.

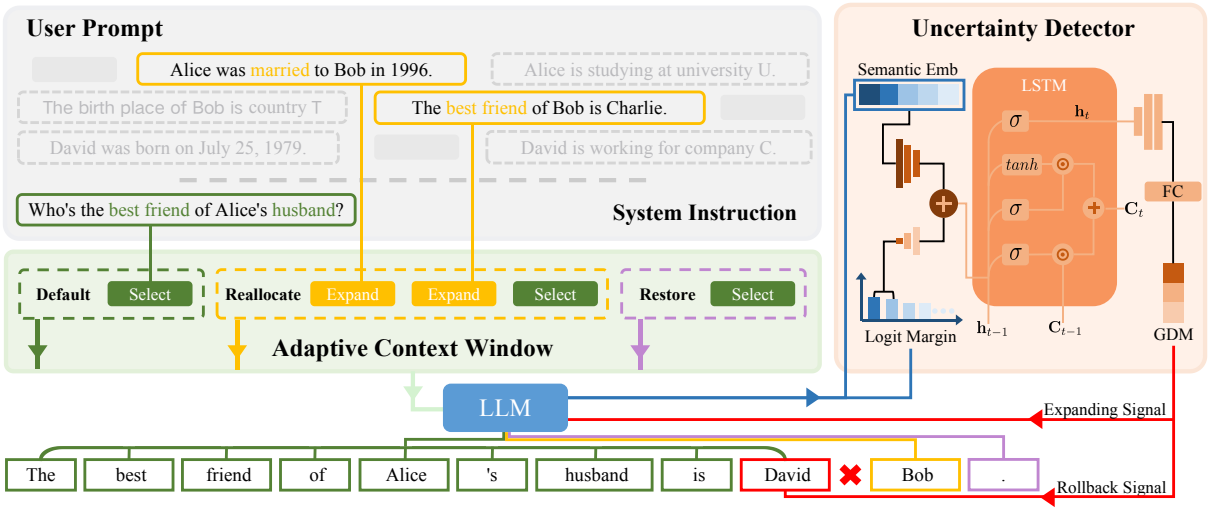

Figure 2: Main workflow of UT-ACA. (a) The user prompt contains the long-context input information. (b) The system instruction specifies the questions or instructions. (c) The uncertainty detector takes the output logits and semantic embeddings to estimate the generation difficulty metric. (d) The adaptive context window receives the detector signal, expands the context window when needed, and triggers regeneration.

The model then regenerates the token under an expanded context window, allowing rectification when the original context allocation is insufficient.

3 Method

Long-context inference is prone to degraded reliability when LLMs are deployed beyond their pretraining length (Li et al., 2024b; Gao et al., 2024). To address this limitation and enable token-wise adaptive context window, we propose **Uncertainty-Triggered Adaptive Context Allocation (UT-ACA)**, an inference-time framework that dynamically adjusts the context window during decoding based on token-level uncertainty detection. Figure 2 illustrates the overall workflow of UT-ACA, including the *uncertainty detector* and the *adaptive context window* modules.

At each decoding step, UT-ACA first performs a tentative generation with a compact context window. The *uncertainty detector* then consumes the LLM output logits together with semantic embeddings extracted from hidden states, and produces a *Generation Difficulty Metric* (GDM) while accounting for uncertainty accumulation across autoregressive steps. Conditioned on this metric, the *adaptive context window* decides whether to keep the compact window or expand it by retrieving more relevant context blocks. When the metric indicates insufficient evidence or elevated hallucination risk, UT-ACA rolls back the tentative token, expands the context window, and regenerates the current step with additional contextual support. Otherwise, UT-ACA accepts the token and continues with a compact window, reducing average context usage while maintaining generation quality in long-context settings.

3.1 Uncertainty Detector

To support uncertainty-aware decoding, we design a lightweight token-level uncertainty estimator that operates at each generation step. The estimator takes two complementary signals as input, namely the logit margin and a semantic embedding extracted from model hidden states. It then fuses these signals with a dual-encoder module and models uncertainty accumulation over time with a Long Short-Term Memory (LSTM) layer, producing a three-way generation difficulty metric that is later used to trigger context reallocation.

Input signals. At decoding step t, let $\boldsymbol{\ell}_t \in \mathbb{R}^{\mathcal{V}}$ denote the output logits, and $\mathcal{V}$ is the length of LLM vocabulary. We adopt the logit margin m_t as a lightweight confidence signal

$$m_t = \boldsymbol{\ell}_t^{\langle 1 \rangle} - \boldsymbol{\ell}_t^{\langle 2 \rangle}, \tag{4}$$

where $\boldsymbol{\ell}_t^{\langle 1 \rangle}$ and $\boldsymbol{\ell}_t^{\langle 2 \rangle}$ are the largest and second largest logits. Margin alone is not a reliable uncertainty proxy because semantically similar candidates may receive relatively identical logits, resulting in subliminal logit margin during generation. To handle this issue, we extract a token semantic embedding from the LLM inner states. Following prior work on hidden-state analysis (Ferrando et al., 2024; Singh et al., 2024), we use the

output embedding from the last attention layer as $\mathbf{emb}_t$ to represent token-level semantic concepts.

Dual-encoder fusion with temporal modeling. We fuse the semantic embedding $\mathbf{emb}_t$ and the logit margin m_t with a dual-encoder module, which preserves the complementary roles of semantic complexity and score-based confidence while aligning them in a shared space for joint decision making. Concretely, the two branches map both signals into a d-dimensional representation and yield

$$\mathbf{z}_t = \mathrm{LN}[\mathbf{W}_e\mathbf{emb}_t + \mathbf{b}_e + \mathrm{MLP}_m[m_t]], \quad (5)$$

where MLP_m lifts the scalar margin to $\mathbb{R}^d$ and LN denotes layer normalization. Since uncertainty can propagate across autoregressive steps and compound after an early mistake, we apply a LSTM to explicitly model this temporal accumulation and aggregate historical evidence

$$\mathbf{h}_t, \mathbf{c}_t = \mathrm{LSTM}[\mathbf{z}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}], \quad (6)$$

where the memory state $\mathbf{c}_t$ captures past uncertainty and the hidden state $\mathbf{h}_t$ serves as the stepwise generation difficulty representation for downstream prediction.

Output as generation difficulty metric. We define the *generation difficulty metric* as a three-way probability vector over token-generation scenarios in Figure 3, namely grounded correct (Correct Answer), unknown style abstention (Unknown), and content fabrication (Hallucination). This tripartite formulation separates sufficient-evidence generation from two insufficient-evidence behaviors. The prediction head maps $\mathbf{h}_t$ to the GDM

$$\begin{aligned} \mathbf{p}_t &= \mathrm{softmax}[\mathbf{W}_g\mathbf{h}_t + \mathbf{b}_g], \\ \mathbf{p}_t &= [p_{t,\mathrm{cor}}, p_{t,\mathrm{unk}}, p_{t,\mathrm{hal}}], \\ \mathbf{p}_t &\in \mathbb{R}^3, \ \sum_{k=1}^{3} p_{t,k} = 1, \end{aligned} \quad (7)$$

where $p_{t,\mathrm{cor}}$ corresponds to grounded correct generation, $p_{t,\mathrm{unk}}$ to unknown style abstention, and $p_{t,\mathrm{hal}}$ to hallucination. A three-way output is needed because correct tokens and unknown tokens may both look confident under local decoding, and unknown predictions can also result from an overly compact context window. In practice, the downstream policy triggers context reallocation when the non-grounded mass is larger

$$p_{t,\mathrm{unk}} + p_{t,\mathrm{hal}} > p_{t,\mathrm{cor}}, \quad (8)$$

which expands the context window and regenerates the current token for rectification.

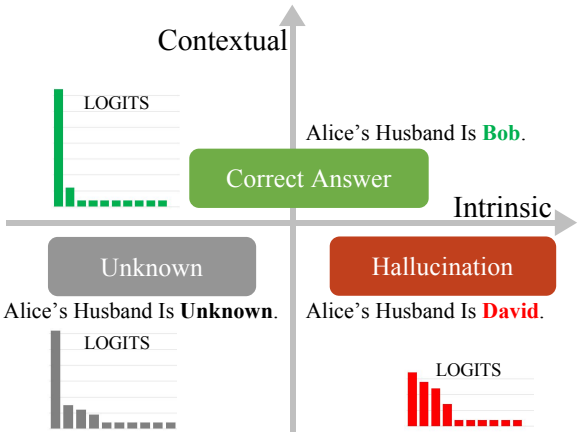

Figure 3: Overview of token-generation scenarios. The axes delineate the sufficiency of contextual versus intrinsic knowledge, while the logit plots depict the corresponding LLM outputs under varying conditions.

3.2 Adaptive Context Window

Inspired by InfLLM (Xiao et al., 2024), we adopt fixed-size blocks as the basic units of our adaptive context window. During prefilling, we partition the KV representations of the long context into blocks that contain the same number of tokens. For each block, we keep a small set of representative tokens to support efficient relevance estimation. During decoding, the query vector at step t retrieves the top-k most relevant blocks, and attention is computed only over the selected blocks to mitigate long-context OOD behaviors.

Since token difficulty varies across steps, the required contextual evidence is non-uniform. UT-ACA therefore starts from a compact window, performs a tentative generation, and feeds the output logits together with the semantic embedding into the uncertainty detector to obtain the GDM. When the GDM indicates Unknown or Hallucination as dominant, UT-ACA rolls back the tentative token, expands the context window by retrieving more relevant blocks, and regenerates the current token. Formally, let $\mathcal{B}$ be the set of all blocks, $\mathbf{r}(B)$ denote the representative keys of block B, and $\mathbf{q}_t$ be the decoding query at step t. Block retrieval under a budget k is written as

$$\mathcal{S}_t(k) = \arg\max_{\substack{\mathcal{S}\subseteq\mathcal{B} \\ |\mathcal{S}|=k}} \sum_{B\in\mathcal{S}} \mathrm{sim}[\mathbf{q}_t, \mathbf{r}(B)], \quad (9)$$

and UT-ACA adaptively expand context window when the non-grounded mass is dominant $p_{t,\mathrm{unk}} + p_{t,\mathrm{hal}} > p_{t,\mathrm{cor}}$. Subsequently, decoding is performed again using $\mathcal{S}_t(k_{\mathrm{large}})$ instead of

$\mathcal{S}_t(k_{\text{small}})$, with rollback applied to remove the tentative token before regeneration. Algorithm 1 presents the detailed pseudocode for our UT-ACA framework.

Algorithm 1 UT-ACA Framework

Require: Model $\mathcal{M}$, Detector $\mathcal{D}$, Input x_0

Initialize cache $\mathbf{C}$, context window size $k \leftarrow K_{\max}$, $t \leftarrow 0$
while $x_t \neq$ <EOS> **do**
  $\mathcal{S} \leftarrow \text{Snapshot}(\mathbf{C})$ ▷ Save state
  // Step1: Tentative Generation
  $\hat{x}, \hat{\mathbf{C}} \leftarrow \mathcal{M}(x_t, \mathbf{C}, \text{topk} = k)$
  // Step2: Uncertainty Check
  $is\_unsafe \leftarrow \mathcal{D}(\hat{x}, \text{activations})$
  if is_unsafe **then**
    $\mathbf{C} \leftarrow \text{Restore}(\mathcal{S})$ ▷ Roll back
    $k \leftarrow K_{\max}$ ▷ Expand
    Regenerate:
    $x_{t+1}, \mathbf{C} \leftarrow \mathcal{M}(x_t, \mathbf{C}, \text{topk} = k)$
  else
    $x_{t+1}, \mathbf{C} \leftarrow \hat{x}, \hat{\mathbf{C}}$
    $k \leftarrow \text{UpdatePolicy}(k)$ ▷ Shrink
  end if
  $t \leftarrow t + 1$
end while

4 Experiments

We assess UT-ACA with a two-stage experimental framework on our synthetic biography dataset. Phase-1: training the uncertainty detector with token-wise LLM output logits, internal states and labels. We conduct eight runs for each training setting and select the best-performing detector. Phase-2: pluging the trained detector to evaluate the adaptive context window on disjoint validation data. We measure uncertainty detection performance using accuracy, recall. And assess UT-ACA applying conceptual accuracy, and computational efficiency (context tokens and latency). Additionally, we provide ablation studies and benchmark results on ∞-Bench (Zhang et al., 2024) and LongBench (Bai et al., 2024). All runs are performed on a single NVIDIA H100 GPU.

4.1 Dataset and Metrics

Dataset. We construct a synthetic biography summarization dataset using a summary-first pipeline. We sample an attribute name from a predefined set, randomly generate its value, and construct a ground-truth summary for a fictitious person, which is then padded or extended into a longer biography. All person names are unique and non-overlapping across splits to avoid data leakage.

Each training and validation instance consists of a single attribute per person. We construct 10000 training samples to serve as LLM inputs. By constraining the context window to be shorter than the summary, unknown or hallucinated tokens are induced designedly. During decoding, we record logits and the semantic embedding for each generated token. Subsequently, we employ GPT-OSS-120B (OpenAI, 2025) to assign one-hot labels (correct, unknown, or hallucinated) based on ground truth summaries, followed by manual verification. The validation set comprises six unseen attributes with 20 summaries each. The long-context test set contains 100 biographies covering all six attributes, extended with Wikipedia text to span lengths from 171k to 400k tokens. Example data instances are provided in Appendix A.2.

Metrics. We evaluate both the uncertainty detector and the downstream context allocation strategy. For uncertainty estimation, we report mean accuracy (**mAcc**) on the validation set, along with recall for uncertain and certain samples (**Recall**$_N$ and **Recall**$_P$). For generation quality, we apply GPT-OSS-120B as automated evaluator to score the conceptual alignment between LLM outputs and the ground-truth, and report the resulting conceptual mean accuracy as **mAcc**$_{conc}$, with an illustrative comparison shown in Figure 4. Finally, we quantify efficiency by reporting the mean number of context tokens consumed (**mTokens**) and the average per-token decoding latency (**mTime**$_{tok}$).

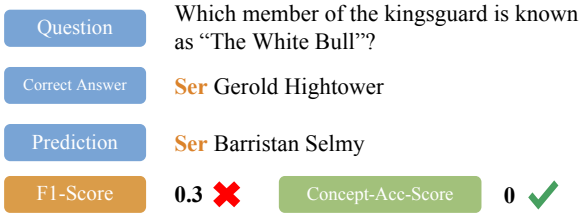

Figure 4: Comparison between f1-score and our conceptual accuracy score.

4.2 Uncertainty Detection Experiments

We train and evaluate the uncertainty detector using logits and embeddings extracted from Llama-3.1-8B-Instruct (Dubey et al., 2024). Specifically, we record token-level output logits and internal hidden states, and adopt GPT-OSS-120B to assign token labels. We then evaluate the trained detector

on the validation set and compare it with heuristic baselines. As summarized in Table 1, the TOP-10 STD baseline uses the standard deviation of the top-10 logits as an uncertainty proxy, while LOGIT MARGIN uses the top-1 versus top-2 margin. Both methods tune a decision threshold on the training set and report validation performance. In contrast, our learning-based detector achieves stronger and more balanced performance, obtaining 89.71% recall on uncertain tokens.

Methods	**mAcc** (%)	**Recall**$_N$ (%)	**Recall**$_P$ (%)
TOP-10 STD	54.47	44.98	58.30
Logit Margin	62.52	45.69	69.31
UT-ACA	**83.43**	**89.71**	**81.27**

Table 1: Comparison between different token-level uncertainty detection methods and UT-ACA.

4.3 Generation Experiments

We evaluate the generation performance of UT-ACA after the training of the uncertainty detector. We mainly use InfLLM (Xiao et al., 2024) as the baseline and keep the experimental settings consistent across methods. For the biography summarization task, we report **mTokens** as the efficiency metric, and conceptual accuracy (**mAcc**$_{conc}$) as the primary quality metric for generation experiments.

Validation set. We implement two update rules for UT-ACA after a window expansion. *Update: Set.1* resets the window length to 1 for the next token, and *Update: Sub.1* decreases the window length by 1 with a minimum of 1. We set the maximum number of context blocks $K_{\max} = 3$ and use a fixed block size of 16 tokens for all methods.

The experimental results are summarized in Table 2, UT-ACA achieves higher generation quality while consuming fewer context tokens on average, reaching 99.08% **mAcc**$_{conc}$ with only 29 context tokens. This gain suggests that uncertainty-triggered expansion allocates additional context only when needed, improving difficult generation steps without increasing average context usage.

Methods	**Settings**	**mTokens** ↓	**mAcc**$_{conc}$(%)
Baseline	$K = 2$	32	91.37
	$K = 3$	48	**99.83**
UT-ACA	Update: Set.1	25	96.59
	Update: Sub.1	**29**	**99.08**

Table 2: Comparison of the baseline and UT-ACA on the validation set.

Long-context test set. The test set consists of biographies ranging from 171k to 400k tokens, with an average length of 252k, which frequently exceeds the maximum sequence lengths supported by Llama-3.1-8B-Instruct at 128k tokens and Qwen2-7B-Instruct (Yang et al., 2024) at 32k tokens. Notably, an independent uncertainty detector is trained for the Qwen model following the same procedure as that used for the Llama model.

Table 3 reports the average number of context tokens usage (**mTokens**) and the average per-token decoding latency (**mTime**$_{tok}$), where K_{tok} denotes the number of tokens selected at each generation step. For LLMLingua-2, K_{com} represents the prompt compression ratio. For UT-ACA, $K_{\max}$ denotes the maximum number of selected context blocks, with the block size fixed to 16 tokens. Compared to the baselines, UT-ACA achieves improved generation quality while consuming fewer context tokens. The rollback mechanism introduces a marginal increase in **mTime**$_{tok}$, while enabling earlier identification of uncertain generation steps and mitigating error accumulation. In addition, UT-ACA allows a larger effective context window for difficult tokens while maintaining a compact average window, making it well suited for long-context inference.

4.4 Ablation Study on Test Set

We conduct an ablation study on the test set using Llama-3.1-8B-Instruct with $K_{\max} = 64$ under the *Update:Sub16* configuration. The contribution of each component in the uncertainty detector is evaluated by individually removing the Logit Margin branch (LogM), the Semantic Embedding branch (SE), and the LSTM module.

The results are reported in Table 4. Removing the Semantic Embedding branch results in substantially degraded uncertainty detection accuracy, which is insufficient to support reliable context reallocation. Consequently, generation performance under these settings is not reported. Overall, the full model that incorporates all components achieves the strongest uncertainty detection performance and consistently yields the highest generation quality across the evaluated configurations. These results suggest that semantic representations and temporal aggregation provide complementary signals that improve the stability of uncertainty estimation and enhance the reliability of the reallocation trigger mechanism.

Models	Methods	Settings	mTokens ↓	mTime$_{\text{tok}}$ (s) ↓	mAcc$_{\text{conc}}$ (%) ↑
Llama-3-8B-it-262k	†RetrievalAttn (Zhu et al., 2024)	$K_{\text{tok}} = 2048$	2k	0.026	32.22
	†SnapKV (Li et al., 2024c)	$K_{\text{tok}} = 1024$	1k	0.208	41.78
	OmniKV (Hao et al., 2025)	$K_{\text{tok}} = 6.7\%$	8k	-	45.83
Llama-3.1-8B-it	†LLMLingua-2 (Pan et al., 2024)	$K_{\text{com}} = 25\%$	32k	-	18.08
	TokenSelect (Wu et al., 2025b)	$K_{\text{tok}} = 1024$	1k	-	39.43
	Baseline (Xiao et al., 2024)	$K = 8$	128	0.064	25.81
		$K = 16$	256	0.072	51.23
		$K = 32$	512	**0.080**	72.26
	UT-ACA (Update: Sub.16)	$K_{\text{max}} = 32$	123	0.069	45.23
		$K_{\text{max}} = 48$	218	0.075	62.17
		$K_{\text{max}} = 64$	344	0.082	70.48
		$K_{\text{max}} = 96$	**498**	0.097	**74.61**
Qwen2-7B-it	Baseline (Xiao et al., 2024)	$K = 8$	128	0.061	37.77
		$K = 16$	256	0.064	44.39
		$K = 32$	512	**0.077**	54.72
	UT-ACA (Update: Sub.8)	$K_{\text{max}} = 16$	119	0.059	37.08
		$K_{\text{max}} = 32$	302	0.066	48.33
		$K_{\text{max}} = 48$	**432**	0.079	**56.64**

Table 3: Comparison of context management methods on the test set. The symbol † indicates methods where long-context samples triggered GPU out-of-memory errors on a single NVIDIA H100 or produced incoherent outputs. These samples are excluded from the evaluation of the corresponding methods.

LogM	SE	LSTM	mAcc (%)	mAcc$_{conc}$(%)
✔	✘	✘	26.82	-
✘	✔	✘	78.95	69.58
✔	✘	✔	53.71	-
✘	✔	✔	80.06	68.58
✔	✔	✘	80.19	69.42
✔	✔	✔	**83.43**	**70.48**

Table 4: Ablation analysis of components in our uncertainty detector.

4.5 Open-Set Benchmark Experiments

We further evaluate UT-ACA on open-set long-context benchmarks to assess its generalization behavior under standardized evaluation protocols. These experiments focus on the trade-off between generation quality and context usage when the model is exposed to unseen tasks and substantially extended inputs.

∞-Bench (Zhang et al., 2024). We evaluate UT-ACA on the ∞-Bench benchmark following the official experimental settings and evaluation metrics, without using the proposed conceptual accuracy score. The block size is fixed to 128 for all methods, and three maximum block budgets $\{8, 16, 32\}$ are considered for UT-ACA, paired with *Update: Sub.4*, *Update: Sub.8*, and *Update: Sub.16*, respectively. Results on the "Longbook Summary English" and "Longbook QA English" subtasks are reported in Table 5. Across both subtasks, UT-ACA attains comparable accuracy while using a smaller average context window, indicating that uncertainty-triggered expansion concentrates computation on more challenging generation steps while maintaining compact context usage for confident predictions.

Methods	Sum Task		QA Task	
	mTokens ↓	Score	mTokens ↓	Score
TokenSelect	1024	27.78	1024	13.71
	2048	28.29	2048	16.93
	4096	28.48	4096	18.44
Baseline	1024	27.24	1024	16.27
	2048	27.87	2048	18.38
	4096	28.58	4096	22.09
UT-ACA	609	27.46	896	16.58
	1610	27.64	1664	18.72
	3508	28.43	**3712**	21.32

Table 5: Performance comparison on ∞-Bench.

LongBench (Bai et al., 2024). UT-ACA is further evaluated on LongBench under the standard evaluation protocols. The block size is fixed to 16 for all methods, and four maximum block budgets $\{4, 8, 16, 32\}$ are examined for UT-ACA, paired with *Update: Sub.2*, *Update: Sub.4*, *Update: Sub.8*, and *Update: Sub.16*, respectively. Results on four LongBench subtasks are presented

Methods	Settings	multifieldqa		narrativeqa		qmsum		samsum	
		mTokens ↓	mAcc	mTokens ↓	mAcc	mTokens ↓	mAcc	mTokens ↓	mAcc
Baseline	K=2	32	20.94	32	13.49	32	17.14	32	19.36
	K=4	64	25.9	64	14.95	64	18.40	64	20.50
	K=8	128	28.5	128	15.75	128	18.62	128	22.30
	K=16	256	**33.83**	256	15.57	256	19.13	256	21.37
UT-ACA	K_{max}=4	31	25.20	39	15.01	24	17.65	58	20.47
	K_{max}=8	51	28.9	59	16.11	47	18.66	88	22.95
	K_{max}=16	95	33.37	109	16.92	**102**	**19.18**	**138**	**23.38**
	K_{max}=32	**133**	32.59	**252**	**17.81**	161	17.74	285	22.18

Table 6: Performance comparison between baseline and our UT-ACA on LongBench.

in Table 6. UT-ACA achieves strong performance across most evaluated settings. In particular, results on "qmsum" and "samsum" show that the highest performance is obtained with a maximum budget size of 16, suggesting that increasing the maximum context budget does not necessarily lead to improved performance and that appropriate budget selection depends on task characteristics.

5 Related Work

Long-context inference has been studied from two complementary directions. One line of work applies training-based methods (Chen et al., 2025; Hu et al., 2025; Gao et al., 2025; Tian et al., 2025) or improves positional encoding (Ding et al., 2024; Jin et al., 2024; Wang et al., 2024) to extend length generalization, while another line focuses on context management to control the effective attention scope and reduce computation (Liskavets et al., 2025; Zhu et al., 2024; Fu et al., 2025, 2024). Since UT-ACA operates at inference time by regulating the usable context, we focus on context management methods and omit other approaches.

Context selection. Context selection methods retain merely the most relevant tokens or blocks to mitigate attention dilution and reduce memory and compute consumption (Zhang et al., 2025a). InfLLM (Xiao et al., 2024) is a training-free framework that performs block-level retrieval over the KV cache to enable length extrapolation at inference time. TokenSelect (Wu et al., 2025b) further explores dynamic token-level KV selection for efficient long-context decoding. Beyond retrieval, XAttention (Xu et al., 2025) designs block-sparse attention guided by structured scoring to capture long-range dependencies more efficiently, and OmniKV (Hao et al., 2025) proposes dynamic KV management to balance efficiency and accuracy. These approaches share a common goal of selecting salient context to reduce unnecessary attention computation.

Context compression. Context compression methods instead shrink the input or internal representations (Zhao et al., 2025a; Liao et al., 2025). At the prompt level, LLMLingua (Jiang et al., 2023) and LongLLMLingua (Jiang et al., 2024) compress prompts using information-theoretic criteria while preserving key semantics. At the KV-cache level, SnapKV (Li et al., 2024c) prunes cache entries by identifying crucial patterns prior to generation, and SCOPE (Wu et al., 2025a) optimizes KV compression to trade off memory savings and generation quality. FocusLLM (Li et al., 2025b) further condenses long-context inputs into compact representations to support precise understanding in long-context settings.

6 Conclusion

This work addresses the problem of attention dilution in long-context language models by rethinking how context is allocated during decoding. Accordingly, we propose an uncertainty-triggered adaptive context allocation framework that dynamically adjusts the effective context window at the token level. The core idea is to apply real-time generation difficulty as a signal to determine when additional contextual evidence is required. During decoding, tokens are tentatively generated under a compact context window and only regenerated with expanded context when high uncertainty is detected. To enable reliable uncertainty estimation, we construct a synthetic biography summarization dataset and train uncertainty detectors for Llama-3.1-8B-Instruct and Qwen2-7B-Instruct. Experimental results show that the proposed method significantly reduces context token usage during long-context decoding while maintaining competitive generation quality.

7 Limitations

The proposed UT-ACA significantly reduces the average number of context tokens, while the per-token decoding latency does not decrease proportionately. We attribute this phenomenon to the computational overhead introduced by the rollback mechanism. Uncertain tokens require regeneration after context expansion, leading to increased inference time, particularly when uncertainty is triggered frequently. Exploring alternative decoding strategies that reduce rollback frequency or amortize its cost remains an important direction for future research.

References

Shengnan An, Zexiong Ma, Zeqi Lin, Dai Zheng, Shuo Wang, Guolei Chen, Chen Niu, and Jian-Guang Wang. 2024. Make your LLM fully utilize the context. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 62160–62188.

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. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 3096–3113.

Jianghao Chen, Junhong Wu, Yangyifan Xu, and Jiajun Zhang. 2025. LADM: Long-context training data selection with attention-based dependency measurement for LLMs. *arXiv preprint arXiv:2503.02502*, pages 1–15.

DeepSeek-AI and others. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, pages 1–22.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. LongRoPE: Extending LLM context window beyond 2 million tokens. In *International Conference on Machine Learning (ICML)*, pages 11091–11104.

Abhimanyu Dubey, Akhil Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, pages 1–92.

Javier Ferrando, Oscar Balcells Obeso, Senthooran Rajamanoharan, and Neel Nanda. 2024. Do i know this entity? knowledge awareness and hallucinations in language models. *arXiv preprint arXiv:2411.14257*, pages 1–36.

Qidong Fu, Xuzhao Han, Zihan Tang, and 1 others. 2024. LazyLLM: Dynamic token pruning for efficient long context LLM inference. *arXiv preprint arXiv:2407.14057*, pages 1–12.

Zihao Fu, Ran Yang, Junxian He, and 1 others. 2025. Squeezed attention: Accelerating long context length LLM inference. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 32631–32652.

Muhan Gao, TaiMing Lu, Kuai Yu, Adam Byerly, and Daniel Khashabi. 2024. Insights into LLM long-context failures: When transformers know but don't tell. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7611–7625.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2025. How to train long-context language models (effectively). In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 7376–7399.

Jitai Hao, Yuke Zhu, Tian Wang, Jun Yu, Xin Xin, Bo Zheng, Zhaochun Ren, and Sheng Guo. 2025. OmniKV: Dynamic context selection for efficient long-context LLMs. In *International Conference on Learning Representations (ICLR)*, pages 1–22.

Coleman Hooper, Sehoon Kim, Amirali Mohammadzadeh, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. 2024. KVQuant: Towards 10 million context length LLM inference with kv cache quantization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1270–1303.

Zhiyuan Hu, Yuliang Liu, Jinman Zhao, Suyuchen Wang, WangYan WangYan, Wei Shen, Qing Gu, Anh Tuan Luu, See-Kiong Ng, Zhiwei Jiang, and Bryan Hooi. 2025. LongRecipe: Recipe for efficient long context generalization in large language models. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 11857–11870.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations (ICLR)*, pages 1–17.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 13358–13376.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 1658–1677.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Fu, Cao Xiao, and Xia Hu Hu. 2024. LLM maybe longlm: Self-extend LLM context window without tuning. *arXiv preprint arXiv:2401.01325*, pages 1–16.

Ammar Khairi, Daniel D'souza, Ye Shen, Julia Kreutzer, and Sara Hooker. 2025. When life gives you samples: The benefits of scaling up inference compute for multilingual LLMs. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 27547–27571.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations (ICLR)*, pages 1–19.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2024a. LooGLE: Can long-context language models understand long contexts? In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 16304–16333.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024b. Long-context LLMs struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*, pages 1–19.

Yiwei Li, Jiayi Shi, Shaoxiong Feng, Peiwen Yuan, Xinglin Wang, Yueqi Zhang, Ji Zhang, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. 2025a. Speculative decoding for multi-sample inference. *arXiv preprint arXiv:2503.05330*, pages 1–8.

Yuhong Li, Yingbing Huang, Bowen Yang, Aman Gill, Jian Luan, Bin Chen, and 1 others. 2024c. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 22947–22970.

Zhenyu Li, Yike Zhang, Tengyu Pan, Yutao Sun, Zhichao Duan, Junjie Fang, Rong Han, Zixuan Wang, and Jianyong Wang. 2025b. FocusLLM: Precise understanding of long context by dynamic condensing. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 31087–31101.

Zihan Liao, Jun Wang, Hang Yu, Lingxiao Wei, Jianguo Li, Jun Wang, and Wei Zhang. 2025. E2LLM: Encoder elongated large language models for long-context understanding and reasoning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 19212–19241.

Barys Liskavets, Maxim Ushakov, Shuvendu Roy, Mark Klibanov, Ali Etemad, and Shane K. Luke. 2025. Prompt compression with context-aware sentence encoding for fast and improved LLM inference. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 24595–24604.

Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. 2024. MiniCache: Kv cache compression in depth dimension for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 139997–140031.

OpenAI. 2024. GPT-4o System Card. *arXiv preprint arXiv:2410.21276*, pages 1–33.

OpenAI. 2025. OpenAI: gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, pages 1–34.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 963–981.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. YaRN: Efficient context window extension of large language models. In *International Conference on Learning Representations (ICLR)*, pages 1–20.

Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo Ponti, and Shay Cohen. 2024. Are large language model temporally grounded? In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 7064–7083.

Aaditya K. Singh, Ted Moskovitz, Felix Hill, Stephanie C. Y. Chan, and Andrew M. Saxe. 2024. What needs to go right for an induction head? a mechanistic study of in-context learning circuits and their formation. In *International Conference on Machine Learning (ICML)*, pages 45637–45662.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. 2024. QUEST: Query-aware sparsity for efficient long-context LLM inference. In *International Conference on Machine Learning (ICML)*, pages 47901–47911.

Junfeng Tian, Da Zheng, Yang Chen, Rui Wang, Colin Zhang, and Debing Zhang. 2025. Untie the Knots: An efficient data augmentation strategy for long-context pre-training in language models. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 1223–1242.

Suyuchen Wang, Ivan Kobyzev, Peng Lu, Mehdi Rezagholizadeh, and Bang Liu. 2024. Resonance RoPE: Improving context length generalization of large language models. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 545–557.

Jialong Wu, Zhenglin Wang, Linhai Zhang, Yilong Lai, Yulan He, and Deyu Zhou. 2025a. SCOPE: Optimizing key-value cache compression in long-context generation. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 10775–10790.

Wei Wu, Zhuoshi Pan, Kun Fu, Chao Wang, Liyi Chen, Yunchu Bai, Tianfu Wang, Zheng Wang, and Hui Xiong. 2025b. TokenSelect: Efficient long-context inference and length extrapolation for LLMs via dynamic token-level kv cache selection. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 21275–21292.

Yuhao Wu, Ming Shan Hee, Zhiqing Hu, and Roy Ka-Wei Lee. 2025c. LongGenBench: Benchmarking long-form generation in long context LLMs. *arXiv preprint arXiv:2409.02076*, pages 1–22.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, Song Han, and Maosong Sun. 2024. InfLLM: Unveiling the intrinsic capacity of LLMs for understanding extremely long sequences with training-free memory. *arXiv preprint arXiv:2402.04617*, pages 1–17.

Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. 2025. XAttention: Block sparse attention with antidiagonal scoring. In *International Conference on Machine Learning (ICML)*, pages 69819–69831.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, pages 1–26.

Hailin Zhang, Xiaodong Ji, Yilin Chen, Fangcheng Fu, Xupeng Miao, Xiaonan Nie, Weipeng Chen, and Bin Cui. 2025a. PQCache: Product quantization-based kvcache for long context LLM inference. *Proc. ACM Manag. Data*, 3(3):1–30.

Haozhen Zhang, Tao Feng, Pengrui Han, and Jiaxuan You. 2025b. AcademicEval: Live long-context LLM benchmark. *arXiv preprint arXiv:2510.17725*, pages 1–32.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Mokul Hao, Xu Han, Thai Ping Lai, Wasila Dong, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞-Bench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 15262–15277.

Yunlong Zhao, Haoran Wu, and Bo Xu. 2025a. Leveraging attention to effectively compress prompts for long-context LLMs. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 26048–26056.

Zhe Zhao, Pengkun Wang, Haibin Wen, Shuang Wang, Liheng Yu, and Yang Wang. 2025b. STEM-LTS: Integrating semantic-temporal dynamics in LLM-driven time series analysis. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 22858–22866.

Di Zhu, Bingyang Wang, Yi Zhao, and 1 others. 2024. RetrievalAttention: Accelerating long-context LLM inference via vector retrieval. *arXiv preprint arXiv:2409.10516*, pages 1–19.

A Dataset Construction Details

In this section, we detail the construction of the synthetic summaries and subsequent biographies. To provide insight into the data's format and diversity, we also present the attribute names and representative examples. Additionally, we include the specific prompts used to generate naturalistic biographies. We intend to release the full dataset publicly in the future.

A.1 Synthetic Dataset Construction Details

Our synthetic dataset is designed primarily for the biography summarization task. As illustrated in Figure 6, we employ a fixed pattern to construct the ground truth summaries. Specifically, each summary consists of a sequence with three components: the person name, the attribute name, and the attribute value. **It is important to note that all person names and attribute values are entirely fictitious. Therefore, there are no privacy concerns regarding real individuals.** To generate the full biography, the summary is extended with filler content or paraphrased into a natural narrative. The dataset is partitioned into training, validation, and test sets, all constructed using synthetic summaries.

Figure 6: Standardized pattern of the synthetic summary. This template is applied across the dataset to ensure structural consistency and controllability.

Training and Validation Set. For the training and validation data construction, each summary is associated with a single attribute, whereas the test set involves six different attributes. To ensure rigorous evaluation, the attributes used in the training set are distinct from those in the validation and test sets. Table 7 lists all attribute names used in our synthetic dataset. Specifically, we utilize 21 attributes for training, and reserve 6 distinct attributes for validation and testing. Additionally, we provide examples for each attribute to demonstrate the diversity of the data.

Dataset	Attribute Names	Example Data
Training Set	• Marry Date • Job Title • Current City • Email Address • Phone Number • Favorite Color • User Agent • Credit Card Provider • Currency Used • Catch Phrase • Street Address • Vehicle License Plate • Favorite File Extension • Domain Name • Cryptocurrency • Timezone • Isbn Code • Lucky Number • Hobby • Marital Status • Personality Type	• The marry date of Laetitia Guyot de Paris is 2010-03-29. • The job title of Franck Jourdan is Magazine features editor. • The current city of Marcial Escalona Yuste is Augerdan. • The email address of Jeremy Wright is catherine40@example.com. • The phone number of Clarence Byrd is 07703087971. • The favorite color of Letizia Fagotto is Black. • The user agent of Inés Novoa Sarabia is Mozilla/5.0 • The credit card provider of Douglas Baker is Maestro. • The currency used of Susann Jacobi Jäckel is Bahraini dinar. • The catch phrase of Olga Torre Rodriguez is Interfaccia estesa ... • The street address of Gabriela Jesus is Bährgasse 62. • The vehicle license plate of Erdal Mälzer is GE 3041 CB. • The favorite file extension of Ornella Tomasini is tiff. • The domain name of Théo Macedo is chretien.com. • The cryptocurrency of Anaïs Gilbert du Rousset is Vertcoin. • The timezone of Eutimio Tejada Ibañez is Europe/Paris. • The isbn code of Jeanne-Sabine Brunet is 978-0-456-54959-9. • The lucky number of Zoé Chauvet is 61. • The hobby of María Manuela Guardia Moles is Sleeping. • The marital status of Gustavo Vallés is Widowed. • The personality type of Salvi Toscanini-Ferrabosco is Extrovert.
Validation Set and Test Set	• Birth Date • Birth Place • Company • Major • University • Work Place	• The birth date of Olivia Garcia is March 22, 1985. • The birth place of Daniel Lee is Seoul, South Korea. • The company of Olivia Chen is Microsoft. • The major of Ethan Clark is Computer Science. • The university of Aiden Gomez is Stanford University. • The work place of Elijah Robinson is Singapore, Singapore.

Table 7: Data samples from our synthetic dataset.

{"role": "system", "content": (
"You are to write a detailed character biography based on the provided information, and you must use the full name of this person all the time!!!\n\n"
"##You will be provided with the following details##: Name: [Person's full name] Birth Date: [Date of birth] Birth Place: [Location where the person was born] University: [Educational institution attended] Major: [Field of study/specialization] Work Place: [Current or primary workplace] Company: [Organization/company where they work]\n\n"\
"##For long biographies (1000+ words)##: Open with engaging anecdote or compelling fact. Describe how birthplace/childhood shaped their interests. Detail their university experience and academic achievements. Explain career decisions and transitions. Include specific projects, innovations, or contributions. Mention professional philosophy or approach. Add personal interests or community involvement if word count allows. Conclude with future goals or ongoing projects.\n\n"\
"##Important Notes##: Accuracy: Ensure dates and locations are logically consistent. Originality: Create unique content for each biography; avoid repetitive phrases. Cultural sensitivity: Be respectful of all locations and institutions mentioned. Professional standards: Write as if this will be published in a professional context."\
"Template: The six information must be described in following template, The [Attribute] of [Name] is [Value].\n\n"\
"You must follow the example following, and XXX is the place you can write on your own: The birth date of Bob Dylan is December 29, 1880. XXX. The birth place of Bob Dylan is William Hayes. XXX. The university of Bob Dylan is Yale University. XXX. The major of Bob Dylan is computer science. XXX. The company of Bob Dylan is Google. XXX. The work place of Bob Dylan is Shenzhen, China. XXX."
)}

Figure 5: The LLM prompt for generating naturalistic biography with ground truth summary.

File List 1	File List 2
Architecture	Bacteria
Art	Biology
Baroque	Black hole
Cinema_disambiguated	Cell (biology)
Classical music	Climate change
Impressionism	DNA_disambiguated
Jazz_disambiguated	Evolution
Literature	General relativity
Modernism	Quantum mechanics
Mona Lisa	Thermodynamics
Painting	Virus
Photography	Communism
Poetry	Culture
Rock music	Democracy_disambiguated
Sculpture	Epistemology
The Starry Night	Human rights
Theatre	Law_disambiguated
Geometry	Logic_disambiguated
Mathematics	Metaphysics
Number	Philosophy
Sport	Psychology
Vaccine	Republic
French Revolution	Socialism
Maya civilization	Ottoman Empire
Roman Empire	World War II
Africa	Algorithm
Amazon River	Artificial intelligence
Amazon rainforest	Blockchain
Antarctica	Cryptography
Atlantic Ocean	Engineering
Australia (continent)	Internal combustion engine
Earth	Internet
Europe	Nanotechnology
Great Barrier Reef	Robotics
Nile	Semiconductor
Ocean_disambiguated	Software
Pacific Ocean	Steam engine
Sahara_disambiguated	Technology
South America	Telecommunication
Volcano	Agriculture
Age of Enlightenment	Calendar
American Civil War	Food
Ancient Egypt	Light
Ancient Greece	Medicine
British Empire	Olympics_disambiguated

Table 8: Source Wikipedia articles used in the test set.

Test Set. We employ GPT-OSS-120B to generate naturalistic biographies based on the summaries provided in the test set. It is important to note that this generation process is model-agnostic; therefore, any capable generative model can be utilized for this task. The specific prompt used to construct the biographies is shown in Figure 5 and can be readily copied and applied. After the module generate naturalistic biographies, we insert common knowledge texts from Wikipedia to further extend the length of synthetic biographies for test set. As shown in Table 8, we present the Wikipedia files used in our test set extending process. The original link of these data can be found in https://en.wikipedia.org/wiki.

A.2 Data Examples from Our Dataset

In our experiments, the training set is utilized to train the uncertainty detector; consequently, the data structure and sequence lengths in this set are relatively simple and short. The validation set is designed to preliminarily evaluate the trained uncertainty detector and the downstream adaptive context window. While the validation data retains the same format as the training set, it comprises distinct attributes and person names. Finally, the test set is constructed to assess our method in a long-context setting. To achieve this, we insert extensive passages from Wikipedia to extend biography lengths, thereby challenging the long-context handling capabilities of the methods.

Table 9 presents examples from our synthetic dataset. While the training and validation sets involve processing only a single attribute per instance, the test set requires the simultaneous ex-

Dataset	Example Summarization	Example Biography
Training Set	The **email address** of Jeremy Wright is catherine40@example.com.	The sky is really blue. ... The **email address** of Jeremy Wright is catherine40@example.com. ... The sky is really blue.
Validation Set	The **birth date** of Olivia Garcia is March 22, 1985.	The sky is really blue. ... The **birth date** of Olivia Garcia is March 22, 1985. ... The sky is really blue.
Test Set	• The **birth date** of Emma Thompson is 12/29/1880. • The **birth place** of Emma Thompson is Tokyo, Japan. • The **university** of Emma Thompson is Harvard University. • The **major** of Emma Thompson is Computer Science. • The **company** of Emma Thompson is Google. • The **work place** of Emma Thompson is San Francisco Bay Area.	... Emma Thompson was **born on** December 29, 1880, **in the** bustling metropolis of Tokyo, Japan. ... She **applied to** Harvard University and was **accepted into** the Computer Science program. ... After careful consideration, she chose to **join** Google, a company known for its innovative culture and cutting-edge technology. ... Emma Thompson's **journey from** Tokyo **to the** San Francisco Bay Area is a testament to her unwavering determination and exceptional talent. ...

Table 9: Representative samples from the training, validation, and test splits.

traction of six attributes. Furthermore, since the biographies in the test set are composed in natural language, the model must perform multi-step reasoning to derive the correct answers.

B Framework Details

This section first introduces the methodology for extracting LLM internal states and provides specific implementation details. Subsequently, we describe the architecture of the uncertainty detector proposed in this work. Finally, we detail the selection of network hyperparameters used in our approach.

B.1 Inner State Extraction of LLMs

To train the uncertainty detector, we leverage internal embeddings from the LLM to capture token-level semantic complexity. Following established protocols, we extract features from the final layer of LLMs. As illustrated in Figure 7, we consider three primary extraction points: (1) the attention mechanism output, (2) the Multi-Layer Perceptron (MLP) block output, and (3) the final residual stream output. We evaluate the efficacy of each position by training independent detectors for comparison. Our results indicate that the extraction site has a negligible impact on detection performance. This consistency stems from the additive nature of the residual stream, where the transformations at these sub-layer stages act as incremental refinements to the same underlying representation, making them functionally equivalent for uncertainty detection.

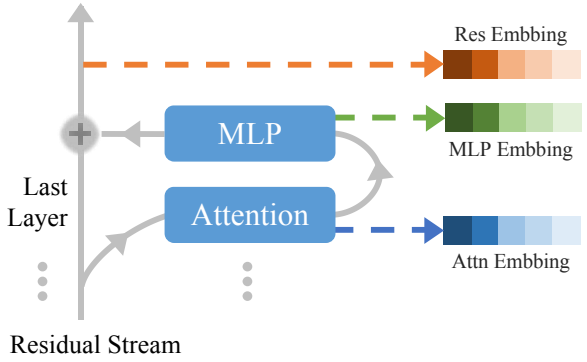

Figure 7: Illustration of extraction points for semantic embeddings. We evaluate features from the final layer (e.g., Layer 32 of Llama-3.1-8B) at three distinct positions: the attention output, the MLP output, and the residual stream.

B.2 Structure of Uncertainty Detector

The proposed model, termed *Uncertainty Detector*, is designed to process sequential data consisting of high-dimensional semantic feature vectors and scalar confidence scores. The architecture comprises three primary modules: dual-branch feature encoding, temporal modeling, and a residual classification head.

Input Encoding and Fusion At each time step t, the model accepts two input streams: a feature vector $\mathbf{emb}_t \in \mathbb{R}^{D_{\text{in}}}$ (where $D_{\text{in}} = 4096$ for Llama-3.1-8B) and a scalar score $m_t \in \mathbb{R}$. The scalar m_t represents the logit margin, calculated by subtracting the top-2 logit from the top-1 logit of the LLM output. The vector input is processed by the `VecEncoder`, which projects the high-dimensional input to a latent space of dimension $d_{model} = 64$. This is achieved via a linear transformation fol-

lowed by a GELU activation and dropout

$$\mathbf{h}_{vec}^{(t)} = \text{Dropout}(\text{GELU}(\mathbf{W}_e\mathbf{emb}_t + \mathbf{b}_e)) \quad (10)$$

where dropout rate is set to 0.5.

Simultaneously, the scalar score is processed by the `ScoreEncoder`, a Multi-Layer Perceptron (MLP) aimed at up-sampling the scalar to match d_{model}. The MLP consists of two linear layers with an intermediate dimension of 32, utilizing GELU activations and dropout ($p = 0.1$).

The `ScoreEncoder` acts as the uncertainty signal branch, employing the logit margin as a lightweight indicator to gauge the generation difficulty of tokens. Conversely, the `VecEncoder` serves as the semantic feature branch, utilizing the hidden states of the LLM to extract conceptual information for enhanced uncertainty estimation. The dual-encoder module aligns the dimensions of these two information sources, after which the features are fused via element-wise addition to form a joint representation ($\mathbf{z}_t$).

Temporal Modeling To effectively model the accumulation of uncertainty throughout the generation process, we employ a temporal modeling approach designed to capture the sequential dependencies between the current token and its predecessors. Specifically, the fused feature sequence $\mathbf{U} = \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$ serves as input to a standard Long Short-Term Memory (LSTM) layer. Configured with a hidden size of d_{model}, the LSTM recursively processes the sequence according to LSTM dynamics.

For the training phase of the uncertainty detector, we preserve the entire sequence of hidden states as supervision signals. In contrast, during the LLM inference stage, only the final hidden state $\mathbf{h}_T$ is extracted to predict the category of the current token, as it aggregates the information of the complete sequence.

Residual Classification Head The output from the LSTM passes through a stack of $N = 3$ `ResidualBlocks`. Each block implements a bottleneck structure with an expansion factor of 2. For an input $\mathbf{h}$, the block operation is defined as

$$\begin{aligned} \hat{\mathbf{h}} &= \text{Dropout}(\text{GELU}(\mathbf{W}_1\mathbf{h} + \mathbf{b}_1)) \\ \mathbf{h}_{out} &= \mathbf{h} + \text{Dropout}(\mathbf{W}_2\hat{\mathbf{h}} + \mathbf{b}_2) \\ \mathbf{p}_{out} &= \text{softmax}(\mathbf{h}_{out}) \end{aligned} \quad (11)$$

Finally, a linear projection layer maps the output of the last residual block to the target class logits (3 classes).

Hyperparameters The specific configurations for our uncertainty detector are detailed in Table 10. Notably, while most hyperparameters remain consistent across different LLMs, certain settings are model-specific. For instance, the feature vector dimension is inherited directly from the eavluated LLM; thus, it is set to 3584 for Qwen2-7B. All other internal parameters of the uncertainty detector are kept uniform to ensure comparability.

Component	Parameter	Value
Input Dimensions	Vector Dim (D_{in})	4096
	Score Dim	1
	Hidden Model Dim (d_{model})	64
Encoders	Vec Dropout	0.5
	Score MLP Inner Dim	32
Residual Head	Number of Blocks	3
	Expansion Factor	2
	Dropout	0.1
Output	Number of Classes	3

Table 10: Hyperparameter configuration for the Llama-3.1-8B uncertainty detector.

C Evaluation Metrics Details

In this section, we detail the implementation of the Conceptual Accuracy Score (CAS) proposed in this study. We then evaluate the reliability of CAS within the context of biography summarization by comparing it against existing evaluation methods and human annotations. Our experimental results demonstrate that CAS exhibits strong alignment with human judgment, establishing it as an efficient, automated, concept-aware metric for evaluating LLM outputs.

C.1 Details of Conceptual Accuracy Score

In our work, we employ GPT-OSS-120B, which can theoretically be replaced by any sufficiently capable generative model, to estimate the Conceptual Accuracy Score (CAS). CAS is computed by comparing the LLM-generated output with the ground-truth summary under the specified attributes. The prompt used for this estimation is shown in Figure 8. The model takes as input the predictions of the evaluated LLMs, together with the corresponding questions and ground-truth answers, and outputs a floating-point score between 0 and 1, indicating the degree of accuracy of the LLM responses.

```
{"role": "system", "content": (
"You are an expert in evaluating the accuracy of large language model predictions."\
Your task is to compare a provided "Answer" with a "Prediction" for a given "Question" and assign an Accuracy Score between 0.0 and 1.0 (inclusive)."\
"1.0 Score: The prediction is identical to, or a complete and accurate restatement of, the answer."\
"0.5 Score: The prediction contains partially correct information, or is conceptually related but not fully accurate. Also applies if the prediction is semantically correct but phrased very differently, requiring deeper interpretation."\
"0.0 Score: The prediction is completely incorrect, irrelevant, or includes hallucinated information not present in the answer or implied by the question."
)}
```

Figure 8: The LLM prompt for evaluating conceptual accuracy score with the input LLM prediction, ground truth answer, and question from each subtask.

C.2 Evaluations of Conceptual Accuracy Score

To evaluate the reliability of the proposed CAS, we conduct a human annotation study. Specifically, human annotators are asked to evaluate 30 samples drawn from the real evaluation workload in our test set experiments. The annotators receive the same prompt used with GPT-OSS-120B and are instructed to assign a floating-point score independently. We then compare the results obtained from three evaluation methods: string-based matching (the standard metric of ∞-Bench and LongBench used in our open-set benchmark experiments), human evaluation, and CAS estimated by GPT-OSS-120B. As each test sample requires the extraction of six attributes, we report both the average accuracy scores under the three evaluation settings and the number of correctly identified attributes, as determined by the human annotators.

Quantitative Evaluation. As reported in Table 11, we randomly select 30 samples from our test set to conduct a comparative analysis. The experimental setup is as follows: we utilize Llama-3.1-8B-Instruct as the model to be evaluated and employ our UT-ACA as the context management method for long-context inference. The specific parameters of the UT-ACA are set to a fixed block size of 16 tokens, with a maximum budget of 96 blocks for the adaptive context allocation. The update policy involves subtracting 16 blocks after the generation of tokens with certainty, maintaining a minimum window size of 1 block. The results demonstrate that our CAS aligns closely with human annotators, showing a difference margin of only 1.3%.

Qualitative Evaluation. Table 12 presents representative examples that demonstrate the effectiveness of our conceptual accuracy score. Specifically, in the evaluation samples, CAS can automatically detect formatting differences while correctly capturing the underlying semantic correctness. For example, in **Case #3**, the LLM predicts “1888-06-03” whereas the ground truth is “June 3, 1888”. A string-matching baseline would

ID	Name	Str	CAS	Hum	Cor	ID	Name	Str	CAS	Hum	Cor
1	Emma Thompson	.000	.000	.000	0/6	16	Lucas Scott	.262	1.00	1.00	6/6
2	Liam Carter	.127	.167	.167	1/6	17	Harper Coleman	.114	.167	.167	1/6
3	Olivia Bennett	.230	1.00	1.00	6/6	18	Henry Green	.260	.500	.667	3/6
4	Noah Wright	.278	.830	.833	5/6	19	Evelyn Perry	.308	1.00	1.00	6/6
5	Ava Sullivan	.299	.667	.667	4/6	20	Oliver Hughes	.358	1.00	1.00	6/6
6	Ethan Brooks	.364	.833	.833	5/6	21	Abigail Kelly	.333	.833	.833	5/6
7	Sophia Reed	.341	.830	1.00	6/6	22	Jack Ross	.268	.830	.833	5/6
8	Mason Foster	.389	1.00	1.00	6/6	23	Emily Howard	.355	1.00	1.00	6/6
9	Isabella Parker	.246	.830	.833	5/6	24	Daniel Gray	.311	.830	.833	5/6
10	William Hayes	.231	.500	.500	3/6	25	Sofia Ellis	.159	.330	.417	2/6
11	Mia Evans	.181	.830	.830	5/6	26	Matthew Bell	.379	1.00	1.00	6/6
12	James Turner	.385	.833	.830	5/6	27	Aria Wood	.417	1.00	1.00	6/6
13	Amelia Fisher	.227	.833	.833	5/6	28	Jacob King	.271	.830	.917	5/6
14	Benjamin Ward	.350	.500	.500	3/6	29	Scarlett Price	.242	.750	.750	4/6
15	Charlotte Morgan	.323	.830	.830	5/6	30	Logan Lee	.361	.830	.917	5/6
	Avg (1–30):	Str=0.279,		CAS=**0.779**,		Hum=**0.766**,		Cor=4.5/6			

Table 11: Evaluation results on 30 test samples using String matching (**Str**), Conceptual accuracy score (**CAS**), and Human annotation (**Hum**). **Cor** denotes the correct number of six attributes, which is count by human.

Case #3: Olivia Bennett (Str: 0.230, LLM: 1.000, Hum: 1.000)				Case #17: Harper Coleman (Str: 0.114, LLM: 0.167, Hum: 0.167)			
Attr	**GT**	**Pred.**	**Hum**	**Attr**	**GT**	**Pred.**	**Hum**
Birth Date	1888-06-03	June 3, 1888	✔	Birth Date	10 Nov, 1907	1976	✘
Birth Place	Sydney, NSW	Sydney, NSW	✔	Birth Place	Mexico City	Madrid, Spain	✘
University	MIT	Mass. Inst. Tech.	✔	University	NUS	Complutense Madrid	✘
Major	Medicine	Medicine	✔	Major	Intl. Rel.	Chem. Eng.	✘
Company	Mayo Clinic	Mayo Clinic	✔	Company	UN	UN	✔
Work Place	Beijing, CN	Mayo's Beijing	✔	Work Place	Seattle, WA	Pacific NW	✘

Table 12: Case study comparison. Case #3 demonstrates semantic robustness despite format variations, whereas Case #17 illustrates prediction failures that are accurately identified by our conceptual accuracy score. (✔=correct, ✘=incorrect)

judge this prediction as incorrect (or assign it an artificially low score) due solely to the surface-form discrepancy. In contrast, CAS correctly assigns a score of 1. In **Case #17**, CAS accurately identifies genuine semantic errors and appropriately assigns a zero score.