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.

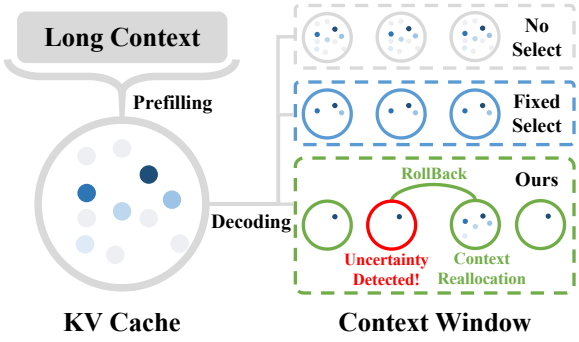

Figure 1: Comparison of context management strategies. The figure contrasts full-context decoding, fixed-size context selection, and the proposed adaptive context allocation strategy. Full-context decoding attends to all available KV entries, fixed selection uses the same retrieved budget at every step, and UT-ACA expands the window only when uncertainty indicates that additional evidence is needed.

1 Introduction

Large language models (LLMs) are increasingly expected to operate in long-context settings, such as long-document question answering, multi-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 introduce out-of-distribution (OOD) dependencies and positional patterns that diverge from pretraining data (Ding et al., 2024; Peng et al., 2024; Li et al., 2024b), resulting in degraded calibration and unstable reasoning during length extrapolation.

Context selection has emerged as an effective strategy for mitigating long-context inference challenges by restricting attention to relevant Key-Value (KV) entries, thereby reducing attention dilution and computational overhead. Nevertheless, most existing approaches (Wu et al., 2025b; Hao et al., 2025; Tang et al., 2024) maintain a fixed context budget throughout decoding, which implicitly assumes that generation difficulty remains uniform across tokens. This assumption is often violated in practice: many tokens can be reliably predicted from local context, whereas others depend on long-range evidence dispersed across the prompt. Consequently, fixed-size context windows may be unnecessarily costly for easy generation steps yet inadequate for difficult ones, highlighting the need for adaptive token-wise context allocation during decoding.

A natural implication is that context require-

ments vary at the token level and should be adjusted dynamically during decoding. Token-wise generation difficulty can be inferred from uncertainty signals available at inference time, enabling adaptive context control: decoding generally defaults to a compact window, and when high uncertainty or hallucination risk is detected, the process reverts, expands the context window, and regenerates 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 **UT-ACA**, an inference-time framework that dynamically adjusts the context window based on token-wise uncertainty. As shown in Figure 1, UT-ACA generally decodes with a compact context window and expands it only when insufficient evidence or elevated hallucination risk is detected. Uncertainty is estimated via a learned dual-encoder detector combining logit margin and semantic embeddings from hidden states, a lightweight strategy that avoids the overhead of multi-sample decoding (Khairi et al., 2025; Li et al., 2025a; Kuhn et al., 2023). To train the detector, we construct a synthetic dataset of 10,000 training, 120 validation, and 100 test samples from LLM output logits and hidden states, and adopt a dual-encoder architecture that fuses semantic and confidence features with an LSTM-based temporal module. Experiments show that UT-ACA generalizes across diverse long-context settings, achieving comparable or improved generation quality while reducing context token consumption.

2 Preliminaries

At inference time, LLMs maintain a KV cache of past token representations across attention layers (Hooper et al., 2024; Liu et al., 2024). At each decoding step t, we define the *context window* $\mathcal{C}_t \subseteq \{(K_i, V_i)\}_{i=1}^{t-1}$ as the subset of KV entries participating in attention computation, and self-attention is computed as

$$\mathrm{Attn}(Q_t, \mathcal{C}_t) = \mathrm{softmax}\left(\frac{Q_t K_{\mathcal{C}_t}^\top}{\sqrt{d}}\right) V_{\mathcal{C}_t}. \quad (1)$$

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 $|\mathcal{C}_t|$.

Rollback and Regeneration. Autoregressive generation is susceptible to error accumulation (Huang et al., 2024): uncertain or incorrect predictions at earlier steps 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, a rollback operation restores the state from ξ_t to ξ_{t-1} by discarding x_t and its KV cache entries, after which the token is regenerated under an expanded context window.

Problem Formulation. Given a long-context input and an autoregressive LLM, this work studies how to adaptively allocate the context window $\mathcal{C}_t$ at each decoding step. Rather than committing to a fixed context budget throughout decoding, the goal is to default to a compact window when local evidence is sufficient, and to detect uncertain steps that warrant rollback, context expansion, and regeneration. In this way, the model aims to reduce average context token consumption while preserving generation quality in long-context inference.

3 UT-ACA Framework

We propose **UT-ACA**, an inference-time framework consisting of two core modules, the *uncertainty detector* and the *adaptive context window*, which jointly enable token-wise dynamic context allocation during decoding, as illustrated in Figure 2. At most decoding steps, UT-ACA performs a tentative generation with a compact context window and feeds both the output logits and hidden-state embeddings into the *uncertainty detector* to obtain the *Generation Difficulty Metric* (GDM). Conditioned on the GDM, the *adaptive context window* module either accepts the tentative token and proceeds with the compact window, or triggers rollback, context expansion, and regeneration with enriched contextual support.

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 (Zhao et al., 2025c; Qiu et al., 2024), producing a three-way generation difficulty metric that is later used to trigger context reallocation.

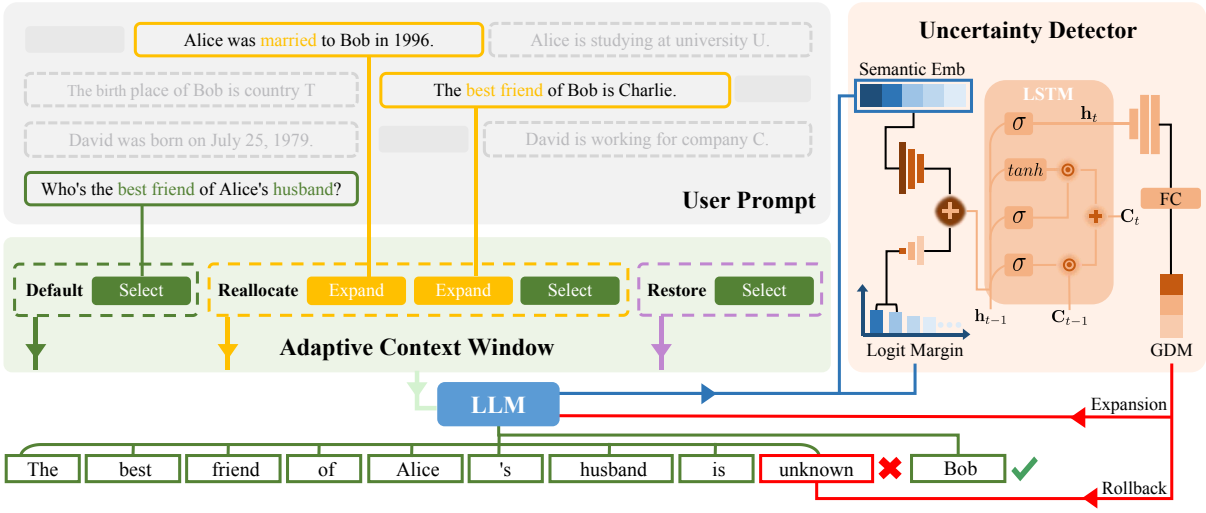

Figure 2: Workflow of our framework. The figure illustrates how UT-ACA connects long-context input, token-level uncertainty estimation, and adaptive context allocation during decoding. (a) The prompt provides the long-context input and target instruction. (b) The uncertainty detector converts output logits and semantic embeddings into a Generation Difficulty Metric (GDM). (c) The adaptive context window accepts low-uncertainty tokens, but rolls back, expands the retrieved context, and regenerates the token when the GDM indicates insufficient support.

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

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

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 logits with similar values, resulting in a comparatively small logit margin during generation. To handle this issue, we extract a token semantic embedding from the inner states of the LLM. Following prior work on hidden-state analysis (Ma et al., 2025; Ferrando et al., 2024; Singh et al., 2024; Chen et al., 2024), we use the output of the attention sub-layer in the final transformer 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

$$\mathbf{z}_t = \mathrm{LN}[\mathbf{W}_e \mathbf{emb}_t + \mathbf{b}_e + \mathrm{MLP}_m[m_t]], \tag{3}$$

where MLP_m lifts the scalar margin to $\mathbb{R}^d$ so as to bridge the dimensionality gap between the logit margin and the semantic embedding, ensuring both signals contribute comparably to the final GDM, and LN denotes layer normalization. Since uncertainty can propagate across autoregressive steps and compound after an early mistake, we apply an LSTM to explicitly model this temporal accumulation and aggregate historical evidence

$$\mathbf{h}_t, \mathbf{c}_t = \mathrm{LSTM}[\mathbf{z}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}], \tag{4}$$

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

Output as generation difficulty metric. We define the *generation difficulty metric* (GDM) as a three-way probability vector over the token-generation scenarios shown in Figure 3, including grounded correct generation (Correct Answer), unknown-style abstention (Unknown), and unsupported content generation (Hallucination). This three-way design is central to UT-ACA because it separates reliable generation from two distinct failure modes under insufficient contextual evidence. The prediction head maps $\mathbf{h}_t$ to the GDM:

$$\begin{aligned} \mathbf{p}_t &= \mathrm{softmax}[\mathbf{W}_g \mathbf{h}_t + \mathbf{b}_g], \\ \mathbf{p}_t &= [p_{t,\mathrm{cor}}, p_{t,\mathrm{unk}}, p_{t,\mathrm{hal}}], \\ \mathbf{p}_t &\in \mathbb{R}^3, \quad \sum_{k \in [3]} p_{t,k} = 1, \end{aligned} \tag{5}$$

where $p_{t,\mathrm{cor}}$ corresponds to grounded correct generation, $p_{t,\mathrm{unk}}$ to unknown-style abstention, and

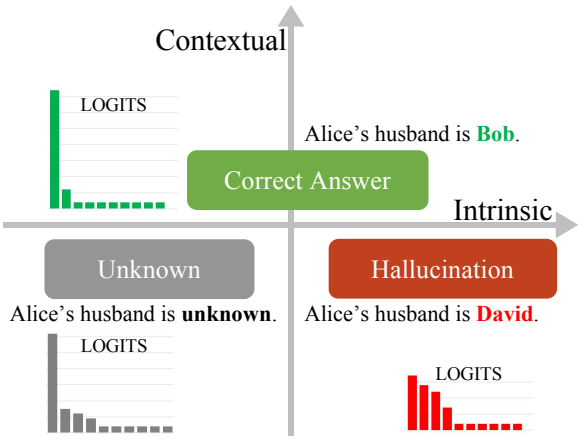

Figure 3: Token-generation scenarios for the Generation Difficulty Metric. The figure categorizes grounded answers, abstentions, and hallucinations by whether the required information is supported by the context or by the intrinsic knowledge of the model.

$p_{t,\text{hal}}$ to hallucination.

A binary formulation is insufficient for this setting because it merges *Unknown* and *Hallucination* into a single negative class, despite their different confidence patterns. Unknown tokens often correspond to confident abstention phrases and can exhibit high logit margins comparable to correct predictions, whereas hallucinated tokens introduce unsupported content. This mismatch introduces label noise and weakens the ability of the detector to identify when additional context is needed. In practice, the downstream policy triggers context reallocation when the non-grounded probability mass exceeds the grounded mass:

$$p_{t,\text{unk}} + p_{t,\text{hal}} > p_{t,\text{cor}}, \tag{6}$$

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

3.2 Adaptive Context Window

Inspired by InfLLM (Xiao et al., 2024a), 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 initialized at K_{max} for the first token to establish contextual grounding and subsequently adjusted by the update policy. At each step, the output logits and hidden-state embeddings are fed 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

$$\mathcal{S}_t(k) = \arg\max_{\substack{\mathcal{S}\subseteq\mathcal{B} \\ |\mathcal{S}|=k}} \sum_{B\in\mathcal{S}} \text{sim}[\mathbf{q}_t, \mathbf{r}(B)]\,, \tag{7}$$

where $\text{sim}[\cdot,\cdot]$ denotes the dot-product similarity between the decoding query and the representative keys of each block, and the expansion condition is formally defined as $p_{t,\text{unk}} + p_{t,\text{hal}} > p_{t,\text{cor}}$. 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 $k \leftarrow K_{\text{max}}$, $t \leftarrow 0$
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
    $k \leftarrow K_{\text{max}}$ ▷ 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 evaluate UT-ACA from three perspectives, including uncertainty detection, long-context generation, and cross-domain generalization. We first describe the synthetic biography dataset and eval-

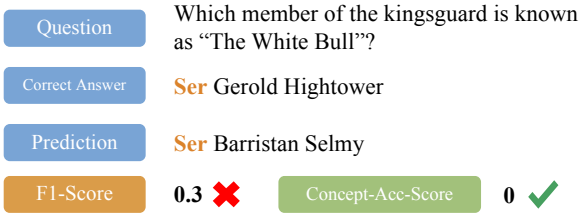

Figure 4: Example comparison between F1 score and Conceptual Accuracy Score (CAS). F1 score gives partial credit for surface overlap, while CAS assigns zero because the prediction is semantically incorrect.

uation metrics, then compare the learned uncertainty detector with logit-based heuristic baselines. Finally, we integrate the detector into the adaptive context allocation framework and evaluate its efficiency-quality trade-off against fixed-budget context-management methods.

4.1 Dataset and Metrics

We construct a synthetic biography summarization dataset using a summary-first pipeline, where each instance contains a randomly generated attribute value for a fictitious person and is expanded from a ground-truth summary into a longer biography. Person names are unique across splits to prevent data leakage. For token-level supervision, we constrain the context window during decoding, record output logits and hidden-state embeddings, and label each generated token as correct, unknown, or hallucinated using GPT-OSS-120B (OpenAI, 2025) with manual verification. The dataset includes 10,000 training samples, a validation set with six unseen attributes and 20 summaries each, and 100 Wikipedia-extended long-context test biographies of 171k-400k tokens. Data examples are provided in Appendix A.2.

Metrics. For the uncertainty detector, we report mean accuracy (**mAcc**) and per-class recall for uncertain and certain samples (**Recall**$_N$ and **Recall**$_P$). For generation quality, GPT-OSS-120B scores conceptual alignment between model outputs and ground truth, reported as Conceptual Accuracy Score (**mAcc**$_{\text{conc}}$ a.k.a. CAS). An illustrative comparison is shown in Figure 4. Efficiency is quantified by mean context token consumption (**mTokens**) and average per-token decoding latency (**mTime**$_{\text{tok}}$).

4.2 Uncertainty Detection Experiments

We train and evaluate the uncertainty detector on logits and hidden-state embeddings extracted from Llama-3.1-8B-Instruct (Dubey et al., 2024), with token labels assigned by the GPT-OSS-120B described above. As summarized in Table 1, we compare against two heuristic baselines: TOP-10 STD, which uses the standard deviation of the top-10 logits as an uncertainty proxy, and LOGIT MARGIN, which uses the top-1 versus top-2 margin. Both tune a decision threshold on the training set. 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: Token-level uncertainty detection performance. The table compares the learned UT-ACA detector with two heuristic logit-based baselines. mAcc denotes overall accuracy; Recall$_N$ and Recall$_P$ denote recall on uncertain and certain tokens, respectively.

4.3 Generation Experiments

We evaluate the generation performance of UT-ACA with InfLLM (Xiao et al., 2024a) as the closest comparable baseline, under consistent experimental settings across all methods.

Validation set. We compare two post-expansion update rules, *Update: Set.1* and *Update: Sub.1*. The former resets the context block budget to 1 for the next token, whereas the latter decreases the budget by 1 while maintaining a minimum value of 1. All methods use a fixed block size of 16 tokens and a maximum block budget of $K_{\max} = 3$. As shown in Table 2, UT-ACA achieves 99.08% **mAcc**$_{\text{conc}}$ with only 29 context tokens on average, approaching InfLLM's peak performance (99.83% at 48 tokens) while reducing context usage by approximately 40%. These results indicate that uncertainty-triggered expansion selectively allocates additional context without unnecessarily increasing the average context budget.

Long-context test set. We further evaluate UT-ACA on a long-context test set consisting of biographies with 171k-400k tokens, with an average length of 252k tokens. These inputs exceed the nominal context lengths of both Llama-3.1-8B-Instruct and Qwen2-7B-Instruct (Yang et al., 2024), providing a challenging setting for context-management methods. For each backbone, we train a separate uncertainty detector using the same pro-

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

Table 2: Generation performance on the validation set. UT-ACA is compared with fixed-budget InfLLM in terms of average context tokens (mTokens) and conceptual accuracy ($\text{mAcc}_{\text{conc}}$), showing that adaptive allocation preserves high accuracy with fewer tokens.

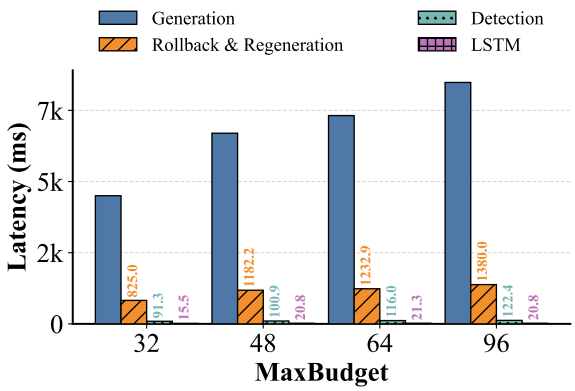

Figure 5: Per-component latency breakdown across `MaxBudget` settings (base generation, rollback & regeneration, uncertainty detection, and LSTM forward pass).

cedure as above. Table 3 reports context usage, per-token decoding latency, and conceptual accuracy across compression-, token-selection-, and block-retrieval-based baselines. For token-selection methods, K_{tok} denotes the number of tokens selected per decoding step, while K_{com} denotes the compression ratio for LLMLingua-2. InfLLM-V2 follows its original block size of 64 with $K = 16$ retrieved blocks. For a direct comparison between InfLLM and UT-ACA, both methods use a block size of 16, where K denotes the fixed block budget of InfLLM and $K_{\max}$ denotes the maximum block budget available to UT-ACA.

As shown in Table 3, UT-ACA consistently achieves a favorable efficiency-quality trade-off across both backbones. On Llama-3.1-8B-Instruct, UT-ACA with $K_{\max} = 96$ attains 74.61% $\text{mAcc}_{\text{conc}}$ at 498 tokens, outperforming the best InfLLM (72.26% at 512 tokens); at $K_{\max} = 64$, it reaches 70.48% with only 344 tokens, a 33% reduction at a 1.78% accuracy cost. On Qwen2-7B-Instruct, UT-ACA ($K_{\max} = 48$) similarly exceeds InfLLM's peak (56.64% vs. 54.72%, 432 vs. 512 tokens), confirming the consistent efficiency-quality advantage of adaptive allocation.

4.4 Latency Analysis

We conduct a detailed latency breakdown to assess the practical overhead introduced by UT-ACA's uncertainty detection, rollback, and regeneration mechanisms. As shown in Figure 5, end-to-end runtime is decomposed into four components across varying `MaxBudget` settings (block size = 16).

Base generation dominates overall latency (4.50–8.48 s), confirming that UT-ACA's control logic does not alter the decoder's fundamental latency profile. Uncertainty detection contributes only 91.3–122.4 ms and is insensitive to `MaxBudget`, consistent with the lightweight LSTM-based detector (1.39M parameters, 15.5–21.3 ms). Rollback-based regeneration adds 0.83–1.38 s but scales more slowly than base generation: as `MaxBudget` grows from 32 to 96, base generation increases by 3.98 s while rollback overhead grows by only 0.55 s, confirming that UT-ACA incurs predictable and bounded overhead across all budget settings.

4.5 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: Sub.16* configuration. The contribution of each component in the uncertainty detector is evaluated through systematic ablation by removing and combining the three components.

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.

4.6 Open-Set Benchmark Experiments

We further evaluate UT-ACA on three open-set long-context benchmarks, including ∞-Bench, LongBench, and RULER, to examine whether the learned uncertainty detector generalizes beyond the synthetic biography domain. The detector is

Models	Methods	Settings	mTokens ↓	$\text{mTime}_{\text{tok}}$ (s) ↓	$\text{mAcc}_{\text{conc}}$ (%) ↑
Llama-3.1-8B-it	†RetrievalAttn (Zhu et al., 2024)	$K_{\text{tok}} = 2048$	2k	0.021	15.25
	†LLMLingua-2 (Pan et al., 2024)	$K_{\text{com}} = 25\%$	32k	-	18.08
	†SnapKV (Li et al., 2024c)	$K_{\text{tok}} = 1024$	1k	0.210	23.43
	TokenSelect (Wu et al., 2025b)	$K_{\text{tok}} = 1024$	1k	-	39.43
	OmniKV (Hao et al., 2025)	$K_{\text{tok}} = 6.7\%$	8k	-	40.47
	†InfLLM-V2 (Zhao et al., 2025a)	$K_{\text{tok}} = 1024$	1k	-	40.29
	FastKV (Jo et al., 2026)	$K_{\text{tok}} = 4096$	4k	0.212	51.40
	H_2O (Zhang et al., 2023)	$K_{\text{tok}} = 1024$	1k	0.123	65.81
	InfLLM (Xiao et al., 2024a)	$K = 8$	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	InfLLM (Xiao et al., 2024a)	$K = 8$	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: Context-management performance on the long-context test set, where Update: Sub.*N* denotes the policy of decreasing the block budget by *N* after each accepted token, with a minimum of 1. The table compares token selection, compression, fixed-budget retrieval, and UT-ACA across two backbones using context usage (mTokens), per-token latency ($\text{mTime}_{\text{tok}}$), and conceptual accuracy ($\text{mAcc}_{\text{conc}}$). Methods marked with † exclude samples with out-of-memory errors or incoherent outputs.

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

Table 4: Component ablation of the uncertainty detector. The table evaluates the contribution of the Logit Margin branch (LogM), the Semantic Embedding branch (SE), and the LSTM module by removing them in different combinations. mAcc measures detector accuracy, and $\text{mAcc}_{\text{conc}}$ measures downstream conceptual generation accuracy.

directly reused from the version trained on our synthetic dataset, without any benchmark-specific retraining. We follow the official evaluation protocol of each benchmark. For ∞-Bench, we report the benchmark-provided scores for summarization and QA tasks. For LongBench, we evaluate narrativeqa with normalized token-level F1 and qmsum with ROUGE-L $\times$ 100. For RULER, we report the official string-match accuracy, which measures whether any normalized gold answer appears in the model prediction.

∞-Bench (Zhang et al., 2024). Following the official settings with block size 128 and maximum block budgets $\{8, 16, 32\}$, UT-ACA attains comparable accuracy to InfLLM on the Sum and QA subtasks while consistently consuming a smaller context window across all budget levels (Table 5).

LongBench (Bai et al., 2024). With block size 16 and maximum budgets $\{4, 8, 16\}$, UT-ACA outperforms InfLLM on both narrativeqa and qmsum under comparable or smaller token budgets (Table 5). On qmsum in particular, UT-ACA achieves its peak score of 19.18 at K_{max}=16 using only 102 tokens, versus InfLLM's 19.13 at 256 tokens, representing a 60% reduction in token consumption.

RULER (Hsieh et al., 2024). On QA-16k and QA-8k (block size 16, maximum budgets $\{8, 16, 32\}$), UT-ACA consistently achieves a more favorable efficiency–quality trade-off than InfLLM (Table 5): matching its peak on QA-16k (21.65 vs. 21.70, 385 vs. 512 tokens) and surpassing it on QA-8k (26.85 vs. 26.75, 352 vs. 512 tokens).

Method	∞-Bench				LongBench				RULER			
	Sum Task		*QA Task*		*narrativeqa*		*qmsum*		*QA-16k*		*QA-8k*	
	mTok↓	Score↑	mTok↓	Score↑	mTok↓	Score↑	mTok↓	Score↑	mTok↓	Score↑	mTok↓	Score↑
H_2O	1024	11.25	1024	1.14	64	5.35	64	13.84	128	12.00	128	14.30
	2048	11.52	2048	0.93	128	4.68	128	14.25	256	11.00	256	16.60
	4096	10.59	4096	1.27	256	4.44	256	14.81	512	10.70	512	21.00
InfLLM	1024	27.24	1024	16.27	64	14.95	64	18.40	128	14.05	128	14.90
	2048	27.87	2048	18.38	128	15.75	128	18.62	256	18.75	256	24.30
	4096	**28.58**	4096	**22.09**	256	15.57	256	19.13	512	**21.70**	512	26.75
UT-ACA	609	27.46	896	16.58	39	15.01	24	17.65	48	10.09	51	12.60
	1610	27.64	1664	18.72	59	16.11	47	18.66	105	14.65	108	14.40
	3508	28.43	**3712**	21.32	**109**	**16.92**	**102**	**19.18**	**385**	21.65	**352**	**26.85**

Table 5: Open-set benchmark results on ∞-Bench, LongBench, and RULER. The table compares H_2O, InfLLM, and UT-ACA under the official evaluation protocols of the corresponding benchmarks, without retraining the UT-ACA uncertainty detector. mTok denotes the average number of selected tokens, Score follows the corresponding benchmark metric, and bold values indicate the best result in each column.

Taken together, results across all three benchmarks consistently demonstrate that UT-ACA selectively retains task-relevant context, achieving competitive or superior accuracy while reducing average token consumption relative to InfLLM and H_2O, thereby validating the generalization of its uncertainty-driven context allocation strategy beyond the synthetic evaluation setting.

5 Related Work

Long-context inference has been studied from two complementary directions. One line applies training-based methods (Chen et al., 2025; Hu et al., 2025; Gao et al., 2025; Tian et al., 2025) or improved positional encodings (Ding et al., 2024; Jin et al., 2024; Wang et al., 2024) to extend length generalization. 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 the latter.

Context selection. These methods retain only the most relevant tokens or blocks to mitigate attention dilution (Zhang et al., 2025a; Xiao et al., 2024b). Training-free approaches such as InfLLM (Xiao et al., 2024a) and TokenSelect (Wu et al., 2025b) perform block- or token-level retrieval over the KV cache, while structured sparse attention methods such as XAttention (Xu et al., 2025) and OmniKV (Hao et al., 2025) design scoring-guided selection to balance efficiency and accuracy. InfLLM-V2 (Zhao et al., 2025a) and FastKV (Jo et al., 2026) further push efficiency via trainable sparse attention and accelerated eviction strategies, respectively. Despite their diversity, all these methods share a fixed-budget assumption and do not adapt context allocation to token-level generation difficulty.

Context compression. Rather than selecting existing entries, compression methods shrink the input or KV representations (Zhao et al., 2025b; Liao et al., 2025; Wang et al., 2025; Li et al., 2023). Prompt-level approaches such as LLMLingua (Jiang et al., 2023) and LongLLMLingua (Jiang et al., 2024) reduce token count using information-theoretic criteria. At the KV-cache level, SnapKV (Li et al., 2024c), SCOPE (Wu et al., 2025a), and H_2O (Zhang et al., 2023) prune or evict entries based on attention patterns, while FocusLLM (Li et al., 2025b) condenses long inputs into compact representations. Like context selection, these methods apply a uniform strategy across all decoding steps, leaving token-wise difficulty unaddressed.

6 Conclusion

We propose UT-ACA, an uncertainty-triggered adaptive context allocation framework that dynamically adjusts the context window at the token level during decoding. Tokens are tentatively generated under a compact window and regenerated with expanded context only when the uncertainty detector signals insufficient evidence, enabling selective allocation without a fixed budget. Experiments across synthetic and open-set long-context benchmarks confirm that UT-ACA substantially reduces average context token consumption 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.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. INSIDE: LLMs' internal states retain the power of hallucination detection. In *International Conference on Learning Representations (ICLR)*, pages 1–21.

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.

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.

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.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, pages 1–27.

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.

Dongwon Jo, Jiwon Song, Yulhwa Kim, and Jae-Joon Kim. 2026. FastKV: Decoupling of context reduction and KV cache compression for prefill-decoding acceleration. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 1–20. Association for Computational Linguistics.

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.

Yucheng Li, Bo Dong, Chenghua Lin, and Frank Guerin. 2023. Compressing context to enhance inference efficiency of large language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6342–6353.

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.

Huan Ma, Jingdong Chen, Joey Tianyi Zhou, Guangyu Wang, and Changqing Zhang. 2025. Estimating LLM uncertainty with evidence. *arXiv preprint arXiv:2502.00290*, pages 1–14.

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.

Guangtao Wang, Shubhangi Upasani, Chen Wu, Darshan Gandhi, Jonathan Li, Changran Hu, Bo Li, and Urmish Thakker. 2025. LLMs know what to drop: Self-attention guided KV cache eviction for efficient long-context inference. In *International Conference on Learning Representations (ICLR)*, pages 1–7.

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. 2024a. InfLLM: Unveiling the intrinsic capacity of LLMs for understanding extremely long sequences with training-free memory. *arXiv preprint arXiv:2402.04617*, pages 1–17.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024b. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, pages 1–21.

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.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang "Atlas" Wang, and Beidi Chen. 2023. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 34661–34710.

Weilin Zhao, Zihan Zhou, Zhou Su, Chaojun Xiao, Yuxuan Li, Yanghao Li, Yudi Zhang, Weilun Zhao, Zhen Li, Yuxiang Huang, Ao Sun, Xu Han, and Zhiyuan Liu. 2025a. Infllm-v2: Dense-sparse switchable attention for seamless short-to-long adaptation. *Preprint*, arXiv:2509.24663.

Yunlong Zhao, Haoran Wu, and Bo Xu. 2025b. 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. 2025c. 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 7, 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 7: Standardized pattern of synthetic summaries. The figure illustrates the controlled person–attribute–value structure used to define ground-truth facts. This standardized structure keeps the ground truth unambiguous while allowing the surrounding biography text to vary in wording and length.

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 6 lists all attribute names used in our syn-

Dataset	Attribute Names	Example Data
Training Set	• Marriage 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 marriage 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 6: Attribute sets and example facts in the synthetic dataset. The table lists representative attribute names and example facts for the training, validation, and test splits. Training uses diverse single-attribute facts, while validation and test focus on six held-out biography attributes.

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

Table 7: Wikipedia source articles used for test-set context extension. These passages are inserted around synthetic biographies to create long distractor contexts, testing whether methods can retrieve relevant biography facts from long, heterogeneous inputs.

thetic 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.

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 6 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 7, 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 8 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 extraction 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 Implementation Details

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

B.1 Training Details

We implement our framework in Python 3.11 using PyTorch 2.4. Training is conducted on a single NVIDIA H100 GPU with eight concurrent runs, each taking approximately 1.5 GPU-hours on average. We train up to 10,000 epochs per run with a batch size of 16. For checkpoint selection, we adopt F1 score as the evaluation metric and select the checkpoint that achieves the highest F1 on the validation set.

{"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 6: Prompt template for generating naturalistic biographies. The figure shows the LLM instruction used to convert structured ground-truth facts into fluent biography passages. The prompt asks the LLM to preserve the provided summary facts while embedding them in natural narrative text, producing test passages that require extraction from natural language rather than template matching.

B.2 Architecture 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 followed by a GELU activation and dropout

$$\mathbf{h}_{vec}^{(t)} = \text{Dropout}(\text{GELU}(\mathbf{W}_e\mathbf{emb}_t + \mathbf{b}_e)) \quad (8)$$

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

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 8: Representative examples from the training, validation, and test splits. The table shows how the three splits differ in attribute complexity and context length. Training and validation examples contain one explicit fact in a short template-like passage, whereas test examples contain six biography attributes expressed in natural language and surrounded by long-context distractors.

an input $\mathbf{h}$, the block operation is defined as

$$\begin{aligned} \hat{\mathbf{h}} &= \text{Dropout}(\text{GELU}(\mathbf{W}_1\mathbf{h} + \mathbf{b}_1)) \\ \mathbf{h}_{out} &= \mathbf{h} + \text{Dropout}(\mathbf{W}_2\hat{\mathbf{h}} + \mathbf{b}_2) \\ \mathbf{p}_{out} &= \text{softmax}(\mathbf{h}_{out}) \end{aligned} \tag{9}$$

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 9. 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 evaluated 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 9: Hyperparameter configuration of the Llama-3.1-8B uncertainty detector. The table reports the input dimensions, encoder settings, residual-head configuration, and number of output classes. The detector projects high-dimensional LLM features into a compact 64-dimensional hidden space before classification.

B.3 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 8, 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. Based on these results, we adopt the attention output (option 1) as the default extraction point in all experiments, as it is the most directly accessible in standard implementations.

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

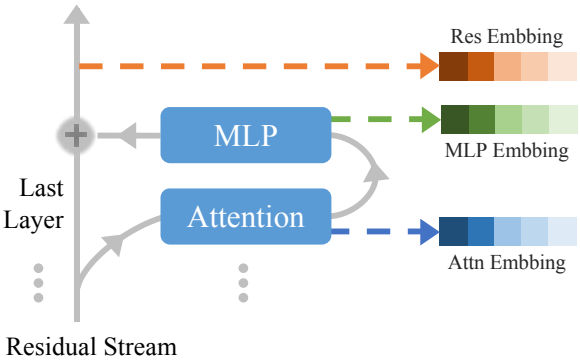

Figure 8: Extraction points for semantic embeddings in the final transformer layer. The figure marks the attention output, MLP output, and residual stream in the final layer (e.g., Layer 32 of Llama-3.1-8B). These positions are evaluated as candidate internal states for token-level semantic features.

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 9. 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.

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 10, 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 11 presents representative examples that demonstrate the effectiveness of our conceptual accuracy score. Specifically, in the evaluation samples, CAS can automatically

```
{"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 9: Prompt template for evaluating Conceptual Accuracy Score (CAS). The figure shows the LLM-judge instruction used to score semantic correctness rather than surface overlap. For each subtask, the prompt provides the question, the model prediction, and the ground-truth answer, then requests a floating-point score between 0 and 1 that reflects conceptual accuracy.

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 10: Human validation of Conceptual Accuracy Score on biography extraction. The table compares string matching (**Str**), conceptual accuracy score (**CAS**), and human annotation (**Hum**) on 30 test samples. **Cor** reports the number of correctly extracted attributes out of six according to human annotators.

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 11: Case study of CAS under format variation and semantic errors. Case #3 shows that date and entity-format differences can still be semantically correct, whereas Case #17 contains wrong attribute values that should receive low scores. ✔ and ✘ indicate human judgments for each attribute.

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 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.

D Additional Results and Analysis

D.1 Ablation Study on Two-Way vs. Three-Way Difficulty Metric

To examine the contribution of the proposed three-way difficulty metric, we conduct an ablation study in which the *Unknown* and *Hallucination* categories are merged into a single class, yielding a two-way difficulty metric. Under this setting, we perform eight independent training runs using the same experimental protocol. For each run, the checkpoint achieving the highest F1 score on the validation set is retained. The selected detectors are then evaluated on the test set with the following configuration: `MaxBudget` $= 64$, `block_size` $= 16$, and `SubPolicy` $= 16$.

As shown in Table 12, merging *Unknown* and *Hallucination* into a single category leads to consistent performance degradation across both detection and downstream generation metrics. At the detector level, mean accuracy (mAcc) decreases by 0.69 points (83.43% $\rightarrow$ 82.74%), and the recall for negative samples drops by 1.67 points (89.71% $\rightarrow$ 88.04%). More critically, the degradation propagates to the downstream generation stage: conceptual generation accuracy ($\text{mAcc}_{\text{conc}}$) declines substantially by 2.49 points (70.48% $\rightarrow$ 67.99%),

Setting	mAcc (%) ↑	Recall_N (%) ↑	Recall_P (%) ↑	mTokens ↓	mAcc_conc (%) ↑
Two-Way Metric	82.74 (↓0.69)	88.04 (↓1.67)	80.60 (↓0.67)	348	67.99 (↓2.49)
Three-Way Metric	**83.43**	**89.71**	**81.27**	**344**	**70.48**

Table 12: Ablation study of two-way versus three-way difficulty metrics. The two-way metric merges Unknown and Hallucination into one negative class, while the three-way metric keeps them distinct. mAcc denotes detector accuracy, Recall_N and Recall_P denote recall on negative and positive samples, mTokens denotes average context token consumption, and mAcc_conc denotes conceptual generation accuracy.

while average context token consumption remains nearly unchanged (344 vs. 348 tokens), confirming that the gain is attributable to improved detection quality rather than increased token expenditure.

This degradation stems from the distinct generative regimes underlying *Unknown* and *Hallucination* responses. Under compact context windows, the model tends to produce *Unknown* responses with disproportionately high confidence, whose logit margins are closer to correct outputs than to hallucinated ones. Merging them into a single negative class therefore conflates heterogeneous behaviors, introducing inconsistent supervision signals that increase label ambiguity and destabilize training. Moreover, the two failure modes warrant distinct rollback strategies: *Unknown* tokens benefit from context expansion and regeneration, whereas *Hallucination* reflects intrinsic model limitations less amenable to retrieval-based correction. Preserving this distinction enables more fine-grained rollback decisions, ultimately yielding higher-quality generation under the same token budget.

D.2 Case Study: Effect of Context Expansion on Token Probability

To provide a concrete and interpretable analysis of how context window expansion affects the model's predictive confidence, we present a single-generation case study. The example involves extracting the birth date of a person from a long-context passage, a task that exhibits a pronounced uncertainty spike at the day token.

Under the original compact setting (`top-k` $= 1$, `block_size` $= 16$), the correct token "`29`" receives a log-probability of -5.13 and is ranked third among the candidate tokens {"`9`", "`15`", "`29`"}. After context window expansion (`top-k` $=$ 64), the token "`29`" becomes the top-ranked candidate, with its log-probability improving to -4.56. Table 13 summarizes the token rankings and log-probabilities before and after expansion.

Setting	Top-1	Rank of "29"	Log p("29") ↑
Before Expansion	"9"	3rd	-5.13
After Expansion	"29"	1st	-4.56

Table 13: Token ranking and log-probability before and after context expansion. The table reports the candidate ranking for the day token in a birth-date extraction example. Before expansion, the correct token "`29`" ranks third; after expansion, it becomes top-ranked with log-probability improving from -5.13 to -4.56.

This case study illustrates that context expansion can effectively mitigate uncertainty by shifting probability mass toward the correct token. The improvement in both rank and log-probability of "`29`" after expansion supports the hypothesis that insufficient context leads to degraded token-level confidence, and that restoring broader context allows the model to recover a more discriminative distribution over candidates.

E Artifact and Data Usage

Pre-trained Models. We use Llama-3.1-8B-Instruct (Dubey et al., 2024) and Qwen2-7B-Instruct (Yang et al., 2024) solely for academic research, consistent with their respective licenses (Meta Llama 3.1 Community License and Apache 2.0).

Baseline Implementations. All baseline methods are used strictly for non-commercial academic comparison, consistent with their intended use as research artifacts.

Synthetic Dataset. Our synthetic biography summarization dataset is constructed entirely in-house and intended for research purposes only. The long-context test set incorporates Wikipedia text retrieved for academic evaluation; Wikipedia content is licensed under CC BY-SA 4.0, and our use is strictly within a non-commercial research context.

Automated Evaluation. An LLM-as-judge protocol is used for evaluation, consistent with the model's intended research usage terms.