DOWNSTREAM-AWARE CONTEXT SELECTION FOR ONLINE IN-CONTEXT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In-context reinforcement learning (ICRL) enables large language model agents to adapt to new environments using their interaction history without updating model parameters. However, repeatedly conditioning on growing histories can lead to substantial token cost. We propose a bounded-history context-management framework that predicts the task-dependent downstream effect of removing historical interactions to guide history selection and determine a decision-dependent context budget. Formally, our framework uses the full rolling history as a reference. The predictor evaluates removal effects, defines a deletion ordering, and applies a shared selection criterion to determine how much history to retain at each decision. We evaluate the method in closed-loop SUMO driving under held-out in-distribution, unseen-domain, and unseen-route settings, and in ScienceWorld under a continual ICRL protocol. Relative to a baseline using the full context, our method reduces total token usage by 25.7%, 25.8%, and 23.2% across the three driving settings while maintaining comparable closed-loop driving performance. In ScienceWorld, it reduces total token usage by 52.1% compared to full context and uses 30.2% and 37.8% fewer tokens than the Recent and Similarity baselines, respectively, while maintaining performance.

1 INTRODUCTION

Large language models (LLMs) have increasingly been used for sequential decision-making tasks, where previous interactions can be retained in context to support adaptation during deployment. In-context learning is useful for these tasks since it allows models to use accumulated task-specific experience and feedback during interaction to adapt behaviors. In-context reinforcement learning (ICRL) extends this idea by conditioning a policy on past state-action-reward interactions, allowing the policy to adapt during inference without updating its model parameters For instance, ICRL has enabled policy adaptation and self-improvement by providing reward feedback Song et al. (2026); Monea et al. (2025). This capability is particularly relevant to closed-loop autonomous systems, where recent ICL/ICRL methods have been used for inference-time adaptation and decision making under changing conditions Khurram et al. (2025); Yuan et al. (2024). However, as an agent continues to interact with the environment, its history grows and more information must be repeatedly included in the policy prompt.

Large context can increase computational cost Huang et al. (2024) and may affect adaptation in complex environments Schmied et al. (2025). Therefore, context selection and compression is meaningful. Existing approaches reduce long contexts through temporal abstraction, external-memory retrieval, and demonstration selection based on signals such as similarity, diversity, and learned relevance Huang et al. (2024); Schmied et al. (2025); Liu et al. (2022); Levy et al. (2023); Scarlatos & Lan (2024). Recent methods further move beyond fixed-size retrieval. Context-Picker learns variable-size sufficient evidence sets using removal-based supervision Zhu et al. (2026), while PACE varies how much of each past memory is retained, according to predicted relevance to the next action Wei et al. (2026). General prompt-compression methods such as LongLLMLingua instead reduce long inputs through token-level compression Jiang et al. (2024). Removal-based influence has also been studied for retrieved contexts in question-answering and RAG settings, where importance is estimated from the performance change induced by removing a candidate context item Deng et al. (2026). However, existing context selection methods address two neighboring but distinct problems. Influence-based and minimal-sufficient context selectors learn which retrieved histories matter for a given query, including approaches that supervise selection through the downstream effect of removing candidate contexts. Separately, long-horizon agent methods manage evolving interaction histories through relevance-guided retrieval, multi-level compression, or learned removal. However, in online ICRL, the context is an evolving state–action–reward trajectory generated by the same pol-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

icy being conditioned, and context reduction must be performed repeatedly as each action changes the history available to subsequent decisions.

We introduce a bounded-history context-selection framework for ICRL. The full rolling history serves as a reference, and a task-dependent decision-difference signal measures the downstream change induced by replacing it with a reduced subset. Directly evaluating this quantity for many candidate subsets would require repeated downstream-policy queries, so we train a lightweight predictor to estimate the effect of context removal. The predictor then defines a deletion order, and a selection criterion then selects how much history to retain at each decision. This allows the context budget to adapt online without separately tuning a fixed history size for each condition, without requiring a fixed context cardinality as an input to the selector as demonstrated in Fig.1.

We evaluate our method in two distinct sequential decision-making domains. In a closed-loop SUMO driving environment based on the ICRL framework of Khurram et al. (2025). we consider held-out in-distribution episodes, an unseen traffic domain, and unseen routes. Relative to Full-H20, our method reduces total token usage by 23.2–25.8% across these settings while maintaining comparable closed-loop driving performance.

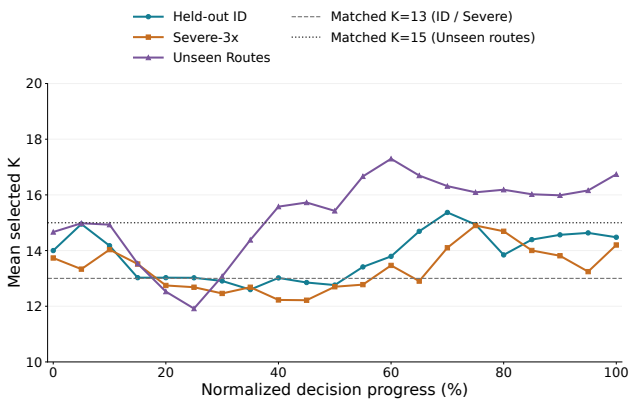

Figure 1: Mean selected context size K across normalized decision progress under held-out ID, Severe-3x, and unseen-route evaluation. Dashed and dotted lines denote the matched fixed-K baselines.

We further evaluate the framework on ScienceWorld following the continual ICRL protocol of Wang et al. (2022); Song et al. (2026). Our method reduces average total token usage by 52.1% relative to Full-H20, while achieving a return of 96.06 compared with 95.50 for Full-H20. It also reduces token usage by 30.2% relative to Recent and 37.8% relative to Similarity. A matched-budget comparison with LongLLMLingua Jiang et al. (2024) shows comparable ScienceWorld performance with substantially lower observed context-management overhead. These results show that the downstream-consequence-based context selection can provide effective context management across different forms of sequential decision making and downstream supervision.

Our main contributions are:

1. We formulate the online context selection problem for ICRL, with the goal of reducing the token cost of a growing interaction history during inference, while maintaining downstream task performance. We formulate it around the predicted downstream consequence of removing interaction history. Unlike selecting over a fixed retrieved set, the selector operates repeatedly inside the policy–environment loop.
2. We propose a context management framework which predicts the effect of removing historical interactions, aggregates predictor rankings into a deletion order that also guides redundancy compression within the retained context, and adaptively selects the retained history size online based on a selection criterion.
3. We evaluate the method under two distinct environments: a closed-loop autonomous driving environment under held-out in-distribution, unseen-domain, and unseen-route settings, and ScienceWorld. Across both domains, the method reduces token usage relative to full-context and heuristic baselines while maintaining competitive downstream performance, and matches a prompt-compression baseline under a comparable token budget at substantially lower selection cost.

2 Related Work

In-Context Learning/In-Context Reinforcement Learning In-context learning (ICL) allows pretrained language models to adapt to new tasks using examples and demonstrations provided in the prompt without updating model parameters or fine-tuning Brown et al. (2020). In reinforcement learning, Algorithm Distillation trains a causal sequence model on learning histories generated by a source RL algorithm, while the Decision-Pretrained Transformer predicts actions from a query state and an in-context dataset of past interactions Laskin et al. (2022); Lee et al. (2023)

More recent work studies ICRL directly with large language models. For instance, Monea et al. (2025) showed that LLMs can adapt online from external reward feedback in contextual bandit problems, while Song et al. (2026) introduced a multi-round ICRl prompting to guide LLM self-improvement with only numerical scalar reward. Chen et al. (2025) studied learning-history filtering and showed that reweighting and filtering suboptimal histories during ICRL pretraining can improve downstream performance. Interaction histories and memory have also been used in LLM-based autonomous driving for shared experience and reflection, inference-time domain adaptation, risk-sensitive memory retrieval, and closed-loop reasoning and memorization Jiang et al. (2024); Khurram et al. (2025); Fu et al. (2024b); Zhou et al. (2026a)

We focus on managing the accumulated context history during deployment. Rather than filtering learning histories before training or retrieving experiences using a fixed relevance criterion, we use a task-dependent estimate of the effect of context removal to construct the policy context online.

Context Selection for ICL The performance of ICL is sensitive to the included demonstrations in the prompt. Early retrieval methods select query-relevant examples using similarity or Language Model(LM)-scored relevance Liu et al. (2022); Rubin et al. (2022) Some recent work also considered selection beyond direct similarity. For example, Levy et al. (2023) designed a model to select diverse demonstrations to improve compositional generalization, while An et al. (2023) proposed Skill-KNN, which selects demonstrations based on task-related skill descriptions.

More recent methods make demonstration selection more dependent on the behavior of the language model (LM). Peng et al. (2024) indicated that demonstration quality depends on both the data and the model, and proposed a model-dependent selection method. Wang et al. (2024) proposed Relevance-Diversity Enhanced Selection (RDES) to select demonstrations based on relevance and diversity. Purohit et al. (2025) formulated Challenger Arm Sampling for Exemplar selection (CASE) framework to select from top m best-arm by a linear scoring function to reduce the number of LLM evaluations needed during subset search. Zhang et al. (2025) proposed GENICL that leverages LLM feedback to learn demonstrations preferred by the model. Other work moves beyond independent fixed-budget selection. For instance, Scarlatos & Lan (2024) solved sequential example selection via Markov Decision Process (MDP) and trained example retriever using RL, while Zhu et al. (2026) dynamically selects a sufficient context set through a two-stage RL framework. Deng et al. (2026) further explore contextual influence through the performance degradation caused by removing retrieved history and train model to predict this influence at inference time.

However, these methods improve how in-context examples are chosen, including through sequential and adaptive selection. These methods also prove that downstream or removal-based signals can outperform semantic criteria, but build on a candidate collection associated with a query rather than repeatedly managing a trajectory generated by an online policy.

Memory and Context Management Long-horizon language agents often use memory to retain information from previous interactions. Huang et al. (2024) reduce sequence length by replacing low-level action sequences with high-level decisions for interactions. MemGPT manages information across different memory levels to extend the effective context available to an LLM Packer et al. (2023). Other works directly reduce long prompts such as Jiang et al. (2023) and Jiang et al. (2024) compress input prompts while preserving important information, reducing the token cost of long-context inference, and addressing position bias. Additionally, Hu et al. (2025) used subgoals formulated by LLM before each step to increase success while reducing context. MEM1 Zhou et al. (2026b) used reinforcement learning to maintain a compact shared internal state that jointly supports memory consolidation and reasoning. ACON Kang et al. (2025) learns compression guidelines from failure analysis in smaller models, while Zhang et al. (2026) treats context deletion and insertion as learnable policy actions to optimize both information retention and task performance. More recently, PACE adjusts how much historical memory is retained according to predicted relevance for the next action Wei et al. (2026), while Agent-Omit trains agents to selectively remove redundant thoughts and observations during multi-turn interactions Ning et al. (2026).

These methods address evolving histories, but manage them through relevance, alternative memory representations, learned agent-side deletion, or budget constraints. Different from methods that compress interaction history into a summary or a learned compact memory state, our method preserves the original state–action–reward interaction structure. It first uses a lightweight external predictor of task-dependent removal effects to repeatedly select from the evolving history, without retraining the downstream policy to perform memory selection.

3 METHODOLOGY

Problem Formulation At each decision time t, the policy receives a current query q_t together with a current history rolling window $H_t = \{h_1, \ldots, h_N\}$, where $N = |H_t|$ denotes the number of available history items. We use $K \leq N$ to denote the number of history items retained after context selection.

We treat the policy conditioned on the full history, $\pi_{\text{Full}}(\cdot \mid q_t, H_t)$ as the reference policy. We maintain a fixed-capacity rolling window H_t of the most recent interactions; once full, each new interaction enters the window and the oldest interaction is removed.

The goal is to reduce the total tokens used by the downstream policy while preserving its decision behavior and task performance. For a retained history subset $S \subseteq H_t$, let $\pi_S(\cdot \mid q_t, S)$ denote the policy conditioned on the current query and the retained history S. We measure the effect of context reduction through a task-specific decision-difference signal

$$D_t(S) = \mathcal{D}(q_t, H_t, S), \tag{1}$$

where smaller values indicate that the selected context S better preserves the behavior induced by the full history. The specific form of $\mathcal{D}$ depends on the downstream task: it can measure either distributional changes in the policy output or degradation in the resulting task outcome.

Therefore, the selected context is not necessarily the most recent or the most state-similar history, but a compact set with a small decision gap relative to the reference policy. However, directly evaluating the decision gap for many subsets would require repeated LLM queries at every decision step, and the number of possible subsets grows exponentially with N. We therefore learn a predictor that produces a behavior-aware deletion ordering over the full history. Let

Algorithm 1: Adaptive Context Selection

Require: Current query q_t; rolling history $H_t = \{h_1, \ldots, h_N\}$; predictor ensemble $\{g_m\}_{m=1}^{M}$; minimum context size $K_{\min}$; window size W

if $N < W$ **then**
  return the policy prompt constructed from q_t and all histories in H_t
end if
for $h_i \in H_t$ **do**
  for $m = 1, \ldots, M$ **do**
    $\hat{y}_m^{(i)} \leftarrow g_m(q_t, H_t, H_t \setminus \{h_i\})$
  end for
end for
for $m = 1, \ldots, M$ **do**
  Rank deletion candidates to obtain $r_i^{(m)}$
end for
$B_i \leftarrow \sum_{m=1}^{M} r_i^{(m)}$
Sort B_i to obtain $o_t = (o_1, \ldots, o_N)$
for $K = N-1, \ldots, K_{\min}$ **do**
  $S_K \leftarrow H_t \setminus \{h_{o_1}, \ldots, h_{o_{N-K}}\}$
  for $m = 1, \ldots, M$ **do**
    $\hat{y}_m(S_K) \leftarrow g_m(q_t, H_t, S_K)$
  end for
end for
$K_t^\star \leftarrow \mathcal{C}(\{\hat{y}_m(S_K)\}_{m,K})$ ($\mathcal{C}$ is selection rule)
$S_t^\star \leftarrow S_{K_t^\star}$
$P_t \leftarrow \Gamma(q_t, \widetilde{S}_t^\star, \rho_t)$
Execute policy using P_t
Append the new interaction to H_t; remove the oldest item once the window is full

$$o_t = (o_{t,1}, \ldots, o_{t,N})$$

denote the order in which history items are removed. For each history item h_i, let $\rho_{t,i}$ denote its position in this deletion order,

$$\rho_{t,i} = \text{pos}_{o_t}(i), \quad \rho_t = (\rho_{t,1}, \ldots, \rho_{t,N}),$$

where a larger $\rho_{t,i}$ means that the history item is removed later. The learned deletion order is used in two places. First, it determines which historical items are kept and therefore the adaptive context size. Second, after selecting the context, the same order is reused to reduce repeated contextual information within those histories. For each repeated value, one retained occurrence keeps the complete text, while the other occurrences are replaced by references to that copy. Thus, the final prompt depends jointly on which interactions are retained and how repeated information is represented within them.

3.1 Decision Gap Predictor

Decision gap supervision The predictor is trained to estimate the effect of removing selected context subsets.

For autonomous driving, let $\mathcal{A}$ denote the discrete action space. For a prompt p, let $\ell_a(p)$ denote the log probability of action token a at the first action position. We construct a normalized action distribution over the valid action set as

$$\pi_p(a) = \frac{\exp(\ell_a(p))}{\sum_{b \in \mathcal{A}} \exp(\ell_b(p))}$$

This distribution represents the policy's decision under prompt p and is used to compute the decision gap between the full-context and selected-context prompts. For a current query q_t and selected context S, the predictor target is

$$y_t(S) = \log\left(1 + D_{\text{KL}}\left(\pi_{\text{Full}}(\cdot \mid q_t, H_t) \,\|\, \pi_S(\cdot \mid q_t, S)\right)\right). \tag{2}$$

Thus, the supervision measures the effect of context removal on the downstream action distribution rather than semantic similarity between the current state and a historical experience.

For ScienceWorld, we instead measure the terminal outcome degradation induced by context reduction:

$$y_t^{\mathrm{SW}}(S) = \frac{\bar{R}_{\mathrm{Full}} - \bar{R}_S}{200}, \tag{3}$$

where $\bar{R}_{\mathrm{Full}}$ and $\bar{R}_S$ denote the mean terminal scores obtained using the full and selected contexts, respectively.

Thus, in both domains, the supervision directly measures the effect of context removal on downstream behavior or task performance.

Representation We define the current query as q_t and each history item as h_i. They are encoded by a frozen text embedding model, $e_t = E(q_t), e_i = E(h_i)$. Each history item is additionally associated with a structured task-dependent feature vector r_i containing decision-time information. The semantic and structured features are contextualized to obtain representations u_t for the current query and u_i for the history items. The Transformer inputs are constructed as

$$x_t = W_q e_t + p_0,$$

$$x_i = W_c([\bar{e}_i; \tilde{r}_i]) + E_{\mathrm{status}}(s_i) + p_i$$

where W_q and W_c are learned projections, E_{status} is a learned status embedding, and p_i denotes the learned embedding of the corresponding sequence position. The construction of $\tilde{e}_i$ and $\tilde{r}_i$ is task-specific. In autonomous driving, histories removed in earlier deletion steps are masked, while the current deletion candidate remains visible. In ScienceWorld, the original attempt representations remain available and the status embedding indicates whether each attempt is selected or excluded.

The resulting sequence is processed by a Transformer encoder with one layer, four attention heads, and hidden dimension 64:

$$(u_t, u_1, \ldots, u_N) = \mathrm{Transformer}(x_t, x_1, \ldots, x_N),$$

where u_t and u_i are the contextualized representations of the current query and history item h_i, respectively.

For a selected context S, we construct a candidate representation $\Phi_t(H_t, S)$ from the contextualized history representations. It summarizes the relationship between the full history and the selected context, together with task-dependent decision information when available. The predictor then estimates the task-specific decision difference as

$$\hat{y}_t(S) = g_\theta(u_t, \Phi_t(H_t, S)). \tag{4}$$

Task-specific representation are provided in Appendix C

Training objective We train the predictor with both regression and ranking terms. Firstly, for a candidate context S, the prediction target is

$$y_t(S) = \log(1 + D_t(S)).$$

for autonomous driving and

$$y_t^{(S)} = \frac{\bar{R}_{\mathrm{Full}} - \bar{R}_S}{200},$$

for ScienceWorld.

The regression loss is

$$\mathcal{L}_{\mathrm{reg}} = \mathrm{Huber}(\hat{y}_t(S), y_t(S)).$$

Secondly, for two selected candidates i and j from the same parent context (or between equal-cardinality candidates from the same history pool in ScienceWorld.), we also match their target difference,

$$\mathcal{L}_\Delta = \mathrm{Huber}(\hat{y}_i - \hat{y}_j,\ y_i - y_j),$$

Finally, we add a pairwise ranking loss that prefers the candidate with the smaller true decision gap,

$$\mathcal{L}_{\mathrm{rank}} = |y_i - y_j|\ \mathrm{BCEWithLogits}(\hat{y}_j - \hat{y}_i,\ \mathbf{1}[y_i < y_j]).$$

The final objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{reg}} + 0.5\mathcal{L}_\Delta + 0.25\mathcal{L}_{\mathrm{rank}}. \tag{5}$$

This gives the predictor an absolute decision gap target while also training the ordering that will be used by the selector.

Predictor training. We train a five-member task-specific predictor ensemble in each domain. For driving, supervision is the log-transformed KL divergence between the full- and reduced-context action distributions, with labels generated using query and history representations use frozen 3,072-dimensional `text-embedding-3-large` embeddings. The driving predictor is trained on 1,956 labeled masks, with a disjoint 89-mask split used for checkpoint selection. For ScienceWorld, supervision is the normalized terminal-score degradation between full and reduced contexts, estimated using three repeated rollouts per condition under the frozen rollout protocol. The ScienceWorld predictor uses 1,540 training labels and label-disjoint 150 checkpoint-selection labels, with frozen `all-MiniLM-L6-v2` embeddings. Both predictors are optimized with AdamW using the regression, difference, and ranking losses in Eq. (5); complete training parameters are provided in Appendix D.

3.2 Deletion-Structured Context Construction

Deletion Ranking The deletion candidates are ranked based on predicted decision gap from smallest to largest by each predictor. Let $r_i^{(m)}$ be the rank of candidate i under member m. We combine the five rankings with a Borda score,

$$B_i = \sum_{m=1}^{5} r_i^{(m)}. \tag{6}$$

Candidates with smaller B_i are deleted earlier. If two candidates have the same score, we break the tie using the mean predicted gap, then the history length, and finally the history index. This gives one fixed deletion order $o_t = (o_1, o_2, \ldots, o_N)$

The items near the end of this order will be deleted last.

Adaptive History Selection Based on the deletion order o_t, we construct a sequence for the contexts. For a context size K, we remove the first $N - K$ items in the deletion order:

$$S_K = H_t \setminus \{h_{o_1}, \ldots, h_{o_{N-K}}\}, \qquad K \in \{10, \ldots, 19\}. \tag{7}$$

Though the deletion ranking is fixed for the current decision, the predicted gap will be recomputed at each context size while excluding the items that have already been removed. For example, when evaluating S_{18}, the predictor knows that the first-ranked item has already been removed and evaluates the next deletion under this reduced context

For each candidate context S_K, the five predictor outputs are evaluated independently. We take the average of the five predictors:

$$\bar{y}_t(S_K) = \frac{1}{5} \sum_{m=1}^{5} \hat{y}_t^{(m)}(S_K).$$

A candidate context passes the predicted-gap threshold if

$$\begin{cases} \bar{y}_t(S_K) \le \log(1+\tau), & \text{Driving}, \\ \dfrac{1}{5} \displaystyle\sum_{m=1}^{5} \left[\hat{y}_t^{(m)}(S_K) - \min_{j \in \{10,\ldots,20\}} \hat{y}_t^{(m)}(S_j) \right] \le \tau, & \text{ScienceWorld}, \end{cases} \tag{8}$$

where $\tau = 0.05$.

We set $K_{\min} = 10$ as a fixed search hyperparameter and then evaluate candidate context sizes from $K = 19$ to $K = 10$ and select the smallest context that satisfies the threshold mentioned above.

If none of the reduced contexts satisfies the threshold, it will automatically keep full window ($N = 20$). Then, the same deletion order remains part of the context representation and determines how repeated contextual values are allocated within the retained chunks. Since we do not assume that the predicted gap changes monotonically with K, an intermediate context size that fails the threshold does not stop the search.

Window-Size We choose $W = 20$ as the capacity of the rolling history window. The window is updated throughout an episode: before it is full, all available interactions are retained; once $N = W$,

each new interaction enters the window while the oldest one is removed, and adaptive selection is applied over the current rolling history. Each driving episode contain approximately 37–42 decisions on average across the evaluated settings, while ScienceWorld contains 40 interaction rounds. A substantially larger W would leave relatively few decisions after the history window becomes full. With $W = 20$, roughly half of trajectories remains available for selection-active evaluation.

Order-guided Redundancy Compression After selecting the context chunks, repeated contextual information may still appear across multiple retained interactions. We therefore apply an additional compression step using the same deletion order. This step reduces repeated text within the selected history by storing an identical contextual value once. The repeated text itself is omitted at that occurrence, while the reference allows the original value to be recovered from its retained source. Here, a reference is a textual marker inserted in place of a repeated value that identifies where the complete value is retained. Detailed field definitions and compression rules are provided in Appendix B

4 Results and Analysis

Environment Setup We test our method under two environments. Firstly, we evaluate in a closed-loop SUMO driving environment built on the framework introduced in Khurram et al. (2025), which is based on the LimSim++ closed-loop driving platform Fu et al. (2024a). We define the discrete driving action space $\mathcal{A} = \{1, 2, 3, 4, 8\}$, corresponding to acceleration, deceleration, left lane change, right lane change, and idle, respectively. All methods are tested on a rolling window of the $W = 20$ most recent interactions. Before the window is full, all available histories are kept; once full, the window advances as new interactions are recorded and context selection is applied at each subsequent decision. Following Khurram et al. (2025), predictor development uses nine seen domains defined by three weather conditions and three traffic densities. We evaluate on (i) 23 held-out in-distribution episodes, (ii) 13 episodes from the unseen Severe-3x domain, and (iii) 27 unseen-route conditions covering three held-out routes, three seen traffic domains, and three random seeds.

Secondly, we evaluate our method under ScienceWorld, following the continual ICRL evaluation protocol of Song et al. (2026). It is a text-based interactive environment whose action space includes high-level operational commands for movement between locations, object manipulation, and scientific experimentation. We include it to test whether the proposed context-selection mechanism transfers beyond the closed-loop driving setting, using a different form of downstream supervision. We use two initial attempts followed by 40 interaction rounds, with exploration and exploitation instructions alternating across rounds. We evaluate randomly selected 18 of the 30 ScienceWorld tasks and a maximum of 15 environment steps per episode. Since our framework targets bounded-history context selection, we maintain a candidate window of at most the 20 most recent experiences; our method adaptively selects and compresses this history, while the Full baseline retains all experiences within the H20 window. Recent and Similarity use approximately matched context cardinality as ours. The downstream policy model is GPT-4o for driving and GPT-4.1 mini for ScienceWorld.

Baselines We also compare our method with three context baselines: Full H20, Similarity-K, and Recent-K. Compression-controlled baselines: Recent+Redundancy Compression (Recent + LORC). We additionally compare our method with LongLLMLingua, a general long-context prompt-compression baseline, on ScienceWorld Jiang et al. (2024). Detailed definitions are shown in A

Evaluation Metrics We evaluate baselines performance under several metrics:

1. We evaluate driving performance using **Safety, Comfort, and Efficiency** Khurram et al. (2025); Fu et al. (2024a). All three driving metrics range from 0 to 1, with higher values indicating better performance. Safety measures collision-related risk, Comfort measures trajectory smoothness based on acceleration and jerk, and Efficiency measures driving efficiency with respect to traffic conditions and road-speed constraints. Detailed metric definitions and aggregation follow the original evaluation protocol in Khurram et al. (2025); Fu et al. (2024a)
2. For ScienceWorld, we compare our method with baselines via **Return**. We follow the same evaluation metrics as Song et al. (2026). Score achieved has been averaged across the evaluated tasks for all-round(Round 1-40)/post-selection(Round 19-40). The two bootstrap trials are excluded.
3. We also evaluate our method under tokens. **Post-selection tokens** measure the total number of tokens provided to the downstream policy over decision steps and generated output tokens after the rolling history reaches its full capacity ($N = W = 20$) and context selection becomes active. The metric includes the current query and the selected/compressed history, and is accumulated over the post-H20 portion of each episode before averaging across episodes.

Table 1: Evaluation results across the held-out in-distribution, unseen OOD, and unseen-route settings. Token counts are rounded to the nearest thousand (K), and performance scores to two decimals. Lower token usage is better, while higher Safety, Comfort, and Efficiency scores are better. Bold indicates the best value based on the unrounded results.

Evaluation Type	Method	Mean K	Post-selection Tokens↓	Avg Total↓	Safety↑	Comfort↑	Efficiency↑
Held-out ID	Full-H20	20.00	130K	218K	0.88	0.76	0.79
	Recent-13	13.00	92K	181K	**0.88**	0.76	0.80
	Similarity-13	13.00	91K	180K	0.88	**0.77**	0.80
	Ours	13.63	**73K**	**162K**	0.88	0.76	**0.81**
Severe-3x	Full-H20	20.00	151K	247K	0.82	0.78	0.83
	Recent-13	13.00	103K	201K	0.87	0.77	0.84
	Similarity-13	13.00	103K	200K	**0.87**	**0.78**	**0.84**
	Ours	13.02	**86K**	**184K**	0.86	0.78	0.83
Unseen Routes	Full-H20	20.00	122K	201K	**0.98**	0.74	0.87
	Recent-15	15.00	96K	175K	0.98	**0.75**	**0.88**
	Similarity-15	15.00	98K	177K	0.98	0.74	0.88
	Ours	15.05	**75K**	**154K**	0.98	0.74	0.87

Total Tokens measure the total number of tokens consumed by the downstream policy over the entire episode, including both input and generated output tokens across all decision steps:

$$T_{\text{total}} = T_{\text{in}} + T_{\text{out}}.$$

4.1 AUTONOMOUS-DRIVING

Held-out In-distribution For in-distribution test, our method substantially reduces context cost while maintaining driving performance. Compared with Full-H20, Ours reduces total token usage by 25.7%, with a paired-bootstrap 95% confidence interval of **[**-29.2%**,** -22.2%**]**, using an average of 13.63 histories. For Recent-13 and Similarity-13, which use comparable history budgets, Ours reduces total tokens by 10.3% and 9.7%, respectively. These reductions are also consistent under paired bootstrap, with 95% confidence intervals of **[**-13.8%**,** -6.6%**]** and **[**-13.8%**,** -5.4%**]**.

Unseen Domain OOD Under out-of-distribution testing, our method continues to reduce token usage relative to the baselines. Compared with Full-H20, Ours reduces average total token usage by 25.8%, from 247,490 to 183,521, with a route-clustered 95% confidence interval of $[-31.2\%, -19.4\%]$, while using an average context size of $K = 13.02$. Ours also uses fewer total tokens than Recent-13 and Similarity-13 by 8.6% and 8.5%, respectively. Driving performance remains comparable overall: relative to Full-H20, Ours improves Safety from 0.8211 to 0.8625, maintains the same Comfort score of 0.7759, and yields a slightly lower Efficiency score (0.8256 vs. 0.8268). These results suggest that the complete context-selection mechanism remains effective under the unseen traffic domain.

Unseen Routes OOD We further evaluate the method on three routes that are not used in predictor training, covering three traffic domains (Clear $1\times$, slightly inclement $2\times$, and moderately inclement $3\times$), and three random seeds for a total of 27 matched conditions. These routes follow distinct edge sequences and cover complementary maneuver patterns: straight driving, a required left lane change followed by a left turn, and a required right lane change. Compared with Full-H20, Ours reduces the average post-selection tokens from 121,753 to 75,138, a 38.3% reduction, while total token usage decreases by 23.2%. Ours uses fewer total tokens in all 27/27 matched route-domain-seed conditions, with a paired-bootstrap 95% confidence interval of $[-24.98\%, -21.40\%]$. Although Ours retains slightly more history on average ($\bar{K} = 15.05$) than Recent-15 and Similarity-15, it uses 21.6% and 23.0% fewer post-selection tokens, and 11.9% and 12.7% fewer total tokens, respectively, with paired-bootstrap 95% confidence intervals of $[-13.59\%, -10.12\%]$ and $[-14.71\%, -10.67\%]$. Driving performance also remains close to Full-H20, with only small changes in Safety, Comfort, and Efficiency. Paired-bootstrap confidence intervals for performance are reported in Appendix

4.2 SCIENCEWORLD EVALUATION

As shown in Table 2, Ours achieves the lowest token usage while matching the observed running-max return of the recency-based baseline. Ours and Recent both achieve a return of 96.06, while Ours reduces average total token usage by 30.2%. Relative to

Table 2: ScienceWorld evaluation. Late-round statistics are averaged over rounds 19–40, when context selection is active for our method, whereas all-round statistics are averaged over rounds 1–40. Because histories grow over the episodes, late-round token usage can exceed the all-round average. "Total" includes both input and output tokens. "Return" are average over all-round.

Method	Mean K	Late-Round Total/Trial↓	Avg Total /Trial↓	Return↑
Full-H20	20.00	352,809	282,879	95.50
Recent-12	12.00	202,710	194,188	**96.06**
Similarity-12	12.00	227,433	217,706	94.78
Ours	12.70	**87,100**	**135,470**	**96.06**

Full, Ours reduces total tokens by 52.1% and achieves a slightly higher observed return (96.06 vs. 95.50). Compared with Similarity, it reduces total tokens by 37.8% while obtaining a higher observed return (96.06 vs. 94.78). Task-level paired bootstrap analysis yields mean total-token differences per formal trial of -147.4k relative to Full (95% CI [-215.5k, -95.1k]), -58.7k relative to Recent ([-80.8k, -37.2k]), and -82.2k relative to Similarity ([-127.8k, -44.7k]). These results indicate substantial context-cost reductions while maintaining competitive observed task performance. We additionally report mean return, post-selection return (round 19-40), trial-level return trajectories and running-max performance in Appendix E. During the selection-active phase, Ours maintains competitive task performance, achieving a post-selection running-max/mean return of 95.94/74.22 compared with 95.50/74.34 for Full reference policy, 95.78/76.02 for Recent, 94.78/69.37 for Similarity.

4.3 Controlled Selection and Budgeting Baselines

We additionally apply the same learned order- guided redundancy compression (LORC) to the Recent-K baseline. It retains the K most recent interactions while using our learned deletion order during compression. We report post-selection total tokens to isolate the selection-active phase. A redundancy-controlled Recent-K comparison shows larger gains under distribution shift: Ours reduces post-selection token usage by 12.7% on Severe-3x and 7.8% on unseen routes. A similar controlled advantage is observed in ScienceWorld, where Ours reduces post-selection by 7.6%.

We further construct matched-K fixed-budget baseline using the same method as ours, where the fixed context size is chosen to match the mean retained history size observed for Ours in each setting instead of selecting independently. At comparable average context cardinalities, the variants yield similar operating points on held-out ID and unseen routes, while Ours uses 7.38% fewer post-selection tokens on Severe-3x. Detailed results refer to Appendix F

4.4 Comparison with General Prompt Compression

We additionally compare with LongLLMLingua under an approximately matched historical-context budget. LongLLMLingua achieves all-round/post-selection return of 94.78/94.67, compared with 96.06/95.94 for Ours. However, its official compression core requires 27.7 s per mature decision on average in our ScienceWorld setting, whereas the complete context-management pipeline of Ours requires 1.24 s on a two-thread CPU. Detailed calibration, task-level results, and latency protocols are provided in Appendix G.

Table 3: ScienceWorld comparison with LongLLMLingua.

Metric	Ours	LongLLMLingua
Avg Total / Trial ↓	135,470	**131,467**
Return ↑	**96.060**	94.780
History Proc. Latency (s) ↓	**1.244**	27.715

4.5 Feasibility Threshold Sensitivity Analysis

We additionally perform an offline sensitivity analysis over post selection driving decisions by recomputing context selection and input-token usage on the evaluation trajectories. Across all three settings, increasing τ consistently reduces context budget and post-selection input tokens, while the threshold $\tau = 0.05$ remains away from either budget extreme, supporting that the feasibility threshold provides controllable adaptive budgeting. Detailed results refer to Appendix H

5 Conclusion

In this work, we study online context selection for in-context reinforcement learning and propose a method for reducing accumulated interaction context during deployment. The method learns the effect of context reduction, constructs a deletion ordering, adaptively selects the retained history size, and reuses the same ordering to reduce redundant information within the retained context.

In closed-loop SUMO driving, our method reduces total token usage by 23.2–25.8% relative to the full-context policy across held-out in-distribution, unseen-domain, and unseen-route evaluations while maintaining comparable driving performance. In ScienceWorld, it reduces total token usage by 52.1% relative to Full and by 30.2% and 37.8% relative to Recent and Similarity, respectively. Overall, across both domains, the learned selector reduces context cost while maintaining comparable downstream performance.

However, our marginal deletion ranking does not explicitly model higher-order dependencies among multiple interaction histories. Consequently, the nested subsets defined by the deletion order may not contain the globally best subset at every context size. Extending the selector to capture such interactions is a direction for future work.

AI USE STATEMENT

Generative AI tools were used to support literature discovery, assist with troubleshooting and code editing, and aid in drafting and editing portions of the manuscript. All AI-assisted suggestions and outputs were reviewed and verified by the authors, who take full responsibility for the final content of the paper.

REFERENCES

Shengnan An, Bo Zhou, Zeqi Lin, Qiang Fu, Bei Chen, Nanning Zheng, Weizhu Chen, and Jian-Guang Lou. Skill-based few-shot selection for in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13472–13492, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Weiqin Chen, Xinjie Zhang, Dharmashankar Subramanian, and Santiago Paternain. Filtering learning histories enhances in-context reinforcement learning. *arXiv preprint arXiv:2505.15143*, 2025.

Jiale Deng, Yanyan Shen, Ziyuan Pei, Youmin Chen, and Linpeng Huang. Influence guided context selection for effective retrieval-augmented generation. *Advances in Neural Information Processing Systems*, 38:29225–29252, 2026.

Daocheng Fu, Wenjie Lei, Licheng Wen, Pinlong Cai, Song Mao, Min Dou, Botian Shi, and Yu Qiao. Limsim++: A closed-loop platform for deploying multimodal llms in autonomous driving. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1084–1090. IEEE, 2024a.

Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. Drive like a human: Rethinking autonomous driving with large language models. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pp. 910–919. IEEE, 2024b.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. HiAgent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32779–32798, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1575. URL https://aclanthology.org/2025.acl-long.1575/.

Sili Huang, Jifeng Hu, Hechang Chen, Lichao Sun, and Bo Yang. In-context decision transformer: Reinforcement learning via hierarchical chain-of-thought. *arXiv preprint arXiv:2405.20692*, 2024.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.825. URL https://aclanthology.org/2023.emnlp-main.825/.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.91. URL https://aclanthology.org/2024.acl-long.91/.

Minki Kang, Wei-Ning Chen, Dongge Han, Huseyin A Inan, Lukas Wutschitz, Yanzhi Chen, Robert Sim, and Saravan Rajmohan. Acon: Optimizing context compression for long-horizon llm agents. *arXiv preprint arXiv:2510.00615*, 2025.

Aleesha Khurram, Amir Moeini, Shangtong Zhang, and Rohan Chandra. Prompt-driven domain adaptation for end-to-end autonomous driving via in-context rl, 2025. URL https://arxiv.org/abs/2511.12755.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, et al. In-context reinforcement learning with algorithm distillation. *arXiv preprint arXiv:2210.14215*, 2022.

Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 36:43057–43083, 2023.

Itay Levy, Ben Bogin, and Jonathan Berant. Diverse demonstrations improve in-context compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1401–1422, 2023.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd workshop on knowledge extraction and integration for deep learning architectures*, pp. 100–114, 2022.

Giovanni Monea, Antoine Bosselut, Kianté Brantley, and Yoav Artzi. Llms are in-context bandit reinforcement learners, 2025. URL https://arxiv.org/abs/2410.05362.

Yansong Ning, Jun Fang, Naiqiang Tan, and Hao Liu. Agent-omit: Adaptive context omission for efficient llm agents. *arXiv preprint arXiv:2602.04284*, 2026.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. Revisiting demonstration selection strategies in in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9090–9101, 2024.

Kiran Purohit, V Venktesh, Sourangshu Bhattacharya, and Avishek Anand. Sample efficient demonstration selection for in-context learning. *arXiv preprint arXiv:2506.08607*, 2025.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 2655–2671, 2022.

Alexander Scarlatos and Andrew Lan. Reticl: Sequential retrieval of in-context examples with reinforcement learning, 2024. URL https://arxiv.org/abs/2305.14502.

Thomas Schmied, Fabian Paischer, Vihang Patil, Markus Hofmarcher, Razvan Pascanu, and Sepp Hochreiter. Retrieval-augmented decision transformer: External memory for in-context rl, 2025. URL https://arxiv.org/abs/2410.07071.

Kefan Song, Amir Moeini, Peng Wang, Lei Gong, Rohan Chandra, Shangtong Zhang, and Yanjun Qi. Reward is enough: Llms are in-context reinforcement learners. In *International Conference on Learning Representations*, volume 2026, pp. 112747–112770, 2026.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298, 2022.

Xubin Wang, Jianfei Wu, Yichen Yuan, Deyu Cai, Mingzhe Li, and Weijia Jia. Demonstration selection for in-context learning via reinforcement learning. *arXiv preprint arXiv:2412.03966*, 2024.

Lei Wei, Xiao Peng, Tt, Guannan Zhang, Chenhao Jiang, Hongyu Li, Lanbo Lin, Yuanwu Xu, Jiayao Liu, Kesu Wang, and Bin Wang. PACE: Predictive adaptive context extraction for long-horizon LLM agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27184–27199, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1252. URL https://aclanthology.org/2026.acl-long.1252/.

Zijiang Yan, Hao Zhou, Hina Tabassum, and Xue Liu. Hybrid llm-ddqn-based joint optimization of v2i communication and autonomous driving. *IEEE Wireless Communications Letters*, 14(4): 1214–1218, 2025.

Jianhao Yuan, Shuyang Sun, Daniel Omeiza, Bo Zhao, Paul Newman, Lars Kunze, and Matthew Gadd. Rag-driver: Generalisable driving explanations with retrieval-augmented in-context learning in multi-modal large language model. *arXiv preprint arXiv:2402.10828*, 2024.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, et al. Proagent: building proactive cooperative agents with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17591–17599, 2024.

Yuxiang Zhang, Jiangming Shu, Ye Ma, Xueyuan Lin, Shangxi Wu, and Jitao Sang. Memory as action: Autonomous context curation for long-horizon agentic tasks. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 19149–19164, 2026.

Zheng Zhang, Shaocheng Lan, Lei Song, Jiang Bian, Yexin Li, and Kan Ren. Learning to select in-context demonstration preferred by large language model. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11345–11360, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.592. URL https://aclanthology.org/2025.findings-acl.592/.

Zhiyuan Zhou, Heye Huang, Boqi Li, Shiyue Zhao, Yao Mu, and Jianqiang Wang. Safedrive: Knowledge-and data-driven risk-sensitive decision-making for autonomous vehicles with large language models. *Accident Analysis & Prevention*, 224:108299, 2026a.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Bryan Kian Hsiang Low, and Paul Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. In *International Conference on Learning Representations*, volume 2026, pp. 58413–58438, 2026b.

Siyuan Zhu, Chengdong Xu, Kaiqiang Ke, and Chao Yu. Context-picker: Dynamic context selection using multi-stage reinforcement learning, 2026. URL https://arxiv.org/abs/2512.14465.

A Baselines and Evaluation Metrics

We compare our method with three baselines:

1. Full H_{20}: Conditions on the entire current rolling window of up to 20 recent interactions at every decision, without selection or compression. It does not perform history selection or context compression and serves as the full-context reference.
2. Similarity-K: keeps the K history items that are most similar to the current state. Similarity is computed using the state representations of the current and historical observations, such as ego speed, lane information, nearby vehicles, conflict-related information, junction state, and traffic-light state for autonomous driving and actions taken, resulting observations, available objects, and rewards recorded in each historical attempt for ScienceWorld. Let z_t and z_i denote the feature vectors of the current state and historical state h_i, respectively. Let $e_t = E(q_t), e_i = E(h_i)$ be text embedding of query and history. Each feature is standardized as
$$\tilde{z}_{i,j} = \frac{z_{i,j} - \mu_j}{\sigma_j},$$
where μ_j and σ_j are the mean and standard deviation of feature j from the dataset.
We compute the similarity using Euclidean distance for autonomous driving:
$$d_i = \|\tilde{z}_i - \tilde{z}_t\|_2 .$$
and cosine distance for ScienceWorld:
$$d_i = 1 - \cos(e_i, e_t).$$
The K histories with the smallest distances are selected:
$$S_t^{\text{sim}} = \arg \min_{S \subseteq H_t,\, |S|=K} \sum_{h_i \in S} d_i.$$
If ties, we will select the more recent chunk. This baseline tests whether similarity-based selection is sufficient for closed loop evaluation.

3. Recent-K: keeps the K most recent history items and removes the older histories. This baseline tests whether using a simple recency-based context is sufficient.

Following prior work on recency-based memory retrieval Zhang et al. (2024) and state-similarity-based driving-experience retrieval, we implement Recent-K and Similarity-K as task-adapted baselines Yan et al. (2025). For Recent-K and Similarity-K, we choose K to approximately match the average number of history items kept by Ours based on results in each evaluation setting. This allows us to compare different context-selection strategies under a similar history budget.

We additionally compare our method with compression-controlled baselines: Recent-K + Redundancy Compression (Recent-K + LORC). These baselines retains the K most recent history items and applies the same learned order-guided redundancy compression used by Ours.

Furthermore, we compare our method to a related context management benchmark, LongLLMLingua, on ScienceWorld Jiang et al. (2024). We calibrate its compression target to the average post-H20 historical-context budget of Ours and fix the resulting compression setting before the formal evaluation. Because LongLLMLingua and the downstream policy use different tokenizers, we report average total context usage using the downstream tokenizer across all trials.

B Order-Guided Redundancy Compression

We divide the information in each history item into two groups. Decision-specific dynamic information, such as the previous decision, action, and reward, is always retained. For ScienceWorld, these fields include the observation text and available-object list within each retained attempt summary. For autonomous driving, we apply the operation only to predefined contextual fields that can repeat across retained interaction histories.

Let v_i^f denote the value of compressible field f in selected history item h_i, and let v_t^f denote the corresponding value in the current query. We process each contextual value according to where the same information is already represented. If $v_i^f = v_t^f$, the historical occurrence can be deleted because the complete value is already available there. If a value v is not present in the current query and occurs in only one retained history, it is left unchanged. If the same value v appears in multiple selected histories, we keep one complete occurrence.

The retained history that keeps the complete occurrence is selected using the deletion order:

$$c_{t,f,v} = \arg \max_{i:\, h_i \in S_{K_t^*},\, v_i^f = v} \rho_{t,i}, \tag{9}$$

where $\rho_{t,i}$ is the position of h_i in the deletion order. Because a larger $\rho_{t,i}$ means that h_i is removed later, $h_{c_{t,f,v}}$ is the retained history with the largest deletion-order position among those containing value v. We therefore keep the complete value in $h_{c_{t,f,v}}$ and replace its other occurrences with references to this retained history based on environment-specific compression rule. In this way, the same deletion order used to select histories also determines which retained history stores each repeated contextual value.

The final policy prompt is therefore

$$P_t = \Gamma\left(q_t, S_{K_t^*}, \rho_t\right).$$

where K_t^* is the adaptively selected history size and Γ constructs the final prompt using the deletion order.

Two deletion orderings can produce the same retained subset but assign repeated values to different chunks, resulting in different final prompts.

C Task-Specific Representation

In autonomous driving, the candidate representation incorporates the current deletion candidate and driving decision information; For a candidate deletion, let P denote the current parent context and S denote the resulting selected context. We compute their average representations as

$$\bar{u}_P = \frac{1}{|P|} \sum_{i \in P} u_i, \qquad \bar{u}_S = \frac{1}{|S|} \sum_{i \in S} u_i.$$

In ScienceWorld, it summarizes the selected, excluded, and full attempt histories. The predictor serves the same role in both settings: estimating the effect of retaining S instead of the full history.

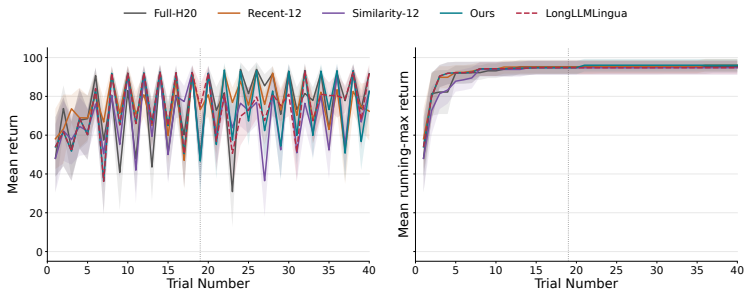

Figure 2: ScienceWorld performance over 40 interaction trials. Left: mean trial return. Right: mean running-max return. Shaded regions denote SEM across tasks.

Table 5: Comparison of Recent, Recent+LORC, and Ours across evaluation scenarios. For driving, total tokens denote post-selection tokens and performance is Safety/Comfort/Efficiency (S/C/E). For ScienceWorld, total tokens denote post-selection tokens per trial and performance is return. Lower token usage is better; higher performance is better.

Scenario	Method	Mean K	Post-selection Tokens↓	Performance↑
Held-out ID	Recent-13	13.00	91,730	**0.8838**/0.7617/0.7981
	Recent-13 + LORC	13.00	74,458	0.8808/0.7602/0.7825
	Ours	13.63	**73,233**	0.8836/**0.7645**/**0.8071**
Severe-3x	Recent-13	13.00	102,971	**0.8673**/0.7715/**0.8351**
	Recent-13 + LORC	13.00	98,165	0.8561/**0.7802**/0.7996
	Ours	13.02	**85,747**	0.8625/0.7759/0.8256
Unseen Routes	Recent-15	15.00	95,859	**0.9828**/**0.7457**/**0.8803**
	Recent-15 + LORC	15.00	81,468	0.9771/0.7401/0.8598
	Ours	15.05	**75,138**	0.9827/0.7412/0.8706
ScienceWorld	Recent-12	12.00	202,710	**96.06**
	Recent-12 + LORC	12.00	94,254	94.78
	Ours	12.70	**87,100**	**96.06**

D Predictor Training

Our predictor uses a five-member predictor ensemble. Every member contains a one-layer Transformer encoder with four attention heads, hidden dimension 64, and dropout 0.1. Predictors are optimized with AdamW at a learning rate of 3×10^{-4} and gradient clipping at 1.0. The training objective combines absolute Huber regression, same-parent (or same-cardinality) difference regression, and pairwise ranking with weights 1, 0.5, and 0.25, respectively. Driving uses weight decay 10^{-4} and ScienceWorld uses weight decay 10^{-2}. Checkpoint selection is performed only on the reserved development split and is separate from predictor weight fitting.

E ScienceWorld Performance across Rounds

Fig.2 reports the mean return at each formal trial and the corresponding running maximum across the 40 interaction rounds. While trial-level returns exhibit variation across tasks and rounds, the running-max curves converge to similar high-return levels for all methods. This analysis complements the running-max return reported in Table 2 by showing the evolution of observed performance throughout the interaction process. Additionally, we report the running-max/mean return over post-selection period (Round 19-40) in Table 4. Ours achieves 95.94/74.22, compared with 95.78/76.02 for Recent, 94.78/69.37 for Similarity, 94.78/70.27 for Recent+LORC, and 94.67/73.74 for LongLLMLingua.

Table 4: ScienceWorld Post-selection return comparison. Higher is better.

Method	Mean Return↑	Running-Max Return↑
Full-H20	74.34	95.50
Recent-12	**76.02**	95.78
Similarity-12	69.37	94.78
Recent-12 + LORC	70.27	94.78
LongLLMLingua (matched-budget)	73.74	94.67
Ours	74.22	**95.94**

Table 6: Sensitivity to the feasibility threshold τ. Post-selection input tokens are estimated from the 21st decision onward on fixed trajectories and averaged per episode. Changes are reported relative to $\tau = 0.05$.

Setting	τ	Mean K	Estimated Post-selection Input Tokens↓	Change vs. $\tau = 0.05$
Held-out ID	0.02	17.58	81,734	+17.79%
	0.03	16.50	78,402	+12.99%
	0.05	**13.63**	**69,387**	**0.00%**
	0.07	11.64	63,183	−8.94%
	0.10	10.64	59,685	−13.98%
	0.15	10.14	57,931	−16.51%
Severe-3x	0.02	17.63	99,111	+22.28%
	0.03	16.62	94,618	+16.74%
	0.05	**13.02**	**81,050**	**0.00%**
	0.07	11.26	74,569	−8.00%
	0.10	10.47	71,851	−11.35%
	0.15	10.15	70,677	−12.80%
Unseen Routes	0.02	19.62	83,729	+18.26%
	0.03	18.85	81,569	+15.21%
	0.05	**15.05**	**70,803**	**0.00%**
	0.07	12.03	62,174	−12.19%
	0.10	10.39	57,455	−18.85%
	0.15	10.00	56,429	−20.30%

F Controlled Baseline Analysis

Redundancy Compression Controlled Recency Baseline To isolate the effect of learned subset selection, we additionally apply the same learned order-guided redundancy compression (LORC) to the Recent-K baseline. We report post-selection total tokens to isolate the selection-active phase as shown in table 5.

The benefit is most pronounced under distribution shift. Relative to Recent-K+LORC, Ours reduces post-selection total tokens by 12.7% on Severe-3x and 7.8% on unseen routes, respectively. On held-out ID, the controlled gap is smaller. The larger gap under the two OOD settings suggests that downstream-aware history selection is particularly useful when simple recency is less aligned with the context needed for the current decision.

A similar controlled advantage appears in ScienceWorld. Ours reduces post-selection total tokens by 7.6% and average total tokens by 7.4% relative to Recent-12+LORC. Notably, applying LORC to Recent-12 lowers its observed running-max return from 96.06 to 94.78, whereas Ours retains a return of 96.06 while using fewer tokens. This indicates that redundancy compression is not performance-neutral when applied to a fixed recency-selected subset, and that its interaction with history selection matters. Together, these controlled comparisons show that the gains over recency-based selection are not explained by redundancy compression alone.

Matched fixed-budget Baseline To isolate the effect of online budget determination, we additionally compare our method with fixed K baseline which follows the exactly same method as ours. For each setting, the fixed K is chosen to match the mean retained history size observed for Ours. At matched context budget, the fixed and online-budget variants achieve similar operating points on held-out ID and unseen routes. On Severe-3x, Ours reduces post-selection total tokens by 7.4% relative to Fixed-K13, with comparable driving performance. These results are intended as a controlled comparison of budget determination.

Budget Determination The fixed-budget variants require a separately specified K for each setting. We calibrate these values to the mean context sizes produced by our adaptive method. Our method achieves comparable performance using a selection threshold, without separately specifying a fixed K for each setting. The comparison therefore highlights the ability to determine context budgets automatically, alongside the setting-dependent additional savings from decision-dependent budgeting.

Decision-Level Budget Variation Although the adaptive and matched fixed-K variants have similar mean context budget, their decision-level budgets differ substantially. Across ID, Severe-3x, and unseen routes, 89.4%, 89.7%, and 88.1% of selection-active decisions, respectively, differ from the corresponding matched fixed K by at least two history items. The variation also occurs within individual trajectories: on unseen routes, all 27 episodes show variation in K, and the retained budget changes between 64.9% of selection-active decisions.

G LongLLMLingua Baseline and Matched-Budget

Implementation and budget matching. We use the official LongLLMLingua v0.2.2 implementation without modifying its compression core Jiang et al. (2024). The compressor is calibrated to the average post-H20 historical-context budget of Ours, measured using the downstream `o200k_base` tokenizer. Because LongLLMLingua uses its own tokenizer internally, this calibration yields a fixed

compressor-side target of 9,042 tokens. The target is fixed before formal evaluation, and all reported token usage is recomputed using the downstream tokenizer. LongLLMLingua is evaluated under the same ScienceWorld protocol as the other methods.

Latency measurement. For Ours, context-management latency is measured from history embedding generation through adaptive selection, redundancy compression, final prompt, and validation. For LongLLMLingua, we time the official compression core after model warm-up, with GPU synchronization immediately before and after each compression call. Both measurements exclude model loading, network communication, downstream policy inference, and environment execution. LongLLMLingua processes raw histories of approximately 31.8k compressor-tokenizer tokens on average. The reported timings characterize the observed deployment overhead and are not hardware-normalized speed comparisons.

H Feasibility Threshold Sensitivity Analysis

We analyze the sensitivity of the adaptive context budget to the feasibility threshold τ. Using the recorded predictor outputs from all 1,114 post-H20 driving decisions, we recompute context selection offline for multiple threshold values while keeping the original evaluation trajectories fixed. For each τ, we report the resulting mean selected context size K and the post-selection input tokens accumulated from the 21st decision onward and averaged across episodes as shown in Table 6

I Paired-bootstrap confidence intervals for performance

We additionally report paired-bootstrap 95% confidence intervals for the performance differences between Ours and Full-H20, with Δ = Ours − Full-H20. Bootstrap resampling is performed over matched route–domain–seed episodes. On held-out ID, the differences are $+0.0060$ for Safety (95% CI $[-0.0090, +0.0208]$), $+0.0065$ for Comfort ($[-0.0027, +0.0160]$), and $+0.0132$ for Efficiency ($[+0.0020, +0.0252]$). On Severe-3x, the corresponding differences are $+0.0414$ ($[-0.0013, +0.0963]$), $+0.0000$ ($[-0.0168, +0.0156]$), and -0.0012 ($[-0.0226, +0.0192]$). On unseen routes, they are -0.0007 ($[-0.0039, +0.0024]$), -0.0005 ($[-0.0132, +0.0124]$), and $+0.0018$ ($[-0.0111, +0.0137]$) for Safety, Comfort, and Efficiency, respectively. For ScienceWorld, we perform paired bootstrap resampling over the tasks. The mean-return 95% confidence interval is $[-5.0, +4.3]$. All driving confidence intervals include zero except for the positive Efficiency difference on held-out ID, while the ScienceWorld mean-return interval also includes zero. Overall, we find no statistically detectable degradation in any driving or ScienceWorld metric relative to Full-H20.