DRSR: LEARNING SET-LEVEL DELETION RISK FOR EFFICIENT LONG-HORIZON AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon language-model agents accumulate reasoning traces, tool exchanges, and observations whose relevance changes with the current decision. Existing compression strategies often score historical units independently, but the risk of deleting several units is generally not determined by their singleton scores: redundant evidence, accumulated small effects, and the information that remains after deletion all matter. We introduce **Direct Relational Set-Risk Pruning (DRSR)**, which formulates agent-history compression as risk-constrained selection over deletion sets. Offline, DRSR constructs fixed-output counterfactual supervision by jointly deleting protocol-valid history Blocks and measuring the change in teacher-forced likelihood of the same recorded next output. A lightweight scorer then predicts set-level harm from online-visible relations between candidate history and the current pre-action state, together with deleted and retained set structure and pairwise interactions. At deployment, DRSR evaluates structurally valid deletion candidates with the lightweight scorer and removes the largest feasible set under recency, protocol, budget, and learned-risk constraints, abstaining when no set meets the risk gate. Across four domains of long horizon agent tasks, DRSR increases mean reward from 0.699 to 0.802 while reducing total model tokens by 20.82%. On the fixed Eval40 comparison, it obtains 0.794 reward at 1.211M tokens per task, using 35.85% fewer tokens than the uncompressed agent. Diagnostic analyses and ablations further show that decision-conditioned relations, retained-context information, pair interactions, and abstention each contribute to reliable pruning.

1 INTRODUCTION

Agents solve long horizon tasks by interleaving reasoning with tool use and incorporating observations into later decisions (Yao et al., 2022). During execution, their context accumulates task constraints, explored alternatives, intermediate conclusions, repeated observations, and completed tool interactions. Keeping the full history avoids information loss, but it also requires the model to repeatedly process an increasingly long context, much of which may no longer matter to the current decision at later stages of execution. As context grows, separating useful evidence from obsolete or redundant content becomes more difficult (Liu et al., 2024; Shi et al., 2023). Existing prompt compression and memory management methods reduce active context (Jiang et al., 2023; Pan et al., 2024; Packer et al., 2023), but risk-aware history pruning requires identifying which past information is still needed at the current agent state.

Historical value is inherently state dependent. The importance of a past record is not a fixed property of the record itself, but changes as the agent progresses through the task, motivating recent work on adaptive context management for long-horizon agents (Li et al., 2026a; Zhang et al., 2026b; Kang et al., 2025). An error trace may be essential while a failure is being diagnosed, yet become redundant once the cause has been identified and the relevant conclusion has been recorded elsewhere. In contrast, an early task constraint may remain important until the task is completed. Historical information therefore cannot be judged reliably from age, position, or a fixed context window alone. Its value depends on the current decision state and on what information would remain after pruning.

History pruning is also a set level problem. Two historical Blocks may each appear removable in isolation because either one preserves the same fact, while removing both may eliminate the only remaining evidence. More generally, removal-based attribution and interaction methods show that

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.

the effect of a set need not be recoverable from independently measured singleton effects (Covert et al., 2021; Grabisch & Roubens, 1999; Sundararajan et al., 2020; Tsai et al., 2023). Even without such interaction, several small deletion effects may accumulate into substantial harm. Independent Block scores therefore cannot reliably recover the risk of the set that is actually removed.

The main difficulty is that fixed-output deletion harm is unavailable when the online decision must be made. The effect of removing history is defined with respect to the agent's actual next output, which has not yet occurred. This distinction is related to learning with privileged information and distillation, where training may exploit information unavailable at deployment (Vapnik & Vashist, 2009; Hinton et al., 2015; Lopez-Paz et al., 2015). Repeatedly querying the language model for every candidate deletion is also unsuitable, because different queries may produce different continuations and would make pruning itself expensive. Related learned eviction methods likewise transfer future or privileged signals into lightweight online selectors (Dong et al., 2026). Completed trajectories make this quantity available for supervision. Once the actual next output is known, we can keep it fixed, jointly remove a candidate set, and measure how its likelihood changes under the edited context. This converts an online unavailable quantity into an offline training signal. A lightweight predictor can then learn deletion risk from information available before the next action.

Based on this idea, we introduce **Direct Relational Set Risk Pruning (DRSR)**. DRSR represents historical value through relations between candidate history and the current decision state, while modeling both the information proposed for removal and the information that remains. Instead of assigning each Block a fixed importance score, it directly predicts the risk of a candidate deletion set. During deployment, candidate sets are evaluated in a batch by a lightweight scorer, using shared pre-action features extracted from the intact context before candidate-set evaluation. Pruning is allowed only when the candidate satisfies protocol constraints, recent history protection, deletion limits, and a learned risk gate. If no candidate meets the risk gate, the original request is retained, following the broader idea of selective prediction with abstention (Chow, 1970; Geifman & El-Yaniv, 2019). Figure 1 summarizes the offline supervision, relational set-risk prediction, and online risk-gated selection pipeline. Our main contributions are:

- **State conditioned set level supervision.** We construct deletion labels from completed trajectories by jointly removing candidate Blocks and rescoring the same recorded next output. Each label therefore measures the proposed deletion set directly rather than combining independently evaluated singleton scores.
- **Relational set risk prediction.** We introduce a lightweight scorer that jointly models the current decision state, candidate history, deleted information, retained information, and interactions within the deletion set. The representation backbone and task model remain frozen.
- **Risk constrained online pruning with abstention.** We use predicted risk as a feasibility condition and select the feasible deletion set with the largest direct token saving. If no candidate passes the risk gate, the policy abstains from pruning.

2 Related Work

Agent memory and context management. Long-horizon agents increasingly manage their own active context. MemGPT organizes information across memory tiers (Packer et al., 2023), while Sculptor, Memory-as-Action, Git-Context-Controller, context folding, and AgentFold expose or learn explicit history editing operations (Li et al., 2026a; Zhang et al., 2026b; Wu et al., 2025; Sun et al., 2025; Ye et al., 2025). ACON optimizes compression policies from trajectory outcomes (Kang et al., 2025), whereas LLMLingua and LLMLingua-2 compress prompts under a token budget (Jiang et al., 2023; Pan et al., 2024). These approaches are commonly evaluated end to end on interactive benchmarks (Trivedi et al., 2024; Jimenez et al., 2024). DRSR instead supervises each protocol-valid deletion set by its measured effect on a fixed recorded next output, without replacement summaries or retrieval stores.

Structured pruning and KV-cache eviction. Reasoning traces can be shortened by generating less per step, skipping low-importance tokens, or removing low-information steps (Xu et al., 2025; Xia et al., 2025; Li et al., 2026d). Related inference-time methods evict KV entries using attention, recency, adaptive budgets, or learned future contribution (Zhang et al., 2023; Xiao et al., 2024; Li et al.,

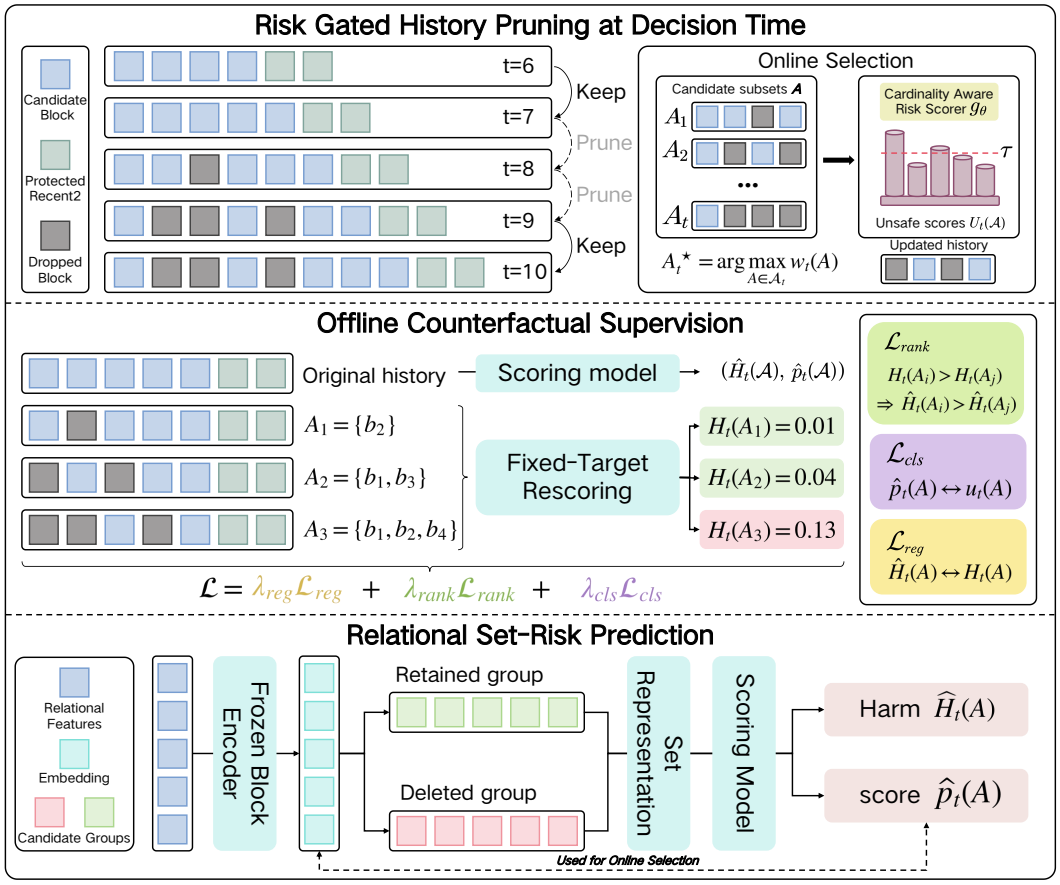


Figure 1. **Overview of DRSR.** Offline, DRSR constructs exact set-level counterfactual supervision by jointly deleting protocol-valid history Blocks and rescoring the same recorded next output. A relational set-risk predictor models both the proposed removal set and the information that remains, and outputs predicted harm and an unsafe score. Online, the learned score gates candidate deletion sets, after which DRSR selects the feasible set with the largest direct token saving and abstains when no set meets the risk gate.

2024; Feng et al., 2026; Dong et al., 2026). DRSR differs in operating on serialized agent-history Blocks and directly predicting the risk of a jointly deleted set rather than assigning independent importance scores to individual units.

Removal-based measures and interactions. Removal-based explanation studies how behavior changes when subsets of features are removed (Covert et al., 2021). The interaction residual used by DRSR is closely related to discrete Shapley interaction terms (Grabisch & Roubens, 1999; Lundberg et al., 2018; Sundararajan et al., 2020; Tsai et al., 2023; Fumagalli et al., 2023). However, DRSR does not seek symmetric attribution over all historical Blocks; it measures and predicts the harm of the specific deletion sets permitted by the agent protocol.

Privileged supervision and selective prediction. Using training-only information to learn a deployable predictor is related to learning with privileged information and knowledge distillation (Vapnik & Vashist, 2009; Hinton et al., 2015; Lopez-Paz et al., 2015). DRSR also abstains when no candidate passes its learned risk gate, connecting it to reject-option and risk-controlled prediction (Chow, 1970; El-Yaniv et al., 2010; Geifman & El-Yaniv, 2019; Bates et al., 2021; Angelopoulos et al., 2024; Mohri & Hashimoto, 2024). Semantic-uncertainty methods provide a complementary line of work (Kuhn et al., 2023; Farquhar et al., 2024; Nikitin et al., 2024); DRSR does not estimate semantic uncertainty or provide a conformal guarantee.

3 PROBLEM FORMULATION

3.1 STRUCTURED HISTORY AND DELETION ACTIONS

At decision time t, let P_t denote the protected system instructions and task specification, and let $S_1, \ldots, S_{t-1}$ denote the completed interaction Steps. We write

$$C_t = P_t \oplus S_1 \oplus \cdots \oplus S_{t-1}, \qquad S_i = B_{i,0} \oplus \cdots \oplus B_{i,m_i}, \tag{1}$$

where $\oplus$ denotes serialization in the original order. A Block is either the complete assistant content associated with a Step (`assistant_state`) or an atomic tool exchange that contains a tool call together with its matched response (`tool_exchange`). Pruning never separates a tool call from its response, and every rewritten request must preserve a valid conversation and tool protocol.

Let $E_t = \{B_1, \ldots, B_n\}$ denote the eligible candidate pool and let $A \subseteq E_t$ denote a proposed deletion set. The operator $D_A(C_t)$ jointly removes all Blocks in A and serializes the remaining context in its original order. We define the direct token saving as

$$w_t(A) = \mathrm{Tok}(C_t) - \mathrm{Tok}(D_A(C_t)). \tag{2}$$

The saving is measured on the actual rewritten request. For each deletion set, token counts are compared before and after joint reserialization, so the saving reflects the complete edit to the request.

3.2 COUNTERFACTUAL DELETION HARM

For an external trajectory, let y_t denote the recorded next assistant output of length m. The target excludes later environment responses and is held fixed when comparing the original context with a deleted context. Given a frozen scoring language model p_ω, we define

$$\mathcal{L}_\omega(y_t \mid C) = -\frac{1}{m}\sum_{k=1}^{m} \log p_\omega(y_{t,k} \mid C, y_{t,<k}), \qquad H_t(A) = \mathcal{L}_\omega(y_t \mid D_A(C_t)) - \mathcal{L}_\omega(y_t \mid C_t). \tag{3}$$

The quantity $H_t(A)$ measures how much harder it becomes to reproduce the same recorded decision after deleting A. Holding y_t fixed makes the comparison depend on the deletion rather than on a newly sampled continuation. Normalization by output length places decisions of different lengths on a common scale. Since all candidate sets at a state are measured from the same intact context, the original context score can be reused when constructing the corresponding labels.

We define an offline unsafe event by

$$u_t(A) = \mathbf{1}[H_t(A) > \delta_H], \tag{4}$$

where δ_H is a fixed harm tolerance. Importantly, deletion harm is indexed by the current decision state. The same historical Block may be removable at one stage of execution and necessary at another. Historical value is therefore treated as state dependent rather than as a fixed property of the Block itself. The value used for δ_H and its interpretation are reported in Appendix B.

3.3 WHY SINGLETON SCORES DO NOT DETERMINE SET RISK

For a single Block, define $H_{i,t} = H_t(\{B_i\})$. For $B_i \notin A$, its conditional effect after deleting A is

$$H_t(B_i \mid A) = H_t(A \cup \{B_i\}) - H_t(A).$$

If $A_k = \{B_{i_1}, \ldots, B_{i_k}\}$ and all intermediate rewrites are valid, then

$$H_t(A) = \sum_{k=1}^{|A|} H_t(B_{i_k} \mid A_{k-1}), \qquad A_0 = \varnothing. \tag{5}$$

Thus, joint harm is a sum of effects evaluated under changing retained contexts. It is not generally the sum of singleton effects measured in the original context. Equation (5) holds for every deletion order, while the final quantity $H_t(A)$ is independent of that order. The conditional terms change because each deletion changes what information remains.

To measure this difference directly, we define

$$I_t(A) = H_t(A) - \sum_{B_i \in A} H_{i,t}, \qquad J_{ij,t} = H_t(\{B_i, B_j\}) - H_{i,t} - H_{j,t}. \tag{6}$$

When $I_t(A) > 0$, the actual joint deletion is more harmful than singleton addition suggests. This can occur when two Blocks contain redundant evidence: either one appears safe to remove while the other remains, but removing both eliminates the last copy of that evidence. When $I_t(A) < 0$, singleton addition overestimates joint harm because it counts part of the same effect more than once. Only $I_t(A) = 0$ permits exact additive reconstruction from singleton harms.

Interaction is not the only limitation of singleton scoring. Even when $I_t(A) = 0$, several individually acceptable harms can accumulate beyond the allowed tolerance. DRSR therefore treats history pruning as a bounded set selection problem. Direct set-level prediction addresses interactions among deleted Blocks and the effect of what remains, while structural constraints limit excessive deletion. These two failure modes motivate predicting risk for the proposed set rather than deriving it from independently scored historical units.

3.4 Risk-Constrained Deletion Objective

Let $\mathcal{A}_t$ denote the deletion sets that satisfy the structural requirements of the agent protocol. If exact deletion harm were observable at decision time, an ideal pruning policy would solve

$$A_t^{\text{oracle}} \in \arg\max_{A \in \mathcal{A}_t} w_t(A) \qquad \text{subject to} \qquad H_t(A) \le \delta_H. \tag{7}$$

This objective separates compression benefit from deletion risk. The quantity $w_t(A)$ measures how much context is removed, while $H_t(A)$ determines whether the proposed edit remains within the allowed harm tolerance.

The exact constraint cannot be evaluated online. Computing $H_t(A)$ requires the realized next output y_t, which does not yet exist when the pruning decision is made. Evaluating candidate deletions through separate model generations would also fail to recover the fixed-target quantity in Eq. (3), because different contexts can produce different continuations. Completed trajectories make exact supervision possible offline: after y_t is observed, candidate sets can be deleted jointly and the same recorded output can be rescored.

DRSR learns an online scorer from these offline measurements. At deployment, the scorer estimates deletion risk using only the current causal context and the proposed deletion set during each online pruning decision. It therefore replaces the unavailable oracle constraint, while direct token saving remains the quantity optimized by the pruning policy.

4 Direct Relational Set-Risk Pruning

4.1 Candidate Construction and Relational Features

At each eligible decision, DRSR protects recent complete Steps and constructs a small candidate pool from older protocol-valid history. Candidates are selected across the trajectory using chronological stratification, with preference given to edits that offer meaningful direct token saving. The candidate rule is fixed before any deletion outcome is observed, using chronological structure and token saving independently of deletion labels. Exact candidate counts, protection rules, alignment checks, and context limits are reported in Appendix C.1.

A frozen representation backbone encodes the intact causal prefix. DRSR represents each historical Block in relation to the current decision state. The feature vector for candidate B_i therefore encodes its decision-conditioned relation to the available history. Let Ω_i denote the token span of B_i. The feature vector $x_{i,t}$ combines structural controls such as age, position, type, length, lexical overlap, tool overlap, and redundancy with attention and hidden-state relations to the current pre-action state.

For a selected attention layer ℓ and head h, we define attention mass and density as

$$a_{i,t}^{\ell,h} = \sum_{k \in \Omega_i} \alpha_{t,k}^{\ell,h}, \qquad d_{i,t}^{\ell,h} = \frac{a_{i,t}^{\ell,h}}{|\Omega_i|}. \tag{8}$$

Attention mass measures how much attention from the current pre-action position falls on the Block, while density reduces the direct effect of Block length.

Hidden-state relations provide a complementary view of the same Block to state relation. DRSR compares the mean hidden representation of the Block with the hidden representation at the current pre-action position using cosine similarity, normalized distance, relative norm, and boundary-token relations across selected layers. These features describe whether the historical content aligns with the current decision in representation space, rather than describing the Block in isolation. Fixed, label-independent projections retain additional directional information that a single similarity value cannot capture. Exact formulas, layers, dimensions, and projection details are given in Appendix C.

Attention, hidden-state relations, and projected semantic features jointly provide the candidate and state inputs used by the learned set-risk model. Attention and similarity describe relations between candidates and states, while the learned set-risk scorer combines these signals with the proposed deletion and the remaining candidate information. The final prediction is supervised by measured set-level deletion harm and also depends on the information that survives the proposed edit.

4.2 Set-Risk Representation and Learning

For a proposed deletion $A \subseteq E_t$, let $D = A$ denote the candidates to be removed and let $R = E_t \setminus A$ denote the candidates that remain. DRSR represents the proposed edit as a whole rather than assigning each Block a fixed deletion score:

$$r_t(A) = \mathrm{SetEnc}_\theta\left(\{x_{i,t} : B_i \in D\}, \{x_{i,t} : B_i \in R\}, b_t(A)\right). \tag{9}$$

The encoder represents four sources of information: what is removed, what remains, relations among jointly deleted candidates, and relations between deleted and retained evidence. Deletion risk therefore depends not only on the content being removed but also on whether equivalent or supporting information survives elsewhere in the context.

A lightweight pairwise summary gives the scorer direct information about relations within the deleted set. This provides an interaction-aware bias while allowing the learned scorer to represent joint effects beyond pairwise addition. The exact pooling operators, deleted and retained summaries, and budget features are given in Appendix C.6.

A lightweight cardinality-aware scorer maps the set representation to predicted harm and an unsafe score for each candidate deletion set:

$$\left(\widehat{H}_t(A), \widehat{p}_t(A)\right) = g_{\theta,|A|}(r_t(A)). \tag{10}$$

The two outputs serve complementary roles. Predicted harm preserves the continuous ordering of deletion effects, while the unsafe score directly supports the thresholded online decision. Training combines harm regression, ranking among deletion sets, and unsafe-set classification. The unsafe score predicts the defined next-output harm event and supports threshold-based decisions about whether a proposed deletion should proceed. Exact model dimensions, loss terms, and optimization settings are reported in Appendices C and G.

4.3 Risk-Gated Selection and Abstention

At deployment, the learned risk score is used as a feasibility condition rather than as a reward for compression. Candidate subsets are batch-scored from shared intact-prefix representations by the lightweight set-risk scorer, with the language model evaluated before candidate-set scoring rather than after each proposed deletion. Protocol validity, recent-history protection, deletion size, and the removal budget are enforced directly, while the learned score estimates whether an otherwise valid edit is sufficiently low risk.

Let $\mathcal{A}_t$ denote the sets that satisfy the structural constraints, and let $U_t(A)$ denote the aggregate unsafe score produced by the scorer ensemble. DRSR selects

$$A_t^\star \in \arg\max_{A \in \mathcal{A}_t} w_t(A) \qquad \text{subject to} \qquad U_t(A) \le \tau. \tag{11}$$

This is the online counterpart of Eq. (7). The unavailable exact harm constraint is replaced by a learned score that can be computed before the next action under the current causal context, while

direct token saving remains the objective. If no candidate set passes the risk gate, DRSR abstains and returns the original request unchanged.

The candidate space is deliberately small, so all admissible subsets can be scored together by the lightweight network. Online execution shares intact-prefix representations across candidate subsets and evaluates each proposed deletion using the lightweight set-risk scorer. The exact candidate limit, removal budget, ensemble rule, validation-selected threshold, tie breaking, and computational cost are reported in Appendix J and Algorithm 1.

5 Experiments

We evaluate DRSR from both offline prediction and end-to-end agent performance for a comprehensive system-level assessment. The offline scorer is trained on 566 external trajectories containing 1,698 decision states and 24,126 jointly measured deletion-set labels, split 80:10:10 by source trajectory and disjoint from WorkBuddyBench. End-to-end evaluation uses WorkBuddyBench Full260 (80 Code, 50 Office, 60 Security, and 70 Web tasks) and a fixed Eval40 subset with ten tasks per domain. Offline analyses report meaningful-gap pair accuracy, within-state Spearman correlation, unsafe-event AUROC, and Brier score; agent evaluation reports task reward and input-plus-output model tokens. Full260 aggregates are task-count weighted, whereas Eval40 uses the equal-domain mean. Additional dataset, token-accounting, task-manifest, and information-boundary details are provided in Appendices B, G, and J.

5.1 Mechanistic Analysis of History and Set Risk

State-conditioned historical relevance. The granularity analysis in Appendix F shows that the median within-state harm range is 0.312 at Step level but only 0.028 at Block level, and 71.400% of unsafe parent Steps contain at least one safe constituent Block. This motivates Block-level actions. More importantly, historical value is decision-conditioned. Using the matched pre-action state yields 70.920% meaningful-gap pair accuracy, compared with 62.020% under the mismatched-state control, an 8.900-point gap with the candidate Block held fixed. The full-context relation reaches 66.390%, while the best combined relation reaches 72.130%, indicating that relating a candidate to the correct current decision state is more informative than adding context indiscriminately. Complete granularity, layerwise, and harm-margin results are reported in Appendix F.

Joint-deletion effects. The central structural question is whether singleton importance can recover the risk of deleting a *set*. We first test the most direct alternative to set-level prediction: singleton harms are aggregated and used to rank candidate deletion sets. Table 1 shows that this approximation is unreliable. Even the strongest simple rule, summation, reaches only 58.940% meaningful-gap pair accuracy. Mean aggregation performs similarly, while maximum and minimum aggregation deteriorate further. Thus, the tested singleton aggregators provide limited accuracy for predicting set risk.

Table 1. Singleton aggregation.

Rule	Pair Acc.
Sum	**58.94%**
Mean	56.81% (-3.61%)
Maximum	46.17% (-21.67%)
Minimum	14.89% (-74.74%)

The failure of singleton aggregation is consistent with the measured interaction residual. Figure 2 examines this effect over the 200-source confirmatory range. Mean absolute non-compositionality increases from 0.000 for singleton deletion to 0.008, 0.016, and 0.027 for deletion sets of size two, three, and four, respectively.

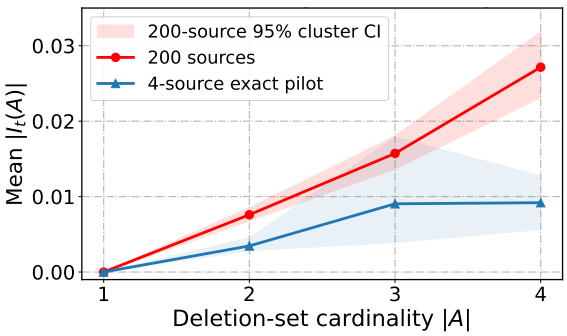


Figure 2. Interaction grows with cardinality.

The discrepancy between independent singleton effects and actual joint deletion therefore becomes more pronounced as several Blocks are removed together. At three Blocks, the mean interaction magnitude is already close

to the $\delta_H = 0.020$ harm-label threshold, while at four Blocks it exceeds that threshold. This makes additive singleton scoring increasingly unreliable in the multi-Block regime targeted by DRSR.

The complete analysis, including the exploratory extension to larger sets and the per-domain pilot, is reported in Appendix F.

We next ask whether explicit interaction information can explain the discrepancy between singleton and joint deletion effects.

Table 2 provides a diagnostic reconstruction test. The singleton-additive approximation obtains an MAE of 0.018, whereas including the exact measured pair interaction reduces the error to 0.008, improving joint-harm reconstruction. Thus, second-order interactions account for a substantial portion of the information that singleton scores fail to capture.

Table 2. Joint-harm reconstruction.

Method	**MAE**
Singleton additive	0.018
+ exact pair interaction	**0.008**

This reconstruction experiment uses exact pair effects separately from the deployed set-risk policy. DRSR uses learned interaction features during online selection. Specifically, it learns set-level deletion risk for each set from online-visible relations between candidates and states together with deleted, retained, and pair-pooled representations.

Taken together, the poor singleton aggregation accuracy, the increasing interaction magnitude with deletion-set cardinality, and the large reconstruction improvement from pair information provide complementary evidence for modeling deletion risk directly at the set level.

5.2 Full260 End-to-End Results

Table 3 reports the principal Full260 result together with *Structural Pruning*, our simple rule-based reference. Structural Pruning reaches a task-weighted overall reward of 0.698 while reducing the token mean from 2.685M to 2.292M per task. DRSR improves substantially beyond this reference, reaching 0.802 reward at 2.126M tokens per task. The gain is especially pronounced in Security, where Structural Pruning reaches 0.640 and DRSR reaches 0.756.

Table 3. Full260 comparison by domain. R denotes reward and T denotes mean total tokens per task in millions. Overall reports the corresponding Full260 aggregate for each method.

Method	**Code**		**Office**		**Security**		**Web**		**Overall**	
	$R\uparrow$	$T\downarrow$	$R\uparrow$	$T\downarrow$	$R\uparrow$	$T\downarrow$	$R\uparrow$	$T\downarrow$	$R\uparrow$	$T\downarrow$
Baseline	0.770	1.370	0.818	1.198	0.478	5.977	0.721	2.429	0.699	2.685
Structural Pruning	0.703	1.380	0.814	0.893	0.640	4.961	0.659	2.046	0.698	2.292
DRSR	**0.858**	1.235	**0.824**	0.997	**0.756**	4.569	**0.760**	1.858	**0.802**	2.126

The domain breakdown shows variation in reward gains and token reductions across the four evaluated task categories. Security has the largest baseline context and the largest reward improvement, while Office changes only modestly. DRSR nevertheless reduces processed tokens in all four domains, consistent with selective history removal rather than a fixed context budget.

5.3 Context-Management Comparison on Eval40

Table 4 compares DRSR with the uncompressed agent, simple context controls, and published context-management methods on the same fixed Eval40 task manifest. Each domain contains ten tasks. We report reward on its native scale from 0 to 1 and mean input-plus-output tokens per task in millions; auxiliary summarization-model tokens are included where applicable.

On Eval40, DRSR achieves 0.794 overall reward at 1.211M tokens per task, compared with 0.675 reward and 1.887M tokens for the uncompressed agent. This is a reward gain of 0.119 while reducing total tokens by 35.85%. CoMem and PACE use fewer tokens but obtain substantially lower reward, while the strongest non-DRSR reward (Periodic Summary, $n = 5$) remains below DRSR while using nearly twice as many tokens.

5.4 Component Ablation

We ablate the three design choices that distinguish DRSR from independent pruning: retained-context features, pair pooling inside the deleted set, and the abstention gate. Table 5 reports reward and total token usage in each Eval40 domain, together with the equal-domain overall mean. Direct-removal statistics are reported in the discussion below and detailed in Appendix I.

Table 4. Eval40 context-management comparison. R is reward and T is mean model tokens per task in millions.

Method	Code		Office		Security		Web		Overall	
	$R\uparrow$	$T\downarrow$	$R\uparrow$	$T\downarrow$	$R\uparrow$	$T\downarrow$	$R\uparrow$	$T\downarrow$	$R\uparrow$	$T\downarrow$
Baseline	0.729	1.438	<u>0.816</u>	1.040	0.446	3.973	0.710	1.098	0.675	1.887
Sliding Window ($K=5$)	0.668	3.123	0.674	3.982	0.323	2.126	0.590	2.577	0.564	2.952
Sliding Window ($K=10$)	0.744	1.756	0.731	1.816	0.401	0.819	0.610	3.547	0.621	1.985
Sliding Window ($K=20$)	0.793	1.419	**0.851**	2.231	0.403	1.132	0.700	3.300	0.687	2.020
Periodic Summary ($n=3$)	0.654	1.913	0.677	2.135	0.263	0.990	<u>0.827</u>	3.819	0.605	2.214
Periodic Summary ($n=5$)	<u>0.836</u>	1.758	0.757	2.552	0.400	1.292	0.813	3.883	<u>0.702</u>	2.371
PACE (Wei et al., 2026)	0.704	0.941	0.558	1.013	0.435	0.573	0.640	1.885	0.584	<u>1.103</u>
LLMLingua-2 (Pan et al., 2024)	0.646	1.630	0.733	3.412	0.401	3.487	0.700	1.090	0.620	2.405
SelfCompact (Li et al., 2026b)	0.768	1.571	0.796	0.850	0.362	3.136	0.710	0.712	0.659	1.567
ACON-Core (Kang et al., 2025)	0.718	1.476	0.686	2.154	<u>0.480</u>	1.511	0.650	0.473	0.633	1.403
Self-GC (Hao et al., 2026)	0.635	1.167	0.717	1.178	0.437	3.438	0.710	0.837	0.625	1.655
LRE (Jahan Lia & Mazumder, 2026)	0.626	2.258	0.634	3.470	0.271	2.850	0.680	0.586	0.553	2.291
CoMem (Zhang et al., 2026a)	0.564	0.550	0.599	0.745	0.025	1.004	0.580	0.764	0.442	**0.766**
SAM (Hu et al., 2026)	0.676	1.412	0.816	1.094	0.472	1.910	0.670	0.969	0.659	1.346
SWE-Pruner (Wang et al., 2026)	0.635	2.172	0.726	1.431	0.409	1.822	0.650	0.582	0.605	1.502
Sculptor (Li et al., 2026a)	0.512	1.029	0.816	0.985	0.353	2.453	0.690	0.708	0.593	1.294
ACM (Li et al., 2026c)	0.387	2.540	0.559	1.526	0.062	5.196	0.340	0.193	0.337	2.364
DRSR	**0.856**	1.216	0.788	0.724	**0.700**	1.616	**0.833**	1.288	**0.794**	1.211

Table 5. Eval40 component ablation. R denotes reward and T denotes mean total model tokens per task in millions. Overall is the equal-domain mean. Direct is the mean number of tokens directly removed per task in thousands across the fixed evaluation tasks.

Variant	Code		Office		Security		Web		Overall		
	$R\uparrow$	$T\downarrow$	$R\uparrow$	$T\downarrow$	$R\uparrow$	$T\downarrow$	$R\uparrow$	$T\downarrow$	$R\uparrow$	$T\downarrow$	**Direct**
DRSR	**0.856**	1.216	<u>0.788</u>	0.724	<u>0.700</u>	1.616	**0.833**	1.288	**0.794**	1.211	25.155
w/o retained	0.722	0.716	0.779	0.266	0.698	1.254	<u>0.800</u>	1.731	0.750	0.992	12.633
w/o pair pooling	0.756	0.567	0.678	0.606	**0.796**	1.094	0.796	2.535	<u>0.757</u>	1.201	32.205
w/o abstention	<u>0.833</u>	1.205	**0.801**	0.447	0.573	0.963	<u>0.800</u>	1.553	0.752	1.042	30.093

The ablation effects are domain dependent, but every removal lowers the overall reward. Removing abstention increases direct deletion from 25.155K to 30.093K tokens per task while lowering overall reward from 0.794 to 0.752. Removing retained-context features makes the policy more conservative in direct deletion (12.633K) yet still lowers overall reward to 0.750, indicating that knowing what survives the edit matters for identifying redundancy. Removing pair pooling yields the largest direct deletion, 32.205K tokens per task, while overall reward falls to 0.757. Together with the measured interaction residuals, this pattern is consistent with multi-Block deletions appearing safer when within-set interaction is not represented.

6 Conclusion

We formulate agent-history pruning as selection over the joint deletion risk of structured Blocks. DRSR learns this risk from offline counterfactual supervision and uses a compact scorer to gate online pruning with abstention. Our analyses show that historical relevance is decision-conditioned and that independently scored Blocks provide limited accuracy for recovering joint deletion effects. On WorkBuddyBench Full260, DRSR improves reward from 0.699 to 0.802 while reducing tokens by 20.82fewer tokens. These results support state-dependent, set-level compression that models both the current decision state and joint deletion structure, providing a practical basis for reducing long-horizon agent context.

Reproducibility Statement

The deletion action space, counterfactual harm target, and label threshold are defined in Section 3; candidate construction, relational features, set representation, training objective, and online selection are specified in Section 4. Appendix A gives the complete decision procedure, Appendix C lists the representation and scorer configuration, Appendix G defines the training and evaluation metrics, Appendix E documents the simple structural baseline, Appendix I reports detailed token summaries, and Appendix J specifies the information boundary and validation-frozen deployment rule.

AI Use Statement

Generative AI was used to aid or polish writing and for retrieval and discovery. This included improving language, readability, clarity, organization, and LaTeX formatting, as well as assisting with literature discovery. All AI-assisted text and retrieved references were carefully reviewed, verified, and approved by the authors, who take full responsibility for the entire submission.

References

Anastasios Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International conference on learning representations*, volume 2024, pp. 55198–55218, 2024.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM (JACM)*, 68(6):1–34, 2021.

Chao Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16(1): 41–46, 1970.

Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021.

Zican Dong, Peiyu Liu, Junyi Li, Zhipeng Chen, Han Peng, Shuo Wang, and Wayne Xin Zhao. Foresightkv: Optimizing kv cache eviction for reasoning models by learning long-term contribution. *arXiv preprint arXiv:2602.03203*, 2026.

Ran El-Yaniv et al. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(5), 2010.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S Kevin Zhou. Ada-kv: Optimizing kv cache eviction by adaptive budget allocation for efficient llm inference. *Advances in Neural Information Processing Systems*, 38:113152–113188, 2026.

Fabian Fumagalli, Maximilian Muschalik, Patrick Kolpaczki, Eyke Hüllermeier, and Barbara Hammer. Shap-iq: Unified approximation of any-order shapley interactions. *Advances in Neural Information Processing Systems*, 36:11515–11551, 2023.

Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *International conference on machine learning*, pp. 2151–2159. PMLR, 2019.

Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of game theory*, 28(4):547–565, 1999.

Xubin Hao, Hongjin Meng, Xin Yin, Jiawei Zhu, and Chenpeng Cao. Self-gc: Self-governing context for long-horizon llm agents. *arXiv preprint arXiv:2607.00692*, 2026.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Yuyang Hu, Hongjin Qian, Shuting Wang, Jiongnan Liu, Ziliang Zhao, Jiejun Tan, Zheng Liu, and Zhicheng Dou. Sam: State-adaptive memory for long-horizon reasoning agent. *arXiv preprint arXiv:2605.24468*, 2026.

Nusrat Jahan Lia and Aritra Mazumder. Learning what not to forget: Long-horizon agent memory from a few kilobytes of learning. *arXiv e-prints*, pp. arXiv–2606, 2026.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 13358–13376, 2023.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations*, volume 2024, pp. 54107–54157, 2024.

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.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

Mo Li, LH Xu, Qitai Tan, Long Ma, Hongyong Song, Ting Cao, and Yunxin Liu. Sculptor: Empowering llms with cognitive agency via active context management. In *International Conference on Learning Representations*, volume 2026, pp. 153411–153440, 2026a.

Tianjian Li, Jingyu Zhang, William Jurayj, Xi Wang, Chuanyang Jin, Mehrdad Farajtabar, Eric Nalisnick, and Daniel Khashabi. Self-compacting language model agents. *arXiv preprint arXiv:2606.23525*, 2026b.

Xiaochuan Li, Ryan Ming, Meng Chu, Shuai Shao, Rong Jin, and Chenyan Xiong. Acm: Agentic context management for long horizon tasks. *arXiv preprint arXiv:2607.23809*, 2026c.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

Zeju Li, Jianyuan Zhong, Ziyang Zheng, Xiangyu Wen, Zhijian Xu, Yingying Cheng, Fan Zhang, and Qiang Xu. Making slow thinking faster: Compressing llm chain-of-thought via step entropy. In *International Conference on Learning Representations*, volume 2026, pp. 38331–38349, 2026d.

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

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

Scott M Lundberg, Gabriel G Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

Christopher Mohri and Tatsunori Hashimoto. Language models with conformal factuality guarantees. *arXiv preprint arXiv:2402.10978*, 2024.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *Advances in Neural Information Processing Systems*, 37:8901–8929, 2024.

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.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, et al. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 963–981, 2024.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International conference on machine learning*, pp. 31210–31227. PMLR, 2023.

Weiwei Sun, Miao Lu, Zhan Ling, Kang Liu, Xuesong Yao, Yiming Yang, and Jiecao Chen. Scaling long-horizon llm agent via context-folding. *arXiv preprint arXiv:2510.11967*, 2025.

Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. The shapley taylor interaction index. In *International conference on machine learning*, pp. 9259–9268. PMLR, 2020.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, 2024.

Che-Ping Tsai, Chih-Kuan Yeh, and Pradeep Ravikumar. Faith-shap: The faithful shapley interaction index. *Journal of Machine Learning Research*, 24(94):1–42, 2023.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009.

Yuhang Wang, Yuling Shi, Mo Yang, Rongrui Zhang, Shilin He, Heng Lian, Yuting Chen, Siyu Ye, Kai Cai, and Xiaodong Gu. Swe-pruner: Self-adaptive context pruning for coding agents. *arXiv preprint arXiv:2601.16746*, 2026.

Lei Wei, Xiao Peng, Guannan Zhang, Chenhao Jiang, Hongyu Li, Lanbo Lin, Yuanwu Xu, Jiayao Liu, Kesu Wang, Bin Wang, et al. Pace: Predictive adaptive context extraction for long-horizon llm agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27184–27199, 2026.

Junde Wu, Minhao Hu, Jiayuan Zhu, Jiazhen Pan, Yuyuan Liu, Min Xu, and Yueming Jin. Git context controller: Manage the context of llm-based agents like git. *arXiv preprint arXiv:2508.00031*, 2025.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. Tokenskip: Controllable chain-of-thought compression in llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3351–3363, 2025.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pp. 21875–21895, 2024.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, et al. Agentfold: Long-horizon web agents with proactive context management. *arXiv preprint arXiv:2510.24699*, 2025.

Yuwei Zhang, Chengyu Dong, Shuowei Jin, Changlong Yu, Hejie Cui, Hongye Jin, Xinyang Zhang, Hamed Bonab, Colin Lockard, Jianshu Chen, et al. Comem: Context management with a decoupled long-context model. *arXiv preprint arXiv:2605.30842*, 2026a.

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, 2026b.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in neural information processing systems*, 36:34661–34710, 2023.

A Decision Procedure

Algorithm 1 One DRSR decision

Require: Valid current request C_t; frozen representation model; scorer ensemble; $K = 3$, $q_{\max} = 0.20$, $\tau = 0.0740822119017442$
Ensure: Protocol-valid pruned request or unchanged C_t

Protect the system/task prefix and Recent2; build an aligned representation view within the context cap.
Construct outcome-blind protocol-valid candidates E_t.
if fewer than three candidates, alignment failure, or invalid protocol **then**
    return C_t
end if
Extract structural, attention, hidden-state, and projected semantic features from the intact representation view.
Enumerate all nonempty $A \subseteq E_t$ with $|A| \leq K$ and batch-score them to obtain $\bar{H}_t(A)$ and $U_t(A)$.
Map each subset to the full request, measure $w_t(A)$, and form the feasible set using protocol, recency, removal-budget, and risk constraints.
if no feasible set remains **then**
    return C_t
end if
Select $A_t^\star$ by maximum direct saving, breaking ties by lower unsafe score and then lower predicted harm.
Jointly delete $A_t^\star$, reserialize the request, and validate the final tool protocol.
return the valid rewritten request; otherwise return C_t.

B Supervision and Task Sets

Table 6. Offline supervision and benchmark task counts.

External-trajectory supervision					
Source trajectories	**Decision states**	**Set-level labels**	**Train / validation / test**		
566	1,698	24,126	80:10:10 by trajectory		
Task evaluation					
Set	**Code**	**Office**	**Security**	**Web**	**Total**
Full260	80	50	60	70	260
Eval40	10	10	10	10	40

The supervision set contains 24,126 set-level deletion labels and is split 80:10:10 by source trajectory.

All states and deletion sets generated from one source trajectory remain in the same split. This grouping prevents closely related deletion rows from crossing train, validation, and test partitions. Task identity is specified by the fixed benchmark manifest within each evaluation set. Acting-model configurations and token-accounting rules provide the remaining protocol details for interpreting comparisons across methods.

The numerical value 0.020 appears in two different roles. The meaningful-gap threshold selects which pairs are counted in pair-ordering evaluation, while δ_H in Eq. (4) defines the offline unsafe event. The two tests use the same numerical margin but serve different purposes.

C Representation and Scorer Details

C.1 Candidate Construction and Structural Controls

Candidate construction is frozen before any deletion outcome is observed. The two most recent complete Steps are protected. Earlier protocol-valid atoms are grouped into at most four chronological

bins according to the position of their parent Step, and the longest atom in each bin is selected. This produces three or four candidates for an eligible request. If fewer than three candidates remain, alignment fails, or a rewrite is invalid, the request is left unchanged.

For each candidate Block B_i, structural controls describe properties that are directly visible before the next action. They include parent-Step age and relative position, Block type and component index, character count, word count, removable token count, lexical overlap with the task and recent history, tool-related overlap, redundancy with other candidates, and simple content indicators such as error, success, path, numeric, and nonzero-exit density. These controls provide context for the learned scorer. Their influence is learned jointly with the relational features.

Representation extraction uses the frozen Qwen3.5-9B backbone. If the full request exceeds 55,296 tokens, DRSR takes the longest exactly aligned suffix that still contains Recent2. Candidate spans extracted from this view are mapped back to the complete serialized request before any token-saving or removal constraint is evaluated.

C.2 Attention Relations

Let t denote the final pre-action position in the intact causal prefix. For attention layer ℓ, head h, and historical token position k, the standard attention weight is

$$\alpha_{t,k}^{\ell,h} = \mathrm{softmax}_k \left(\frac{q_t^{\ell,h} (k_k^{\ell,h})^\top}{\sqrt{d_h}} \right). \tag{12}$$

For the token span Ω_i of Block B_i, DRSR uses the attention mass and density

$$a_{i,t}^{\ell,h} = \sum_{k \in \Omega_i} \alpha_{t,k}^{\ell,h}, \qquad d_{i,t}^{\ell,h} = \frac{a_{i,t}^{\ell,h}}{|\Omega_i|}. \tag{13}$$

Mass measures the total attention assigned to the Block, while density normalizes that quantity by Block length. The implementation records head-level mass and density together with summary statistics across heads, including mean and maximum mass, mean and maximum density, and the mean of the largest token-level attention values within the Block.

Formal attention features are extracted only from the standard full-attention layers

$$\ell \in \{4, 8, 12, 16, 20, 24, 28, 32\}.$$

Other backbone layers use a different attention implementation and are not treated as measurements of the same attention quantity. Any interpolated points used only for visualization are excluded from the formal features and from the reported attention analysis.

Attention supplies one input to the set-risk predictor. A Block can receive low attention and still carry a constraint whose removal changes the next decision. Conversely, a highly attended Block can be redundant when similar evidence remains elsewhere. Attention therefore enters the candidate representation only as one online-visible relation.

C.3 Hidden-State Relations

The hidden-state features are designed to measure how a historical Block relates to the current decision state. The target is relational rather than intrinsic to the historical Block.

Let $h_k^\ell \in \mathbb{R}^{d_\ell}$ be the hidden state of token k at layer ℓ. For Block B_i with token span Ω_i, define its mean hidden representation as

$$\bar{h}_i^\ell = \frac{1}{|\Omega_i|} \sum_{k \in \Omega_i} h_k^\ell, \tag{14}$$

and let h_t^ℓ denote the hidden state at the final pre-action position. DRSR compares the two representations through

$$\cos(\bar{h}_i^\ell, h_t^\ell), \qquad \frac{\|\bar{h}_i^\ell - h_t^\ell\|_2}{\sqrt{d_\ell}}, \qquad \log \frac{\|\bar{h}_i^\ell\|_2}{\|h_t^\ell\|_2}. \tag{15}$$

The cosine term measures directional similarity, the normalized Euclidean distance measures separation in representation space, and the log norm ratio records differences in representation magnitude without allowing hidden dimension to dominate the scale.

Mean-span features can hide information concentrated at the boundaries of a Block. The implementation therefore also includes cosine relations between h_t^ℓ and the first and last token states of the Block:

$$\cos(h_{i,\text{first}}^\ell, h_t^\ell), \qquad \cos(h_{i,\text{last}}^\ell, h_t^\ell). \tag{16}$$

Together, the mean and boundary relations provide several views of the same state-conditioned question: whether the representation of historical content remains aligned with the representation immediately before the next action.

These quantities describe relations between candidates and states used by the learned scorer. Their usefulness is learned from offline deletion labels. Retention and deletion decisions combine these features with the remaining components of the learned set-risk representation. Numerical implementations stabilize zero norms consistently before cosine and norm-ratio computation.

C.4 Fixed Projected Semantic Relations

Scalar similarities compress a high-dimensional relation into a small number of values. DRSR therefore retains additional directional information through fixed random projections. At semantic layers

$$\ell \in \{8, 16, 24, 32\},$$

let $R_\ell \in \mathbb{R}^{d_\ell \times 64}$ be a Gaussian projection whose entries are drawn independently with variance $1/64$. The projection seed is fixed to $73000 + \ell$. The projected Block and current-state pair is

$$s_{i,t}^\ell = \left[\frac{\bar{h}_i^\ell}{\|\bar{h}_i^\ell\|_2} R_\ell,\ \frac{h_t^\ell}{\|h_t^\ell\|_2} R_\ell \right]. \tag{17}$$

The projection matrices are fixed before training and are independent of all deletion labels. They therefore provide a compact view of semantic directions without fitting the representation backbone or the projection to downstream outcomes.

C.5 Candidate Feature Assembly

The complete candidate vector concatenates the structural, attention, hidden-state, and projected semantic components:

$$x_{i,t} = \left[x_{i,t}^{\text{struct}}, x_{i,t}^{\text{attn}}, x_{i,t}^{\text{hidden}}, x_{i,t}^{\text{proj}} \right] \in \mathbb{R}^{1154}. \tag{18}$$

Every component is computed from the intact causal prefix. No candidate deletion is executed to construct $x_{i,t}$, and no realized future output, measured harm, or task reward is included. The feature order is fixed across training and deployment.

The candidate features retain temporal information through structural position variables even though the later set encoder is permutation invariant within the deleted and retained groups. The representation view can be shorter than the complete request because of the 55,296-token cap, but token saving, Recent2 protection, protocol validation, and removal limits are always evaluated after mapping candidate spans back to the full request.

C.6 Exact Set Representation

A two-layer Linear/GELU encoder maps the 1,154-dimensional candidate vector to a 64-dimensional embedding

$$e_i = \phi_\theta(x_{i,t}) \in \mathbb{R}^{64}. \tag{19}$$

For deleted candidates $D = A$ and retained candidates $R = E_t \setminus A$, let μ_G and m_G denote the mean and elementwise maximum of the embeddings in a nonempty group G. Empty groups use zero-padded summaries.

The deleted-set pair statistic is

$$p_D = \frac{\left(\sum_{i\in D} e_i\right)^{\odot 2} - \sum_{i\in D} e_i^{\odot 2}}{2\max\left\{\binom{|D|}{2}, 1\right\}}. \tag{20}$$

For $|D| \geq 2$, this equals the average elementwise product over unordered pairs of deleted candidates. For a singleton, the numerator is zero and p_D is the zero vector. Appendix D gives the algebraic identity behind this pooling operation.

The model also compares the deleted and retained groups through

$$c_{D,R} = \mu_D \odot \mu_R, \qquad g_{D,R} = |\mu_D - \mu_R|. \tag{21}$$

These terms expose whether the proposed deletion and the surviving candidate information occupy similar or different regions of the learned candidate space. They do not impose a hand-written redundancy rule. Their effect is learned from the set-level supervision.

The budget vector is

$$b_t(A) = \left[\frac{|A|}{|E_t|}, \frac{\log(1 + w_t(A))}{10}, \frac{w_t(A)}{\sum_{B_i\in E_t} w_t(\{B_i\})}\right]. \tag{22}$$

The exact set representation is

$$r_t^{\text{exact}}(A) = [\mu_D, \mu_R, m_D, m_R, p_D, \mu_D \odot \mu_R, |\mu_D - \mu_R|, b_t(A)] \in \mathbb{R}^{451}. \tag{23}$$

Each of the seven pooled embedding terms contributes 64 coordinates and $b_t(A)$ contributes three, giving $7 \times 64 + 3 = 451$ features.

The pair statistic provides information about interactions inside the deleted set, while the deleted and retained geometry provides information about what survives after the edit. The final scorer is not restricted to an additive sum of singleton or pair effects.

C.7 Scorer and Training Objective

A shared 64-dimensional trunk maps $r_t^{\text{exact}}(A)$ to h_A. Cardinality-indexed heads produce

$$\widehat{H}_t(A) = g^{\text{harm}}_{|A|}(h_A), \qquad \widehat{p}_t(A) = \sigma\left(g^{\text{risk}}_{|A|}(h_A)\right). \tag{24}$$

Training includes deletion cardinalities one through four, although online selection permits at most three deleted candidates. Separate heads allow the output mapping to vary with set size, which is useful because the scale and frequency of interaction can change with cardinality.

The two outputs have different roles. The harm head learns a continuous quantity and supports ranking among candidate sets. The unsafe head learns the event in Eq. (4) that is used by the online gate. The unsafe output is a learned decision score and is not interpreted as a calibrated probability of downstream task failure.

The composite training objective is

$$\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{rank}} + 0.30\mathcal{L}_{\text{SmoothL1}} + 0.40\mathcal{L}_{\text{BCE}} + 0.10\mathcal{L}_{\text{Brier}}. \tag{25}$$

The regression term fits measured joint harm. Weighted BCE uses

$$w_+ = \min\left(\frac{N_-}{N_+}, 4\right), \tag{26}$$

and the Brier term penalizes squared error on the unsafe target. The ranking term is defined in Appendix G. Checkpoint selection lexicographically prioritizes unsafe AUROC, meaningful-gap pair accuracy, and negative Brier score. The test split is used for neither fitting nor threshold selection.

C.8 Model and Training Configuration

D Algebraic Properties and Design Implications

This section collects identities used by the main formulation and explains their design implications. The results characterize the defined harm quantities and pooling operators, while recovery of measured deletion harm by the learned scorer is evaluated empirically.

Table 7. Model and training configuration. The language models are frozen; only the set-risk scorer is optimized.

Configuration item	**Value**
Representation and scorer	
Representation backbone	Frozen Qwen3.5-9B
Acting model	DeepSeek-V4-Flash
Candidate features / embedding	1,154 / 64 dimensions
Set representation / shared hidden state	451 / 64 dimensions
Attention layers	4, 8, 12, 16, 20, 24, 28, 32
Semantic layers / projection size	8, 16, 24, 32 / 64
Projection seed at layer ℓ	$73000 + \ell$
Context / target token cap	55,296 / 3,072
Scorer optimization	
Optimizer / learning rate	AdamW / 2×10^{-3}
Weight decay / gradient clipping	10^{-4} / 5.0
Maximum epochs / evaluation interval	300 / every 5 epochs
Early-stopping patience	12 evaluations
Ensemble seeds	11, 29, 47
Online constraints	
Maximum candidates / deletion cardinality	4 / 3
Protected complete Steps	Most recent 2
Per-request removal cap	20.000% of full request
Harm-label threshold δ_H	0.020
Deployment threshold τ	0.0740822119017442
Uncertainty coefficient β	0
No feasible deletion set	Retain the complete request

D.1 Telescoping Decomposition of Joint Harm

For an ordered deletion sequence $A_k = \{B_{i_1}, \ldots, B_{i_k}\}$ with $A_0 = \varnothing$, recall

$$H_t(B_{i_k} \mid A_{k-1}) = H_t(A_k) - H_t(A_{k-1}).$$

Proposition 1. If all intermediate rewrites are valid, then

$$H_t(A_m) = \sum_{k=1}^{m} H_t(B_{i_k} \mid A_{k-1}). \tag{27}$$

Proof. Substituting the definition of the conditional effect gives

$$\sum_{k=1}^{m} H_t(B_{i_k} \mid A_{k-1}) = \sum_{k=1}^{m} [H_t(A_k) - H_t(A_{k-1})] \tag{28}$$

$$= H_t(A_m) - H_t(A_0). \tag{29}$$

Since $A_0 = \varnothing$ and deleting the empty set leaves the context unchanged, $H_t(A_0) = 0$. This yields Eq. (27). □

The identity explains why singleton harm is insufficient in general. Each conditional term is evaluated after a different subset has already been removed. The final joint harm is independent of the chosen deletion order, but the intermediate conditional effects need not be.

D.2 Singleton Additivity and Interaction Residuals

For a set A, the interaction residual is

$$I_t(A) = H_t(A) - \sum_{B_i \in A} H_{i,t}.$$

Therefore

$$H_t(A) = \sum_{B_i \in A} H_{i,t} + I_t(A). \tag{30}$$

Exact singleton additivity for a specific set A holds if and only if $I_t(A) = 0$. Additivity is evaluated separately for each deletion set; zero residual at one set establishes the additive decomposition for that particular set.

For two Blocks,

$$H_t(\{B_i, B_j\}) = H_{i,t} + H_{j,t} + J_{ij,t}, \tag{31}$$

where $J_{ij,t}$ is defined in Eq. (6). Exact pair interaction therefore reconstructs pair harm by definition. For sets with three or more Blocks, higher-order residuals may remain after pairwise correction. The deployed scorer therefore learns set-level harm directly, while exact pair effects support the diagnostic reconstruction experiment.

D.3 Why Retained Context Changes Deletion Risk

For $B_i \notin A$,

$$H_t(B_i \mid A) = H_t(A \cup \{B_i\}) - H_t(A). \tag{32}$$

The effect of deleting B_i is therefore indexed by the set already removed, or equivalently by the context that remains. If another Block contains redundant evidence, deleting B_i can have small conditional effect while that evidence remains and a much larger effect after the redundant source is also removed.

This dependence motivates the deleted and retained partition in the set encoder. Explicit retained-set features expose surviving evidence alongside the deleted content, helping the scorer compare proposals that remove similar content while retaining different supporting information. DRSR therefore includes explicit retained-set summaries and deleted-to-retained geometry.

D.4 Identity Behind the Pair Pooling Statistic

Let

$$S_D = \sum_{i \in D} e_i.$$

Elementwise squaring gives

$$S_D^{\odot 2} = \sum_{i \in D} e_i^{\odot 2} + 2 \sum_{\substack{i<j \\ i,j \in D}} e_i \odot e_j. \tag{33}$$

Rearranging,

$$\frac{S_D^{\odot 2} - \sum_{i \in D} e_i^{\odot 2}}{2} = \sum_{\substack{i<j \\ i,j \in D}} e_i \odot e_j. \tag{34}$$

Dividing by $\binom{|D|}{2}$ when $|D| \geq 2$ yields the average unordered pair product used in Eq. (20). For $|D| = 1$, the numerator is zero, so the implementation defines p_D as the zero vector.

This statistic is permutation invariant and can be computed without explicitly listing every pair. It supplies information about pair structure to the downstream network, which learns joint deletion harm directly from the measured set-level targets.

D.5 Oracle Constraint and Learned Online Gate

The ideal objective in Eq. (7) uses the true event

$$H_t(A) \leq \delta_H.$$

This event cannot be evaluated at online decision time because it depends on the realized next output y_t. DRSR instead learns an unsafe score $\widehat{p}_t(A)$ from offline labels and aggregates an ensemble score $U_t(A)$ for deployment.

The online condition

$$U_t(A) \leq \tau$$

serves as a learned feasibility surrogate for the offline condition $H_t(A) \leq \delta_H$, linking the measured target to the online decision score. The two thresholds act on different variables. The value δ_H defines the offline target, while τ is selected on validation predictions under the policy constraints described in Appendix J.6.

D.6 Accumulation Without Interaction

Set-level reasoning is required even when the interaction residual is zero. Suppose $I_t(A) = 0$. Then

$$H_t(A) = \sum_{B_i \in A} H_{i,t}.$$

It is possible for every singleton to satisfy $H_{i,t} \leq \delta_H$ while the sum exceeds δ_H. Thus independent singleton thresholding can admit a joint deletion whose total harm exceeds tolerance. The cardinality cap and the direct set-level gate address this accumulation failure in addition to nonadditive interaction.

E Structural Pruning Baseline

Structural Pruning. We use a structural pruning implementation as a fixed rule-based reference for comparison. The heuristic protects the two most recent complete Steps, preserves tool calls and matched results, preserves state-bearing content, and removes only short transition narration for which later structure provides a witness. It performs direct deletion through fixed structural rules without learned risk scoring, counterfactual supervision, or external recall.

On Full260, the measured domain rewards are 0.703, 0.814, 0.640, and 0.659 for Code, Office, Security, and Web, respectively, with a task-weighted overall reward of 0.698. Its mean total tokens per task are 1.380M, 0.893M, 4.961M, and 2.046M by domain, with 2.292M overall. At the aggregate level, this corresponds to a 14.64% reduction from the canonical Full260 baseline. The main Full260 comparison presents these reward and token measurements together with the corresponding baseline and learned DRSR results.

F Additional Representation Results

Table 8. History granularity and relational representations. Accuracy measures pairwise ordering of the evaluated historical deletion effects.

Measurement / representation	**Value**
History granularity	
Median within-state Step-harm range	0.312
Median within-state Block-harm range	0.028
Unsafe parent Steps containing a safe Block	71.400%
Controls: pair accuracy at harm margin 0.020	
Current state only	50.000%
Surface features	64.750%
State relations: pair accuracy at harm margin 0.020	
Pre-action relation with matched state	70.920%
Pre-action relation with mismatched state	62.020%
Full-context relation	66.390%
Attention mass, layer 20	71.310%
Combined relations, layers 12 and 16	**72.130%**

Figure 3 examines whether historical relevance is better represented relative to the current decision state than from the historical Block alone. At the primary harm margin, the matched pre-action

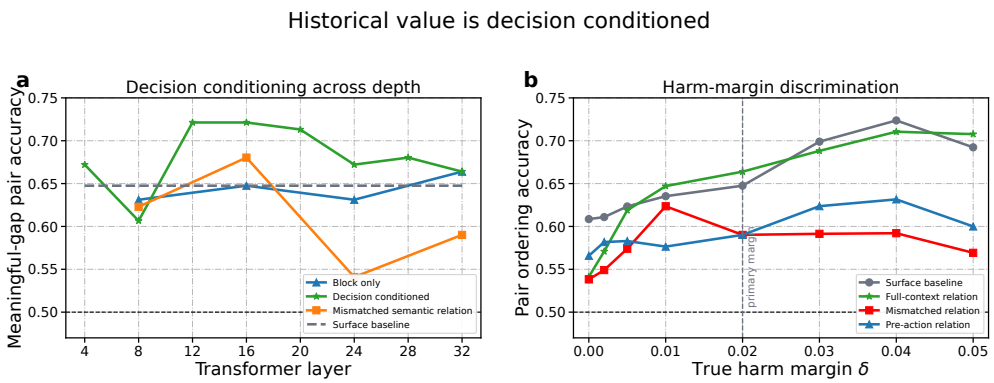


Figure 3. Decision-conditioned historical relevance. (a) Meaningful-gap pair accuracy across representation depth for decision-conditioned, Block-only, and mismatched-state relations, with the surface-feature baseline for reference. (b) Pair-ordering accuracy as the true harm margin varies; the dashed line marks the primary evaluation margin.

relation reaches 70.920% pair accuracy, compared with 62.020% when the same candidate is paired with a mismatched state. The best combined relation reaches 72.130%. The layerwise view further shows where this decision-conditioned signal emerges across representation depth, while the margin sweep verifies that the comparison is not specific to a single ordering threshold. Together, these results support representing historical content through its relation to the current state rather than assigning a fixed state-independent importance score.

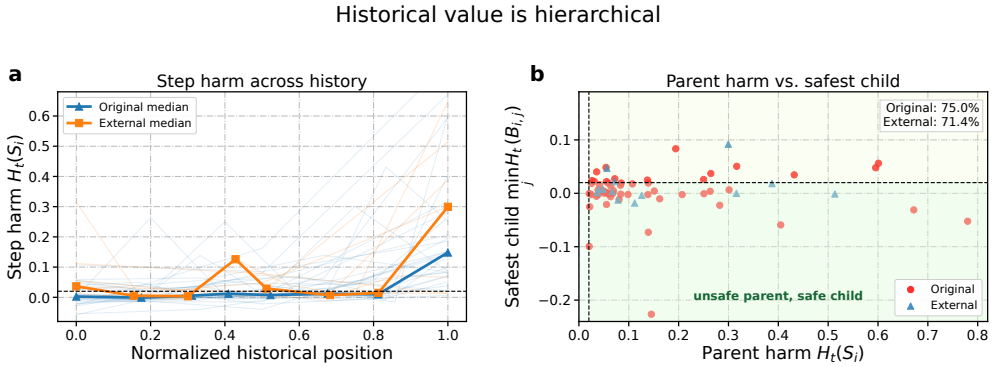


Figure 4. History granularity analysis. (a) Step harm across normalized historical position. (b) Parent-Step harm against the lowest-harm constituent Block; the unsafe-parent/safe-child region shows why Block-level actions can preserve useful parts of an otherwise unsafe Step.

Figure 4 motivates the Block-level action space used by DRSR. Step-level deletion aggregates heterogeneous assistant content and tool exchanges into a single removal decision, producing a substantially wider within-state harm range than Block-level deletion (0.312 versus 0.028). Moreover, 71.400% of unsafe parent Steps contain at least one constituent Block that is individually safe to remove. The comparison between parent and child Blocks in panel (b) therefore shows why deleting an entire Step can discard useful content unnecessarily: a coarse unit may be unsafe even when a finer-grained subcomponent remains removable.

Figure 5 extends the main-text cardinality analysis. Within the confirmatory range, mean absolute non-compositionality increases from 0 for singleton deletion to 0.008, 0.016, and 0.027 for deletion sets of size two, three, and four. The discrepancy between independently measured singleton effects and actual joint deletion therefore grows as more Blocks are removed together. At cardinality four, its mean magnitude already exceeds the $\delta_H = 0.020$ harm threshold used to define unsafe supervision. The additional larger-set and per-domain panels are retained as exploratory diagnostics and are not used to define the deployed policy.

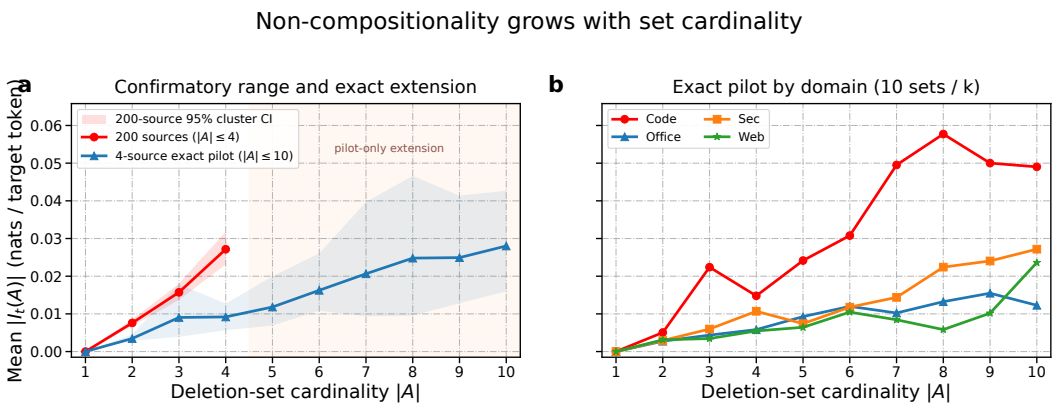


Figure 5. Full non-compositionality analysis. The main paper uses only the 200-source confirmatory range $|A| \leq 4$; this appendix figure retains the exploratory exact extension to larger sets and the per-domain pilot.

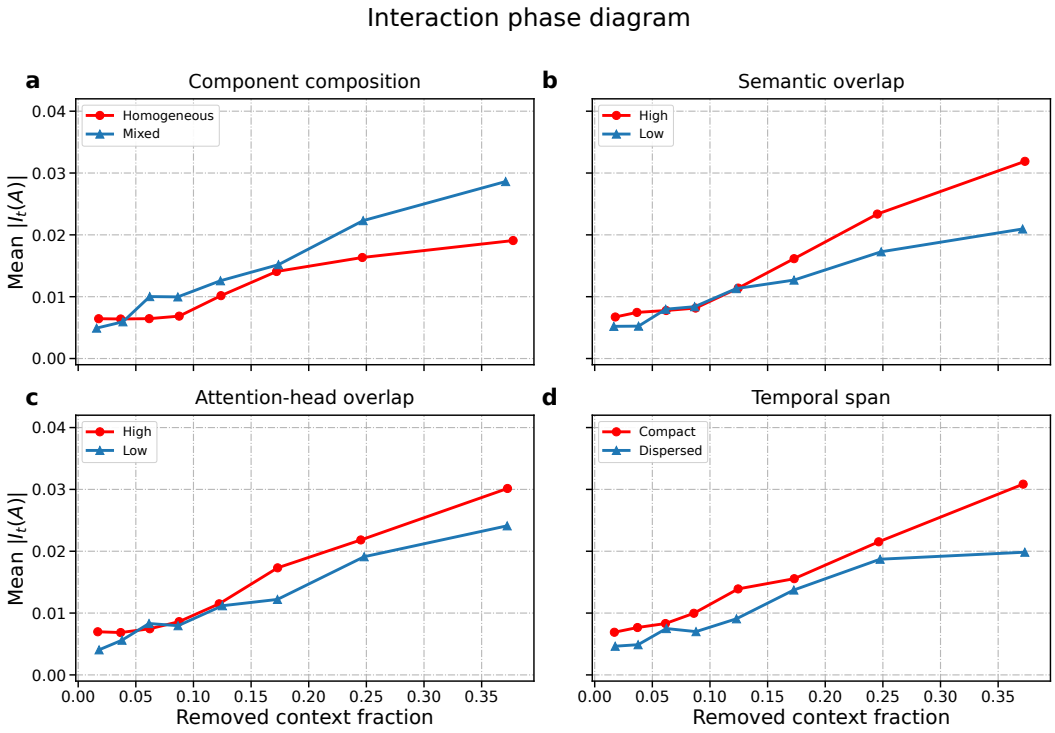


Figure 6. Interaction summaries by component composition, semantic overlap, attention-head overlap, and temporal span.

F.1 Composition, Risk Scores, and Attention

Figure 6 decomposes the measured interaction residual along several observable properties of a deletion set. The panels distinguish component composition, semantic overlap, attention-head overlap, and temporal separation, allowing the non-compositional effect to be inspected beyond set cardinality alone. These analyses are diagnostic: they test whether joint harm is associated with multiple forms of relation among historical Blocks, rather than defining hand-written deletion rules. DRSR therefore learns from the measured set-level labels instead of thresholding any individual overlap or distance statistic.

Figure 7 shows the distribution of the learned unsafe score on validation and external examples. The dashed line marks the frozen deployment threshold τ, making explicit how candidate sets are partitioned by the online risk gate. The figure characterizes score distributions for the labeled deletion

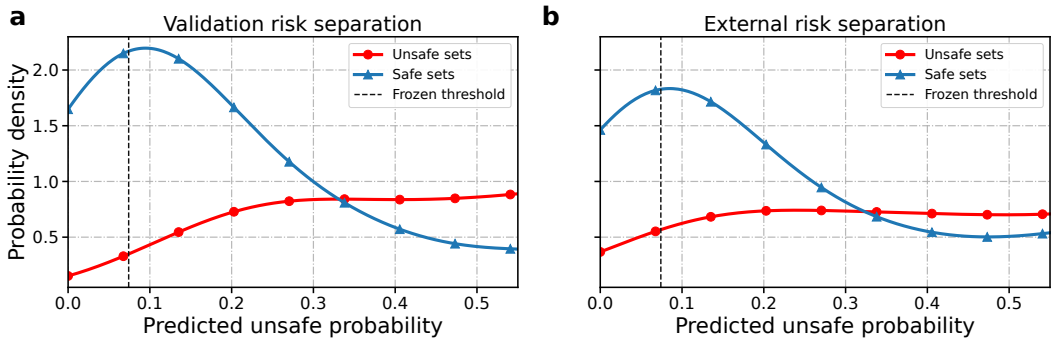


Figure 7. Unsafe-score distributions for validation and external sets. The dashed line marks the deployment threshold; score distributions summarize separation between the labeled deletion sets.

sets: the unsafe output supplies the online decision score, and group separation summarizes how that score distinguishes the measured next-output harm events across validation and external examples.

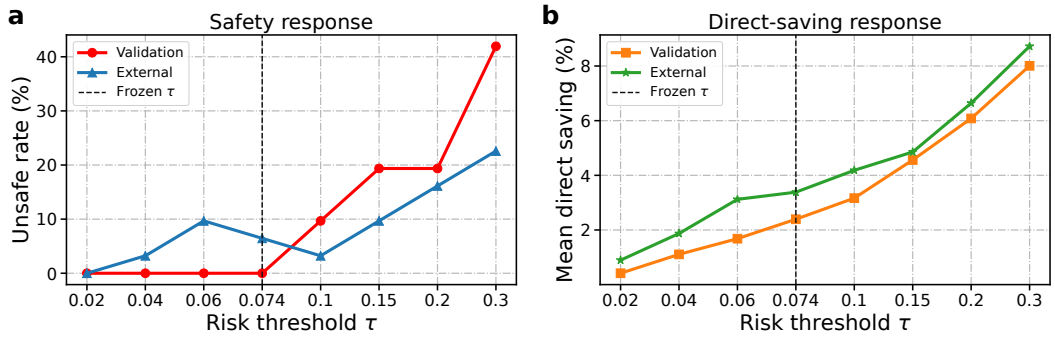


Figure 8. Risk-gate operating curve. Left: empirical unsafe rate. Right: mean direct removal. The dashed line marks the frozen deployment threshold.

Figure 8 visualizes the operating behavior of the risk gate as its threshold changes. The left panel tracks empirical unsafe rate, while the right panel tracks mean direct removal, exposing the trade-off between conservative selection and more aggressive compression. The dashed line marks the threshold selected on validation and subsequently frozen for deployment. Thus, the online threshold is chosen as a policy operating point, rather than being identified with the offline harm threshold δ_H.

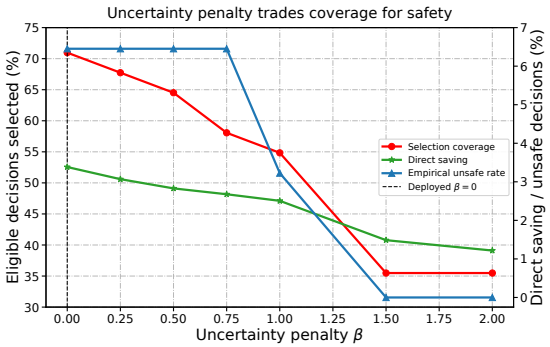


Figure 9. Selection coverage, direct saving, and empirical unsafe rate across the uncertainty coefficient β. The deployed setting is $\beta = 0$.

Figure 9 reports sensitivity to the ensemble uncertainty coefficient β. Increasing β changes the deployment score from the ensemble mean toward a more uncertainty-penalized criterion, which can alter selection coverage, direct removal, and empirical unsafe rate. The validation search selects $\beta = 0$ for the deployed configuration, so the final policy uses the ensemble-mean unsafe score without an additional standard-deviation penalty. This analysis reports the effect of the uncertainty term across the evaluated coefficient settings.

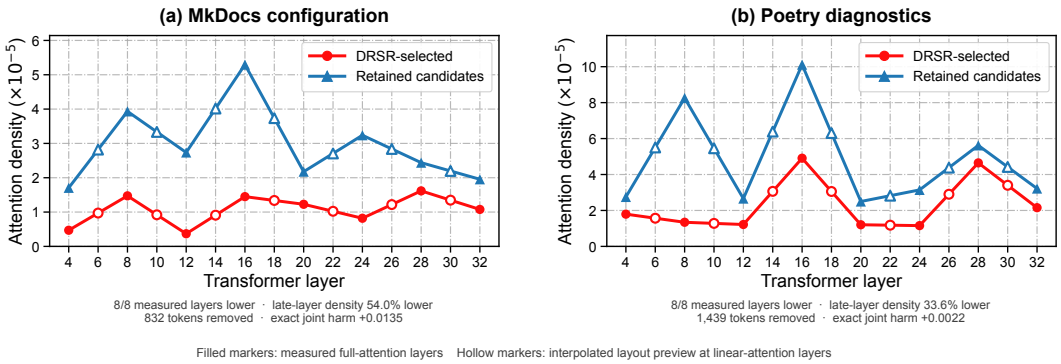


Figure 10. Attention density for selected and retained candidates in two examples. Filled markers are measured full-attention layers; hollow markers are interpolated layout points. Deletion decisions use the full learned set-risk representation.

Figure 10 provides two qualitative examples of attention density across representation depth for candidates that are selected for deletion and candidates that are retained. Filled markers correspond to measured full-attention layers, whereas hollow markers are interpolation points used only for visualization. The examples illustrate why attention is treated as one relational feature rather than a standalone pruning criterion: the deletion decision is produced by the learned set-risk model together with state, retained-context, and interaction information, not by applying a fixed low-attention threshold.

G Optimization and Metric Definitions

G.1 Harm Standardization and Ranking

Let μ_H and σ_H be computed on training labels only. Regression uses

$$\widetilde{H}_t(A) = \frac{H_t(A) - \mu_H}{\max(\sigma_H, 0.02)}. \tag{35}$$

Predictions are restored to the original harm scale for interpretation. Ranking pairs have the same source trajectory and deletion cardinality. For examples $a = (t, A)$ and $b = (t', B)$, define $\Delta_{ab} = H_t(A) - H_{t'}(B)$. Pairs with $|\Delta_{ab}| < 10^{-5}$ are omitted. With

$$\omega_{ab} = 0.1 + 0.9 \min\left(\frac{|\Delta_{ab}|}{0.02}, 1\right), \tag{36}$$

training minimizes the weighted pairwise loss

$$\mathcal{L}_{\text{rank}} = \mathbb{E}_{a,b}\left[\omega_{ab}\,\text{softplus}\left(-\text{sign}(\Delta_{ab})(\widehat{H}_a - \widehat{H}_b)\right)\right]. \tag{37}$$

Trajectory-grouped training pairs and within-state evaluation pairs are different grouping rules. Both preserve the intended source dependence; their metrics retain these distinct grouping conventions.

G.2 Offline Prediction Metrics

For a set of evaluated pairs $\mathcal{P}$ satisfying the specified harm margin, pair accuracy is the fraction with correctly ordered predicted harm. Within-state Spearman correlation compares the predicted and measured set orders at a common decision state and cardinality.

The unsafe target is $u_t(A)$ from Eq. (4). Brier score averages $(\widehat{p}_t(A) - u_t(A))^2$, while AUROC measures the ability to rank unsafe events above safe events. Both evaluate the defined unsafe event. Multiple sets from one trajectory are dependent observations; uncertainty estimates require grouping at the trajectory level rather than treating every deletion row as independent.

Table 9. Eval40 token accounting for comparator methods. Totals cover 40 tasks.

Method	Total (M)	Per task (M)
DeepSeek-V4-Flash	75.50	1.887
Sliding Window ($K = 5$)	118.08	2.952
Sliding Window ($K = 10$)	79.39	1.985
Sliding Window ($K = 20$)	80.82	2.020
Periodic Summary ($n = 3$)	88.57	2.214
Periodic Summary ($n = 5$)	94.85	2.371
PACE	44.12	1.103
LLMLingua-2	96.18	2.405
SelfCompact	62.70	1.567
ACON-Core	56.13	1.403
Self-GC	66.21	1.655
LRE	91.64	2.291
CoMem	**30.64**	**0.766**
SAM	53.85	1.346
SWE-Pruner	60.07	1.502
Sculptor	51.75	1.294
ACM	94.56	2.364
DRSR	**48.43**	1.211

G.3 Direct and Complete-Task Token Accounting

For task j and requests t, direct deletion is

$$D_{\text{direct},j} = \sum_{t} \left[\text{Tok}(C_{j,t}) - \text{Tok}(D_{A_{j,t}}(C_{j,t})) \right]. \tag{38}$$

Complete-task processed tokens are instead $T_j = \sum_r (I_{j,r} + O_{j,r})$. Cache-read and cache-creation counts, when reported as input components, are already included in $I_{j,r}$ and are not added again. For benchmark means,

$$\Delta_T(\%) = 100 \left(\frac{\bar{T}_{\text{DRSR}}}{\bar{T}_{\text{Base}}} - 1 \right). \tag{39}$$

A negative token change denotes a decrease. A positive "reduction" percentage would use the opposite sign. Monetary cost requires weighting these token counts by the applicable prices for input, output, and cache components.

Changes in action paths, retries, termination, and request counts can separate direct removal from the observed difference between two full rollouts. One may define a residual $\Delta T_{\text{rollout}}$ so that

$$T_{\text{Base}} - T_{\text{DRSR}} = D_{\text{direct}} + \Delta T_{\text{rollout}}, \tag{40}$$

where the residual records the remaining rollout-level token difference. Local representation computation is accounted for separately from the API token totals used in the complete-task comparison.

H Comparator Token Accounting

Table 9 reports the Eval40 token quantities used in Table 4. For methods with an auxiliary summarization model, its reported input and output tokens are included in the total. Totals are shown in millions to keep the appendix compact; the per-task column uses the same accounting rule as the main comparison.

I Detailed Task and Token Summaries

This section provides the domain-level quantities underlying the Eval40 ablation and the Full260 token summaries. To improve readability, token magnitudes are reported in millions (M) or thousands (K) rather than as long raw counts. This changes only presentation, not the underlying values.

Table 10. Per-domain Eval40 ablations. T is mean total model tokens per task in millions; Direct is directly removed tokens per task in thousands.

Domain	Reward	T (M)	Direct (K)
DRSR			
Code	0.856	1.216	46.6
Office	0.788	0.724	6.8
Security	0.700	1.616	37.6
Web	0.833	1.288	9.7
w/o retained			
Code	0.722	0.716	22.7
Office	0.779	0.266	0.0
Security	0.698	1.254	17.7
Web	0.800	1.731	10.1
w/o pair pooling			
Code	0.756	0.567	40.3
Office	0.678	0.606	2.3
Security	0.796	1.094	52.1
Web	0.796	2.535	34.1
w/o abstention			
Code	0.833	1.205	46.9
Office	0.801	0.447	1.6
Security	0.573	0.963	45.7
Web	0.800	1.553	26.3

Table 11. Equal-domain token decomposition on Eval40. Input, cache, and total columns are in millions of tokens per task; output is in thousands.

Variant	Input (M)	Output (K)	Cache read (M)	Cache create (M)	Total (M)
DRSR	1.188	22.7	1.067	0.121	1.211
w/o retained	0.970	21.3	0.873	0.097	0.992
w/o pair pooling	1.178	22.2	1.052	0.126	1.201
w/o abstention	1.018	23.7	0.872	0.147	1.042

I.1 Domain-Level Ablations

Table 10 preserves the four-domain Eval40 ablation results. Each domain has equal weight in the Eval40 summary; for example, the full DRSR reward is

$$(0.856 + 0.788 + 0.700 + 0.833)/4 \approx 0.794. \tag{41}$$

The ablations are visibly domain dependent. Removing retained-context features reduces direct deletion in several domains but still lowers overall reward; removing pair pooling produces the largest direct deletion in Security; and removing abstention is particularly harmful in Security. These domain-level patterns are consistent with the main-text conclusion that the three components affect different parts of the pruning decision rather than acting as redundant copies of the same mechanism.

I.2 Token Components

Table 11 gives the equal-domain token decomposition used by the Eval40 ablation summary. Input and cache quantities are shown in millions of tokens per task, while output is shown in thousands because it is an order of magnitude smaller. Cache read and cache creation are components of input and are therefore not added to the total a second time.

The decomposition shows that the observed token differences are driven mainly by input-side processing rather than output length. This is expected for history pruning: changing what remains in the active context primarily changes how much prior context is repeatedly processed. The online controller itself uses two local representation forwards at eligible decisions; that local workload is separate from the API token totals reported here.

For completeness, Table 12 retains the per-domain token components in the same compact units.

Table 12. Per-domain token components for Eval40 ablations. Input and cache columns are in millions of tokens per task; output is in thousands.

Domain	Input (M)	Output (K)	Cache read (M)	Cache create (M)
DRSR				
Code	1.205	11.2	1.052	0.153
Office	0.702	22.0	0.642	0.060
Security	1.589	26.4	1.429	0.161
Web	1.256	31.0	1.146	0.110
w/o retained				
Code	0.706	10.1	0.605	0.101
Office	0.247	19.4	0.196	0.050
Security	1.232	21.8	1.095	0.137
Web	1.697	33.9	1.596	0.100
w/o pair pooling				
Code	0.558	9.0	0.463	0.095
Office	0.583	23.6	0.519	0.064
Security	1.072	21.7	0.881	0.191
Web	2.501	34.5	2.345	0.156
w/o abstention				
Code	1.194	11.4	1.014	0.179
Office	0.425	22.1	0.349	0.076
Security	0.940	23.2	0.769	0.171
Web	1.515	37.9	1.355	0.160

Table 13. Full260 reward and token accounting by domain. Token columns are per-task means in millions; Direct is per-task direct removal in thousands.

Domain	N	Base R	DRSR R	Base T (M)	DRSR T (M)	ΔT	Direct (K)
Code	80	0.770	0.858	1.370	1.235	-9.91%	46.9
Office	50	0.818	0.824	1.198	0.997	-16.82%	6.8
Security	60	0.478	0.756	5.977	4.569	-23.55%	31.0
Web	70	0.721	0.760	2.429	1.858	-23.52%	38.2
Overall	**260**	**0.699**	**0.802**	**2.685**	**2.126**	-20.82%	**33.2**

I.3 Full260 Token Accounting

Table 13 reports the Full260 reward and token quantities in the same compact notation. T is the complete-task model-token mean per task, whereas Direct measures only the tokens removed from the rewritten request at the pruning decision. The two quantities capture different effects because pruning decisions can also change later action paths, request counts, termination, and subsequent context growth.

The Full260 table makes the domain dependence of compression explicit. Security has by far the largest baseline context and also shows the largest absolute reward improvement, while Office changes only modestly. DRSR reduces processed tokens in all four domains, but the amount of direct same-request removal is not identical to the complete-rollout token difference.

J Deployment and Supervision Details

J.1 Training and Deployment Information Boundary

Offline supervision may use quantities that are unavailable when the online pruning decision is made. Table 14 makes this separation explicit. The realized next output y_t and exact deletion harm are used only to construct supervision. The deployed scorer receives only the current causal prefix, candidate features, and the proposed deletion mask.

Table 14. Information boundary for DRSR. "Label only" means that a quantity may construct supervision but is not an online feature.

Quantity	Offline training	Online selection	Post-hoc audit
Current query and observed history	Yes	Yes	Yes
Pre-action hidden states / attention	Yes	Yes	Yes
Candidate mask and token saving	Yes	Yes	Yes
Realized next output y_t	Label only	No	Yes
Exact joint deletion harm $H_t(A)$	Supervision	No	Optional remeasurement
Task reward	Not a feature	No	System evaluation
Counterfactual deletion forward	Label generation	0	Optional audit

J.2 Exact Online Risk Aggregation and Feasible Set

For ensemble member s, let $\widehat{H}_t^{(s)}(A)$ and $\widehat{p}_t^{(s)}(A)$ denote the predicted harm and unsafe score. DRSR aggregates the three members as

$$\bar{H}_t(A) = \frac{1}{3}\sum_s \widehat{H}_t^{(s)}(A), \qquad U_t(A) = \bar{p}_t(A) + \beta\ \mathrm{Std}_s\left[\widehat{p}_t^{(s)}(A)\right]. \tag{42}$$

The deployed setting uses $\beta = 0$, so $U_t(A)$ is the ensemble-mean unsafe score. The exact feasible set is

$$\begin{aligned}\mathcal{F}_t = \{A \subseteq E_t : 1 \le |A| \le K, \quad 0 < w_t(A) \le q_{\max}\,\mathrm{Tok}(C_t),\\ A \cap \mathrm{Recent2} = \varnothing, \quad U_t(A) \le \tau, \quad \mathrm{ProtocolSafe}(D_A(C_t))\}.\end{aligned} \tag{43}$$

The main-text set $\mathcal{A}_t$ contains the same structural constraints without the risk condition. Equation (11) then chooses the largest-saving member that also satisfies $U_t(A) \le \tau$. If several sets remove the same number of tokens, the policy prefers lower unsafe score and then lower predicted harm.

With at most four candidates and $K = 3$, the number of nonempty subsets considered online is

$$\sum_{k=1}^{3}\binom{4}{k} = 14. \tag{44}$$

All candidate subsets are batch-scored by the lightweight network using shared features extracted from the intact causal prefix.

J.3 Online Computational Cost

The online controller estimates deletion risk from shared intact-prefix features instead of running a language model per candidate. At an eligible decision, the implementation performs two local representation passes over the intact causal prefix: one pass extracts attention and hidden-state relations, and the second provides the projected semantic representation used by the feature pipeline. Candidate-specific features are then assembled from these intact-prefix representations.

With $n \le 4$ candidates and $K = 3$, at most

$$\sum_{k=1}^{3}\binom{n}{k} \le 14$$

candidate sets are formed. Their 451-dimensional set representations are scored together by the lightweight scorer ensemble. The number of online candidate-specific counterfactual deletion forwards is therefore exactly zero. Only the single deletion set selected by the policy is applied to the request that is passed to the acting model.

This distinction is important for both meaning and cost. The offline label $H_t(A)$ compares the likelihood of one fixed recorded output under intact and edited contexts. Online candidate scoring instead uses shared pre-action features and the lightweight network, transferring the offline supervision to deployment while avoiding separate model generations for each proposed deletion.

J.4 Protocol-Preserving Deletion

The implementation enforces the following invariants at every eligible decision: (i) the two latest complete Steps are protected; (ii) a tool call and its matched result are atomic; (iii) no rewrite may create dangling tool identifiers or an empty assistant message; (iv) actual removal on the full serialized request is at most 20%; (v) only subsets passing the frozen risk gate are eligible; (vi) if no subset is feasible, the complete request is retained; (vii) alignment or protocol-validation failure also returns the request unchanged; and (viii) online selection never reads a future target and never executes a counterfactual deletion forward. These checks are logically separate from the learned risk score.

J.5 Exact Set-Level Supervision

For a selected decision state, candidate construction is frozen before any deletion outcome is observed. Three candidates generate all seven nonempty subsets, while four candidates generate all fifteen. Every subset is jointly deleted from the original intact context, the request is reserialized, and the same recorded next output is teacher-forced under the full and deleted contexts. Joint labels are therefore measurements of the actual joint rewrite, not sums of singleton labels.

All states and deletion sets originating from one source trajectory stay in the same 80:10:10 train/validation/test partition. This prevents deletion rows from a common trajectory from crossing splits. The supervision corpus contains 566 external trajectories and 1,698 target states, with 24,126 set-level labels. The cardinality breakdown is 6,624 singletons, 9,684 pairs, 6,288 triples, and 1,530 quadruples.

Table 15. Set-level supervision by deletion cardinality.

Deletion size	Labelled sets	Mean interaction
1	6,624	0
2	9,684	0.008
3	6,288	0.016
4	1,530	0.027

The interaction statistic summarizes measured joint-deletion effects. Labels are obtained by rescoring the same recorded output after each actual set-level rewrite.

J.6 Validation Selection and Frozen Deployment Policy

Validation searches the maximum online deletion cardinality, an optional ensemble uncertainty penalty, and the deployment threshold. A candidate policy is admissible only when its empirical validation unsafe rate is at most 0.10 and its mean selected-set exact harm is at most 0.02. Among admissible policies, validation maximizes the mean fraction of the full request removed, then prefers lower unsafe rate and lower harm. Test labels and WorkBuddyBench outcomes are not used for this selection.

The harm-label threshold $\delta_H = 0.02$ and the deployment threshold $\tau = 0.0740822119017442$ act on different quantities. The first thresholds measured teacher-forced NLL change to create an offline binary target. The second thresholds the scorer's predicted unsafe score during online selection. Their values reflect separate label-construction and deployment-selection criteria.

Table 16. Validation search space and frozen deployment values.

Item	Validation search / rule	Frozen value
Maximum cardinality K	$\{2, 3\}$	3
Uncertainty coefficient β	$\{0, 0.25, 0.5, 1, 2\}$	0
Risk threshold τ	observed validation scores	0.0740822119017442
Empirical unsafe-rate constraint	≤ 0.100	enforced in selection
Mean selected exact-harm constraint	≤ 0.020	enforced in selection
Full-request removal cap	fixed	20%
Protected history	fixed	Recent2
Fallback	fixed	strict abstention

K EVAL40 TASK MANIFEST

Eval40 contains ten tasks from each of Code, Office, Security, and Web. The task identifiers below are the frozen manifest used by the reported context-management comparison. Listing the identifiers makes the 40-task comparison reproducible and distinguishes it from other 40-task pilots.

Domain	#	Task ID
Code	1	`api_contract-hard-markup_errors`
Code	2	`api_contract-hard-openapi_params`
Code	3	`api_contract-hard-token_errors`
Code	4	`api_contract-hard-validation_errors`
Code	5	`bug_fix-easy-a_crash_in_local`
Code	6	`bug_fix-easy-filtered_relation_queryset_arg`
Code	7	`bug_fix-easy-invalid_filterwarnings_regex_error`
Code	8	`bug_fix-medium-error_key_uses_data_key`
Code	9	`bug_fix-medium-errors_from_earlier_indices`
Code	10	`bug_fix-medium-incorrect_linenos_on_fstring`
Office	11	`analyst-forecast-extract-L3-018`
Office	12	`api-usage-explain-cli-l3-001`
Office	13	`board-material-update-timeline-excel`
Office	14	`calendar-dida-sync-state`
Office	15	`channel-period-compare-L4-017`
Office	16	`cloudagent-sdk-doc-validation-report`
Office	17	`contract-extract-L3-014`
Office	18	`cross-week-dashboard-migration`
Office	19	`crypto-backtest-chain-L4-002`
Office	20	`daily-creation-checkpoint-recovery`
Security	21	`agent-to-agent-injection-hard-multistep`
Security	22	`apt-multi-source-correlation-hard-multistep`
Security	23	`bb-bin-dns-parse-010`
Security	24	`bb-bin-firmware-audit-007`
Security	25	`bb-bin-format-log-004`
Security	26	`bb-bin-int-length-005`
Security	27	`bb-bin-ipc-cache-001`
Security	28	`bb-bin-media-parse-008`
Security	29	`bb-bin-oob-read-003`
Security	30	`bb-bin-parse-crash-006`
Web	31	`animated-explainer-L3-028`
Web	32	`atmosphere-game-L4-035`
Web	33	`blog-editor-draft-recovery-L4-059`
Web	34	`browser-clipper-extension-L4-005`
Web	35	`canvas-webgl-scene-L4-026`
Web	36	`chart-generation-L2-025`
Web	37	`checkout-incident-analysis-L4-049`
Web	38	`city-article-theme-variants-L4-066`
Web	39	`claims-drawer-state-review-report-L3-071`

Domain	#	Task ID
Web	40	`cohort-retention-dashboard-L4-054`