WHO PAYS FOR PRIVACY? CONTROLLING UTILITY COSTS IN ADAPTIVE RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Training retrieval policies to hide task identity can improve average evidence recall while removing useful evidence from individual inputs. Population averages conceal this cost because gains on other inputs offset the losses. We introduce policy-induced evidence cost: the positive decrease in expected support recall relative to a matched utility-only policy. This definition separates evidence lost through a policy change from baseline task difficulty and makes that loss directly optimizable. We instantiate the objective in two-step adaptive retrieval with an upper-quarter penalty that prioritizes the most affected training questions. Exact expectations over both adaptive actions provide the cost signal without action-sampling noise, while inference retains the same controller and retrieval budget. On 6,000 held-out questions from 2WikiMultiHopQA, the penalty reduces group-level evidence-loss frequency by 31.9–48.9% and mean loss among the worst 5% of questions by 24.5–57.9% relative to adversarial privacy training across sparse, MiniLM, and Qwen3 retrieval. All three settings retain positive task-recovery reductions relative to Utility-only under the evaluated sketch attackers; richer raw-ID attacks reveal a measurable privacy price of repair in the sparse settings. A scale-matched positive-mean penalty achieves similar evidence repair, showing that the benefit extends across aggregation rules applied to the same cost definition. These findings establish policy-induced evidence cost as a practical training target for controlling who bears the utility cost of privacy.

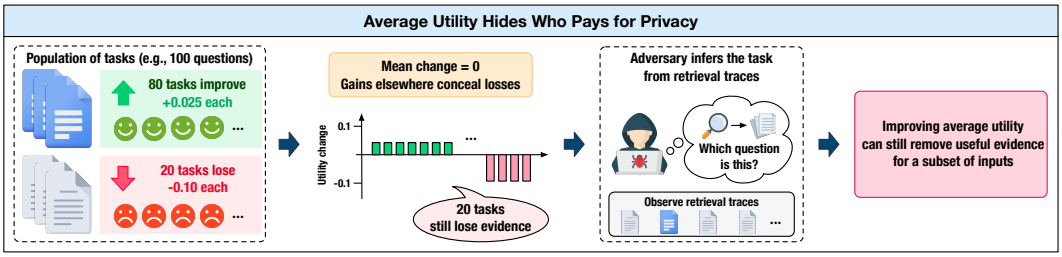


Figure 1: Average utility can conceal the evidence cost of privacy training. In this illustrative population, gains of 0.025 on 80 tasks offset losses of 0.10 on 20 tasks: mean utility is unchanged, but a subset loses evidence. An observer infers task identity from retrieval traces, motivating privacy training whose utility costs must be measured at the affected inputs. The chart is schematic; the numerical example is not an experimental result.

1 INTRODUCTION

An auxiliary objective can improve a learned policy's mean utility while removing useful evidence from individual inputs, because gains elsewhere offset their losses. Controlling this burden requires distinguishing evidence lost through a policy change from evidence that was already difficult to retrieve. The relevant learning signal is therefore the utility decrease relative to attainable task-only behavior on the same input.

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.

Adaptive retrieval makes this mismatch concrete. Retrieval supplies evidence to language models, and adaptive systems use earlier observations to choose subsequent queries (Lewis et al., 2020; Trivedi et al., 2023; Asai et al., 2024). These decisions expose access patterns that can support task inference even when other information is concealed (Cash et al., 2015; Oya & Kerschbaum, 2021; Nie et al., 2024). Training a policy to suppress inference changes both the observed trace and the evidence available to the client. Privacy–utility learning typically measures retention through population objectives (Makhdoumi et al., 2014; Bertran et al., 2019; Xiao et al., 2020); these averages do not identify which tasks bear the cost.

We define this cost as the positive loss in expected evidence recall relative to a matched utility-only policy. Clipping gains at zero prevents cancellation across inputs, while the reference distinguishes induced cost from difficulty shared by both policies. In adaptive retrieval, measuring this change also requires accounting for how the first retrieval affects the second decision. We compute expected recall over the joint two-step paths, obtaining an exact cost signal for training. An upper-tail penalty then prioritizes the largest quarter of current costs, using established tail-sensitive optimization (Rockafellar & Uryasev, 2000; Fan et al., 2017). Deployment retains the same controller and retrieval budget.

On 2,000 held-out 2WikiMultiHopQA groups containing 6,000 questions (Ho et al., 2020), adversarial privacy training improves mean sparse-retrieval recall but reduces it by more than 0.02 in 273 groups. Cost control lowers this count to 186, with 99 originally affected groups leaving the loss set and 12 entering it. Evidence repair persists across two dense retrievers, a recurrent controller, and MuSiQue. The 2WikiMultiHopQA policies retain positive privacy benefit relative to Utility-only, and direct policy comparisons quantify the privacy price of repair.

The contribution is a task-level formulation of policy-induced utility cost and its exact adaptive-retrieval instantiation. A scale-matched positive-mean penalty achieves similar evidence repair, separating the cost definition from the choice of aggregator. The results support optimizing induced cost directly rather than inferring its distribution from the population trade-off.

2 Policy-Induced Utility Cost Hidden by Averages

Consider a policy π trained with an auxiliary objective and an attainable task-only reference π_0. For an input x, let $U_\pi(x)$ denote task utility under the corresponding policy. We define the utility cost induced by changing the policy as

$$c_\pi(x) = [U_{\pi_0}(x) - U_\pi(x)]_+, \qquad [z]_+ = \max(z, 0). \tag{1}$$

The positive part assigns zero cost when the new policy matches or improves upon its reference, including inputs with equally low utility under both policies. We instantiate U_π as evidence recall and the auxiliary objective as protection against task recovery from retrieval traces.

2.1 Adaptive retrieval and task recovery

A client receives a question x and constructs a bounded set of candidate queries. A policy π selects a first query, observes the retrieved documents, and then selects a second query. The server-side observation considered here is the ordered document-access trace, together with access overlap and query repetition. The private question and client state are not provided to the attacker. The observation model excludes query plaintext, network timing, and answer text. It assumes the observer has no query-channel information; the experiments do not implement a separate query-protection mechanism. Reporting attackers encode the measured access behavior in a 258-dimensional sketch.

We compare task identities within groups of three questions sharing the same full annotated support-title set. Denote this set by $C(x)$. Knowing $C(x)$ alone leaves all three identities equally likely, with recovery rate $1/3$. An attacker receives the evidence-set representation and the access trace and ranks the three possible questions. Privacy measures whether the observer can distinguish the task; utility measures whether the client still retrieves its evidence.

For this instantiation, $U_\pi(x)$ is the expected fraction of annotated support documents retrieved by policy π, where the expectation is over both controller actions. The matched reference π_0 is trained only for utility with the same architecture, retriever, training population, and initialization seed.

Utility-only permits repeated queries and omits the aggregate constraints used by the other policies. Equation 1 measures the entire reference-relative policy change, including these differences, rather than isolating the adversarial training term. Comparisons with the existing No-repeat and Alignment references assess this attribution sensitivity.

2.2 Why preserving the mean leaves local cost unconstrained

Consider an illustrative population of 100 tasks. A policy loses 0.10 recall on 20 tasks and gains 0.025 on the other 80. Its mean recall is unchanged, yet one fifth of the tasks have less evidence. This is a consequence of cancellation, rather than uncertainty in measuring the mean. Writing the signed change as gains minus induced costs makes the distinction explicit:

$$\mathbb{E}[U_\pi - U_{\pi_0}] = \mathbb{E}[(U_\pi - U_{\pi_0})_+] - \mathbb{E}[c_\pi]. \tag{2}$$

A nonnegative left-hand side can coexist with substantial positive cost on the right. An average-retention constraint therefore controls a different quantity from the losses caused at individual inputs.

The distribution of c_π also matters beyond its mean. Two policies can impose the same total positive cost while concentrating it on different tasks. Once the cost is explicit, an aggregator determines how training priority is allocated across those tasks. The next section develops an exact cost signal for adaptive retrieval and instantiates this allocation with an upper-tail objective.

3 Controlling Policy-Induced Evidence Cost

The method separates the quantity to control from the rule used to allocate training effort. We first compute positive reference-relative evidence cost over adaptive retrieval paths, then instantiate cost control with an upper-quarter penalty. Section 5 tests whether evidence repair depends on this aggregation rule.

3.1 Exact task utility for adaptive actions

For a question x, let $\mathcal{A}(x)$ be the candidate-query set and $D_x(a)$ the documents returned by action a. After the first action a, the client forms a state $s_x(a)$ from the retrieved document embeddings. The joint action probability is

$$p_\pi(a, b \mid x) = \pi(a \mid x)\, \pi(b \mid x, s_x(a)). \tag{3}$$

The second factor uses the state produced by each particular first action. For no-repeat policies, the second distribution excludes a.

With the retrieved documents cached for the candidate queries, expected evidence recall is

$$U_\pi(x) = \sum_{a,b \in \mathcal{A}(x)} p_\pi(a, b \mid x) \frac{|(D_x(a) \cup D_x(b)) \cap C(x)|}{|C(x)|}. \tag{4}$$

The retrieval sets are fixed and the probabilities are differentiable functions of the policy parameters. Enumerating at most eight candidate queries requires at most 64 ordered pairs per question. The expectation removes action-sampling noise from the utility cost, including its ordering in the tail. It leaves the discrete, observed traces available for privacy evaluation.

Credit assignment across the two decisions. A first query determines both the collected documents and the state guiding the second query. Equation 4 retains both effects: the union counts each support document once, while the conditional probability accounts for how the first observation redirects retrieval. The resulting gradient favors complementary retrieval histories, because a second query contributes utility when it supplies evidence missing from the first result set.

3.2 Allocate updates by positive cost

We first train π_0 for utility and store $U_{\pi_0}(x)$ on the training questions. These values are held fixed. For a batch of N training questions, compute the costs in Equation 1 and sort them in descending

order as $c_{(1)}, \ldots, c_{(N)}$. Our allocation rule averages the largest current costs:

$$\mathcal{T}_\alpha(\pi) = \frac{1}{k} \sum_{i=1}^{k} c_{(i)}, \qquad k = \lceil \alpha N \rceil. \tag{5}$$

We use $\alpha = 0.25$, so each update emphasizes the largest quarter of added training costs. The identity of those questions changes with the policy. The positive-part and top-k operations are piecewise differentiable; automatic differentiation follows their active branches, with implementation-selected generalized gradients at zero costs and ranking ties. For the full training population, this selects 450 of 1,800 question-level costs at each update.

Equation 5 is the empirical upper-tail average used in conditional-value-at-risk and top-k loss optimization (Rockafellar & Uryasev, 2000; Fan et al., 2017). The cost definition specifies what is charged to the policy change; the upper-tail operator allocates training pressure across questions.

Let $\mathcal{L}_{\text{base}}$ be the adversarial privacy objective, including its utility and alignment terms. The corrected controller minimizes

$$\mathcal{L}_{\text{tail}}(\pi) = \mathcal{L}_{\text{base}}(\pi) + \lambda \mathcal{T}_{0.25}(\pi). \tag{6}$$

The coefficient λ controls the emphasis on local evidence cost. We compare $\lambda = 1$ and 3 on development groups and use 3 for all test comparisons. This adds one training term and no new inference module.

How the cost term directs learning. Away from ranking ties and zero-cost boundaries, let S_α contain the selected top-k questions and let $S_\alpha^+ = \{i \in S_\alpha : c_i > 0\}$. Because the reference is fixed, the contribution of the penalty to the parameter gradient is

$$\nabla_\theta(\lambda \mathcal{T}_\alpha) = -\frac{\lambda}{k} \sum_{i \in S_\alpha^+} \nabla_\theta U_{\pi_\theta}(x_i). \tag{7}$$

The penalty increases expected-recall pressure on selected questions below their reference. A question leaves the active set when it reaches its reference or other questions incur larger costs, so training priority adapts to current losses. The base objectives continue to update the shared parameters over the full training population.

Because membership depends on the decrease from the reference, even a question with relatively high recall can receive additional training pressure when the policy has lost much of its attainable evidence. The penalty's scale depends jointly on λ and the selected fraction, motivating the scale-matched comparison in Section 5.

3.3 Privacy learning and controller execution

The base policy combines expected evidence recall, alignment of traces within a shared-support group, and adversarial task obfuscation. Alignment penalizes squared distances between trace representations and their group mean. Conditional adversaries receive the trace and support-set embedding and learn to distinguish the three questions. The policy is updated to increase their contrastive prediction loss, following the adversarial privacy–utility learning approach (Bertran et al., 2019; Xiao et al., 2020). This asks the controller to make task-dependent access patterns less distinguishable while continuing to collect evidence.

Privacy and alignment use straight-through sampled actions, with the Gumbel–Softmax estimator (Jang et al., 2017). Evidence utility and the tail penalty use the exact expectation in Equation 4. All non-reference policies share aggregate targets for evidence recall, action entropy, and effective action support, so the Tail-aware comparison isolates the added cost penalty under the same constraints.

The default controller projects question, candidate-query, and retrieved-state embeddings into a 192-dimensional space. Candidate scores are scaled inner products followed by a masked softmax. The recurrent variant replaces the additive state projection with a GRU cell that updates a question-initialized hidden state after the retrieved observation. Both controllers use frozen 384-dimensional MiniLM semantic features (Wang et al., 2020; Reimers & Gurevych, 2019).

At deployment, the controller selects a query, updates its state from the retrieved documents, and selects its second query. Gold support sets and the utility-only reference are used offline, so cost

control changes action selection without adding retrieval calls or controller parameters. Appendix B gives the complete objective and optimization settings.

4 Policy-Induced Cost in Adaptive Retrieval

We first test whether cost control repairs losses hidden by mean recall and quantify the associated privacy trade-off. We then examine whether the repair persists across retrieval backends, controller architectures, and a second dataset.

4.1 Matched evaluation on held-out support groups

We construct 2,900 groups from 2WikiMultiHopQA, with three distinct source questions per group and an identical full annotated support-title set within each group. We allocate 600 groups to training, 300 to development, and 2,000 to testing. Complete support sets, question identities, and normalized question texts are disjoint across these splits. The test population contains 6,000 questions; individual support titles may recur across groups.

All retrieval settings search the same corpus of 369,280 documents and use the same question-derived candidate queries. Each action returns up to five documents. We compare sparse BM25 retrieval (Robertson & Zaragoza, 2009) with dense retrieval using MiniLM and Qwen3-Embedding-0.6B (Wang et al., 2020; Zhang et al., 2025). Dense indexes use cosine similarity; controller features remain MiniLM-based in every setting. Appendix A specifies candidate construction and retrieval settings.

The five policies form a matched progression: Utility-only supplies the reference, No-repeat adds distinct-query selection and aggregate constraints, Alignment adds within-support-group trace alignment, and Adversarial adds conditional task obfuscation. Tail-aware adds Equation 5 to Adversarial. Thus the last comparison isolates the tail term within the same privacy-training recipe. The primary four-setting experiment trains each policy for 160 epochs with three controller seeds; learning rates, dimensions, and privacy weights are shared within each setting. A supplementary Sparse FF experiment adds five training seeds (Appendix H). Hyperparameters are selected on development groups.

4.2 Measure the auxiliary benefit and induced cost separately

For privacy, ridge regression, distance-weighted nearest neighbors, an MLP, and a GRU predict the question embedding from the trace and support-set embedding. Each attacker is fitted on training traces from the policy it evaluates. It ranks the three distinct question identities within each test group. We average accuracy over that group's questions and controller seeds, take the maximum over the four attackers and the $1/3$ evidence-only comparator, and then average ten fitting repetitions. The privacy benefit is the decrease in this recovery score relative to the matched Utility-only policy. Positive values mean less task recovery. Appendix C gives the aggregation order and attacker specifications.

For task utility, we average exact expected evidence recall over the three trained controllers and three questions in each group. A group incurs local evidence loss if this mean is more than $\delta = 0.02$ below its Utility-only value. We also report δ from 0.01 to 0.05. This is a group-level evidence outcome, distinct from the question-level costs used in training. The primary utility endpoint is support-document recall; it directly measures the evidence available to a downstream reader.

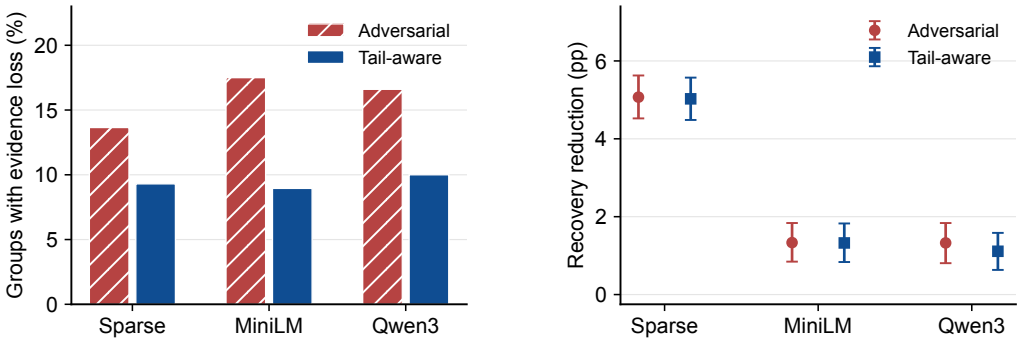
All intervals use paired comparisons. Test groups that share any support title form 910 connected components, the largest containing 64 groups. We resample whole components with their constituent groups; privacy intervals additionally resample attacker fits (Efron & Tibshirani, 1993; Owen, 2007). The reported 95% intervals use 10,000 draws and condition on the training data and the three fitted controllers. The same 2,000 groups are reused across settings.

4.3 Aggregate improvement still leaves local losses

Adversarial raises mean recall from 0.4640 to 0.4730, yet 273 groups lose more than 0.02 relative to Utility-only (Table 1). Tail-aware reduces this count to 186, a 31.9% relative decrease, while raising recall to 0.4772. The paired frequency decrease is 4.35 [3.25, 5.50] percentage points. Privacy

Table 1: Sparse retrieval on 2,000 held-out groups. Mean evidence improves under Adversarial, yet 13.65% of groups lose evidence relative to Utility-only. Tail-aware training reduces this burden. Privacy benefit is the recovery reduction relative to Utility-only with support-component and fit-resampled 95% intervals. Local loss uses expected recall and tolerance 0.02.

Policy	Evidence ↑	Privacy benefit [95% CI] ↑	Loss groups (%) ↓
Utility-only	0.4640	—	0.00
No-repeat	0.4847	-0.0438 [-0.0502, -0.0377]	1.35
Alignment	0.4739	0.0274 [0.0221, 0.0327]	12.35
Adversarial	0.4730	0.0507 [0.0452, 0.0563]	13.65
Tail-aware	0.4772	0.0503 [0.0448, 0.0557]	9.30



(a) Evidence-loss frequency, lower is better

(b) Privacy benefit, higher is better

Figure 2: Tail-aware training reduces local evidence cost while retaining positive privacy benefit. Each setting uses the same 2,000 groups and three controller seeds. (a) Percentage of groups whose expected recall falls more than 0.02 below Utility-only. (b) Recovery reduction relative to Utility-only under the sketch attacker family, in percentage points; error bars are 95% intervals over support components and attacker fits.

benefit remains 0.0503 [0.0448, 0.0557], compared with Adversarial's 0.0507. No-repeat has the highest recall but makes task recovery easier than Utility-only, illustrating why evidence repair must be assessed together with privacy. The lower loss count also reflects repair of originally affected inputs: 99 groups leave Adversarial's loss set and 12 enter it. Section 5 examines loss severity and tracks these affected inputs.

Privacy at the repaired operating point. Having established evidence repair, we compare Tail-aware directly with Adversarial to measure its privacy price. Positive benefit over Utility-only can coexist with a reduction in the benefit achieved by Adversarial. Under the sketch family, this trade-off is detectable for Sparse GRU (0.384 [0.052, 0.729] percentage points); the feed-forward intervals cross zero. Adding a raw-ID/rank attacker detects a cost of 0.548 [0.133, 0.974] points for Sparse FF and 1.183 [0.794, 1.568] for Sparse GRU, with both retaining positive benefit over Utility-only. Appendix J.1 gives the expanded-family results, ID coverage, and reconstruction protocol.

4.4 The correction persists across retrievers and controllers

We next test whether the repair observed with sparse retrieval persists when the retrieval backend or controller changes. Figure 2 pairs evidence-loss frequency with retained privacy benefit across the three retrievers.

MiniLM and Qwen3 reduce loss counts from 350 to 179 and 332 to 200, respectively (Table 2). Their paired frequency decreases are 8.55 [7.21, 9.95] and 6.60 [4.86, 8.40] percentage points. Both retain positive privacy benefit over Utility-only with unchanged tail fraction and coefficient. Sparse recurrence reduces loss counts from 252 to 194 and retains benefit 0.0498 [0.0445, 0.0554].

Table 2: The tail correction reduces local evidence loss across retrieval and controller choices. Each setting uses the same 2,000 test groups, matched Utility-only reference, three controller seeds, and ten attacker fits. Privacy intervals resample support components and fits. Loss counts use expected recall with tolerance 0.02.

Setting	Policy	Evidence ↑	Privacy benefit [95% CI] ↑	Loss groups ↓
Sparse / feed-forward	Adversarial	0.4730	0.0507 [0.0452, 0.0563]	273
	Tail-aware	0.4772	0.0503 [0.0448, 0.0557]	186
MiniLM / feed-forward	Adversarial	0.4624	0.0133 [0.0085, 0.0184]	350
	Tail-aware	0.4748	0.0133 [0.0084, 0.0183]	179
Qwen3 / feed-forward	Adversarial	0.5052	0.0133 [0.0081, 0.0184]	332
	Tail-aware	0.5124	0.0111 [0.0063, 0.0158]	200
Sparse / recurrent	Adversarial	0.4734	0.0537 [0.0481, 0.0594]	252
	Tail-aware	0.4770	0.0498 [0.0445, 0.0554]	194

Table 3: The most costly evidence losses become smaller. We average the largest 300 positive question-level losses among 6,000 test questions, ranking each policy separately after averaging its three controller seeds. Loss is measured relative to Utility-only. Brackets give paired 95% support-component intervals for the relative decrease.

Setting	Adversarial	Tail-aware	Decrease (%)
Sparse, feed-forward	0.2219	**0.1675**	24.5 [19.6, 29.2]
MiniLM dense	0.2478	**0.1042**	57.9 [54.3, 61.3]
Qwen3 dense	0.1854	**0.1071**	42.3 [37.7, 46.6]
Sparse, recurrent	0.2278	**0.1661**	27.1 [22.4, 31.6]

Consistency across training seeds. The eight-seed Sparse FF evaluation improves mean recall by 0.344–0.449 percentage points and reduces loss frequency by 3.6–4.8 points in every seed. Mean positive-cost repair ranges from 0.468 to 0.561 points. Across the three original CUDA runs and five additional CPU runs, evidence repair is consistent, whereas privacy differences vary in sign. Appendix H reports the seed-specific outcomes and their aggregation separately from the pooled three-seed endpoint.

4.5 Second-dataset transfer of the cost objective

To assess transfer beyond 2WikiMultiHopQA, we use MuSiQue (Trivedi et al., 2022) and transfer the same hyperparameters ($\lambda = 3$, $\alpha = 0.25$), three controller seeds, and retrieval budget to 240 training, 60 development, and 175 test groups, keeping support-connected components within splits. Tail-aware reduces loss counts from 50 to 35, a decrease of 8.571 [5.532, 11.476] percentage points, and raises mean recall by 0.601 [0.127, 1.052] points. Every seed improves recall and reduces loss frequency. This extends evidence repair to a second dataset; positive privacy benefit on MuSiQue remains unresolved because its interval crosses zero (Appendix I).

5 How Cost Control Repairs Evidence Loss

5.1 How much the most affected questions lose

Frequency alone does not show whether the largest losses become smaller. Table 3 measures their magnitude directly, averaging the worst 5% of positive question-level costs across the full test population. The mean falls from 0.2219 to 0.1675 with sparse retrieval, from 0.2478 to 0.1042 with MiniLM, and from 0.1854 to 0.1071 with Qwen3. These are reductions of 24.5–57.9%, with all paired intervals above zero. The recurrent controller follows the same pattern. Ranking questions separately for each policy measures its own highest-cost tail. Fixing the 300 worst Adversarial questions instead, Tail-aware lowers their mean positive cost from 0.2219 to 0.1508 (sparse), 0.2478 to 0.0932 (MiniLM), 0.1854 to 0.0988 (Qwen3), and 0.2278 to 0.1513 (recurrent). Between 89.0%

Table 4: At matched scale, positive-mean and upper-quarter penalties give similar evidence outcomes. Privacy is benefit relative to Utility-only; Table 6 gives $\beta = 3$ and paired intervals.

	Positive mean, $\beta = 12$		Upper quarter	
Setting	Loss groups	Privacy	Loss groups	Privacy
Sparse FF	187	0.0470	186	0.0503
MiniLM FF	177	0.0124	179	0.0133
Qwen3 FF	203	0.0107	200	0.0111
Sparse GRU	195	0.0481	194	0.0498

and 99.3% of these same questions improve. This post-hoc cohort analysis tracks original victims, while Appendix E also shows the full-population cost scatter.

The improvement also persists across sampled trajectories, loss tolerances, and all eight Sparse FF training seeds (Appendices D.1 and H). On matched reader contexts, the number of groups losing either evidence or answer F1 falls from 336 to 282; Appendix D.2 reports the full answer metrics and decoding protocol.

5.2 Repair reaches the originally affected groups

A smaller loss set could still contain newly affected inputs, so we track both departures and entries. The sparse result above—99 departures and 12 entries at tolerance 0.02—also holds directionally for dense retrieval: MiniLM yields 174 departures and 3 entries, while Qwen3 yields 141 departures and 9 entries. Together with the fixed worst-question cohort, these transitions show that the aggregate reduction is accompanied by repair on inputs that the original privacy policy harmed.

Tail-aware therefore reduces both its remaining worst-5% burden and losses on the original Adversarial cohort. This test cohort serves descriptive tracking; training uses costs from the training split.

Agreement between question-level severity and group-level frequency connects the training target to the reported benefit: repair remains visible after averaging questions into support groups. Because those groups share an evidence target, utility and conditional task recovery describe complementary outcomes of the same retrieval task.

5.3 Separate positive-cost penalties from tail weighting

Does evidence repair require prioritizing the upper quarter, or does it also occur when all positive costs are averaged? To distinguish these possibilities, we compare aggregation rules by replacing the tail term with $\beta N^{-1} \sum_i c_i$, keeping the full training budget and all other objectives unchanged in every retrieval and controller setting. We report $\beta = 3$ and 12. The latter matches the tail term's scale when all positive cost lies in the largest quarter, since then $\mathcal{T}_{0.25} = 4N^{-1} \sum_i c_i$.

Both aggregators repair evidence cost (Table 4). With $\beta = 12$, their outcomes differ by at most three loss groups and 0.00013 mean recall across the four settings. Upper-quarter weighting yields 0.0033 [0.0006, 0.0059] more privacy benefit in Sparse FF; the other three privacy-difference intervals cross zero. Evidence repair therefore persists under both allocation rules, while upper-quarter weighting makes prioritization of the most affected questions explicit. At $\beta = 3$, upper-quarter weighting reduces loss frequency by a further 2.1–5.7 percentage points. Table 6 reports both coefficients and their privacy trade-offs.

The near equality at matched scale is consistent with the recorded cost distribution: positive costs outside the upper quarter account for little total mass (Appendix F). The comparison varies aggregation while holding the positive reference-relative definition fixed.

5.4 Sensitivity to the evaluation reference

To assess dependence on the evaluation reference, we evaluate the same policies against No-repeat and Alignment. Both change the measured cost but preserve the repair direction in all four settings (Appendix E); Utility-only remains the training reference.

6 Related Work

Privacy–utility learning places a task objective alongside a penalty for sensitive inference. The privacy funnel formalizes an information trade-off (Makhdoumi et al., 2014), while adversarial representation learning makes related objectives trainable (Bertran et al., 2019; Xiao et al., 2020). These objectives aggregate utility and privacy over a population. We measure each input's utility change from a matched task-only policy, separating induced cost from absolute difficulty.

Tail-sensitive and robust learning provide tools for allocating training pressure across heterogeneous losses. Conditional value-at-risk emphasizes a high-loss fraction (Rockafellar & Uryasev, 2000), average top-k objectives interpolate between mean and maximum loss (Fan et al., 2017), and group-robust learning targets performance on predefined groups (Sagawa et al., 2020). Demographic-free DRO bounds unknown-group risk using a lower bound on group mass (Hashimoto et al., 2018); Blind Pareto Fairness combines worst-subgroup risk with Pareto efficiency for the remaining population (Martinez et al., 2021). These methods optimize prediction risk rather than the positive utility change from a fixed policy reference. Selective learning likewise studies how decision rules redistribute reliability across inputs and groups (Gangrade et al., 2021; Lee et al., 2021). We apply upper-tail averaging to positive utility loss from a fixed policy reference, recomputing the training tail as the policy changes.

Counterfactual no-harm policy learning bounds the fraction harmed by treatment relative to non-treatment, using potential-outcome assumptions and estimated harm constraints (Li et al., 2023). Our cached retrieval setting permits evaluating both policies' expected evidence utility directly. This yields a directly computable reference-relative recall cost, with empirical repair assessed on held-out questions.

Retrieval-augmented generation couples external evidence to language-model output (Lewis et al., 2020). Adaptive approaches interleave retrieval with reasoning or learn when to retrieve and critique evidence (Trivedi et al., 2023; Asai et al., 2024). We vary dense retrieval geometries (Reimers & Gurevych, 2019; Wang et al., 2020; Zhang et al., 2025) while fixing controller features and query candidates, using 2WikiMultiHopQA's annotated evidence (Ho et al., 2020) to compare task identities with a shared support set.

Remote retrieval also leaks through access and search patterns even when content is encrypted (Curtmola et al., 2006; Cash et al., 2015; Oya & Kerschbaum, 2021; Nie et al., 2024). Defenses reshape query frequencies, perturb queries, or conceal queries and accesses cryptographically (Grubbs et al., 2020; Bollegala et al., 2022; Cheng et al., 2025; Liang et al., 2026). Privacy work on retrieval-augmented systems studies database extraction, leakage from retrieval-based language models, and agent memory (Huang et al., 2023; Zeng et al., 2024; Wang et al., 2025). We instead hold the observable trace channel fixed and study how a learned client policy redistributes evidence utility while suppressing task inference.

7 Conclusion

Positive reference-relative evidence cost makes the distributional burden of privacy training explicit. Exact two-step expectations turn this cost into a training signal without changing deployment. Across three 2WikiMultiHopQA retrievers, upper-quarter training reduces local loss frequency by 31.9–48.9% and worst-5% loss by 24.5–57.9% relative to Adversarial, while retaining positive recovery reductions under the evaluated sketch attackers relative to Utility-only. Repair reaches originally affected inputs and persists across controller architectures, datasets, and aggregation rules. Direct privacy comparisons quantify its price. Explicit policy-induced cost thus makes the burden on affected inputs measurable and directly optimizable.

Reproducibility statement

Appendices A–D specify group construction, query candidates, controller objectives, attacker fitting, and uncertainty estimation. Tables and figures are generated from completed machine-readable evaluations. The appendices distinguish the original evaluations from local reproduction experiments and identify incomplete downstream evaluations.

AI use statement

We used generative AI tools to polish the language of the manuscript for clarity and readability. We also used these tools for retrieval and discovery, including finding related work and locating relevant literature.

Ethics statement

The experiments use public question-answering data and involve no collection of private user traces. Task-recovery measurements characterize an empirical attack surface; they do not provide differential-privacy or cryptographic guarantees.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

Martin Bertran, Natalia Martinez, Afroditi Papadaki, Qiang Qiu, Miguel Rodrigues, Galen Reeves, and Guillermo Sapiro. Adversarially learned representations for information obfuscation and inference. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 614–623. PMLR, 2019.

Danushka Bollegala, Tomoya Machide, and Ken ichi Kawarabayashi. Query obfuscation by semantic decomposition. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 6200–6211. European Language Resources Association, 2022.

David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. Leakage-abuse attacks against searchable encryption. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 668–679, 2015. doi: 10.1145/2810103.2813700.

Yihang Cheng, Lan Zhang, Junyang Wang, Mu Yuan, and Yunhao Yao. RemoteRAG: A privacy-preserving LLM cloud RAG service. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3820–3837. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.197.

Reza Curtmola, Juan A. Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 79–88, 2006. doi: 10.1145/1180405.1180417.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, 1993.

Yanbo Fan, Siwei Lyu, Yiming Ying, and Baogang Hu. Learning with average top-k loss. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Aditya Gangrade, Anil Kag, and Venkatesh Saligrama. Selective classification via one-sided prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2179–2187. PMLR, 2021.

Paul Grubbs, Anurag Khandelwal, Marie-Sarah Lacharité, Lloyd Brown, Lucy Li, Rachit Agarwal, and Thomas Ristenpart. Pancake: Frequency smoothing for encrypted data stores. In *29th USENIX Security Symposium*, pp. 2451–2468. USENIX Association, 2020.

Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 1929–1938. PMLR, 2018.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.580.

Yangsibo Huang, Samyak Gupta, Zexuan Zhong, Kai Li, and Danqi Chen. Privacy implications of retrieval-based language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14887–14902. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.921.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017.

Joshua K. Lee, Yuheng Bu, Deepta Rajan, Prasanna Sattigeri, Rameswar Panda, Subhro Das, and Gregory W. Wornell. Fair selective classification via sufficiency. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6076–6086. PMLR, 2021.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Haoxuan Li, Chunyuan Zheng, Yixiao Cao, Zhi Geng, Yue Liu, and Peng Wu. Trustworthy policy learning under the counterfactual no-harm criterion. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 20575–20598. PMLR, 2023.

Xiaojian Liang, Lushan Song, Shishuai Du, Weicheng Zhu, Tan Faith, Jun Jie Sim, Haibing Jin, Zhenghao Wu, Yingting Liu, Xin Zhang, Jiangming Yang, and Pu Duan. Pisces: Cryptography-based private retrieval-augmented generation with dual-path retrieval. In *International Conference on Learning Representations*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Ali Makhdoumi, Salman Salamatian, Nadia Fawaz, and Muriel Médard. From the information bottleneck to the privacy funnel. In *2014 IEEE Information Theory Workshop*, pp. 501–505, 2014. doi: 10.1109/ITW.2014.6970882.

Natalia L Martinez, Martin A Bertran, Afroditi Papadaki, Miguel Rodrigues, and Guillermo Sapiro. Blind pareto fairness and subgroup robustness. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 7492–7501. PMLR, 2021.

Hao Nie, Wei Wang, Peng Xu, Xianglong Zhang, Laurence T. Yang, and Kaitai Liang. Query recovery from easy to hard: Jigsaw attack against SSE. In *33rd USENIX Security Symposium*, pp. 2599–2616. USENIX Association, 2024.

Art B. Owen. The pigeonhole bootstrap. *The Annals of Applied Statistics*, 1(2):386–411, 2007. doi: 10.1214/07-AOAS122.

Simon Oya and Florian Kerschbaum. Hiding the access pattern is not enough: Exploiting search pattern leakage in searchable encryption. In *30th USENIX Security Symposium*, pp. 127–142. USENIX Association, 2021.

Qwen Team. Qwen3 technical report, 2025.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of EMNLP-IJCNLP*, pp. 3982–3992, 2019. doi: 10.18653/v1/D19-1410.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2(3):21–41, 2000. doi: 10.21314/JOR.2000.038.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 10014–10037, 2023. doi: 10.18653/v1/2023.acl-long.557.

Bo Wang, Weiyi He, Shenglai Zeng, Zhen Xiang, Yue Xing, Jiliang Tang, and Pengfei He. Unveiling privacy risks in LLM agent memory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 25241–25260. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1227.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Taihong Xiao, Yi-Hsuan Tsai, Kihyuk Sohn, Manmohan Chandraker, and Ming-Hsuan Yang. Adversarial learning of privacy-preserving and task-oriented representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 12434–12441, 2020. doi: 10.1609/aaai.v34i07.6930.

Shenglai Zeng, Jiankun Zhang, Pengfei He, Yue Xing, Yiding Liu, Han Xu, Jie Ren, Shuaiqiang Wang, Dawei Yin, Yi Chang, and Jiliang Tang. The good and the bad: Exploring privacy issues in retrieval-augmented generation (RAG). In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4505–4524. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.267.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

A Data and Retrieval Construction

A.1 Shared-support questions and splits

The source is the 2WikiMultiHopQA training population in the public multi-hop retrieval data format used by IRCoT (Ho et al., 2020; Trivedi et al., 2023). We group questions by their sorted set of all supporting document titles. Each selected group contains three distinct source questions with distinct normalized question text. Previously used and reserved groups are excluded before sampling. A random selection with seed 20260830 assigns 600 groups to training, 300 to development, and 2,000 to testing. These correspond to 1,800, 900, and 6,000 questions. Among the 2,900 groups, 2,309 have two support titles and 591 have four.

The full support-title sets have no overlap between splits, and neither do question IDs or normalized question text. Individual support documents can be shared by different support sets. There are 99 shared titles between training and development, 299 between training and test, and 193 between development and test. Within the test population, connecting groups that share any title produces 910 connected components; the largest has 64 groups. Component-resampled intervals preserve these connections. This split tests generalization to held-out questions and complete support sets, not to an entirely unseen document collection.

The group annotation is the complete supporting-title set supplied for these questions. It establishes a matched evidence comparison at the document-title level. Because all three questions in a group have the same annotation, a classifier given only that annotation has average recovery $1/3$ under the uniform question prior. We use this as the evidence-only comparator in task recovery.

A.2 Candidate queries and global retrieval

Candidate queries are constructed deterministically from the question text. The ordered candidate list starts with the whitespace-normalized question, followed by capitalized spans, three-word and two-word windows of non-stopword content tokens, and individual content tokens. Candidates are deduplicated without regard to case, and the first eight are retained. This construction uses neither support annotations nor answers. The implementation ensures at least two candidates by appending a question-based evidence suffix when needed. The candidate set remains fixed during the two-action episode; the second choice adapts to the retrieved observation.

All indexes search the same 369,280-document corpus. Documents are encoded or tokenized as their title followed by their text. Sparse retrieval uses BM25 with $k_1 = 1.5$ and $b = 0.75$, with negative inverse-document-frequency values floored at one quarter of the mean inverse document frequency. Dense retrieval searches normalized document and query vectors by cosine similarity. MiniLM uses the frozen all-MiniLM-L6-v2 sentence encoder shared with the controller. Qwen3 retrieval uses Qwen3-Embedding-0.6B (Zhang et al., 2025) with a 256-token encoding limit and the query prefix *Instruct: Given a web search query, retrieve relevant passages that answer the query* followed by *Query:*. Documents receive no query instruction. The maximum retrieval size is five documents per action. There are 27,964 unique candidate-query strings across the three splits.

The retrieved state is the mean of the returned document embeddings. It always uses the same 384-dimensional MiniLM representation, including when the retrieval index uses Qwen3. Thus the cross-retriever experiments change the retrieval geometry without changing the controller's semantic feature dimension.

B Training Details

B.1 Controller and trace representations

The feed-forward controller maps the question q, candidate embedding e_a, and observed state s into a hidden dimension $h = 192$. Its score for a candidate after an observation is

$$\ell_a = h^{-1/2}\big(\tanh(W_q q + b_q) + \tanh(W_s s + b_s)\big)^\top \tanh(W_a e_a + b_a). \tag{8}$$

The state term is omitted on the first action. A masked softmax converts scores into probabilities. The recurrent controller instead initializes its hidden state from the projected question and updates it

with a 192-unit GRU cell after the first observation. The same candidate-key projection and scaled inner product produce its action scores.

For each access set, a fixed signed document-ID sketch has 128 coordinates. The contribution of a document is weighted by the reciprocal of its rank and the completed vector is normalized to unit norm. The trace concatenates the two ordered sketches, the fraction of first-step documents accessed again at step two, and an indicator of repeated query selection. The resulting trace dimension is 258. The support-set embedding is computed from the joined sorted support titles and concatenated only for attacker input, yielding 642 input features. The controller does not receive the support-set embedding.

B.2 Base objective and local-cost penalty

Let $\bar{U}$ denote mean expected evidence recall, A the mean squared trace distance to the within-group trace centroid, and ℓ_j^{attack} the contrastive loss of training adversary j. For the adversarial policy, the controller objective before aggregate constraint penalties is

$$1 - \bar{U} + 3A - \frac{1}{3}\sum_{j=1}^{3} \ell_j^{\text{attack}} + 10^{-3}\bar{H}, \tag{9}$$

where $\bar{H}$ is the mean action entropy across the two decisions. Each adversary is a 192-unit GELU MLP mapping the concatenated trace and support embedding to a 384-dimensional prediction. It scores normalized question embeddings after centering them by the shared-support group's mean. Cross-entropy over the three question candidates uses cosine-score temperature 0.07. Adversaries minimize this loss; the controller maximizes their average loss through the negative term in Equation 9.

Every non-reference policy also has aggregate training targets. Mean evidence recall targets 95% of the matched Utility-only value. Mean action entropy and effective action support target 80% of their reference values. Effective support is the number of actions with probability at least 0.05, averaged over both decisions. Writing the three target shortfalls as v, we add

$$d^\top v + 5\|[v]_+\|_2^2, \qquad d \leftarrow [d + 0.5v]_+. \tag{10}$$

Dual coefficients start at one. These are the same training penalties for No-repeat, Alignment, Adversarial, and Tail-aware.

Utility-only optimizes $1 - \bar{U} + 10^{-3}\bar{H}$ without the additional constraints. No-repeat adds the distinct-action mask and aggregate penalties. Alignment further adds $3A$. Adversarial adds the adversary term. Tail-aware adds $\lambda\mathcal{T}_{0.25}$, using the reference policy trained first with the same initialization seed. Its stored expected utilities are detached from gradient computation. The tail is computed over all 1,800 training questions on each update. Weights 1 and 3 are compared on development groups; weight 3 is used for test evaluation. There is no test-time risk threshold or selective policy assignment.

The primary experiment uses 160 full-batch epochs and controller seeds 7, 42, and 77. The supplementary Sparse FF repetition retains this epoch budget and adds seeds 11, 23, 101, 137, and 211. AdamW (Loshchilov & Hutter, 2019) uses learning rate 0.003 and weight decay 10^{-4}. Gradient norms are clipped at 5. Each of the three adversaries takes three updates before a controller update. Straight-through Gumbel–Softmax temperature is 1 for the sampled privacy and alignment path. The evidence and tail terms enumerate the two-action distribution exactly. Both dense settings and the recurrent setting use the same training budget and loss weights.

B.3 Training and inference cost

The expected utility calculation evaluates up to 64 ordered action pairs using cached retrievals. It is quadratic in the candidate count rather than the corpus size. The tail operation selects the largest 450 costs from the 1,800-question training population. The reference utilities are computed once after training Utility-only. Tail-aware and Adversarial have identical controller parameter counts and both make two retrieval calls at inference. The cost of training the reference is an offline cost; no reference-policy execution or evidence-label access is needed after training.

C Attacker Fitting and Statistical Estimation

C.1 Finite-family task recovery

Reporting attackers are trained separately from the adversaries used for policy learning. The family comprises ridge regression with regularization 10, distance-weighted 11-nearest-neighbor regression, a 192-unit GELU MLP, and a 64-unit GRU. All predict the question's 384-dimensional embedding. The GRU reads the two ordered access sketches and concatenates its final state with the support-set embedding and the two scalar trace features before a linear prediction head. The MLP has one hidden layer. Input scaling is fitted on training traces only.

The neural reporters use 80 full-batch AdamW epochs, learning rate 0.002, weight decay 10^{-4}, and cosine prediction loss. Fit seeds are 101, 211, 307, 401, 503, 601, 701, 809, 907, and 1009. Initialization is paired between policy comparisons. Ridge and nearest neighbors are deterministic under their fixed settings and are reused across these repetitions. For each policy, its three controller seeds yield 5,400 training traces from the 1,800 training questions. The test population similarly has repeated observations of 6,000 question identities, not 18,000 distinct identities.

Within a group, an attacker's predicted vector is compared by cosine similarity with the embeddings of the three distinct question candidates. Stable candidate ordering resolves ties. For policy π, attacker family a, and fit s, let $r_{a,s}(\pi, g)$ be recovery averaged over the three questions and three controller seeds in group g. We aggregate by

$$R_s(\pi, g) = \max\{1/3,\ \max_a r_{a,s}(\pi, g)\}, \qquad B(\pi, g) = \frac{1}{10}\sum_s \big[R_s(\pi_0, g) - R_s(\pi, g)\big]. \tag{11}$$

Reported privacy benefit is the mean of $B(\pi, g)$ over groups. The family maximum occurs within each group and fit, before averaging. It is therefore an empirical finite-family envelope, not the accuracy of one universally strongest trained model. All main results use attackers trained on each evaluated policy's own training traces.

C.2 Evidence outcomes

For each policy seed and question, Equation 4 is evaluated on test retrievals. The group mean averages these exact expected recalls over three questions and three controller seeds. Let this average be $\bar{U}_\pi(g)$. The local loss frequency at tolerance δ is

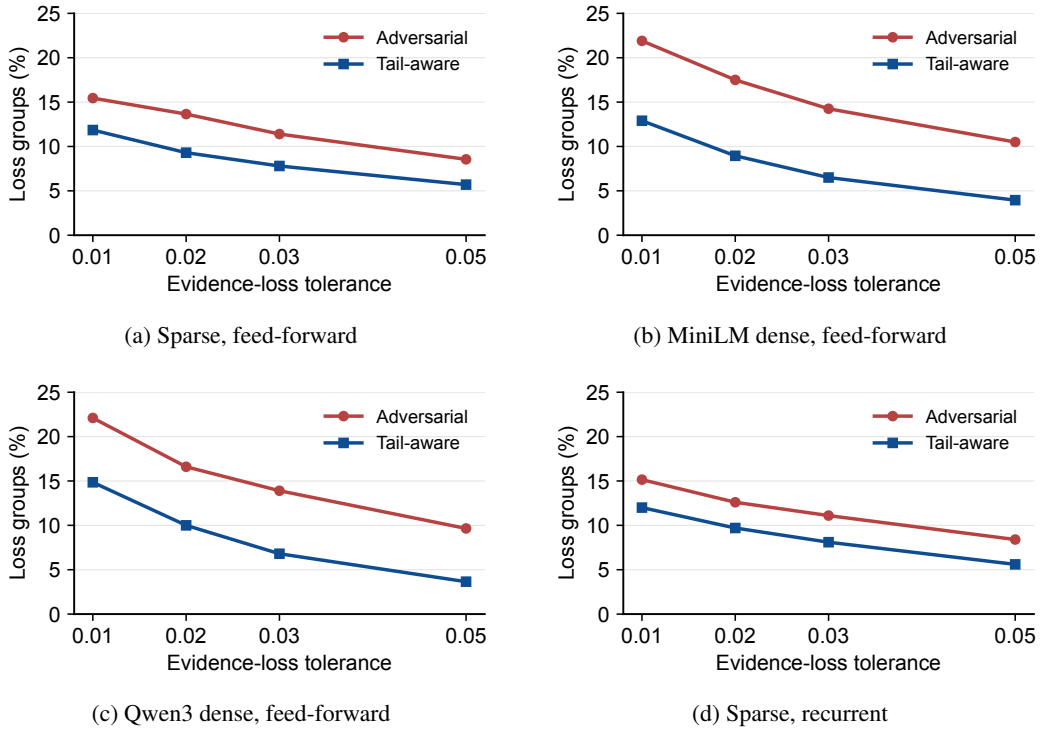
$$\frac{1}{G}\sum_{g=1}^{G} \mathbf{1}\{\bar{U}_{\pi_0}(g) - \bar{U}_\pi(g) > \delta\}. \tag{12}$$

Test tables report $\delta = 0.02$, with 0.01, 0.03, and 0.05 in the sensitivity comparison. This definition differs from the training tail in Equation 5, which operates on individual questions before group averaging. Sampled evidence comparisons use the same grouping and seeds but replace exact expected recall with recall on a sampled two-action trajectory.

C.3 Paired uncertainty

Support components are the transitive connected components formed by shared annotated document titles. For evidence comparisons, we draw 910 components with replacement, retain all their groups, and compute the group-weighted mean. Thus a component containing more groups contributes more observations rather than receiving the same total weight as a singleton. Privacy intervals independently draw ten fitting repetitions with replacement as well. The same sampled components and fits are used for both policies in a contrast. This crossed-resampling construction represents component-level dependence and reporter-fitting variation (Owen, 2007).

We take the 2.5th and 97.5th percentiles of 10,000 draws. For the paired change in local evidence-loss frequency, the group-level value is the difference between the two indicators, each defined relative to the common Utility-only reference. The intervals condition on the available training set, fitted feature encoders, and three trained controllers. The cross-setting comparisons reuse the same test questions, allowing matched comparisons rather than independent-population pooling.



(a) Sparse, feed-forward

(b) MiniLM dense, feed-forward

(c) Qwen3 dense, feed-forward

(d) Sparse, recurrent

Figure 3: Fewer groups lose evidence across all four reported tolerances. Each point is the fraction of 2,000 groups whose expected evidence recall decreases by more than the horizontal-axis tolerance relative to Utility-only. Training uses the continuous tail cost, not these evaluation thresholds.

D Expanded Matched Results

D.1 Severity, transitions, and controller seeds

For the severity analysis, we first average each question's exact expected recall over the three controller seeds, then take its positive loss relative to the same average for Utility-only. Table 3 averages the largest 300 losses among the 6,000 questions. The 5% reporting tail provides a severity summary alongside the original frequency endpoint; training uses the 25% tail throughout. Each bootstrap draw resamples support components with all their questions, reranks each policy's costs within the resulting population, and averages its largest $\lceil 0.05N \rceil$ costs. We compute paired decreases from the same draw and use 10,000 draws.

The all-question mean positive cost falls from 0.0157 to 0.0105 for sparse feed-forward retrieval, from 0.0205 to 0.0076 for MiniLM, from 0.0154 to 0.0077 for Qwen3, and from 0.0156 to 0.0105 for sparse recurrence. Loss-group counts for controller seeds 7, 42, and 77 change from (271, 266, 272) to (185, 184, 186) in sparse feed-forward retrieval; from (335, 347, 358) to (187, 178, 174) in MiniLM; from (331, 335, 341) to (189, 206, 200) in Qwen3; and from (255, 255, 252) to (194, 191, 193) in sparse recurrence. With the three seeds averaged, the recurrent comparison moves 74 groups below the loss tolerance and 16 above it.

D.2 Downstream reader on matched retrieved contexts

We evaluate Utility-only, Adversarial, and Tail-aware with Qwen3-8B on every sparse feed-forward test question. For each question and policy, the context uses the most frequent sampled action history across the three controller seeds; lexicographic order breaks ties. Documents are deduplicated in retrieval order. This produces 18,000 policy–question inputs and 13,542 distinct prompts. Identical prompts share one generated answer. The reader receives the question and retrieved documents, not gold support annotations or the reference answer. It uses greedy decoding, thinking disabled, a

Table 5: Qwen3-8B answers on all 6,000 sparse-retrieval test questions. Each question uses its modal action history across three controller seeds. Evidence and answer metrics therefore use the same retrieved context. Loss counts use group-average evidence recall or answer F1 more than 0.02 below Utility-only, among 2,000 groups.

Policy	Evidence recall	Answer EM	Answer F1	Loss groups
Utility-only	0.4623	0.1582	0.1771	—
Adversarial	0.4751	**0.1757**	**0.1944**	336
Tail-aware	**0.4801**	0.1718	0.1910	**282**

Table 6: Replacing upper-tail aggregation by the mean of the same positive reference-relative costs. Each setting uses 160 epochs, three controller seeds, and ten attacker fits. $\beta = 3$ matches the numeric tail coefficient; $\beta = 12$ matches its scale when positive cost is concentrated in one quarter of the population. No coefficient is selected from test results.

Setting	Penalty	Evidence	Loss groups	Worst 5% cost	Privacy benefit
Sparse FF	None	0.4730	273	0.2219	0.0507
	Mean, $\beta = 3$	0.4748	244	0.2041	0.0502
	Mean, $\beta = 12$	**0.4773**	187	**0.1629**	0.0470
	Upper quarter	0.4772	**186**	0.1675	**0.0503**
MiniLM FF	None	0.4624	350	0.2478	0.0133
	Mean, $\beta = 3$	0.4677	292	0.1894	**0.0157**
	Mean, $\beta = 12$	0.4748	**177**	**0.1017**	0.0124
	Upper quarter	**0.4748**	179	0.1042	0.0133
Qwen3 FF	None	0.5052	332	0.1854	0.0133
	Mean, $\beta = 3$	0.5080	284	0.1581	**0.0134**
	Mean, $\beta = 12$	0.5123	203	**0.1067**	0.0107
	Upper quarter	**0.5124**	**200**	0.1071	0.0111
Sparse GRU	None	0.4734	252	0.2278	**0.0537**
	Mean, $\beta = 3$	0.4749	236	0.2079	0.0531
	Mean, $\beta = 12$	**0.4771**	195	**0.1605**	0.0481
	Upper quarter	0.4770	**194**	0.1661	0.0498

maximum of 64 generated tokens, and an 8,192-token context budget; no input is truncated. Answer exact match and token F1 lowercase text and remove punctuation, articles, and extra whitespace.

Evidence recall in Table 5 is measured on the reader's actual context, rather than the expected recall used for policy evaluation. We average evidence and F1 over each group's three questions and define a joint loss when either decreases by more than 0.02 relative to Utility-only. The correction moves 94 groups below this joint-loss criterion and 40 above it, for a net reduction of 54 groups. Intervals resample the same 910 support components as the main evaluation and condition on this reader and context-selection rule.

D.3 Positive-mean objective replacement

This follow-up comparison replaces the upper-quarter aggregation with the mean of the same positive reference-relative costs. The mean-penalty coefficients are 3 and 12, and both are reported without test-set selection. Coefficient 3 matches the numerical tail coefficient; 12 matches its scale when at most one quarter of questions have positive cost. Each variant uses the original utility-only checkpoint for its seed, 160 full-batch epochs, the same 1,800 training and 900 development questions, and unchanged privacy and aggregate-constraint objectives. Evaluation uses all 6,000 test questions and all ten reporter fits. Original policy traces and recovery scores are reproduced exactly in this comparison.

Against the $\beta = 3$ replacement, upper-quarter weighting reduces loss frequency by 2.90 [1.98, 3.80], 5.65 [4.52, 6.83], 4.20 [2.91, 5.52], and 2.10 [1.28, 2.93] percentage points for sparse feed-forward, MiniLM, Qwen3, and sparse recurrent retrieval. The corresponding privacy-benefit differences are

Table 7: Development-set evidence losses decline as the tail weight increases. Counts are out of 300 groups and use sampled recall averaged over three controller seeds, with tolerance 0.02 relative to Utility-only. The zero-weight policy is Adversarial.

Setting	Weight 0	Weight 1	Weight 3
Sparse / feed-forward	43	36	32
MiniLM / feed-forward	67	56	40
Qwen3 / feed-forward	64	54	41
Sparse / recurrent	40	35	31

0.0001 [-0.0034, 0.0035], -0.0024 [-0.0055, 0.0006], -0.0023 [-0.0054, 0.0009], and -0.0033 [-0.0065, -0.0002].

At $\beta = 12$, the loss-frequency differences are 0.05 [-0.27, 0.37], -0.10 [-0.54, 0.35], 0.15 [-0.32, 0.61], and 0.05 [-0.27, 0.37] percentage points. Mean-recall differences are no larger than 0.00013 in magnitude. The paired privacy-benefit differences are 0.0033 [0.0006, 0.0059], 0.0009 [-0.0017, 0.0035], 0.0004 [-0.0021, 0.0029], and 0.0017 [-0.0009, 0.0044]. All intervals use 10,000 paired support-component draws, additionally crossing reporter fits for privacy.

D.4 Full test and development comparisons

Table 8 includes every non-reference test policy and evidence tolerance, using the ten-fit privacy endpoint in Equation 11. Table 9 includes all six development candidates. Development privacy scores summarize one ridge/kNN/MLP fit per controller seed, separately from the test endpoint.

Table 8: Test comparisons against each setting's Utility-only policy. All loss counts use exact expected recall and a denominator of 2,000 groups. Columns give four evidence-loss tolerances. Privacy intervals resample support components and fits.

Setting	Policy	Privacy benefit [95% CI]	0.01	0.02	0.03	0.05
Sparse / feed-forward	No-repeat	-0.0438 [-0.0502, -0.0377]	40	27	22	11
Sparse / feed-forward	Alignment	0.0274 [0.0221, 0.0327]	297	247	191	122
Sparse / feed-forward	Adversarial	0.0507 [0.0452, 0.0563]	309	273	228	171
Sparse / feed-forward	Tail-aware	0.0503 [0.0448, 0.0557]	237	186	156	114
MiniLM / feed-forward	No-repeat	-0.0108 [-0.0161, -0.0054]	17	9	5	0
MiniLM / feed-forward	Alignment	0.0094 [0.0045, 0.0146]	433	321	244	155
MiniLM / feed-forward	Adversarial	0.0133 [0.0085, 0.0184]	438	350	285	210
MiniLM / feed-forward	Tail-aware	0.0133 [0.0084, 0.0183]	258	179	130	79
Qwen3 / feed-forward	No-repeat	-0.0271 [-0.0328, -0.0213]	12	5	4	1
Qwen3 / feed-forward	Alignment	0.0070 [0.0020, 0.0120]	431	312	232	122
Qwen3 / feed-forward	Adversarial	0.0133 [0.0081, 0.0184]	442	332	278	193
Qwen3 / feed-forward	Tail-aware	0.0111 [0.0063, 0.0158]	297	200	136	73
Sparse / recurrent	No-repeat	-0.0480 [-0.0543, -0.0419]	39	24	17	11
Sparse / recurrent	Alignment	0.0301 [0.0248, 0.0354]	282	221	176	112
Sparse / recurrent	Adversarial	0.0537 [0.0481, 0.0594]	303	252	222	168
Sparse / recurrent	Tail-aware	0.0498 [0.0445, 0.0554]	240	194	162	112

E Direct Trade-offs and Reference Sensitivity

All analyses in this section reuse the evaluated policies and test population. Question costs take the positive part after averaging expected recall over the three controller seeds. The fixed cohort contains the 300 questions with largest observed Adversarial costs, with ties resolved by task order. These are post-hoc descriptive results, conditional on the selected cohort. Across all 6,000 questions, 971/125 (sparse), 678/43 (MiniLM), 511/84 (Qwen3), and 520/127 (recurrent) leave/enter the positive-cost set. These counts use question cost $c > 0$, distinct from the group-level 0.02 tolerance. No-repeat includes the action restriction and aggregate constraints; Alignment additionally includes trace alignment. The cost levels depend on the reference, while Tail-aware reduces mean positive cost and loss frequency

Table 9: Complete development comparison on 300 groups. Evidence is sampled recall; excess recovery is the development attacker aggregate averaged over three controller seeds. Tail-aware without a weight suffix uses weight 3.

Setting	Policy	Evidence	Excess recovery	Loss groups
Sparse / feed-forward	Utility-only	0.4548	0.0322	0
Sparse / feed-forward	No-repeat	0.4783	0.0722	8
Sparse / feed-forward	Alignment	0.4630	0.0156	36
Sparse / feed-forward	Adversarial	0.4619	0.0122	43
Sparse / feed-forward	Tail-aware	0.4670	0.0104	32
Sparse / feed-forward	Tail-aware, weight 1	0.4639	0.0019	36
MiniLM / feed-forward	Utility-only	0.4660	0.0104	0
MiniLM / feed-forward	No-repeat	0.4804	0.0156	3
MiniLM / feed-forward	Alignment	0.4594	0.0067	62
MiniLM / feed-forward	Adversarial	0.4504	0.0033	67
MiniLM / feed-forward	Tail-aware	0.4684	0.0063	40
MiniLM / feed-forward	Tail-aware, weight 1	0.4619	0.0052	56
Qwen3 / feed-forward	Utility-only	0.5120	0.0030	0
Qwen3 / feed-forward	No-repeat	0.5184	0.0211	6
Qwen3 / feed-forward	Alignment	0.5046	0.0048	54
Qwen3 / feed-forward	Adversarial	0.4994	0.0019	64
Qwen3 / feed-forward	Tail-aware	0.5086	0.0104	41
Qwen3 / feed-forward	Tail-aware, weight 1	0.5047	0.0085	54
Sparse / recurrent	Utility-only	0.4542	0.0337	0
Sparse / recurrent	No-repeat	0.4755	0.0804	12
Sparse / recurrent	Alignment	0.4633	0.0111	35
Sparse / recurrent	Adversarial	0.4631	0.0033	40
Sparse / recurrent	Tail-aware	0.4672	0.0081	31
Sparse / recurrent	Tail-aware, weight 1	0.4650	0.0089	35

Table 10: Costs on the same Adversarial worst-5% cohort. Improved and worsened compare positive costs for each question; no question in these cohorts has unchanged cost.

Setting	Adv. cost	Tail cost	Improved (%)	Worsened (%)
Sparse FF	0.2219	0.1508	89.00	11.00
MiniLM FF	0.2478	0.0932	99.33	0.67
Qwen3 FF	0.1854	0.0988	97.67	2.33
Sparse GRU	0.2278	0.1513	95.67	4.33

under all three references in all four settings. These comparisons change the evaluation reference; they do not test training with a different reference.

F Recorded Objective Differences

For $N = 1800$ and $k = 450$, the relative difference between $12\,\mathrm{mean}(c)$ and $3\mathcal{T}_{0.25}$ equals the fraction of total positive cost lying outside the largest quarter. Thus more than 450 positive entries can coexist with nearly equal objective values. The 24 replacement runs each record nine states, at epochs 0, 20, 40, 60, 80, 100, 120, 140, and 159. None of these 216 states satisfies the sufficient equality condition of at most 450 strictly positive costs. These observations do not establish the condition at unrecorded updates. We additionally restore the sparse retrieval cache and MiniLM inputs locally and recompute the original Sparse FF final checkpoints. The largest question/support-embedding differences from saved inputs are below 2.3×10^{-7}. For Tail-aware seeds 7, 42, and 77, the positive-cost counts are 592, 574, and 557, while the relative penalty gaps are 0.0023%, 0.0013%, and 0.0009%. These are final-checkpoint measurements, separate from the pre-update log records.

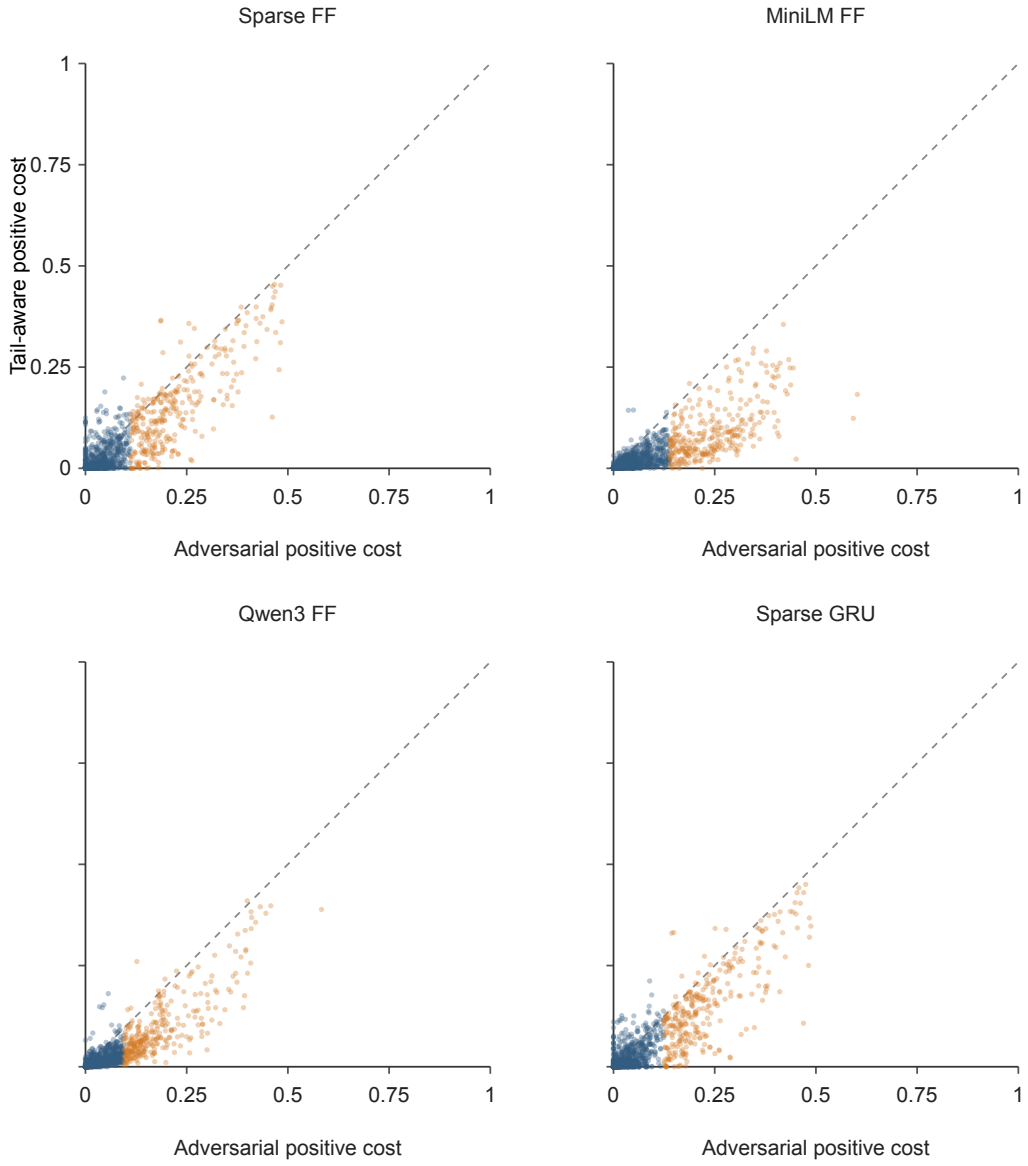


Figure 4: Positive costs for all 6,000 questions in each setting. Orange points mark the fixed Adversarial worst-5% cohort; the diagonal indicates unchanged cost.

G Local Evaluation and Reproduction Status

Local restoration uses PyTorch 2.8.0 on an Apple Silicon machine with 16 GiB unified memory. Sparse retrieval was rebuilt from the 369,280-document corpus in 918 seconds; 27,964 unique queries had identical opaque-ID and plain-ID rankings. MiniLM encoding of the missing input texts took 79 seconds including setup and saving. Original checkpoint evaluation uses the restored embeddings, whose saved question and support targets agree within 2.4×10^{-7}.

MiniLM dense retrieval was rebuilt on MPS in 91 seconds. At least one historical test sketch could not be matched to this cache, so we reran all three original policies and four sketch reporters on MPS (181 seconds) before comparing attacks. The raw-ID attack uses exactly these new observations, with zero sketch-reconstruction error, and takes a further 16 seconds excluding reconstruction. Raw-ID/rank recovery gives Tail-aware a benefit of 1.078 [0.764, 1.410] percentage points over Utility-only and

Table 11: Evaluation-reference sensitivity for fixed Adversarial and Tail-aware policies. Loss counts use a 0.02 group-recall tolerance. Cost repair is the decrease in mean positive question cost. Tail-aware was trained against Utility-only in every row.

Setting	Reference	Adv. losses	Tail losses	Cost repair
Sparse FF	Utility-only	273	186	0.00520
Sparse FF	No-repeat	383	323	0.00579
Sparse FF	Alignment	172	111	0.00254
MiniLM FF	Utility-only	350	179	0.01292
MiniLM FF	No-repeat	485	334	0.01339
MiniLM FF	Alignment	195	47	0.00596
Qwen3 FF	Utility-only	332	200	0.00764
Qwen3 FF	No-repeat	379	243	0.00766
Qwen3 FF	Alignment	151	36	0.00379
Sparse GRU	Utility-only	252	194	0.00516
Sparse GRU	No-repeat	364	327	0.00537
Sparse GRU	Alignment	192	108	0.00358

Table 12: Ranges over three seeds at the last logged state (epoch 159, before its parameter update) of each positive-mean replacement. Gap is a percentage of $12\,\mathrm{mean}(c)$.

Setting	Penalty	Positive questions	Relative gap (%)
Sparse FF	$\beta = 3$	750–780	0.0675–0.0926
Sparse FF	$\beta = 12$	619–626	0.0042–0.0069
MiniLM FF	$\beta = 3$	1076–1102	1.1246–1.2452
MiniLM FF	$\beta = 12$	950–976	0.7749–0.8544
Qwen3 FF	$\beta = 3$	1207–1218	1.2983–1.3847
Qwen3 FF	$\beta = 12$	1166–1194	1.4609–1.5922
Sparse GRU	$\beta = 3$	570–581	0.0062–0.0102
Sparse GRU	$\beta = 12$	497–518	0.0002–0.0008

0.294 [0.000, 0.600] over Adversarial. The family maximum, reported in the main text, remains the privacy endpoint. Across the three policies, 9.2–10.4% of distinct test IDs were seen in training. The maximum change in per-question expected support recall from the historical evaluation is 1.3×10^{-6} across the three policies. These local measurements retain the original consumed test population and are not an independent confirmation sample.

For per-seed downstream evaluation, we prepared 54,000 contexts from the original three Sparse FF seeds. Every action sequence and sampled support recall matches its original policy record. Exact system/user prompts and decoding settings permit reuse of 34,163 historical reader answers. Completing the remaining rows requires 14,165 unique new prompts. The original Qwen3-8B reader weights are unavailable locally; the cached 4-bit variant has not been substituted into the historical results. Consequently, the modal-context reader table remains the completed downstream experiment, and per-seed answer metrics are pending.

The initial local capacity audit found at most seven HotpotQA and 56 MuSiQue groups in the public development sources. We subsequently obtained the official MuSiQue training source from the authors' released IRCoT data package and constructed the separate second-dataset experiment in Appendix I. The original local capacity counts do not describe that larger source.

H Supplementary Training Repetitions

We fixed five additional Sparse FF seeds (11, 23, 101, 137, and 211) before scoring new test outcomes. Each trains Utility-only, Adversarial, and Tail-aware for the original 160 epochs with unchanged hyperparameters, adding 15 complete controllers. Local CPU training and development evaluation took 1,819 seconds. An initial MPS run developed non-finite parameters; it was excluded and the full fixed set was trained on CPU. The original seeds 7, 42, and 77 were trained on CUDA, so hardware and seed cohort are confounded.

All eight checkpoints per policy are evaluated on the same consumed test population with new local rollouts. Four sketch reporters are refitted to pooled training observations, with ten fitting repetitions. For the seed-specific privacy endpoint, we maximize reporter recovery within each controller seed and fit before averaging groups and fits. This differs from maximizing after averaging controller seeds in the primary endpoint. All seeds share data and reporter fits; their distribution is descriptive and does not supply a full training-and-refitting confidence interval.

Table 13: Sparse FF training repetitions. Values are percentage points; all differences are Tail minus Adversarial except positive-cost repair, which is Adversarial minus Tail. Privacy uses the four sketch reporters with the seed-specific family maximum. CUDA/CPU denotes controller training hardware; all rows use local re-evaluation.

Seed	Training	ΔB	ΔU	ΔL	Cost repair
7	CUDA	-0.362	0.449	-4.300	0.561
42	CUDA	-0.080	0.420	-4.100	0.512
77	CUDA	0.008	0.411	-4.300	0.487
11	CPU	0.280	0.406	-3.900	0.488
23	CPU	-0.225	0.344	-3.650	0.467
101	CPU	-0.293	0.441	-4.800	0.541
137	CPU	-0.508	0.392	-4.150	0.512
211	CPU	0.100	0.388	-3.600	0.489
Mean	All	-0.135	0.406	-4.100	0.507

Every seed improves mean expected recall (0.344–0.449 percentage points), reduces mean positive cost (0.468–0.561 points), and lowers loss frequency (3.6–4.8 points). Seed-specific sketch privacy differences range from -0.508 to 0.280 points, with five negative and three positive values. Thus the evidence direction is consistent across these eight repetitions while privacy differences vary in sign. The complete evaluation took 402 seconds. Seed-conditioned recovery values average back to pooled recovery within 1.2×10^{-16} because both use the same row-level ranking decisions.

I Second-Dataset Evaluation on MuSiQue

We obtain the authors' processed MuSiQue training source through the official IRCoT data release (Trivedi et al., 2022; 2023). The existing loader retains 19,560 of 19,938 source questions with a nonempty question and answer and at least two support titles. Among these, 475 natural groups contain at least three distinct question texts sharing the exact normalized support-title set. Comparing question IDs, normalized question text, and complete support sets excludes no overlap with the previously available public development source. A fixed component allocation assigns 240 groups to training, 60 to development, and 175 to evaluation, with three questions per group. The splits contain 53, 27, and 34 support-connected components, respectively, with no support title shared across splits. We use the full 70,408-document source-context corpus for every split. Where a title has several paragraph texts, the existing corpus convention retains the longest text with a lexical tie-break; recall therefore measures title-level evidence coverage.

The experiment trains Sparse FF Utility-only, Adversarial, and Tail-aware with seeds 7, 42, and 77 on CPU for 160 epochs each. We retain the original dimensions, learning rates, privacy losses, two-action budget, eight candidates, five returned IDs, $\lambda = 3$, and $\alpha = 0.25$. No dataset-specific hyperparameter search is performed. Attackers are refitted separately to each policy's training observations. The four sketch families use ten fitting repetitions; the added raw-ID/rank ridge follows the same training-component holdout and regularization grid as the 2Wiki expansion. Intervals resample evaluation support components and, for sketch-family comparisons, reporter fits, conditional on the fitted controllers and training data.

A separate small experiment completed on the originally available public development source uses 28/8/20 train/development/evaluation groups. Tail-aware reduces mean positive cost by 0.384 [0.070, 0.701] percentage points, while loss counts remain 6/20 for both privacy policies and mean recall changes by 0.148 [-0.302, 0.599] points. Its expanded-family privacy benefit over Utility-only is -0.111 [-7.778, 6.311] points. This small-source experiment does not establish privacy transfer; its

observations are not pooled with the larger training-source experiment. The larger protocol was fixed before analyzing these small-source test results.

Completed training-source results. All nine controller runs completed 160 epochs with finite parameters; training and development evaluation took 448 seconds. Every raw trace reproduces its saved sketch exactly. Between 13.3% and 13.9% of distinct test IDs appear in the training traces for the two privacy policies. The test set has 34 support components, so intervals retain within-component dependence.

Table 14: MuSiQue privacy by attacker representation. Values are percentage points with conditional paired 95% intervals.

Attackers	B_{Tail}	$B_{\text{Tail}} - B_{\text{Adv}}$
Sketch	1.073 [-0.865, 2.596]	-0.400 [-1.456, 0.599]
ID/rank	-0.825 [-1.903, 0.000]	-0.063 [-0.911, 0.847]
Expanded	0.692 [-1.277, 2.236]	-0.349 [-1.626, 1.038]

The raw ridge alone has a negative estimated benefit for Tail-aware relative to Utility-only. Its interval reaches zero; it does not support a positive privacy gain. Expanding the family restores a positive point estimate but its interval still crosses zero. The evidence benefit and uncertainty in privacy transfer are reported together in the main text.

J Extended comparisons and training diagnostics

J.1 Raw-ID attacks and additional controller seeds

We add an attacker using document IDs at their two-step rank positions, all within-trace ID-equality indicators, return counts, the original overlap/repeat scalars, and the support embedding. It receives no document text or query string. Reconstruction matches every saved training observation to a unique raw sequence; saved test actions resolve 14–18 step-level sketch ambiguities per policy. A support-component holdout within the original training split selects ridge regularization before test scoring.

With this reporter added to the original family, Tail-aware retains privacy benefit over Utility-only of 7.891 [7.252, 8.555] percentage points for sparse feed-forward retrieval and 7.684 [7.014, 8.380] for sparse recurrence. Its benefit relative to Adversarial decreases by 0.548 [0.133, 0.974] and 1.183 [0.794, 1.568] points, respectively (Table 17). The additional attack therefore makes the privacy price of evidence repair detectable in both sparse settings. Only 9.0–9.7% of distinct test IDs appear in training; equality features preserve relationships between unseen IDs, but their individual linear coefficients remain untrained. This finite attack expansion tests richer observations without establishing unrestricted full-trace privacy.

A separate local MiniLM re-evaluation uses the original checkpoints with a locally reconstructed dense cache and newly sampled observations. The cache does not exactly reproduce every historical observation, so these results are not combined with the original CUDA rollouts. With the expanded family, Tail-aware's benefit is 1.977 [1.447, 2.509] percentage points relative to Utility-only and -0.073 [-0.525, 0.369] relative to Adversarial. The latter interval crosses zero. Appendix G describes this local evaluation.

Additional training seeds. A supplementary eight-seed Sparse FF evaluation preserves the evidence direction in every seed: Tail-aware increases mean recall by 0.344–0.449 percentage points and reduces loss frequency by 3.6–4.8 points relative to Adversarial. Seed-specific privacy differences under the four sketch reporters range from -0.508 to 0.280 points. These repetitions combine three historical CUDA-trained controllers and five new CPU-trained controllers; their descriptive distribution and distinct aggregation order are reported in Appendix H.

Table 15: Direct Tail-aware minus Adversarial effects in percentage points with paired 95% intervals. $\Delta B > 0$ means more privacy benefit under the adapted sketch attackers, $\Delta U > 0$ means higher evidence recall, and $\Delta L < 0$ means fewer loss groups. Intervals condition on the fitted controllers.

Setting	ΔB	ΔU	ΔL
Sparse FF	-0.043 [-0.425, 0.320]	0.427 [0.298, 0.550]	-4.350 [-5.497, -3.250]
MiniLM FF	-0.009 [-0.361, 0.342]	1.242 [1.048, 1.457]	-8.550 [-9.948, -7.209]
Qwen3 FF	-0.215 [-0.566, 0.139]	0.722 [0.593, 0.852]	-6.600 [-8.401, -4.857]
Sparse GRU	-0.384 [-0.729, -0.052]	0.362 [0.236, 0.481]	-2.900 [-3.876, -1.974]

Table 16: MuSiQue evaluation on 175 groups (525 questions), with three controller seeds and unchanged training hyperparameters. B is privacy benefit over Utility-only under the expanded family, in percentage points with paired 95% support-component/fit intervals. Evidence columns average controller seeds.

Policy	Mean recall	Positive cost	Loss groups	Expanded B
Utility-only	0.28443	0.00000	0/175	0
Adversarial	0.28291	0.02141	50/175	1.041 [-1.161, 3.128]
Tail-aware	0.28892	0.01341	35/175	0.692 [-1.277, 2.236]

J.2 Recorded penalty mass and development selection

The positive-mean replacement runs record both penalty components at nine training states per run. Across 24 runs, none of these 216 recorded states has at most 450 positive costs. At the final recorded states, however, the relative gap $(12\,\mathrm{mean}(c) - 3\mathcal{T}_{0.25})/(12\,\mathrm{mean}(c))$ ranges from 0.0002% to 1.5922%. Thus many positive costs lie outside the top quarter while carrying little total mass. The original Tail histories omit these components; the replacement logs describe their recorded states, not every update or the Tail training trajectory. Appendix F reports these log ranges and a separate recomputation of the original Sparse FF final checkpoints.

The original development sweep (Table 7) likewise separates evidence cost from privacy benefit. Increasing λ from zero to one and three reduces loss counts in all four settings, whereas privacy is not monotone. The test value $\lambda = 3$ is kept unchanged in this ablation.

J.3 Transitions, sampled trajectories, and downstream answers

The group-level transitions show where the frequency improvement comes from. In sparse retrieval, 99 previously affected groups fall below the 0.02 loss tolerance, while 12 previously unaffected groups cross above it. The corresponding counts are 174 versus 3 for MiniLM and 141 versus 9 for Qwen3. The net decrease therefore reflects many more groups leaving the loss set than entering it. All three controller seeds reduce loss counts in every setting. Appendix D.1 gives the continuous-cost estimates, seed-level counts, and the tolerance scan in Figure 3.

Sampled two-action trajectories agree with the exact-expectation results. Their loss counts fall from 281 to 229 for sparse feed-forward retrieval, 338 to 208 for MiniLM, 338 to 210 for Qwen3, and 266 to 229 for sparse recurrence.

Downstream answers. We also run Qwen3-8B (Qwen Team, 2025) on all 6,000 sparse-retrieval test questions, using the modal action history across the three controller seeds for each policy. Tail-aware reaches answer F1 0.1910, compared with 0.1771 for Utility-only and 0.1944 for Adversarial. Its gain over Utility-only is 0.0138 [0.0076, 0.0204]; its difference from Adversarial is -0.0035 $[-0.0076, 0.0009]$. Thus the evidence-tail improvement coexists with positive answer benefit over the utility reference, without establishing higher answer F1 than the adversarial policy. Counting groups that lose either evidence or answer F1 relative to Utility-only gives 336 for Adversarial and 282 for Tail-aware, a paired decrease of 2.70 [1.55, 3.84] percentage points. Appendix D.2 reports the matched-context evaluation and decoding settings.

Table 17: Privacy under raw-ID/rank ridge and the expanded attacker family on the original sparse-policy rollouts. Values are percentage points with paired 95% intervals. B_{Tail} is relative to Utility-only; $\Delta B = B_{\mathrm{Tail}} - B_{\mathrm{Adv}}$. Expanded takes the group/fit maximum over the original four reporters and the deterministic new ridge reporter.

Setting	Attackers	B_{Tail}	ΔB
Sparse FF	Sketch	5.027 [4.484, 5.572]	-0.043 [-0.425, 0.320]
Sparse FF	ID/rank	6.728 [6.099, 7.371]	-0.717 [-1.092, -0.342]
Sparse FF	Expanded	7.891 [7.252, 8.555]	-0.548 [-0.974, -0.133]
Sparse GRU	Sketch	4.984 [4.451, 5.538]	-0.384 [-0.729, -0.052]
Sparse GRU	ID/rank	6.533 [5.866, 7.231]	-1.261 [-1.589, -0.919]
Sparse GRU	Expanded	7.684 [7.014, 8.380]	-1.183 [-1.568, -0.794]

J.4 Computing the expected utility during training

From cached retrievals to a policy update. For each training question, the returned document sets and their support overlaps are fixed. Each first-action state is computed from its retrieved embeddings; the controller then scores the available second actions in that state. Weighting the cached union recalls by the joint probabilities yields one expected utility per question. The reference utility is fixed throughout this process, so changes in positive cost track changes in the learned policy. Privacy learning still uses sampled traces, allowing its observation channel and the exact utility calculation to share the same action space.