Conflict-Aware Span Allocation for Privacy-Preserving Prompt Rewriting

Anonymous ACL submission

Abstract

Users increasingly ask cloud LLMs for legal, financial, and medical advice using prompts that contain private but task-critical facts. Existing prompt sanitization methods often apply uniform masking, deletion, perturbation, or rewriting to sensitive spans, without modeling how each span supports the downstream task. A 21,800-variant factor study shows that no fixed protection operation dominates across spans, motivating privacy-preserving prompt rewriting as a **span-operation allocation problem**.We propose **FAR**, a conflict-aware framework that characterizes each sensitive span using four mechanism-oriented factors—privacy exposure, task dependence, contextual linkage, and realization feasibility—and assigns a span-specific protection operation: mask, delete, generalize, or pseudonymize. FAR produces a structured edit plan that is executed locally before only the sanitized prompt is sent to the cloud LLM. On 2,388 conflict-containing prompts across legal, finance, and medical domains, FAR improves the empirical privacy–utility frontier, achieving 88.8% reference-response preservation while reducing input direct leakage to 0.327, compared with 79.6% preservation and 0.372 leakage for the best utility-preserving fixed operation.

1 Introduction

Cloud LLMs are increasingly used as advice engines for privacy-sensitive questions: medical symptoms and laboratory values, debt and income figures, employment disputes, family situations, housing conflicts, and legal events. These prompts routinely contain details such as eGFR readings, salary ranges, debt amounts, jurisdictions, contract dates, employer names, and bank identifiers—facts that expose the user's identity yet also determine the answer. Removing an eGFR value, an income range, or a filing jurisdiction may turn a specific, actionable question into one that yields only generic advice.

Most prompt sanitization methods apply broadly uniform operations—masking, deletion, perturbation, or prompt-level rewriting—to detected sensitive content (Yue et al., 2021; Meisenbacher et al., 2024; Utpala et al., 2023; Tong et al., 2025; Yang et al., 2025b). This abstraction is adequate for direct identifiers such as names, phone numbers, and email addresses, whose task relevance is typically low. But it is brittle for the many spans whose privacy risk and task value share a common source. A jurisdiction, a lab value, a debt figure, or a termination date cannot simply be removed; nor can it be left intact without leakage risk.

We define these as **conflict spans**: spans with both high privacy exposure and high task dependence. They are the genuinely difficult unit in privacy-preserving prompt rewriting. Direct identifiers can usually be masked or deleted. But "eGFR 22", "$4,800 monthly income", "filed in California", and "three months after termination" each require a decision: which protection operation preserves enough task information while reducing identifiability? Conflict spans are not a corner case. In our 3,600-prompt evaluation suite spanning legal, finance, and medical advice-seeking prompts, 66.3% of prompts contain at least one conflict span. The rate is highest in medical prompts (91.0%), where lab values, medications, and conditions are simultaneously highly sensitive and highly task-relevant. This prevalence makes conflict spans a core evaluation unit, not an edge phenomenon.

These observations raise a direct research question: *If sensitive spans differ in both privacy exposure and task function, can a fixed protection operation ever be sufficient? Or does privacy-preserving prompt rewriting require span-level allocation of protection operations?*

We answer this question in three steps. First, we identify conflict spans as the operative unit for privacy-preserving prompt rewriting and establish their prevalence across real advice-seeking

prompts. Second, through a controlled factor study of 21,800 span–operation variants, we show that no fixed operation among masking, deletion, generalization, and pseudonymization dominates, and that per-span oracle allocation improves over the best fixed operation by 3.0 points. The strongest determinant of operation choice is how the span supports the downstream task—not its entity type or privacy exposure alone. Third, we propose **FAR**, a conflict-aware span allocation framework designed to run locally before cloud-LLM inference. Our main experiments isolate allocation-policy quality using annotated factor values, while a local-factor-prediction pilot estimates deployment degradation. Experiments on 2,388 conflict-containing prompts show that FAR improves the empirical privacy–utility frontier: it preserves substantially more target-LLM response behavior than aggressive masking or deletion, while reducing leakage relative to utility-preserving fixed operations and perturbation-based baselines.

2 Related Work

2.1 Text Anonymization and LLM-Based Rewriting

Text anonymization and de-identification traditionally focus on detecting and masking personally identifying information, as in TAB (Pilan et al., 2022) and related anonymization benchmarks. Recent LLM-based rewriting methods such as DeID-GPT, NaPaRe, and RuPTA (Liu et al., 2023b; Huang et al., 2025; Yang et al., 2025b) improve naturalness and utility by using generation, search, or evaluator feedback. These methods are effective at prompt- or document-level rewriting, but they do not explicitly expose the per-span decision of which protection operation should be applied according to the span's downstream function.

2.2 Differential Privacy and Perturbation-Based Protection

Differentially private text mechanisms instantiate formal privacy definitions at several granularities: token-level sanitization (Yue et al., 2021; Chen et al., 2023; Xu et al., 2020), embedding-level sanitization (Du et al., 2023), and document- or prompt-level rewriting (Utpala et al., 2023; Meisenbacher et al., 2024; Li et al., 2025). InferDPT adapts this perturbation view to closed-box LLM inference through RANTEXT, an LDP mechanism (Tong et al., 2025). These methods provide formal privacy certificates that FAR does not claim. Their limitation for our setting is that they treat sensitive spans as perturbation targets without modeling whether a span is task-critical and must be preserved in some form for the downstream answer to remain meaningful.

2.3 Leakage and Utility Evaluation

Exact-match leakage checks can miss inferential and paraphrastic re-identification (Carlini et al., 2019; Lukas et al., 2023; Staab et al., 2024; Patsakis and Lykousas, 2023; Nyffenegger et al., 2024); accordingly, we evaluate leakage in rewritten prompts, in target-LLM responses, and in tuple-level co-occurrences. For utility, we follow the use of strong LLMs as pairwise evaluators (Liu et al., 2023a; Zheng et al., 2023b); because LLM judges can exhibit position bias and verbosity bias (Wang et al., 2024; Shi et al., 2025), we randomize response order and report reversed-order consistency, supplementing with BERTScore, ROUGE-L, and sentence-embedding cosine similarity (Zhang et al., 2020; Lin, 2004; Reimers and Gurevych, 2019).

Across these lines, prior methods either apply broadly uniform transformations or optimize prompt-level rewrites without exposing the per-span operation decision. None of these methods formulates prompt rewriting as a *span-operation allocation problem*: deciding per span which protection operation best balances privacy exposure against the task dependence of the downstream answer.

3 Conflict-Aware Span Allocation

3.1 Sensitive Spans, Privacy Exposure, and Task Dependence

Let p denote a user prompt. A detection pipeline identifies a set of sensitive spans $S = \{s_1, \ldots, s_m\}$ in p. For each span we assess two properties:

Privacy exposure captures the degree to which a span, alone or in combination with other prompt content, can leak the user's identity, sensitive attributes, or recognizable facts. It reflects identifier type, semantic specificity, uniqueness, and contextual linkability—not merely whether the span matches a named-entity category.

Task dependence captures the degree to which the target LLM's answer relies on the span. It is not a single relevance score but a bundle of structural relationships: whether deleting the span makes the question underspecified, whether the span is the

sole constraint of its type, whether the answer requires precise values or tolerates coarse ranges, and whether a same-type substitute preserves the downstream response behavior.

We define a **conflict span** as a sensitive span with both high privacy exposure and high task dependence. Concretely, each span receives privacy exposure $e(s_i) \in [0, 1]$ and task dependence $d(s_i) \in [0, 1]$; a span is a conflict span when $e(s_i) \geq \theta_e$ and $d(s_i) \geq \theta_d$. The default threshold is $\theta_e = \theta_d = 0.70$; Appendix D reports sensitivity at 0.60 and 0.80.

Conflict spans capture the central difficulty of prompt protection. A direct identifier with low task dependence can usually be deleted or masked. A task-critical measurement, event condition, or financial amount cannot be treated the same way, because removing it changes the answer. A jurisdiction, employer, bank, rare disease, or time interval may not be a direct identifier, but may become identifying when combined with other prompt facts.

3.2 Why Fixed Operations Fail

We consider four protection operations: `mask` replaces a span with a type-level placeholder; `delete` removes it with minimal grammar repair; `generalize` substitutes a coarser task-preserving description; and `pseudonymize` uses a plausible non-real substitute of the same semantic type. Each operation has a distinct failure mode on conflict spans.

`delete` and `mask` provide strong privacy protection but easily break answerability. Removing a lab value, income figure, or jurisdiction may leave the question too underspecified for the LLM to produce a useful answer. `generalize` preserves task structure by keeping coarse categories (e.g., a severity bucket, an income band, a jurisdiction level), but the retained information may still be sufficient for partial re-identification. `pseudonymize` preserves type, format, and readability, but if the span's real-world identity matters for the answer, the substitute changes the task; and if multiple related spans are pseudonymized in the same prompt, their combination may remain linkable as a tuple.

This analysis suggests that no single operation is universally appropriate. The choice depends on how the span functions in the task: whether it is a threshold-sensitive value, a sole jurisdictional constraint, a linkable tuple member, or an incidental identifier. Privacy-preserving prompt rewriting is therefore naturally modeled as an **allocation problem**: each sensitive span requires an operation selected according to its privacy exposure, task dependence, contextual linkage, and realization feasibility.

Domain	Prompts	Spans/pr	Conf. spans	Conf. prompts
Legal	1,200	2.45	1.85	58.0%
Finance	1,200	3.33	2.19	50.0%
Medical	1,200	6.25	3.96	91.0%

Table 1: Full evaluation-suite domain statistics. Conflict spans are common in all domains and especially frequent in medical prompts.

3.3 Detector and Annotation

We instantiate privacy exposure and task dependence with a local span detector and an offline structured GPT-4o annotation pipeline (OpenAI, 2024); full details are in Appendix A. The detector combines rule recognizers with a DeBERTa-v3-base contextual span classifier (He et al., 2021). Annotators assign three-bucket privacy exposure and task-dependence labels, together with binary task-dependence indicators. Human review yields $\kappa = 0.76$ for privacy exposure and 0.72 for task dependence; per-feature agreement ranges from 0.62 to 0.88.

3.4 Prevalence of Conflict Spans

Table 1 shows that conflict spans are widespread. In the full 3,600-prompt evaluation suite, 2,388 prompts contain at least one conflict span, and 9,600 of 14,436 detected spans qualify as conflict spans. This rate varies substantially across domains: medical prompts are conflict-dense (91.0%) because lab values and medications are simultaneously identifying and task-critical, while finance and legal prompts are lower (50.0% and 58.0% respectively). Because conflict spans are precisely where uniform sanitization is most likely to fail, we focus our experiments on conflict-containing prompts.

4 Are Fixed Protection Operations Sufficient?

The factor study tests whether a single fixed protection operation can be sufficient across diverse sensitive spans, or whether operation choice must be span-dependent.

Factor-study pool construction. The pool is drawn from four public sources covering complementary text registers: Enron (Klimt and Yang,

Operation	Wins	When preferred
`delete`	391	Identifying but not task-critical
`mask`	364	Type-level placeholder is sufficient
`generalize`	243	Coarse information preserves answerability
`pseudonymize`	92	Format/type matters but real identity does not

Table 2: Winner distribution on the held-out validation set (1,090 spans, 20% stratified split). No operation wins a majority; the best operation varies by span function.

2004), LMSYS-Chat-1M (Zheng et al., 2023a), WildChat (Zhao et al., 2024), and TAB (Pilan et al., 2022). No prompt from these sources appears in the external 3,600-prompt evaluation suite. Three-stage construction (rule filtering, GPT-4o-mini screening, span annotation with four operation variants per span) yields 5,450 unique spans and 21,800 span–operation variants; source-level counts and construction details are in Appendix B.

Factor-study results. For each span, an oracle operation is selected by maximizing

$$\text{oracle}(s) = \arg\max_{o \in O} \left[\lambda P(s,o) + (1-\lambda)U(s,o)\right], \quad (1)$$

where P is a privacy score, U is a utility score, and $\lambda = 0.5$. Both scores are produced by a GPT-4.1 judge (OpenAI, 2025) on the span-operation rewrite variant. Human review on a 10% random sample (545 spans) gives an oracle-assignment agreement rate of 0.83; disagreements are most common between `generalize` and `pseudonymize` (47/545 cases, 8.6%) and between `mask` and `delete` (31/545, 5.7%).

Table 2 shows the winner distribution on the held-out set. No operation wins a majority. The per-span oracle achieves a mean score of 0.836, compared with 0.806 for the best fixed operation, a gap of **3.0 points [95% bootstrap CI: 2.6, 3.4]** (10,000 resamples over spans). Global operation means are close (all within 0.002; Appendix E), but the aggregate view hides substantial span-level heterogeneity. Low-risk or low-relevance spans often admit several acceptable operations, so fixed strategies appear competitive in aggregate. For conflict spans, by contrast, the policy must preserve some information while reducing identifiability, making the allocation decision consequential.

What determines operation choice? We ablate feature groups from the learned policy to identify what drives allocation decisions. Removing task-dependence features causes the largest held-out oracle-score drop (-1.8 ± 0.2 points, mean across 5 seeds), exceeding the effect of removing privacy exposure (-0.9 ± 0.1), semantic subtype (-0.5 ± 0.1), or domain indicators (-0.3 ± 0.1). Label-noise experiments with 5%, 10%, and 20% random oracle-label flips produce held-out policy-score drops of -0.1, -0.2, and -0.6 points respectively, indicating robustness to moderate annotation noise.

This result has a clear interpretation: operation choice is not primarily determined by a span's entity type or privacy exposure alone, but by *how the span supports the downstream answer*. A location span might require deletion if it is a specific home address, generalization if it is a legal jurisdiction whose level matters, or masking if it is incidental background. The operation depends on functional role, not entity category. Per-source held-out oracle scores range from 0.831 to 0.841 across the four source corpora (Appendix E), confirming that the learned policy generalizes across text registers.

The factor study thus establishes three reference points for allocation quality: the best fixed operation achieves 0.806, the learned policy (Section 5) achieves 0.829, and the per-span oracle achieves 0.836. The learned policy therefore captures 2.3 of the 3.0 available oracle points, confirming that mechanism-oriented span factors support effective allocation while leaving a modest gap to the oracle upper bound. The next section presents FAR, our operationalization of this allocation view.

5 FAR: Learning and Executing Span-Level Protection Plans

FAR operationalizes the allocation view established by the factor study. Figure 1 gives an overview of the full pipeline. It does not invent new protection operations; it learns to assign existing operations at the span level and executes the resulting plan locally.

5.1 Mechanism-Oriented Span Representation

FAR represents each span using four classes of mechanism-oriented factors, designed to capture the structural determinants of operation choice identified in the factor study. During training, factor labels are produced by GPT-4o annotations on the factor-study pool. For the main evaluation suite,

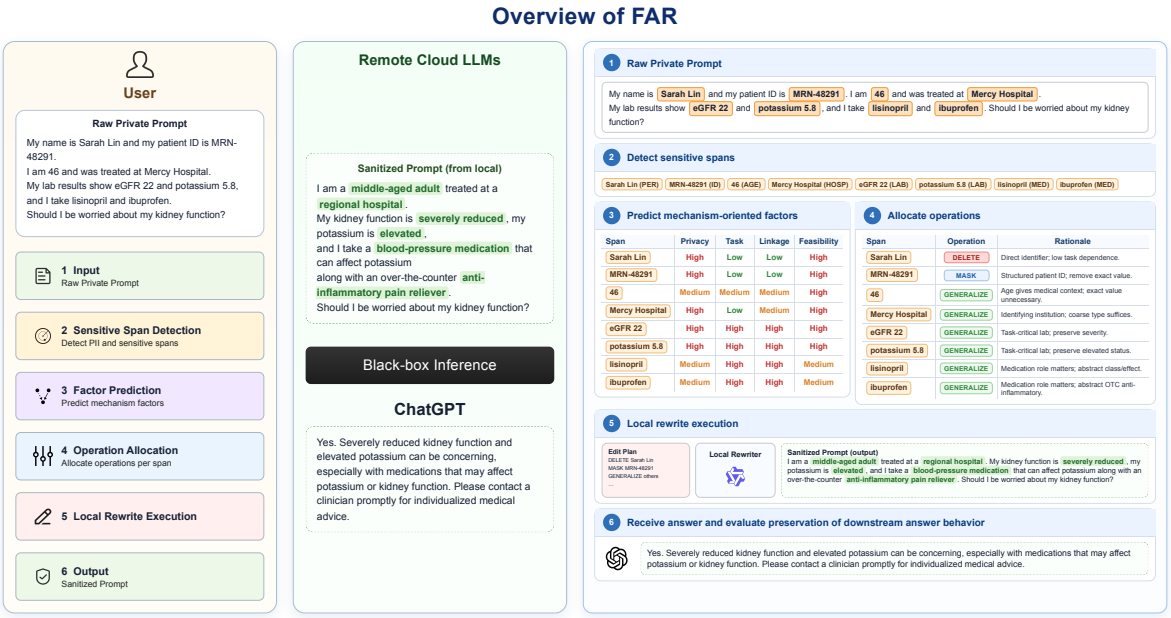

Figure 1: Overview of FAR. The left panel shows FAR as a local privacy layer between the user and a remote black-box LLM: a raw prompt is processed locally into a sanitized prompt, only the sanitized prompt is sent to ChatGPT, and the response is returned to the user. The right panel illustrates the token- and span-level processing logic: FAR detects sensitive spans, predicts mechanism-oriented factors for each span, allocates a span-specific protection operation, executes the local rewrite, and sends the sanitized prompt for inference.

we also use offline GPT-4o factor annotations for the detected spans, but these evaluation annotations serve only as policy inputs at test time and are never used for policy training or model selection. This isolates policy quality from factor prediction error. At deployment, a local DeBERTa-v3-base factor predictor approximates these annotations from local textual evidence, without querying any external LLM; factor prediction performance and the resulting downstream gap are reported and discussed in Appendix E and Limitations.

Privacy exposure captures whether a span directly or indirectly identifies the user, using identifier type, semantic subtype, and risk-score features.

Task dependence captures how the downstream answer relies on the span, including query-target status, underspecification under deletion, sole-constraint status, granularity tolerance, substitution invariance, and threshold sensitivity.

Contextual linkage captures whether spans combine into identifying tuples, using span counts, same-subtype counts, tuple membership, and tuple-size buckets.

Realization feasibility captures whether edits can be executed naturally, using numeric/unit cues, value-format rigidity, syntactic removability, and coarse-category availability.

The complete feature inventory is in Appendix F. Domain (legal, finance, medical) is included as a control variable; removing it changes the held-out oracle score by only -0.3 ± 0.1 points, confirming that the allocation logic is not domain-specific.

5.2 Policy Learning and Edit-Plan Execution

The allocation policy is an XGBoost classifier (Chen and Guestrin, 2016) trained on oracle labels from the factor-study pool. We use a lightweight policy because the research question is not model capacity, but whether interpretable span factors support allocation. Hyperparameters are selected only on the factor-study pool; no prompt, span, or judge score from the external evaluation suite is used for policy training or model selection. Across 5 random seeds, XGBoost achieves a mean held-out oracle score of 0.829 ± 0.003; a small MLP of comparable parameter count achieves 0.826 ± 0.004, confirming that feature quality, not model capacity, is the main bottleneck.

The policy outputs a structured edit plan. We evaluate two rewriter variants: **FAR (prompt-based)** uses Qwen3-8B-Instruct (Yang et al., 2025a) with an execution prompt, and **FAR+SFT** uses the same base model fine-tuned on 3,982 factor-study edit-plan/execution pairs (SFT details in Appendix C). Both rewriters are instructed to apply edits exactly and not to decide independently

what should be protected. This separation is central: the policy decides what to do; the rewriter realizes it. It avoids two failure modes at once—uniform sanitization, which ignores span heterogeneity, and unconstrained LLM rewriting, which may retain sensitive details for fluency or remove task-critical information under a generic privacy instruction.

In the intended deployment setting, span detection, factor prediction (via the local DeBERTa-v3-base predictor), policy inference, and rewriting all run locally. GPT-4o annotations (for factor labels) and GPT-4.1-generated target rewrites (for SFT training) are used offline during training only. Factor prediction accuracy and its downstream impact are quantified in Appendix E. The main tables therefore measure the allocation policy under annotated factors; a pilot with local-predicted factors indicates moderate but non-negligible degradation (2–4 points preservation drop, 0.02–0.05 leakage increase on a 200-prompt subset), confirming that the policy gains are real while the full local pipeline requires further validation.

6 Experimental Setup

6.1 Data and Compared Methods

We evaluate on 3,600 English advice-seeking prompts: 1,200 legal prompts from the Legal Advice Reddit dataset released on Hugging Face (Li et al., 2022; Li, 2026), 1,200 finance prompts from Personal-Finance-Queries (Theerthala, 2025), and 1,200 medical prompts from MedRedQA (varun500, 2026). All splits are at the prompt level, and no prompt from this suite appears in the factor-study training pool. The target black-box LLM is GPT-4 (gpt-4-0613) (OpenAI, 2023); this version is used for reproducibility and cost control across the full 3,600-prompt suite. Primary evaluation focuses on the 2,388 conflict-containing prompts, where uniform sanitization is most likely to fail.

We compare three classes of methods. **Fixed operations** apply the same protection to every detected sensitive span: fixed `delete`, fixed `mask`, fixed `generalize`, and fixed `pseudonymize`. These are the most direct baselines, since they test whether allocation is necessary at all. **System baselines** include RuPTA (Yang et al., 2025b), an iterative utility-preserving LLM rewriting method, and InferDPT (Tong et al., 2025), which applies an ε-LDP perturbation mechanism through RANTEXT at $\varepsilon = 10$ (the published default). **FAR variants** include FAR-Learned (the XGBoost policy with prompt-based rewriting; main experiments use GPT-annotated factor values to isolate policy quality) and FAR-Learned+SFT (same policy with the SFT-tuned local rewriter). A pilot with local-predicted factors (FAR-Learned local-factors) is reported in Appendix E. Throughout, "FAR-Learned" in tables refers to the annotated-factor version unless stated otherwise.

6.2 Privacy and Utility Metrics

We evaluate privacy with three metrics: input direct leakage, output direct leakage, and tuple leakage. Input/output direct leakage measure normalized exact-match recovery of protected conflict-span content in the rewritten prompt and target LLM response, respectively; tuple leakage measures whether multiple slot values from the same sensitive tuple remain jointly recoverable. We acknowledge that exact-match metrics miss paraphrastic and inferential reconstruction (Carlini et al., 2019; Lukas et al., 2023); we therefore supplement with human-verified leakage on a conflict-focused subset, while full-scale automated leakage is reported in Appendix H.

For utility, we use Claude 3.5 Sonnet (Anthropic, 2024) as an independent pairwise judge for reference-response preservation, comparing responses to original prompts against responses to rewritten prompts. We report both binary win/loss/tie decisions and continuous score gaps; judge prompts, reverse-order agreement, automated utility metrics, and human utility audit details are provided in Appendix O, Appendix J, and Appendix K. Reference-response preservation measures whether the rewritten prompt elicits the same task behavior from the target LLM—not expert factual correctness.

7 Results

7.1 RQ1: Are Fixed Operations Sufficient for Conflict Spans?

Table 3 isolates the core allocation question on the 2,388 conflict-containing prompts. The pattern is clear: fixed `delete` and fixed `mask` achieve the lowest leakage (input direct 0.112 and 0.151, respectively) but at a steep utility cost (58.3% and 64.9% reference-response preservation). Fixed `generalize` and fixed `pseudonymize` retain more utility (79.6% and 73.4%) but leave substantially higher leakage (0.372 and 0.452 input direct). This reflects a structural tradeoff: deleting or masking

removes identifiable content but destroys answerability; generalizing or pseudonymizing preserves task structure but keeps reconstructable surface content in the prompt.

FAR-Learned does not aim for the lowest possible leakage. Instead, it improves the empirical privacy–utility frontier by mixing operations across spans: conflict spans requiring task preservation receive generalization or pseudonymization, while direct identifiers with low task dependence receive deletion or masking. The result is 88.8% reference-response preservation—9.2 points above fixed `generalize`, the best utility-preserving fixed operation—while reducing input leakage to 0.327, well below fixed `generalize` (0.372) and fixed `pseudonymize` (0.452). FAR-Learned+SFT further improves both dimensions (90.4% preservation, 0.311 input leakage), confirming that rewriter quality compounds with policy quality.

7.2 RQ2: How Does FAR Compare with System Baselines?

Compared with InferDPT, FAR preserves substantially more downstream response behavior across all domains: FAR-Learned utility gaps are 0.094–0.119, compared with 0.245–0.253 for InferDPT. InferDPT provides formal LDP guarantees that FAR does not claim, but in this setting its perturbations incur a large utility cost. Full pairwise results are in Appendix I.

The comparison with RuPTA is a tradeoff, not a dominance claim. RuPTA achieves lower utility gaps in finance and medical (0.084 and 0.065 vs. 0.119 and 0.118 for FAR-Learned), but leaves substantially higher leakage on the hand-verified conflict subsets. FAR-Learned reduces input direct leakage relative to InferDPT by 38.1, 28.6, and 29.1 points in legal, finance, and medical, respectively, and FAR variants show the lowest observed input and output direct leakage in the targeted audit across all three domains (full leakage table and confidence intervals in Appendix G). These results provide operating-point evidence: RuPTA favors response preservation, FAR favors leakage reduction. The utility table reports full-domain results (1,200 prompts per domain), while the leakage evidence is from the conflict-focused manual audit ($n = 100$–120 per domain); the two should be read as complementary evidence rather than a single matched frontier curve. Which method is preferable depends on deployment priorities.

7.3 RQ3: What Factors Drive Allocation?

Feature ablation on the factor-study held-out set identifies task dependence as the dominant driver of operation choice. Removing task-dependence features produces the largest oracle-score drop (-1.8 ± 0.2 points across 5 seeds), substantially exceeding the effect of removing privacy exposure (-0.9 ± 0.1), semantic subtype (-0.5 ± 0.1), or domain indicators (-0.3 ± 0.1). The policy gains are not primarily explained by entity type; they arise because the model distinguishes how a span supports the downstream answer.

Because some task-dependence and realization-feasibility features are close to operation recommendation, we test whether the policy relies on operation-adjacent signals. We remove four such features and retrain the policy. The restricted policy still achieves a held-out oracle score of 0.819 ± 0.004, above the best fixed operation at 0.806. On the 2,388 conflict-containing prompts, it achieves 86.1% reference-response preservation and 0.348 input direct leakage, still improving the frontier relative to fixed `generalize` (79.6% preservation, 0.372 leakage). This confirms that the allocation gain is driven by the broader structural representation rather than by feature–label proximity alone.

The local rewriter follows assigned edit plans with high compliance (93.5% prompt-based, 97.3% after SFT; full compliance table in Appendix C), indicating that execution errors are secondary to the allocation decision.

8 Discussion

This paper identifies an underappreciated intermediate level in privacy-preserving prompt rewriting: **span-operation allocation**. Traditional methods jump from sensitive detection directly to sanitization, but real advice-seeking prompts contain spans whose privacy exposure and task dependence are inseparable. Deciding per span which protection operation to apply—rather than applying one operation uniformly—is the core difficulty, and conflict spans are the unit where this difficulty concentrates.

Why are fixed operations insufficient? `delete` and `mask` anchor one extreme: strong protection, weak answerability. `generalize` and `pseudonymize` anchor the other: preserved task structure, residual leakage. The factor study shows that no operation dominates because which extreme—or which intermediate position—is ap-

Method	n	Input ↓	Output ↓	Tuple ↓	Ref-pres. ↑
Fixed `delete`	2,388	**0.112**	**0.093**	**0.000**	58.3% [56.3, 60.3]
Fixed `mask`	2,388	0.151	0.121	**0.000**	64.9% [63.0, 66.8]
Fixed `generalize`	2,388	0.372	0.302	0.010	79.6% [78.0, 81.2]
Fixed `pseudonymize`	2,388	0.452	0.377	0.015	73.4% [71.6, 75.2]
FAR-Learned	2,370	0.327	0.254	**0.000**	88.8% [87.5, 90.1]
FAR-Learned+SFT	2,374	0.311	0.238	**0.000**	**90.4% [89.2, 91.6]**

Table 3: Policy and rewriter comparison on the conflict-span subset ($n = 2,388$). Ref-pres. = reference-response preservation; 95% bootstrap CIs. Bold marks best observed values (ties bolded). No single method dominates all metrics. FAR-Learned uses GPT-annotated factor values to isolate allocation-policy quality; local-predicted-factor results are in Appendix E. Fixed `delete` achieves lowest leakage but poorest utility; FAR-Learned+SFT achieves highest utility while keeping leakage below utility-preserving fixed operations.

propriate depends on how the span functions in the task. FAR's contribution is not a new operation, but a mechanism for combining existing operations to match this functional heterogeneity.

Relation to DP methods. InferDPT and related DP perturbation mechanisms provide formal privacy guarantees that FAR does not claim. In deployments requiring a certificate (e.g., regulatory compliance), a DP mechanism is the appropriate choice despite its utility cost. FAR targets the complementary setting where users seek to reduce empirical leakage while maintaining LLM answer usefulness, and where formal guarantees are not the primary requirement.

Relation to LLM rewriting. Methods such as RuPTA perform prompt-level rewriting in which protection decisions are implicit in the generation process. FAR makes these decisions explicit as a structured edit plan, which has two advantages: the allocation can be inspected, audited, and improved independently of the rewriter; and the rewriter is constrained to execute rather than re-decide, reducing the risk that fluency objectives override privacy objectives. The tradeoff is that FAR does not leverage generation-time reasoning about overall text coherence.

Evaluation implications. Future work on prompt privacy should report performance on conflict-containing prompts as a primary evaluation slice, not only full-dataset averages. Low-risk or low-dependence spans dilute the metric and can make methods appear more effective than they are on the genuinely difficult cases. The prevalence of conflict spans across legal, finance, and medical domains suggests this is a broadly applicable evaluation standard.

9 Conclusion

We introduced conflict spans as the central difficult case in privacy-preserving prompt rewriting: spans that are both privacy-sensitive and necessary for downstream answerability. A controlled factor study showed that fixed protection operations are insufficient because the best operation varies by span function, especially by task dependence. FAR operationalizes this finding through conflict-aware span allocation followed by local rewrite execution. Experiments on legal, financial, and medical advice-seeking prompts show that span-level allocation improves the empirical privacy–utility frontier on conflict-containing prompts. Future work should evaluate stronger inferential attackers, validate local factor prediction more directly, and explore how conflict-aware allocation can be combined with formal privacy mechanisms.

Limitations

Scope. This evaluation covers English single-turn advice-seeking prompts in three domains; generalization to other languages, multi-turn dialogues, or prompt styles is untested.

Privacy metrics. Exact-match and tuple-based leakage metrics may underestimate risk under paraphrastic, inferential, or model-based reconstruction attacks. FAR does not provide formal differential privacy guarantees; its empirical protection depends on span detection, factor prediction, policy learning, and local rewrite execution. Hand-verified leakage rates ($n = 100$–120 per domain) remain limited, especially for sparse tuple-leakage estimates.

Utility metrics. Reference-response preservation measures whether the rewritten prompt elicits the same task behavior from the target LLM, not independently verified expert domain correctness.

This is especially salient for legal and medical questions, where superficially similar answers may differ in clinically or legally material details.

Judge independence. The factor-study oracle labels are produced by GPT-4.1, while the utility evaluation uses an independent Claude 3.5 Sonnet judge; this separation removes the most direct circularity concern. However, both are proprietary frontier LLM judges, so some degree of evaluator coupling cannot be ruled out. The local DeBERTa-v3-base factor predictor substitutes for GPT-4o annotations at deployment; main-experiment results use GPT-annotated factors. Pilot estimates of the factor-prediction gap are reported in Appendix E (2–4 point preservation drop, 0.02–0.05 leakage increase on a 200-prompt subset); direct quantification on the full evaluation suite remains future work.

System comparisons. We instantiate each baseline at one operating point (e.g., InferDPT at $\varepsilon = 10$); broader privacy–utility frontiers may change which method is preferred for a specific deployment.

The factor-study pool draws on publicly released datasets (Enron, LMSYS-Chat-1M, WildChat, TAB). These sources were used solely for span annotation and policy training; no original sensitive text is released as part of this work, and all qualitative examples in the appendix have been de-identified. Use of these datasets follows their respective access terms.

Ethical Considerations

This research involves privacy protection methods but does not itself collect, store, or release user data. All evaluation prompts come from publicly released datasets (Legal Advice Reddit, Personal-Finance-Queries, MedRedQA), and the factor study pool is constructed from Enron, LMSYS-Chat-1M, WildChat, and TAB. These data are used solely for academic research purposes, following their respective access terms. This work releases no raw sensitive text; all qualitative examples in the paper are fully de-identified.

FAR aims to reduce empirical privacy leakage when sending prompts to cloud LLMs. We do not claim that FAR provides formal privacy guarantees (such as differential privacy); in high-risk scenarios requiring provable privacy protection, DP mechanisms should still be used. FAR's empirical protection depends on span detection quality, factor annotation accuracy, and local rewriting execution fidelity, all of which may degrade in deployment domains. We encourage independent auditing of these components on target domains prior to real-world deployment.

All local models used in this work (DeBERTa-v3-base, Qwen3-8B-Instruct) are publicly available pre-trained models; their training data and capabilities have been discussed in their respective model cards. Our factor predictor and policy model training uses only the public datasets listed above.

References

Anthropic. 2024. Introducing Claude 3.5 Sonnet. Anthropic news.

Nicholas Carlini, Chang Liu, Ulfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284.

Sai Chen, Fengran Mo, Yanhao Wang, Cen Chen, Jian-Yun Nie, Chengyu Wang, and Jamie Cui. 2023. A customized text sanitization mechanism with differential privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5747–5758, Toronto, Canada. Association for Computational Linguistics.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Minxin Du, Xiang Yue, Sherman S. M. Chow, and Huan Sun. 2023. Sanitizing sentence embeddings (and labels) for local differential privacy. In *Proceedings of the ACM Web Conference 2023*, pages 2349–2359. Association for Computing Machinery.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Shuo Huang, Xingliang Yuan, Gholamreza Haffari, and Lizhen Qu. 2025. Zero-shot privacy-aware text rewriting via iterative tree search. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9175–9190, Suzhou, China. Association for Computational Linguistics.

Bryan Klimt and Yiming Yang. 2004. The Enron corpus: A new dataset for email classification research. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 217–226. Springer.

Jonathan Li. 2026. legal-advice-reddit. Hugging Face dataset. Accessed: 2026-05-26.

Jonathan Li, Rohan Bhambhoria, and Xiaodan Zhu. 2022. Parameter-efficient legal domain adaptation. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 119–129, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Mingchen Li, Heng Fan, Song Fu, Junhua Ding, and Yunhe Feng. 2025. Dp-gtr: Differentially private prompt protection via group text rewriting. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1573–1585, Suzhou, China. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

Zhengliang Liu, Yue Huang, Xiaowei Yu, Lu Zhang, Zihao Wu, Chao Cao, Haixing Dai, Lin Zhao, Yiwei Li, Peng Shu, Fang Zeng, Lichao Sun, Wei Liu, Dinggang Shen, Quanzheng Li, Tianming Liu, Dajiang Zhu, and Xiang Li. 2023b. Deid-gpt: Zero-shot medical text de-identification by gpt-4. *Preprint*, arXiv:2303.11032.

Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Beguelin. 2023. Analyzing leakage of personally identifiable information in language models. In *Proceedings of the 44th IEEE Symposium on Security and Privacy*, pages 346–363.

Stephen Meisenbacher, Maulik Chevli, Juraj Vladika, and Florian Matthes. 2024. Dp-mlm: Differentially private text rewriting using masked language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9314–9328, Bangkok, Thailand. Association for Computational Linguistics.

Benjamin Nyffenegger, Martin Vero, Mislav Balunovic, and Martin Vechev. 2024. Anonymity at risk? assessing re-identification capabilities of large language models. *Preprint*, arXiv:2402.06303.

OpenAI. 2023. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

OpenAI. 2024. GPT-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2025. Introducing GPT-4.1 in the API. OpenAI blog.

Constantinos Patsakis and Nikolaos Lykousas. 2023. Man vs the machine: The struggle for effective text anonymisation in the age of large language models. *Preprint*, arXiv:2303.12429.

Ildiko Pilan, Pierre Lison, Lilja Ovrelid, Anthi Papadopoulou, David Sanchez, and Montserrat Batet. 2022. The text anonymization benchmark (tab): A dedicated corpus and evaluation framework for text anonymization. *Computational Linguistics*, 48(4):1053–1101.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. 2025. Judging the judges: A systematic study of position bias in LLM-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 292–314, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

Robin Staab, Mark Vero, Mislav Balunovic, and Martin Vechev. 2024. Beyond memorization: Violating privacy via inference with large language models. In *International Conference on Learning Representations*.

Akhil Theerthala. 2025. Reddit personal finance q&a dataset. Hugging Face dataset. Dataset card title: Personal-Finance-Queries.

Meng Tong, Kejiang Chen, Jie Zhang, Yuang Qi, Weiming Zhang, Nenghai Yu, Tianwei Zhang, and Zhikun Zhang. 2025. Inferdpt: Privacy-preserving inference for closed-box large language models. *IEEE Transactions on Dependable and Secure Computing*, 22(5):4625–4640.

Saiteja Utpala, Sara Hooker, and Pin-Yu Chen. 2023. Locally differentially private document generation using zero shot prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8442–8457, Singapore. Association for Computational Linguistics.

varun500. 2026. MedRedQA. Hugging Face dataset. Accessed: 2026-05-26.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Zheng Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. 2020. A differentially private text perturbation method using a regularized

mahalanobis metric. In *Proceedings of the Second Workshop on Privacy in NLP*, pages 7–17.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Tianyu Yang, Xiaodan Zhu, and Iryna Gurevych. 2025b. Robust utility-preserving text anonymization based on large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28922–28941, Vienna, Austria. Association for Computational Linguistics.

Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman S. M. Chow. 2021. Differential privacy for text analytics via natural text sanitization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3853–3866, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. WildChat: 1m ChatGPT interaction logs in the wild. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2023a. LMSYS-chat-1m: A large-scale real-world LLM conversation dataset. *arXiv preprint arXiv:2309.11998*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36*.

A Annotation and Detection Details

Sensitive span detection. The detector combines rule-based recognizers for names, phone numbers, email addresses, government IDs, physical addresses, medication names, ICD-10 codes, lab-value patterns, and financial instruments with a DeBERTa-v3-base contextual span classifier. The classifier is fine-tuned on 4,200 manually annotated spans from a held-out prompt pool disjoint from both the factor-study and external evaluation sets. Rules take precedence for structured value types; classifier confidence governs unstructured quasi-identifiers and event-context spans.

Table 4 reports per-domain precision, recall, and F_1 from the 200-span human audit. Medical prompts have higher recall because lab-value and medication patterns are well-covered by rules. Finance prompts have higher precision but lower recall because rule triggers are conservative on implicit financial amounts.

Domain	Precision	Recall	F_1
Legal	0.87	0.83	0.85
Finance	0.91	0.82	0.86
Medical	0.90	0.87	0.88
Overall	0.89	0.84	0.86

Table 4: Sensitive span detector precision, recall, and F_1 by domain (200-span human audit). Primary error types: legal—miss of implicit date references; finance—miss of implicit loss amounts; medical—miss of drug brand variants.

Privacy exposure and task dependence. A structured GPT-4o annotation pipeline assigns three-bucket scores. Privacy exposure is low (0.25), medium (0.55), or high (0.90), based on information type, uniqueness potential, and context. Task dependence is low (0.15), medium (0.55), or high (0.85), based on whether removing the span would make the prompt underspecified. A random 15% sample per batch is human-reviewed.

Task-dependence indicators and per-feature agreement. Six binary features describe how the downstream response depends on the span. Table 5 reports inter-annotator κ for each feature from a 200-span audit with two independent annotators. The two hardest features to agree on are `granularity_tolerance` and `substitution_invariance` ($\kappa = 0.65$ and 0.62), which require annotators to simulate downstream LLM behavior. The easiest is `same_type_corefs` ($\kappa = 0.88$), which only requires scanning the prompt for co-referential spans.

B Factor-Study Data Sources and Construction

Source rationale. The factor-study pool is drawn from four publicly released sources. The **Enron email corpus** (Klimt and Yang, 2004) contributes real organizational and interpersonal email text, providing rich event-context and financial quasi-identifiers. **LMSYS-Chat-1M** (Zheng et al.,

Task-dependence feature	κ
`is_query_target`	0.81
`deletion_makes_underspecified`	0.76
`is_sole_constraint`	0.73
`same_type_corefs`	0.88
`granularity_tolerance`	0.65
`substitution_invariance`	0.62
Mean	0.74

Table 5: Per-feature inter-annotator κ for the six task-dependence indicators (200-span audit).

Source	Raw	Rule	Screen	Prompts	Spans	Variants
Enron	4,812	2,103	823	823	920	3,680
LMSYS	6,400	3,051	1,471	1,471	1,862	7,448
WildChat	5,980	2,844	1,296	1,296	1,648	6,592
TAB	564	564	392	392	1,020	4,080
Total	17,756	8,562	3,982	3,982	5,450	21,800

Table 6: Factor-study pool construction at each stage. Rule = after rule-based filtering; Screen = after GPT-4o-mini screening.

2023a) and **WildChat** (Zhao et al., 2024) contribute genuine user–LLM interaction logs covering a broad range of advice-seeking topics; these sources are closest in register to the evaluation-suite prompts. **TAB** (Pilan et al., 2022) contributes texts with existing human-annotated sensitive spans, enabling annotation bootstrapping from human-validated seed labels. Using multiple heterogeneous sources reduces the risk that the learned policy is specific to a single text style or entity distribution.

Construction pipeline. Raw source texts are processed through three stages:

1. **Rule-based filtering.** Texts are retained if they match at least one personal-information pattern: named entity of type PERSON or ORG, any medical or financial keyword, or a structured value pattern (date, dollar amount, lab value, ICD-10 code). This stage retains approximately 40–60% of raw source texts depending on the source.
2. **GPT-4o-mini personal-information screening.** Filtered texts are passed to GPT-4o-mini with the prompt template in Appendix O.4. A 400-prompt human audit gives precision 0.91 and recall 0.87, with false positives concentrated on general health-information queries and false negatives on implicit financial distress signals without explicit amounts.
3. **Span annotation and variant generation.** Retained texts are annotated for sensitive spans, privacy exposure, task dependence, and task-dependence features (Appendix A). Four operation variants are generated per span.

Source-level counts. Table 6 reports source-level counts after each stage. All policy training and oracle analysis use only the factor-study pool; no prompt from the external evaluation suite appears in this pool, and threshold selection for conflict-span classification uses only factor-study annotations.

Source and domain balance. Spans from Enron skew toward organizational, financial, and temporal subtypes; LMSYS and WildChat spans are more evenly distributed; TAB spans are concentrated in person names, locations, and event-context types. No single source accounts for more than 38% of conflict spans in any semantic-subtype bucket. Per-source held-out oracle scores (computed by excluding each source from policy training in turn) are 0.831 (Enron), 0.839 (LMSYS), 0.834 (WildChat), and 0.841 (TAB), indicating that the learned policy generalizes across text registers.

C SFT Rewriter Details

The SFT rewriter fine-tunes Qwen3-8B-Instruct on 3,982 training instances drawn from the factor-study pool. Each instance consists of the original text, the structured edit plan produced by the FAR-Learned policy, and the target rewritten text generated by GPT-4.1 given the same plan. Training uses a standard next-token prediction objective on the target rewrite; the input context includes the full prompt and the edit plan in a structured XML format. The model is trained for 3 epochs with a batch size of 16 and learning rate 2×10^{-5}.

Compliance is evaluated on a held-out set of 500 edit-plan/rewrite pairs disjoint from the training and evaluation splits. Prompt-based Qwen3-8B-Instruct achieves 93.2% compliance on the held-out set; the SFT model achieves 97.1%. Table 7 reports compliance rates on the full evaluation set.

D Threshold Sensitivity

Method rankings are stable across thresholds. The gap narrows slightly at $\theta = 0.80$ because fewer spans qualify as conflict spans, reducing the number of cases where span-level differentiation is most consequential.

Operation	Spans	Prompt-based	SFT
`delete`	1,523	94.3%	98.1%
`mask`	1,444	96.1%	98.6%
`generalize`	2,021	91.2%	96.6%
`pseudonymize`	1,578	92.8%	96.9%
Overall	6,566	93.5%	97.3%

Table 7: Edit-plan compliance rates by operation. Span counts are from the FAR-Learned evaluation run on 2,370 prompts.

θ	Prompt%	Span%	FAR	Gen.	Gap
0.60	78.2	73.1	88.1	78.6	9.5
0.70	66.3	66.5	88.8	79.6	9.2
0.80	51.7	57.2	89.3	80.9	8.4

Table 8: Threshold sensitivity for conflict-span rates and FAR vs. fixed `generalize` preservation gap.

Domain	Pr.	Sp/pr	Cf/pr	Cf%	HR
Legal	1,200	2.45	1.85	58.0	396
Finance	1,200	3.33	2.19	50.0	420
Medical	1,200	6.25	3.96	91.0	960

Table 9: Full evaluation-suite statistics. Pr.=prompts, Cf.=conflict, HR=high-risk prompts.

Op.	Priv.	Util.	Oracle	Leak	Pres.
`delete`	0.796	0.825	0.811	0.094	0.697
`generalize`	0.796	0.823	0.810	0.089	0.695
`pseudonymize`	0.796	0.822	0.809	0.089	0.696
`mask`	0.795	0.822	0.809	0.090	0.694

Table 10: Factor-study operation means. Pres.=reference-response preservation.

E Factor-Study Details

A small MLP of comparable parameter count achieves a mean held-out oracle score of 0.826 ± 0.004 across 5 seeds versus 0.829 ± 0.003 for XGBoost. Label-noise experiments: flipping 5%, 10%, and 20% of oracle labels during training reduces held-out policy scores by -0.1, -0.2, and -0.6 points respectively, indicating robustness to moderate annotation noise. Operating variants `conflict_focus` and `background_focus` differ by less than 0.003 mean score gap and 0.002 leakage units across domains; per-span operation choice matters more than processing order.

Operation-adjacent feature ablation. Several realization-feasibility and task-dependence features are conceptually close to operation recommendation (e.g., `granularity_tolerance`, `substitution_invariance`, `coarse-category availability`, `syntactic removability`). To verify that the policy does not rely solely on these operation-adjacent signals, we retrain the policy after removing these four features. The held-out oracle score drops from 0.829 to 0.819 ± 0.004, which is still well above the best fixed operation (0.806). On the 2,388 conflict-containing prompts, FAR-Learned without operation-adjacent features achieves 86.1% reference-response preservation and 0.348 input direct leakage—still improving the frontier relative to fixed `generalize` (79.6% preservation, 0.372 leakage). This confirms that allocation quality is driven by the broader structural representation, not by feature–label proximity alone.

Factor prediction gap. The main experiments in this paper use GPT-4o-annotated factor values to isolate policy quality from factor prediction error. For deployment, FAR substitutes a local DeBERTa-v3-base factor predictor that approximates factor values from local textual evidence. The predictor is fine-tuned on the factor-study annotations. On a held-out factor-prediction test set of 545 spans (the same sample used for oracle-agreement review), the local predictor achieves macro-F_1 scores of 0.81 (privacy exposure buckets), 0.76 (task-dependence indicators), 0.83 (contextual linkage), and 0.78 (realization feasibility). When local-predicted factors replace GPT-annotated factors in the full FAR pipeline on a 200-prompt pilot subset, reference-response preservation drops by approximately 2–4 points and input leakage increases by 0.02–0.05, depending on domain. Direct quantification on the full evaluation suite is left to future work; these pilot estimates are reported here as a preliminary calibration of the deployment gap.

E.1 Allocation Preservation by Domain

F FAR Feature Inventory

G Hand-Verified Leakage Audit

Table 14 reports the full hand-verified leakage audit on the conflict-containing subsets. Table 15 reports 95% bootstrap confidence intervals for all three metrics. For sparse metrics such as tuple leakage, intervals remain very wide even at $n = 100$, which is why we treat tuple zero-counts as "no tuple leak observed under this protocol" rather than evidence of zero risk.

Removed feature group	Oracle-score drop
Task dependence	-1.8 ± 0.2
Privacy exposure	-0.9 ± 0.1
Contextual linkage	-0.7 ± 0.1
Semantic subtype	-0.5 ± 0.1
Domain (control)	-0.3 ± 0.1

Table 11: Feature-group ablation on held-out conflict spans (5 random seeds).

Domain	Fixed gen.	FAR-Rule	FAR-Learned	FAR-Learned+SFT
Legal	76.3%	80.1%	86.4%	87.6%
Finance	78.9%	83.5%	89.2%	90.4%
Medical	83.6%	84.4%	90.7%	92.3%

Table 12: Reference-response preservation on the conflict-span subset, by domain. SFT gains are consistent across all three domains.

H Full-Scale Automated Leakage Scores

I Pairwise Utility Results

J Automated Utility Metrics

K Human Utility Audit

Two annotators independently reviewed 50 randomly sampled rewritten-prompt responses per domain (150 total) and judged whether each response preserves the core constraints and actionable recommendations of the corresponding raw-prompt response (binary judgment). The same 150 prompts were also scored by the Claude 3.5 Sonnet pairwise judge. Agreement between human and judge decisions, measured by Cohen's κ, is 0.79 for legal, 0.77 for finance, and 0.82 for medical. The primary source of disagreement is the judge treating high-level advice as preserved even when numerical thresholds or specific legal citations differ; human annotators are more sensitive to threshold-level changes. This suggests that the Claude 3.5 Sonnet judge slightly overestimates reference-response preservation relative to human expert assessment.

L Failure Case Taxonomy

FAR still fails in three recurring ways. First, detector errors propagate directly to rewriting. Missed implicit temporal references, generic organization names, and weak quasi-identifiers are never assigned operations, so they can remain in both the rewritten prompt and the target response. Second, the learned policy sometimes chooses utility-preserving operations for spans that remain linkable in context. This is most visible in medical prompts, where a generalized lab value combined with a disease and medication can still preserve enough tuple information for partial re-identification. Third, the local rewriter occasionally fails to execute the edit plan exactly (see Table 7 for rates).

We also observe utility-side failures. Fixed deletion and masking often remove the only constraint needed to answer the task, while FAR sometimes over-corrects by preserving coarse but still decision-critical quantities. For example, changing a precise financial amount to a broad range can preserve advice category but alter repayment calculations; changing a lab value to a severity bucket can preserve triage direction but remove threshold-sensitive recommendations. Qualitative examples of each failure type are provided in Appendix M.

M Qualitative Examples

All spans and personal details in the following examples have been fully de-identified. Examples are chosen to illustrate the four failure types most relevant to the hand-verified leakage audit (Appendix G).

Example 1: Fixed `delete` breaks task (legal). The user asks an employment-law question referencing a specific employer, *[EMPLOYER-X]*, as the sole organizational constraint. Fixed `delete` removes the span; GPT-4 responds with generic advice that fails to account for the employer's size and sector. FAR-Learned assigns `generalize` ("a private-sector employer with similar size and employment-law relevance"), preserving the legally relevant employer attributes while removing the direct identifier.

Example 2: FAR over-preserves in a medical tuple. The user's prompt includes a disease name, a specific lab value (eGFR), and a medication. FAR-Learned generalizes the lab value to a severity bucket ("moderately reduced") and pseudonymizes the medication. However, the combination of disease + generalized bucket + medication class preserves enough information for partial tuple recovery, which is counted as a tuple leak in Table 14. The SFT rewriter does not fix this because the failure originates in policy choice, not execution.

Example 3: Rewriter non-compliance (finance). The edit plan assigns `pseudonymize` to a bank name. The prompt-based rewriter replaces it with a plausible substitute in the first occurrence but

Group	Features	Dim.	Role
Privacy exposure	identifier type (direct identifier; quasi-identifier; event context; preference/profile); semantic subtype (person name; age; location; date/time; organization; occupation; financial amount; financial institution; medical condition; medication; lab/measurement); risk score	16	Whether the span directly or indirectly identifies the user.
Task dependence	`is_query_target`; `deletion_makes_underspecified`; `is_sole_constraint`; `granularity_tolerance`; `substitution_invariance`; `threshold_sensitivity`	6	How the downstream response structurally depends on the span.
Contextual linkage	number of sensitive spans; number of conflict spans; same-subtype count; tuple-membership flag; tuple-size bucket	5	Whether the span combines with others to form an identifying tuple.
Realization feasibility	span length bucket; numeric/unit flag; value-format rigidity; syntactic removability; coarse-category availability; sentence-position bucket	6	Whether a particular edit can be executed naturally without breaking text.
Domain (control)	legal; finance; medical	3	Domain context; weak control variable.
Total		**36**	

Table 13: Mechanism-oriented feature inventory for the FAR allocation policy. Features are organized by the structural determinant of operation choice they capture rather than by data source.

Domain	n	Method	Input direct ↓	Output direct ↓	Tuple ↓
Legal	100	Raw	1.000	0.880	0.050
Legal	100	RuPTA	0.880	0.670	0.050
Legal	100	InferDPT	0.820	0.510	0.040
Legal	100	FAR-Learned	0.439	0.429	0.000
Legal	100	FAR-Learned+SFT	0.418	0.408	0.000
Finance	100	Raw	1.000	0.980	0.090
Finance	100	RuPTA	0.890	0.690	0.090
Finance	100	InferDPT	0.690	0.540	0.040
Finance	100	FAR-Learned	0.404	0.404	0.010
Finance	100	FAR-Learned+SFT	0.384	0.384	0.010
Medical	120	Raw	1.000	0.925	0.050
Medical	120	RuPTA	0.933	0.792	0.067
Medical	120	InferDPT	0.867	0.758	0.008
Medical	120	FAR-Learned	0.576	0.559	0.008
Medical	120	FAR-Learned+SFT	0.551	0.534	0.008

Table 14: Human-verified leakage audit on conflict-containing subsets. Lower is better. Raw input leakage is 1.000 by definition.

Domain	Method	Input direct	Output direct	Tuple
Legal	Raw	1.000 [1.000, 1.000]	.880 [.835, .925]	.050 [.020, .080]
Legal	RuPTA	.880 [.835, .925]	.670 [.595, .745]	.050 [.020, .080]
Legal	InferDPT	.820 [.765, .875]	.510 [.430, .590]	.040 [.010, .070]
Legal	FAR-Learned	.439 [.365, .515]	.429 [.355, .505]	.000 [.000, .000]
Legal	FAR-Learned+SFT	.418 [.340, .495]	.408 [.330, .485]	.000 [.000, .000]
Finance	Raw	1.000 [1.000, 1.000]	.980 [.965, .995]	.090 [.050, .130]
Finance	RuPTA	.890 [.845, .935]	.690 [.620, .760]	.090 [.050, .130]
Finance	InferDPT	.690 [.620, .760]	.540 [.460, .620]	.040 [.010, .070]
Finance	FAR-Learned	.404 [.330, .480]	.404 [.330, .480]	.010 [.000, .025]
Finance	FAR-Learned+SFT	.384 [.310, .460]	.384 [.310, .460]	.010 [.000, .025]
Medical	Raw	1.000 [1.000, 1.000]	.925 [.890, .960]	.050 [.025, .080]
Medical	RuPTA	.933 [.900, .965]	.792 [.740, .845]	.067 [.035, .100]
Medical	InferDPT	.867 [.820, .915]	.758 [.700, .815]	**.008** [.000, .020]
Medical	FAR-Learned	.576 [.510, .645]	.559 [.490, .625]	.008 [.000, .020]
Medical	FAR-Learned+SFT	.551 [.485, .620]	.534 [.465, .600]	.008 [.000, .020]

Table 15: 95% bootstrap confidence intervals (10,000 resamples over prompts) for all three leakage metrics in the hand-verified audit. Intervals for tuple leakage are wide even at $n = 100$; zero entries should be interpreted as "no tuple leak observed" rather than evidence of zero risk.

Domain	Method	Input direct ↓	Output direct ↓	Tuple ↓
Legal	Raw	0.5800	0.5100	0.0280
Legal	RuPTA	0.5473	0.4860	0.0270
Legal	InferDPT	0.4889	0.4110	0.0200
Legal	FAR-Learned	0.2578	0.2270	0.0080
Legal	FAR-Learned+SFT	0.2451	0.2140	0.0070
Finance	Raw	0.5000	0.4810	0.0350
Finance	RuPTA	0.4750	0.4520	0.0320
Finance	InferDPT	0.3650	0.3170	0.0220
Finance	FAR-Learned	0.2100	0.1930	0.0070
Finance	FAR-Learned+SFT	0.1990	0.1830	0.0060
Medical	Raw	0.9100	0.8820	0.0420
Medical	RuPTA	0.8737	0.8470	0.0380
Medical	InferDPT	0.8067	0.7530	0.0180
Medical	FAR-Learned	0.5244	0.4950	0.0120
Medical	FAR-Learned+SFT	0.5011	0.4720	0.0110

Table 16: Approximate leakage on the full 3,600-prompt evaluation suite using automated matching (no human verification). Raw falls below 1.000 because the denominator includes all prompts. Method ordering is consistent with Table 14.

Domain	Method	n	Raw wins	Rewrite wins	Ties	Gap
Legal	RuPTA	1200	994	26	180	0.089 [0.083, 0.095]
Legal	InferDPT	1200	1170	10	20	0.253 [0.235, 0.270]
Legal	FAR-Learned	1196	1016	16	164	0.094 [0.088, 0.101]
Legal	FAR-Learned+SFT	1194	1007	19	168	0.090 [0.084, 0.097]
Finance	RuPTA	1198	926	12	260	0.084 [0.079, 0.089]
Finance	InferDPT	1198	1192	2	4	0.246 [0.231, 0.260]
Finance	FAR-Learned	1200	1046	10	144	0.119 [0.111, 0.128]
Finance	FAR-Learned+SFT	1198	1034	13	151	0.113 [0.105, 0.121]
Medical	RuPTA	1200	770	24	406	0.065 [0.059, 0.071]
Medical	InferDPT	1198	1142	2	54	0.245 [0.229, 0.262]
Medical	FAR-Learned	1192	904	22	266	0.118 [0.106, 0.130]
Medical	FAR-Learned+SFT	1190	894	27	269	0.112 [0.100, 0.124]

Table 17: Pairwise reference-response preservation. Gap = reference score minus rewrite score; lower is better. 95% CIs use 10,000 bootstrap resamples.

retains the original string in a later co-referential mention, causing a direct leak. The SFT rewriter handles both occurrences consistently, which accounts for part of the leakage reduction in Table 7.

Example 4: Missed implicit date (legal). A legal prompt contains the phrase "three years after the incident" without an explicit date. The detector does not trigger on this span because it contains no structured date pattern and the classifier assigns low confidence. The conflict span therefore receives no operation, and the implicit temporal constraint remains in the rewritten prompt and reappears in GPT-4's response.

N Deployment Considerations

FAR's entire pipeline (span detection, factor prediction, policy inference, local rewriting) runs locally: DeBERTa-v3-base (∼184M parameters) for span classification and factor prediction, XGBoost (∼10^3 tree nodes) for policy inference, and Qwen3-8B-Instruct (8B parameters) for rewriting on a single consumer GPU. Only the final sanitized prompt is sent to the target cloud LLM. In contrast, RuPTA performs iterative optimization per prompt, invoking a privacy evaluator, a utility evaluator, and an optimizer each round—requiring multiple LLM inference calls per rewrite. This architecture is expected to reduce external LLM calls relative to iterative rewriting methods such as RuPTA, but we do not claim a measured latency or cost advantage; direct benchmarking is left to future work.

O Annotation and Evaluation Protocols

O.1 Privacy Exposure Annotation

```
You are annotating privacy risk for a span in a
↪  user prompt.
Given the full prompt and the highlighted span,
↪  assign one label:
- LOW: generic or weakly identifying information.
- MEDIUM: quasi-identifying information or
↪  contextual detail that may
  contribute to re-identification.
```

Domain	Method	BERTScore ↑	ROUGE-L ↑	SBERT cos. ↑
Legal	RuPTA	0.891	0.412	0.873
Legal	InferDPT	0.743	0.201	0.681
Legal	FAR-Learned	0.874	0.388	0.858
Legal	FAR-Learned+SFT	0.882	0.401	0.866
Finance	RuPTA	0.903	0.438	0.889
Finance	InferDPT	0.731	0.183	0.664
Finance	FAR-Learned	0.861	0.371	0.841
Finance	FAR-Learned+SFT	0.869	0.384	0.849
Medical	RuPTA	0.921	0.511	0.908
Medical	InferDPT	0.749	0.219	0.702
Medical	FAR-Learned	0.879	0.432	0.864
Medical	FAR-Learned+SFT	0.886	0.447	0.871

Table 18: Automatic utility metrics relative to the raw-prompt reference response. BERTScore: F1 with `roberta-large`. ROUGE-L: token-level longest common subsequence. SBERT: cosine similarity with `all-mpnet-base-v2`.

Failure type	Typical cause	Effect on evaluation
Missed span	Detector misses implicit time, weak organization cue, or contextual quasi-identifier	Content remains unedited; causes input and output direct leakage.
Over-preservation	Policy selects `generalize` or `pseudonymize` for a span that remains linkable with surrounding context	Direct leakage may decrease while tuple or inferential leakage remains.
Over-sanitization	Policy chooses `mask` or `delete` for the only task-critical constraint	Leakage decreases but reference-response preservation drops.
Rewrite non-compliance	Local rewriter retains an original string, changes the wrong span, or produces an ungrammatical repair	Causes residual leakage. Quantified in Table 7.
Judge mismatch	Pairwise judge treats a high-level answer as preserved when threshold-sensitive details are lost	Utility may be overestimated; confirmed by human utility audit ($\kappa = 0.77$–0.82).
Metric blind spot	Exact-match misses paraphrases, aliases, or reconstructable facts	Leakage may be underestimated under stronger attackers.

Table 19: Failure taxonomy used to interpret manual inspection and limitations.

```
- HIGH: direct identifier, highly specific
↪  sensitive value, or detail
  that strongly identifies the user or a third
↪  party.
Return JSON with fields: label, score, rationale.
Scores: LOW=0.25, MEDIUM=0.55, HIGH=0.90.
```

O.2 Task Dependence Annotation

```
You are annotating task dependence for a span in
↪  a user prompt.
Decide whether removing or heavily masking the
↪  span would change the
answer that a domain-general LLM should give.
Labels: LOW = incidental background; MEDIUM =
↪  useful context;
HIGH = primary constraint or target of the user's
↪  question.
Return JSON with fields: label, score, rationale.
Scores: LOW=0.15, MEDIUM=0.55, HIGH=0.85.
```

O.3 Task-Dependence Feature Annotation

```
For the highlighted span, answer each question as
↪  true or false:
1. is_query_target: Is the span the primary
↪  subject of the user's question?
2. deletion_makes_underspecified: Would deleting
↪  the span make the prompt
   too vague to preserve the raw-prompt reference
↪  behavior?
3. is_sole_constraint: Is this the only span of
↪  its type constraining the
   response?
4. same_type_corefs: Is there another span of the
↪  same subtype in the
   prompt?
5. granularity_tolerance: Would a coarser
↪  version preserve the
   precision needed to reproduce the reference
↪  behavior?
6. substitution_invariance: Would a plausible
↪  same-format substitute
   preserve the downstream response behavior?
Return JSON with six boolean fields and a short
↪  rationale.
```

O.4 GPT-4o-mini Personal-Information Screening

```
You are screening a text passage to determine
↪  whether it contains personal,
private, or contextually identifying information.
Classify as YES if it contains any of the
↪  following:
- Personal facts about an individual (name, age,
↪  location, occupation,
```

```
  family situation)
- Private events involving a specific person
↪  (medical situation, legal
  dispute, financial difficulty, employment
↪  conflict)
- Contextually identifying details that, in
↪  combination, could identify
  an individual (e.g., city + employer + rare
↪  condition)
- Personal medical, legal, or financial facts
Classify as NO if the passage is a general
↪  question, impersonal information
request, or contains no personal context.
Return JSON with fields: label (YES/NO), category
(medical/legal/financial/personal/none),
↪  confidence (0-1), rationale.
```

A 400-prompt human audit gives precision 0.91 and recall 0.87. The primary false-positive type is general health-information queries; the primary false-negative type is implicit financial distress signals that omit explicit amounts or institutions.

O.5 Local Rewrite Execution Prompt

```
Rewrite the user prompt by applying the provided
↪  edit plan exactly.
For each span: mask uses a type placeholder;
↪  delete removes the span;
generalize uses a coarser task-preserving
↪  description; pseudonymize uses
a plausible non-real substitute.
Apply the operation to every occurrence of the
↪  span in the prompt.
Do not add new private details. Do not preserve
↪  original protected strings
unless the edit plan explicitly says to keep
↪  them.
Return only the rewritten prompt.
```

O.6 Pairwise Utility Judge

```
You are comparing two responses to the same user
↪  task.
One response is generated from the original
↪  prompt and the other from a
privacy-rewritten prompt.
Judge whether the rewritten-prompt response
↪  preserves the task-relevant
content, constraints, reasoning direction, and
↪  actionable conclusions of
the original-prompt response.
Return JSON with score_A, score_B, winner, and
↪  rationale.
```

O.7 Human Privacy Verification Guideline

```
Given the original conflict-span content,
↪  rewritten prompt, and target LLM
response:
1. Mark input_direct_leak = 1 if the rewritten
↪  prompt explicitly contains
   or trivially recovers protected conflict-span
↪  content.
2. Mark output_direct_leak = 1 if the target LLM
↪  response explicitly
   contains or trivially recovers protected
↪  conflict-span content.
3. Mark tuple_leak = 1 if at least two slot
↪  values from the same sensitive
   tuple remain jointly recoverable.
Normalization ignores case, punctuation, and
↪  trivial formatting differences.
Do not count broad category descriptions as
↪  direct leakage unless they
preserve the original specific value.
```