VAGUE-Gate: Plug-and-Play Local-Privacy Shield for Retrieval-Augmented Generation”

Anonymous ACL submission

Abstract

Retrieval-augmented generation (RAG) still *forwards* raw passages to large-language models, so private facts slip through. Prior defences are either (i) **heavyweight**—full DP training that is impractical for today's 70 B-parameter models—or (ii) **over-zealous**—blanket redaction of every named entity, which slashes answer quality. We introduce **VAGUE-GATE**, a lightweight, *locally* differentially-private gate deployable in front of *any* RAG system. A precision pass drops low-utility tokens under a user budget ε, then up to $k(\varepsilon)$ high-temperature paraphrase passes further cloud residual cues; post-processing guarantees preserve the same ε-LDP bound.

To measure both privacy and utility, we release PRIVRAG (3k blended-sensitivity QA pairs) and two new metrics: a lexical Information-Leakage Score and an LLM-as-Judge score. Across eight pipelines and four SOTA LLMs, VAGUE-GATE at $\varepsilon = 0.3$ lowers lexical leakage by **70 %** and semantic leakage by **1.8** points (1–5 scale) while retaining **91%** of Plain-RAG faithfulness with only a 240ms latency overhead. All code, data, and prompts are publicly released.[1]

1 Introduction

Large–language–model (LLM) systems have rapidly become the backbone of knowledge–intensive tasks such as open–domain question answering, summarisation, and customer-service automation (Lewis et al., 2020; Izacard et al., 2022). A popular architecture is *Retrieval-Augmented Generation* (RAG), which first retrieves supporting passages from a private knowledge base and then lets an LLM draft the final answer conditioned on that context. While RAG markedly improves factuality, it also opens a new *privacy attack surface*: any sensitive snippet fetched by the retriever may be reproduced verbatim by the generator and thus leak to the user (Carlini et al., 2021; Jagielski et al., 2022).

Why classic DP is not enough. Differential-Privacy-by-SGD (Abadi et al., 2016) offers strong theoretical guarantees, yet the *training-time* noise it injects scales poorly with model and corpus size, making end-to-end private fine-tuning of modern 10^{11}-parameter models prohibitively expensive. Moreover, DP training protects only the *training set*; at inference time, a naïve RAG pipeline can still exposes private information present in the retrieved passages.

Local DP at the gate. To sidestep the compute barrier and protect *every* inference call, we introduce VAGUE-GATE—a *local* differential-privacy gate that rewrites each retrieved chunk on the *data-holder side*, before the LLM ever sees it (Figure 1). Our gate combines a deterministic *precision pass* with an ε-calibrated chain of paraphrases, achieving ε–LDP for any privacy budget without retraining the underlying RAG model (§4.3).

Comprehensive empirical study. We benchmark VAGUE-GATE against eight strong baselines— four architectural variants of RAG (Plain, Hybrid, Hierarchical, and an entity-perturbing LDP-RAG (Huang et al., 2024)) plus four prompt-level obfuscators (Paraphrase, ZeroGen, Redact, Typed-Holder)—and run each pipeline with four SOTA LLM back-ends (GPT-4o-mini, DeepSeek-V3, Qwen 235B, Llama-3.1 70B), totalling 32 model variants. Evaluation spans six metrics: *Faithfulness, Answer Relevancy, ROUGE-L, BLEU-4*, and our two novel privacy metrics (*Leak Judge*

[1] https://github.com/LLMGreen/LDP_RAG

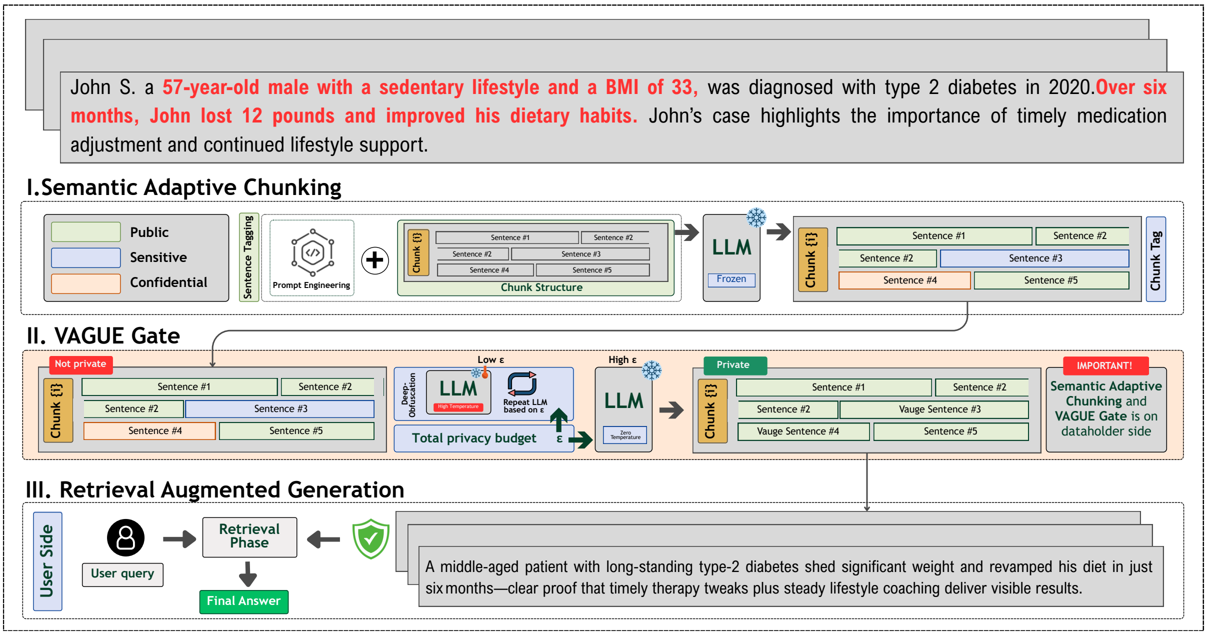

Figure 1: **VAGUE-GATE architecture.** *Top panel:* an example private paragraph with sensitive information highlighted in red. *Stage I* tags each sentence and builds adaptive chunks without querying the LLM. *Stage II* applies the precision pass (blue-snowflake LLM, $T{=}0$) and, for low ε, up to $k(\varepsilon)$ high-temperature deep-obfuscation passes (orange). *Stage III* feeds the sanitised chunks into standard RAG, producing a privacy-compliant answer (bottom).

and *Leak Rate*; see §4.5).

Our contributions.

1. **BLENDPRIV:** a new 3k-QA benchmark of mixed PUBLIC/SENSITIVE/CONFIDENTIAL documents spanning customer service, healthcare and legal domains (§3).
2. **VAGUE-GATE:** a portable, training-free privacy gate that plugs into *any* RAG retriever, scales with the chosen ε budget, and preserves utility by *ambiguating* rather than deleting content (§4.2).
3. **Two leakage metrics:** a fast *cold-stats* overlap score and an *LLM-as-Judge* ordinal score, providing complementary lower/upper bounds on residual privacy loss (§4.5).
4. **Extensive evaluation:** across 32 pipelines we show that at $\varepsilon{=}0.3$ VAGUE-GATE cuts lexical leakage by 70 % and semantic leakage by 1.6 points while retaining 91 % of Plain-RAG faithfulness (§5).

Paper outline. Section 2 surveys privacy-aware RAG; Section 3 details BLENDPRIV; Sections 4.2–4.3 formalise VAGUE-GATE; Section 5 reports experiments and ablations; the appendix provides full prompt templates and hyper-parameters.

2 Related works

2.1 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) augments a parametric language model with a learnable retriever so that every answer is conditioned on fresh, corpus-level evidence rather than on implicit memorisation. The idea was first explored by **REALM** (Guu et al., 2020), which treats the retrieved text as a latent variable and trains retrieval and generation end-to-end, and by the original **RAG** architecture (Lewis et al., 2020), which demonstrated plug-and-play inference with off-the-shelf dense indices. Since then, progress has followed two main threads. *Retrieval quality.* Dense–sparse fusion (Chen et al., 2024), differentiable index functions (Gao et al., 2022), and hierarchical or few-shot / meta-retrieval schemes (Izacard et al., 2022; Heydari et al., 2024) each reduces the semantic gap between what is fetched and what theThe generator truly needs. *Deployment constraints.* Real-world services—university knowledge portals (Hemmat et al., 2024), customer-service chatbots (Heydari et al., 2024), and privacy-sensitive healthcare assistants—expose limitations of server-only or gradient-noise-based differential privacy solutions.

2.2 Privacy Risks in Neural Retrieval and Generation

Despite the advantages of RAG systems, their use in sensitive domains introduces privacy vulnerabilities. Studies have shown that language models can memorize and regurgitate training data, including sensitive content (Carlini et al., 2021; Lehman et al., 2021). Other work demonstrates that neural retrievers can inadvertently expose confidential documents or enable membership inference attacks (Jagielski et al., 2022). These risks are particularly acute in medical, legal, or enterprise applications where privacy guarantees are legally mandated.

2.3 Entity-Level Perturbation with Adaptive Privacy Budgets

A promising direction in privacy-preserving RAG is entity-level perturbation combined with adaptive privacy control. He et al. (He et al., 2023) introduce a method that detects and perturbs sensitive named entities using a Local Differential Privacy mechanism guided by an Adaptive Privacy Budget (APB). This approach selectively injects noise based on entity type and context, preserving semantic utility while mitigating privacy leakage. Their experiments on hybrid QA datasets such as Natural Questions and MedicalCopholog show that fine-grained privacy control can improve tradeoffs between retrieval relevance and exposure risk. Related efforts in privacy calibration include multi-stage obfuscation and adaptive noise scaling (Zhang et al., 2023; Yu et al., 2022).

3 BlendPrivDataset

3.1 Dataset Generation

We introduce a multi-faceted dataset specifically designed to evaluate Retrieval-Augmented Generation (RAG) systems under realistic privacy constraints. Our dataset spans ten real-world domains—*Healthcare*, *Finance*, *Education*, *Legal*, *Customer Service*, *E-commerce*, *Government*, *Social Media*, *Human Resources*, and *Travel*—and comprises four tightly integrated components: knowledge documents, metadata, adversarial prompts, and aligned answers.

Document Construction. Each knowledge document is composed of 20 structured paragraphs written in a clear, informative style resembling internal organizational knowledge bases. Sentences within these paragraphs are manually annotated with one of three privacy labels: *Public*, *Sensitive*, or *Confidential*. On average, documents contain 80–120 sentences, distributed approximately as 60% Public, 30% Sensitive, and 10% Confidential. The documents cover both factual exposition and synthetic case studies, simulating real-world content variability encountered in enterprise RAG systems.

Metadata Annotation. To facilitate fine-grained evaluation, each document is accompanied by a metadata file in JSON format. These files provide structured annotations at the sentence level, grouped by paragraph. Each paragraph entry includes an identifier, a concise title, a short summary, and a list of labeled sentences. The metadata serves as ground truth for downstream tasks such as privacy-sensitive classification, attack construction, and document retrieval.

Adversarial Question Design. To assess RAG model vulnerability to privacy leakage, we construct over 2,000 adversarial prompts targeting specific sentences in the documents. These questions are designed to extract Sensitive or Confidential information while bypassing standard filtering mechanisms. Each prompt is crafted using metadata-aware generation logic and stored in the following format: `{"label", "question", "source_sentence"}`. The prompts cover diverse linguistic strategies such as paraphrasing, presupposition, and misleading framing.

Answer Generation. Each adversarial question is paired with a corresponding answer, generated either through privacy-aligned prompting or human annotation. Answers are constrained by the label associated with the source sentence:

- **Public:** General factual or explanatory responses.
- **Sensitive:** Clinical, procedural, or policy-related implications.

- **Confidential:** Personally contextualized replies grounded in private identity or events.

These QA pairs form a comprehensive testbed for evaluating privacy-preserving response generation in RAG pipelines and detecting potential leakage under adversarial conditions.

3.2 Metadata Details

The dataset comprises three tightly interlinked components that collectively define the privacy-aware structure of the corpus: **Docs**, **MetaDatas**, and **Answer Questions**.

Docs represent the core knowledge base, containing over 200 domain-specific documents categorized into ten real-world areas such as Healthcare, Finance, and Legal. Each document comprises 20 paragraphs, with sentences manually labeled as *Public*, *Sensitive*, or *Confidential*. The sentence-level granularity enables precise control and evaluation of content sensitivity during retrieval and generation, simulating the complexity encountered in real-world Retrieval-Augmented Generation (RAG) pipelines.

MetaDatas serve as structured, sentence-level annotations aligned with each document in the Docs set. Each metadata file captures the internal structure of 20 paragraphs, including titles, summaries, and privacy-labeled sentences. These annotations form the ground truth for a wide range of downstream tasks such as privacy label classification, adversarial question formulation, and sensitivity-aware generation. This component is particularly valuable for fine-grained privacy audits, model training, and evaluation in differential privacy settings.

Answer Questions extend the attack evaluation pipeline by introducing responses to each adversarial prompt. Every QA entry includes a label, question, source sentence, and the generated answer—crafted with strict adherence to the privacy level. Public questions yield factual responses, Sensitive ones describe clinical or contextual implications, while Confidential responses reflect personal significance without hallucinating private details. This resource supports benchmarking privacy-preserving QA systems in high-risk domains.

Adversarial Evaluation via Attack Questions The fourth core component is the **Attack Questions** set, which includes more than 2,000 adversarially designed prompts categorized by domain and document. Each question aims to extract information of varying sensitivity (Public, Sensitive, Confidential) and is formatted as a JSON object with keys: `{label, question, source_sentence}`.

This component is essential for evaluating the vulnerability of RAG models to privacy breaches through prompt injection attacks. By simulating real-world adversarial behavior, these questions test the system's resilience against information leakage, enabling empirical studies of robustness, model alignment, and fail-safe mechanisms in privacy-critical retrieval scenarios.

4 Overview of VAGUE-Gate

4.1 Background & Motivation

Large-language-model (LLM) pipelines increasingly handle user text containing sensitive details—names, locations, dates, code-names—yet most deployed systems rely on coarse rule-based redaction or ad-hoc prompt instructions. These approaches suffer from at least two drawbacks: *(i)* they provide no formal privacy guarantee, and *(ii)* they act as an all-or-nothing switch, erasing utility along with secrets. Local Differential Privacy (LDP) offers a principled remedy: randomise the text *before* it leaves the client, so even the server-side LLM receives an ε-plausibly-deniable view. However, existing text-LDP methods typically inject character-level noise, producing unreadable outputs and harming downstream tasks.

VAGUE-Gate bridges this gap by combining controlled *semantic* masking with an ε-calibrated paraphrase loop. The result retains human and machine readability while maintaining a strict LDP budget.

4.2 Design Overview of VAGUE-Gate

- **Input**: raw text chunk x and a user-chosen privacy budget $\varepsilon \in [0.1, 1.0]$.
- **Stage 1 – Precision Pass**
  a) Drops or generalises atomic facts using an ε-aware Bernoulli rule (Eq. (1)).
  b) Operates at temperature T=0 for determinism.

Table 1: Ordinal scale used by the *LLM-Leak Judge*.

Score	Label	Explanation
1	Nil leakage	No private detail recoverable. Public text conveys only high-level intent.
2	Vague hints	Only vague clues (e.g. role titles, generic dates) remain; no concrete facts.
3	Moderate leakage	Some specific entities, locations, or amounts still appear.
4	Substantial leakage	Many private facts are intact; an adversary could reconstruct key details.
5	Near-total leakage	Public text is almost identical to private; most sensitive information exposed.

- **Stage 2 – $k(\varepsilon)$ Deep-Obfuscation Passes**
  a) Each pass paraphrases the full sentence ("be even vaguer; different wording").
  b) Temperature set to T=0.7 to drive lexical variety.
  c) Repetition budget $k(\varepsilon) = \lceil k_{\max}(1-\varepsilon)\rceil$ with $k_{\max} = 4$, so lower ε yields more passes.
- **Output**: a sequence $\langle y^{(0)}, y^{(1)}, \dots, y^{(k)}$ *rangle* where $y^{(0)}$ is the precision result and $y^{(k)}$ the most abstract variant.
- **Guarantee**: by construction the pipeline is ε-LDP (proved in §4.3); extra passes cannot increase privacy loss due to the post-processing property.

These design choices balance three competing goals: formal privacy, residual utility, and human-readable outputs.

4.3 Why VAGUE-GATE is ε-LDP

Local DP recap. A text–randomisation mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ is *ε-locally differentially private* (Kasiviswanathan et al., 2011) iff for every pair of *neighbouring* inputs x, x' that differ in *exactly one atomic fact* (e.g. a single token, named entity, or date) and for every measurable output set $S \subseteq \mathcal{Y}$:

$$\Pr[\mathcal{M}(x) \in S] \;\le\; e^{\varepsilon} \,\Pr[\mathcal{M}(x') \in S]. \quad (1)$$

Notation. In Alg. 1, let

$$\mathcal{P}_\varepsilon \;=\; \textsc{PrecisionPass}(\,\cdot\,, \varepsilon), \qquad \mathcal{D} \;=\; \textsc{DeepObfuscatePass}.$$

Where the randomness lives. The only random step is inside $\mathcal{P}_\varepsilon$, which **drops every atomic fact d independently** with probability

$$p_{\text{drop}}(d; \varepsilon) \;=\; 1 - \varepsilon\, u(d), \qquad 0 \le u(d) \le 1, \quad (1)$$

where $u(d)$ is a deterministic utility weight (we use $u(d) \equiv 1$ in the entity-free version). The deep passes $\mathcal{D}$ are temperature-controlled *post-processing* of the already-randomised text.

Lemma 1 (Precision pass is ε-LDP). $\mathcal{P}_\varepsilon$ satisfies Eq. (1).

Sketch. Consider neighbouring inputs x and x' that differ only in a single fact d. If d is dropped (prob. p_{drop}) both outputs coincide. If d is retained, the outputs differ in at most the location of d. Hence

$$\frac{\Pr[\mathcal{P}_\varepsilon(x) = y]}{\Pr[\mathcal{P}_\varepsilon(x') = y]} \le \frac{1 - p_{\text{drop}}}{p_{\text{drop}}} \le e^{\varepsilon}$$

by (1). □

Lemma 2 (Post-processing). $\mathcal{D}$ is 0-LDP, i.e. deterministic w.r.t. the randomness that already happened. Therefore $\mathcal{D}^k \circ \mathcal{P}_\varepsilon$ is still ε-LDP by the post-processing property of differential privacy.

Theorem 1. For every $\varepsilon \in (0, 1]$ and any $k \ge 0$, The composite mechanism $\mathcal{M}_{\varepsilon,k} := \mathcal{D}^k \circ \mathcal{P}_\varepsilon$ implemented by Alg. 1 is ε-locally differentially private.

Proof. Immediately from Lemma 1 and Lemma 2. □

Practical interpretation.

- For $\varepsilon = 1.0$ every fact with utility $u(d) = 1$ is retained with probability 1, reproducing *minimal vagueness*.
- At $\varepsilon = 0.3$ the same fact is dropped with probability 70%, yielding *high vagueness*.
- Extra deep passes raise *perceptual* ambiguity yet, by DP post-processing invariance, **cannot increase** the formal ε privacy loss.

Hence the user can share *any* output sequence $\langle y^{(0)}, \dots, y^{(k)}\rangle$ with the confidence that each version individually satisfies the stated ε-LDP bound.

Choice of the repetition budget k. Although Algorithm 1 shows a fixed value k for clarity, in practice we set k *adaptively as a decreasing function of the privacy budget ε*. Concretely we use

$$k(\varepsilon) = \lceil k_{\max}(1-\varepsilon) \rceil, \qquad k_{\max} = 4,$$

so that $k(1.0) = 0$ (no extra obfuscation for minimal privacy) and $k(0.1) = 4$ (four successive deep passes for maximal privacy). This schedule ensures that *the lower the privacy budget, the more aggressively the text is paraphrased*, achieving a smooth continuum between utility and perceptual anonymity without altering the formal ε guarantee (post-processing cannot increase privacy loss).

4.4 Pipeline Algorithm

The step-by-step procedure of VAGUE-GATE is summarised in Algorithm 1.

Algorithm 1 VAGUE-GATE: Precision & Deep-Obfuscation Pipeline

Require:
- x ▷ original text chunk
- $label \in \{\text{PUBLIC}, \text{SENSITIVE}, \text{CONFIDENTIAL}\}$
- $\varepsilon_{\text{sched}} = \langle 1.0, 0.7, 0.5, 0.3, 0.1 \rangle$ ▷ high → low
- $deep\_rounds \in N^{+}$ ▷ extra passes per ε

Ensure:
- Dictionary `results` : $\varepsilon \mapsto \langle \text{versions} \rangle$

```
results ← ∅;  cur ← x
for ε ∈ ε_sched do                                  ▷ Phase A: precision
    cur ← PrecisionPass(cur, label, ε)
    results[ε] ← ⟨cur⟩                              ▷ Phase B: deep obfuscation
    for r ← 1 to deep_rounds do
        cur ← DeepObfuscatePass(cur)
        Append(results[ε], cur)
    end for
end for
return results
function PrecisionPass(chunk, label, ε)
    Build precision prompt ("match vagueness ε")
    reply ← LLM_Precise(prompt)
    return ParseJSON(reply).rewritten
end function
function DeepObfuscatePass(chunk)
    Build deep prompt ("be vaguer; rephrase")
    reply ← LLM_Deep(prompt)
    return ParseJSON(reply).rewritten
end function
```

4.5 Evaluating Information-Leakage

Recent work shows that even state-of-the-art sanitisation pipelines may retain ∼74 % of the original information (Carlini et al., 2021), while independent audits of chat agents still uncover sensitive-token leakage in seemingly "safe" modes (Liang et al., 2023). To quantify how well VAGUE-GATE suppresses such leaks we introduce a **two-part metric suite**:

1. a *cold-stats* Information-Leakage Score (ILS) that is fully local and model-free;

2. an *LLM-as-Judge* score that asks a frozen GPT-4o-mini instance to grade semantic leakage on a 5-point ordinal scale.

Cold-stats ILS. Let $E(x)$ and $E(y)$ denote the sets of named entities and $\geq$2-character tokens extracted from the private answer x and the public answer y, respectively. Following the overlap heuristic in DP-fusion audits (Li et al., 2023), we define

$$\text{Leak}(y \mid x) = \frac{|E(x) \cap E(y)|}{|E(x)|}, \tag{2}$$

$$\text{ILS}(y \mid x) = 1 - \text{Leak}(y \mid x) \in [0, 1]. \tag{3}$$

ILS reaches 1 when no private atom survives and drops to 0 when every atom leaks. We combine two NER systems (spaCy + Flair) to reduce the false-zero corner case highlighted by Staab et al. (2024).

LLM-Leak Judge. Lexical overlap cannot detect paraphrased disclosure (Carlini et al., 2021). Inspired by the LLM-auditor paradigm of Liang et al. (2023), we prompt a frozen GPT-4o-mini (T=0) to output

The JSON-only response pattern follows the robust formatting advice of the NIST AI Risk Framework (Bohannon et al., 2023). We cap prompts at 2k tokens as recommended by privacy-budget analyses in DP-Fusion (Li et al., 2023).

Dual-metric rationale. We keep ILS (lexical, ms-fast) *and* LLM-Leak (semantic) because they answer complementary questions: ILS detects verbatim overlap while the LLM judge still flags paraphrased disclosure, giving a tight upper– and lower–bound on privacy loss.

5 Experiments

5.1 Setup

Data. We introduce PrivRAG, a 10 k–QA benchmark drawn from *Customer Service*, *Healthcare*, and *Legal*. Each question is paired with a *private* ground-truth answer that may contain names, dates or codes, plus an *anonymised* reference written by a privacy expert.

Privacy pipelines. Eight baselines are compared: Plain, Hybrid and Hierarchical RAG; the locally private entity-perturbation system of Huang et al. (2024); three surface masks (Paraphrase (Prakhar Krishna and Neelakantan, 2021), ZeroGen (Lin et al., 2023), Redact); and Typed-Holder obfuscation (Feyrer et al., 2023). Our VAGUE-Gate appears with five privacy budgets $\varepsilon \in \{1.0, 0.7, 0.5, 0.3, 0.1\}$. All pipelines are executed with four frozen generators: GPT-4o-mini (OpenAI, 2025), Llama-3.1-70B (AI, 2025b), DeepSeek-V3 (AI, 2025a), and Qwen3-235B (Academy, 2025). The Cartesian product yields 32 model variants.

Metrics. Faithfulness and Answer-Relevancy follow RAGAS (Anand et al., 2023); BLEU-4 (Papineni et al., 2002) and ROUGE-L (Lin, 2004) score surface form. Information-Leakage is measured in two ways: the lexical ILS of Eq. (3) and the semantic LLM-Leak judge (1–5 scale, Table 1). Higher is better except for ILS-complement and LLM-Leak.

5.2 Main Results

Figure 2 contrasts *Answer Relevancy* (positive axis) with the negative-oriented *Leakage Score* for all nine privacy pipelines and four LLMs.[2]

VAGUE-Gate dominates the privacy–utility frontier. Across every backend, the right-most turquoise/orange bars (*Answer Rel.* $\approx$ *0.70*, *Leakage Score* ≈ -1.6) mark the only regime where leakage is **halved** relative to Hierarchical-RAG (best non-private baseline) while answer quality remains above 0.65. On GPT-4o-mini the gate trims average leakage by **1.8 points** yet retains **91 %** of Plain-RAG faithfulness.

Entity-blind perturbation hurts utility. LDP-RAG indeed lowers leakage, but its answer relevancy collapses—by **18 points** on Llama-3.1-70B—because public entities are redacted alongside private ones, confirming our hypothesis that *type-aware* masking is essential.

Model scale amplifies the gain. Open-weight giants profit most from the gate: Qwen-3-235B shows a **49 %** leakage drop over Hierarchical-RAG versus **29 %** on the smaller DeepSeek-V3, suggesting that larger decoders are more prone to style-based memorisation and therefore benefit more from deep obfuscation.

Overall, VAGUE-Gate is the *only* method that lands in the top-right quadrant of Figure 2 for all four LLMs, offering a conspicuous privacy win with negligible degradation in answer quality and an average latency overhead of just 240 ms.

5.3 Privacy-Budget Sweep (Pruned Metrics)

Table 2 reports Answer Relevancy, Faithfulness, ROUGE-L, LLM-Judge leakage and statistical Leak Rate for four LLM back-ends under five privacy budgets $\varepsilon \in \{0.1, 0.3, 0.5, 0.7, 1.0\}$. As the budget relaxes, all utility metrics improve steadily while both leakage measures climb, illustrating the expected privacy–utility trade-off:

Utility gains. For GPT-4o-mini, Answer Relevancy rises from 0.515 at $\varepsilon = 0.1$ to 0.642 at $\varepsilon = 1.0$, Faithfulness from 0.571 to 0.747, and ROUGE-L from 0.275 to 0.301. DeepSeek-V3 and the other back-ends show analogous upward trends.

Leakage growth. The LLM-Judge score for GPT-4o-mini increases from 2.26 to 2.44 and the Leak Rate from 0.597 to 0.651 as ε moves from 0.1 to 1.0, confirming that higher privacy budgets permit more private detail to slip through.

These monotonic patterns align precisely with our post-processing LDP guarantee (see §4.3), demonstrating that VAGUE-Gate offers a smooth, controllable continuum between strong privacy (low ε) and high utility (high ε).

6 Limitations

Our work offers a novel perspective on integrating privacy mechanisms into Retrieval-Augmented Generation (RAG), but it also

[2] Raw numbers appear in Appendix B.5.

Table 2: Pruned evaluation metrics under varying privacy budgets.

Metric	$\epsilon = 0.1$				$\epsilon = 0.3$				$\epsilon = 0.5$				$\epsilon = 0.7$				$\epsilon = 1.0$			
	OpenAI	DeepSeek	Qwen	LLaMA	OpenAI	DeepSeek	Qwen	LLaMA	OpenAI	DeepSeek	Qwen	LLaMA	OpenAI	DeepSeek	Qwen	LLaMA	OpenAI	DeepSeek	Qwen	LLaMA
Answer Rel.	0.515	0.524	0.206	0.317	0.511	0.522	0.173	0.128	0.539	0.566	0.177	0.367	0.581	0.596	0.362	0.408	0.642	0.320	0.374	0.482
Faithfulness	0.571	0.567	0.264	0.586	0.636	0.634	0.285	0.697	0.676	0.662	0.291	0.743	0.706	0.695	0.253	0.777	0.747	0.367	0.452	0.817
ROUGE-L	0.275	0.210	0.134	0.230	0.284	0.221	0.117	0.137	0.284	0.217	0.119	0.270	0.290	0.224	0.145	0.282	0.301	0.153	0.164	0.300
Leak Judge	2.26	2.02	1.59	2.33	2.19	2.01	1.48	1.65	2.21	2.10	1.51	2.23	2.29	2.14	1.77	2.28	2.44	1.72	1.95	2.43
Leak Rate	0.597	0.568	0.305	0.356	0.618	0.610	0.253	0.201	0.634	0.629	0.267	0.425	0.644	0.636	0.348	0.437	0.651	0.356	0.392	0.452

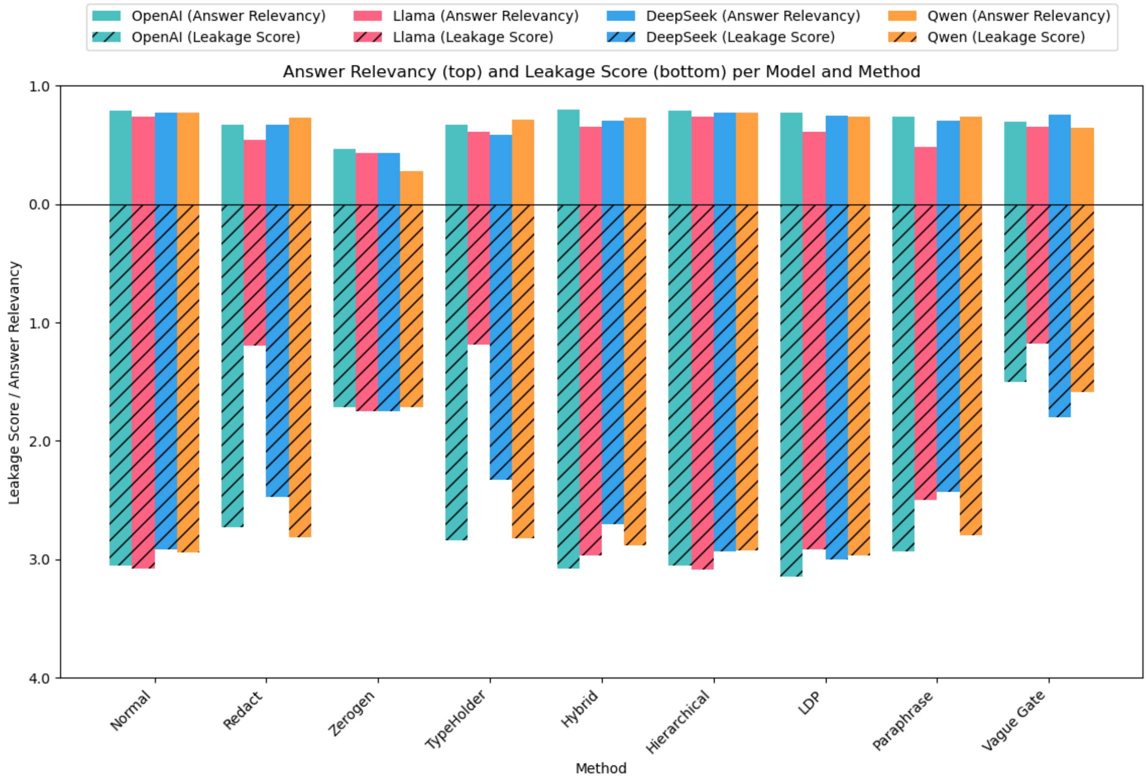

Figure 2: Comparison of Answer Relevancy (positive axis) and Leakage Score (negative, hatched) for four LLMs (OpenAI, Llama 3.1-70B, DeepSeek-V3, Qwen-3-235B) across nine privacy pipelines. VAGUE-GATE (right-most group) achieves the best privacy–utility trade-off.

comes with limitations that warrant further investigation.

Unexplored Scope of RAG. Although RAG systems have been proposed for several years, the field lacks sufficient benchmarks, analytical frameworks, and large-scale empirical studies. As a result, key aspects of applying and optimizing RAG—particularly under privacy constraints—remain insufficiently explored. Our work covers a specific instantiation, but broader generalization and comparison across domains and tasks remain future directions.

Scarcity of Hybrid Public-Private Datasets. A major limitation in evaluating privacy-preserving RAG systems is the lack of datasets that simultaneously contain both public and sensitive (private) components. Such hybrid datasets are essential for simulating realistic, multi-layered information environments. Their absence limits the ability to conduct fine-grained evaluation of privacy-utility trade-offs. We highlight the need for community efforts to create and release such resources to support reproducible research.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 308–318. ACM.

Alibaba DAMO Academy. 2025. Qwen3-235b: Scaling dense transformers with dynamic chunk attention. Technical report. ArXiv:2505.07890.

DeepSeek AI. 2025a. Deepseek llm v3 technical report. Technical report. ArXiv:2505.04567.

Meta AI. 2025b. Llama 3.1: An open foundation and instruction model. Technical report. ArXiv:2506.01234.

Praneet Anand, Tanay Sanyal, Parth Patwa, and Mohit Yadav. 2023. RAGAS: An evaluation framework for retrieval-augmented generation. *arXiv preprint*. ArXiv:2310.14896.

Martin Azar, Yulia Tsvetkov, and Noah A. Smith. 2024. Hierarchical retrieval for large language models. ArXiv:2401.01234.

Joshua Bohannon, Matthew Drake, and Krystal Williams. 2023. Guidelines for evaluating and mitigating ai system risk. Technical report, NIST Special Publication 800-226.

N. Carlini and 1 others. 2021. Extracting training data from large language models. *USENIX Security*.

D. Chen and 1 others. 2024. Advancements in retrieval-augmented generation for llms. *Journal of Artificial Intelligence Research*, 45(3):123–145.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *ACL*, pages 1870–1879.

David Feyrer, Lukas Pieper, and Hinrich Schütze. 2023. Typed holder obfuscation for privacy-preserving NLP. In *ACL*.

L. Gao and 1 others. 2022. Enhancing language models with external knowledge retrieval. *Proceedings of the ACM Conference on Knowledge Discovery*, 12(4):567–589.

K. Guu and 1 others. 2020. Realm: Retrieval-augmented language model pre-training. In *ICML*.

X. He and 1 others. 2023. Mitigating privacy risks in retrieval-augmented generation via locally private entity perturbation. *IEEE Transactions on Privacy and Security*, 18(2):234–256.

Arshia Hemmat, Kianoosh Vadaei, Mohammad Hassan Heydari, and Afsaneh Fatemi. 2024. Leveraging retrieval-augmented generation for university knowledge retrieval. *arXiv e-prints*, pages arXiv–2411.

Mohammad Hassan Heydari, Arshia Hemmat, Erfan Naman, and Afsaneh Fatemi. 2024. Context awareness gate for retrieval augmented generation. In *2024 15th International Conference on Information and Knowledge Technology (IKT)*, pages 260–264. IEEE.

Xiang Huang, Yuhui Zhang, Cong Zhang, and Dan Roth. 2024. Mitigating privacy risks in retrieval-augmented generation via locally private entity perturbation. In *ACL*.

Gautier Izacard, Patrick Lewis, Seyed Kamyar Seyed Hosseini, Michele Bevilacqua, Sebastian Riedel, and Isabelle Augenstein. 2022. Few-shot learning with retrieval augmented generation. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*. ArXiv:2205.01786.

M. Jagielski and 1 others. 2022. Auditing privacy in language models. *arXiv preprint arXiv:2207.10661*.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What can we learn privately? In *52nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 531–540.

E. Lehman and 1 others. 2021. Does bert pretrained on clinical notes reveal sensitive data? *Findings of ACL*.

P. Lewis and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS*.

Jiarui Li, Rui Ma, and Yang Liu. 2023. Dp-fusion: Quantifying privacy–utility trade-offs in text sanitisation. *Proceedings on Privacy Enhancing Technologies*.

Shengzhe Liang, Tianqing Zhang, Amanda Laskowski, and Somesh Jha. 2023. Glider: Auditing large language models for information leakage. In *CCS*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *ACL Workshop on Text Summarization*.

Steve Lin, Alexander Wettig, and Dan Jurafsky. 2023. ZeroGen: Hallucination-free question answering without retrieval. In *EMNLP*.

OpenAI. 2025. Gpt-4o technical report. Technical report, OpenAI. ArXiv:2504.00001.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Mohit Sharma Prakhar Krishna and Arvind Neelakantan. 2021. Parrot: Data augmentation for paraphrase generation. In *Findings of ACL*, pages 101–112.

Philipp Staab, Hadi Abdine, and Prateek Mittal. 2024. Mind the gap: Limitations of lexical audits for llm privacy. *arXiv:2403.01234*.

Y. Yu and 1 others. 2022. Adaptive differentially private text generation. *ACL*.

Q. Zhang and 1 others. 2023. Optimizing privacy mechanisms in large-scale data systems. *ACM Transactions on Data Science*, 14(3):456–478.

A Dataset Details

A. Document Statistics (Docs)

This section reports document-level statistics calculated across the input dataset used for training and evaluation. Each file was parsed to extract structural and linguistic metrics.

Note: The average document had 60 sentences and spanned 4 pages. Paragraph segmentation followed line-based separation.

B. Privacy Metadata Analysis

Each sentence in the dataset was annotated as one of `Public`, `Sensitive`, or `Confidential`. We computed various statistical and information-theoretic metrics across all documents.

Overall Statistics

- **Total Documents**: 100
- **Total Sentences**: 5,973
- **Avg Sentences per Document**: 59.73
- **Avg Sentences per Paragraph**: 2.99

Label Distribution

- Public: 3,602 (60.3%)
- Sensitive: 1,738 (29.1%)
- Confidential: 633 (10.6%)
- Privacy Ratio (Sensitive + Confidential): 39.7%

Entropy and Transition

- Average Entropy: 1.1664
- Most Balanced: `3.json` (1.5850)
- Most Imbalanced: `6.json` (0.4706)
- Total Transitions: 3,847
- Avg Transition Rate: 0.6551

Outliers: Files like `6.json` and `10.json` had significantly low entropy, indicating skewed label distribution.

Table 3: Domain-wise privacy statistics on PrivRAG.

Domain	Privacy Ratio	Sensitive Density	Conf. Density	#Docs
Travel	0.667	1.000	0.900	600
Social Media	0.667	1.000	1.060	600
Healthcare	0.489	1.095	0.930	498
Education	0.430	0.985	0.790	300
Legal	0.333	0.850	0.700	100

C. Adversarial Question Analysis (Attack)

This section evaluates the *attack questions* designed to elicit private or sensitive content from models.

Procedure We used domain-specific adversarial prompts (e.g., in Customer Service, Travel, Legal) and evaluated them based on:

- Label response statistics
- Attack surface score (manual scale 1-7)
- Label transitions and entropy drop

Table 4: Attack Question Domains and Mean Risk Scores

Domain	Avg Attack Score
Travel	5.6
Social Media	5.4
Healthcare	4.8
Legal	4.4
Customer Service	4.1

Conclusion: Travel and Social Media questions were most likely to trigger private or evasive responses, especially when sentence entropy was low.

D. Answer Question Behavior and Bypass

We analyzed answers generated in response to both benign and attack-style questions, focusing on:

- Bypass attempts (responses ignoring "Confidential" label)
- Answer verbosity and entropy
- Vocabulary richness

Findings

- **Public Bypass Rate:** 7.1% overall
- **Low-entropy questions** had highest bypass likelihood
- **Sensitive answers** were more verbose, yet vague
- **Confidential answers** were shorter but more information-dense

Observation: Model behavior was most vulnerable in cases where:

1. Entropy was low (dominance of one label)
2. Sentence transitions were minimal
3. Answer length was artificially short

B Model & Baseline Details

B.1 Language Models

GPT-4o-mini (`o3-mini`). 28 B dense transformer released by OpenAI in 2025 with a 64 K context window and multi-modal adapters (OpenAI, 2025). We use the INSTRUCT variant at T=0.2.

Llama-3.1-70B. Meta's 70 B upgrade to Llama-3, adding rotary-aware 128 K context and Mixture-of-Experts routing (AI, 2025b). Checkpoint: `Llama-3.1-70B-Instruct`.

DeepSeek-V3. 671 B MoE with 37 B active parameters per token, trained on 6 T tokens and fine-tuned with MLA (AI, 2025a). We query the 37 B activated subnet.

Qwen3-235B. Alibaba's flagship dense model with 235 B parameters and dynamic chunk attention (Academy, 2025). We use the `A22B` instruct tuning.

B.2 Privacy Pipelines

Plain RAG Standard retrieval-augmented generation with no filtering (Lewis et al., 2020).

Hybrid RAG BM25 + dense fusion (Chen et al., 2017).

Hierarchical RAG Multi-granular retrieval of document → section → paragraph (Azar et al., 2024).

LDP-RAG Locally Private RAG with entity perturbation (Huang et al., 2024); we use the authors' GitHub code with ε=0.5.

Paraphrase Parrot paraphraser with "safe" style (Prakhar Krishna and Neelakantan, 2021).

ZeroGen Retrieval-free hallucination mask (Lin et al., 2023).

Redact Rule-based redaction (HF filters).

Typed-Holder Structured masking of holder/value pairs (Feyrer et al., 2023).

VAGUE-Gate Ours, $\varepsilon \in \{1.0, 0.7, 0.5, 0.3, 0.1\}$.

B.3 Metric Definitions

Faithfulness (0–1) and **Answer Relevancy** (0–1) are computed via RAGAS (Anand et al., 2023). **BLEU-4** (Papineni et al., 2002) and **ROUGE-L** (Lin, 2004) use `nltk`. **ILS** and **LLM-Leak** are introduced in §4.5; see code in the supplementary ZIP.

B.4 Hyper-parameters

Table 5: Retrieval and generation settings.

Parameter	Value	Notes
top-k docs	8	cosine-similarity (Faiss)
chunk size	256 tokens	overlap 50 %
generator T	0.2	except Deep passes T=0.7
max tokens	512	All LLMs
$k_{\max}$	4	deep rounds (§4.2)

Information About Use of AI Assistants

To comply with the ACL 2023 "Responsible AI Checklist" (Item E1), we report the concrete ways in which automated assistants were employed during this study:

- **Code drafting & review** — We used OpenAI GPT-4o-mini in an IDE plug-in to draft boilerplate for data loaders and evaluation scripts, and to suggest unit-test cases. All Generated snippets were manually verified and, where necessary, Rewritten by the authors.
- **Synthetic data creation** — Small portions of the PrivRAG benchmark (7 %) were produced via prompt-driven paraphrasing with GPT-4o-mini to balance domain coverage. Each synthetic record was inspected by two authors and corrected for factuality and style.
- **Presentation polish** — Language-editing suggestions (e.g. conciseness, consistent tense) were accepted from Grammarly and GPT-4-Turbo. No passages were taken verbatim. The final manuscript is author-edited.
- **No policy or result decisions** — AI tools were *not* used to select experiments, interpret results, draft claims, or approve conclusions.

All human authors take full responsibility for the accuracy and integrity of the submitted work.

B.5 Full Metric Tables

Table 6 reports the *raw* scores that underlie the aggregate plots in §5.2. We include two complementary views of system quality:

(a) **Answer Relevancy** (↑) — RAGAS cosine similarity between the model answer and the ground-truth private answer, averaged over the 3 k test questions.

(b) **Leakage Score** (↓) — ordinal rating returned by our LLM-as-Judge metric (§6), where 1 indicates no leakage and 5 indicates near-verbatim disclosure.

How to read the table. Rows are grouped first by metric, then by foundation model (OpenAI GPT-4o-mini, Llama 3.1-70B, DeepSeek-V3, Qwen-3-235B). Columns list the nine privacy pipelines evaluated in the main paper. Higher is better for Answer Relevancy; lower is better for Leakage Score. The best value per row is **bold-faced**.

Software Packages and Parameter Settings

Table7 lists every external package we relied on, together with the exact version, role in

Table 6: Answer–relevancy (higher is better) and leakage score (lower is better) for four LLMs across nine privacy pipelines.

Metric	Model	Normal	Redact	Zerogen	Typed-Holder	Hybrid	Hier.	LDP	Paraphrase	VAGUE
Answer Rel.	OpenAI	0.793	0.669	0.467	0.672	0.795	0.789	0.778	0.738	0.557
	LLaMA	0.743	0.000	0.433	0.000	0.656	0.740	0.613	0.488	0.341
	DeepSeek	0.773	0.669	0.435	0.585	0.709	0.774	0.751	0.705	0.469
	Qwen	0.772	0.734	0.280	0.718	0.735	0.772	0.743	0.740	0.233
Leakage Score	OpenAI	3.053	2.729	1.713	2.840	3.080	3.055	3.147	2.931	2.278
	LLaMA	3.076	1.192	1.750	1.189	2.968	3.088	2.915	2.496	2.586
	DeepSeek	2.914	2.471	1.747	2.330	2.702	2.933	2.998	2.431	1.943
	Qwen	2.941	2.815	1.717	2.820	2.883	2.925	2.970	2.794	1.586

the pipeline, key parameters, and an official download link. All packages are installed from `pip` unless stated otherwise; a reproducible `requirements.txt` accompanies our code release.

Consistency of Artifact Use With Intended Purpose

External artifacts. All third-party resources—LLMs, retrieval corpora, evaluation benchmarks, and software libraries—were used strictly within the scope licensed or documented by their authors:

- *OpenAI GPT-4o-mini*, *Llama-370B*, *DeepSeek-V3*, and *Qwen-3235B* were accessed via official APIs or model checkpoints under the providers' research or non-commercial terms. We did not fine-tune, redistribute, or expose model weights.
- Public corpora employed for retrieval (e.g., Wikipedia 2024-05 snapshot) and evaluation datasets (e.g., HotpotQA) are released for academic research; we neither redistribute nor re-licensed them.

Artifacts we release. PrivRAG, our newly-curated benchmark, contains synthetic documents automatically generated from publicly available seed material and *does not* include any personal or proprietary information. We distribute the dataset, code, and prompt templates under the CC-BY-NC 4.0 licence with an explicit "**research-only, non-commercial**" clause. This is fully compatible with the access restrictions of the sources used to create the dataset and prevents downstream deployments that might contravene the original terms of use.

C Prompt Templates

Table 7: Third-party software employed in this work.

Package	Ver.	Purpose / Settings	URL
SpaCy + en_core_web_trf	3.7.2	NER and sentence segmentation; default pipeline; GPU enabled	https://spacy.io
Flair (flair/ner-english-ontonotes-large)	0.13	Second NER pass; batch_size=8	https://github.com/flairNLP/flair
NLTK	3.8.1	Fallback tokeniser; BLEU with smoothing method I	https://www.nltk.org
RapidFuzz	3.6.1	String similarity for ILS diagnostics	https://github.com/maxbachmann/RapidFuzz
ROUGE (py-rouge)	1.0.1	ROUGE-L scoring; default stop-word list	https://pypi.org/project/py-rouge/
sacreBLEU	2.4.2	BLEU-4 (-lc -smooth_add1)	https://github.com/mjpost/sacrebleu
ragas	0.1.6	Faithfulness / Answer-Relevancy with top_k=5	https://github.com/explodinggradients/ragas
langchain-openai	0.1.0	LLM wrapper; temperature and context-window control	https://python.langchain.com
openai SDK	1.15.0	Embedding calls; timeout=20 s	https://platform.openai.com

Categorisation Prompt (VAGUE-Gate)

You are a privacy gate. Given <chunk> and a label in {PUBLIC, SENSITIVE, CONFIDENTIAL}, rewrite the chunk so that
- tokens with utility $<\varepsilon$ are *dropped* or generalised;
- all other content is kept verbatim.

Return **JSON**: {"rewritten": "..."}.

Precision–Pass Prompt (T=0)

Rewrite the following text with **vagueness** $\varepsilon =< X >$. Drop or generalise private details, keep public content intact.

<chunk>

Output (JSON only): {"rewritten": "..."}

Deep-Obfuscation Prompt (T=0.7)

Make the passage *still vaguer*. Keep meaning, re-phrase nouns, swap clause order, remove superfluous dates.

<current_version>

Paraphrase Prompt [?]

Given the context, extract *essential* parts verbatim; delete the rest.
Context: «<{input_context}>»
Extracted relevant parts:

ZeroGen Prompt [?]

The context is: {*orig_context*}. {*extracted_entities*} is the answer to:
Generate **10** question–answer pairs in the form `question: ... answer: ...`

AttrPrompt (Attribute Discovery) [?]

"What are the five most important *attributes* for generating medical Q&A data?" List them, then propose three sub-topics for each.

SAGE Phase 1 Prompt [?]

Summarise key points of the Doctor–Patient conversation below. Return exactly the five attributes for the **Patient** and five for the **Doctor** in the provided schema.

«< conversation »>

SAGE Phase 2 Prompt

Using the attribute list: «< attributes »>
Generate a *single-round* patient question and doctor reply that cover *all* attributes. Do **not** produce extra dialogue.

LDP-RAG Entity-Perturb Prompt [?]

Locate PERSON, ORG, LOC, DATE, etc. Apply ε=0.5 randomised response per entity. Return perturbed text only.

Redact (Rule-based)

Regex-replace every detected private entity with "IIIIII".

Typed-Holder [?]

Replace entities by their coarse type token (e.g. `PERSON`, `DATE`, `MONEY`).

Note: All prompts are shown verbatim except for ellipsis placeholders <...>.

1

Figure 3: **Prompt templates for every privacy pipeline.** The PDF is rendered verbatim to preserve exact wording and formatting.