Auditing and Mitigating Privacy Leakage in Cloud-Edge Collaborative Decoding

Kejia Zhang[1] **Tianyuan Zou**[2] **Zixuan Gu**[3] **Yang Liu**[1*]
[1]The Hong Kong Polytechnic University
[2]Institute for AI Industry Research, Tsinghua University
[3]School of Software, Tsinghua University

Abstract

Applications such as personalized assistance and proprietary document analysis require large language models (LLMs) to generate outputs from private data. Yet powerful LLMs typically cannot be deployed on the resource-constrained devices where private data resides, and uploading private data to cloud-hosted LLMs exposes sensitive information. Recent work addresses this tension with a *cloud-edge collaborative decoding* paradigm, where private data are kept on the edge with a small language model (SLM) producing next-token distributions, which are fused with predictions from a cloud LLM operating solely on public data. In this paper, we systematically analyze the privacy risks of such a paradigm with a novel evaluation framework using constructed QA datasets, which show that such collaboration can expose substantial private-context information. To address such privacy leakage, we propose CoVEIL, a defense mechanism which dynamically optimizes transmitted signals to suppress leakage during decoding time while preserving the collaborative quality. Extensive evaluations demonstrate that CoVEIL consistently improves the privacy-utility trade-off over existing baselines by reducing data leakage by up to 87.2%, with minimal accuracy loss.

1 Introduction

Large language models (LLMs) have emerged as powerful tools for reasoning over private data, with applications spanning clinical decision support (Hager et al., 2024), personal assistance (Salemi and Zamani, 2025), and proprietary document analysis (Ma et al., 2024). However, deploying LLMs raises a core conflict between capability and privacy. Powerful LLMs are computationally expensive to run on edge devices that private data often resides (Lu et al., 2025; Fu et al., 2024), while uploading private data to cloud-hosted LLMs exposes it to external service providers (Tong et al., 2025). Prior work mitigates this tension by deploying small language models (SLMs) through knowledge distillation, model compression (Gu et al., 2024; Sun et al., 2024; Hu et al., 2025; Feng et al., 2026; Tuo and Wang, 2026), or by sanitizing private prompts before cloud inference (Tong et al., 2025; Zhang et al., 2025a). However, these approaches limit task utility by relying on smaller local models or removing useful context before cloud inference.

To better leverage LLM capability without uploading the private context, an important research direction couples a cloud-hosted LLM with a locally deployed SLM, allowing the two models to collaborate through decoding signals while the cloud LLM operates only on public or non-sensitive data and the SLM retains the private context (Zhang et al., 2024b; Zheng et al., 2025; She et al., 2025; Liu et al., 2025). We refer to this paradigm as *cloud-edge collaborative decoding*. At each autoregressive step, the edge SLM and the cloud LLM each produce a next-token distribution, and the two distributions are fused to generate the next token.

Depending on where the two next-token distributions are fused, this paradigm can be categorized into **cloud-side fusion**, where the edge SLM uploads its next-token probabilities (probability-level signal) to the cloud (Zhang et al., 2024b), and **edge-side fusion**, where the edge locally fuses the two distributions and synchronizes the sampled token (token-level signal) to the cloud (Zheng et al., 2025; She et al., 2025). Although prior work has shown that next-token probabilities can leak information about input data (Morris et al., 2024; Finlayson et al., 2024; Nazir et al., 2025), a systematic analysis of the distinct privacy risks inherent to both probability-level and token-level signals remains largely absent.

Motivated by this, we systematically analyze

*Corresponding author: yang-veronica.liu@polyu.edu.hk.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 16572–16592
October 24-29, 2026 ©2026 Association for Computational Linguistics

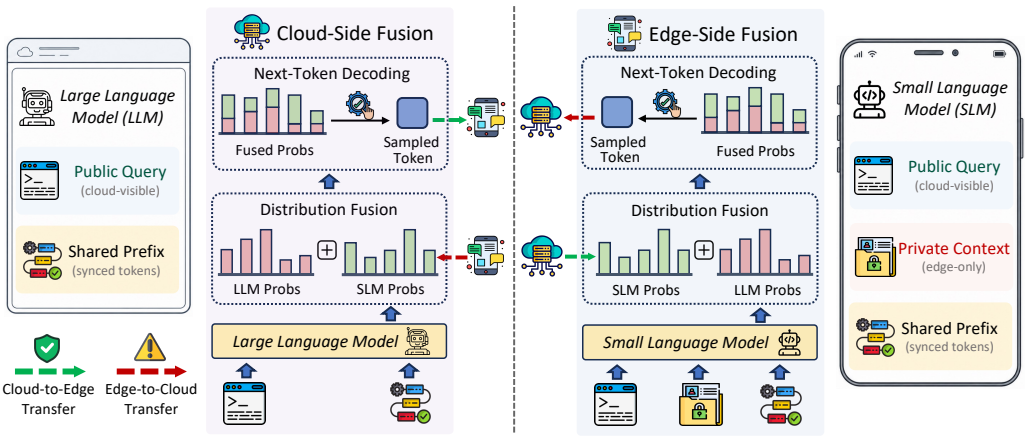


Figure 1: Cloud-edge collaborative decoding, where private context remains on the edge under two fusion modes.

whether the cloud-observed decoding signals during collaborative decoding leak the private edge-side context. We introduce DECODELEAK, a privacy evaluation framework for cloud-edge collaborative decoding, and construct two QA benchmarks from medical and commonsense QA sources (Jin et al., 2021; Huang et al., 2019); each instance pairs a public query with an edge-side private context. Each data sample also includes manually annotated private evidence, defined as a set of critical sensitive text spans in the private context needed for answering the public query, and are used to quantify the level of leakage. For example, for a medication-selection query, private context spans such as "respiratory depression", "pupillary constriction" are annotated as private evidence because they are needed to select the correct medication. DECODELEAK evaluates leakage in two sequential steps: *private evidence recall*, which measures the fraction of annotated private evidence exposed by the top-K token IDs from the per-step next-token distributions; then *private context inversion* which uses the corresponding per-step top-K exposure sequence as the attack input for reconstructing the private context. We find that cloud-observed decoding signals can reveal substantial private evidence using our evaluation framework.

These findings motivate us to develop privacy-preserving collaborative decoding mechanisms. Existing defenses typically apply fixed global controls to uploaded distributions (Zhang et al., 2024b; Thareja et al., 2026), or use privacy-aware rules to choose the synchronized token (Ghoukasian et al., 2025; Vinod et al., 2025). Such controls may suppress utility-critical signals or retain leaky low-utility ones. We therefore propose COVEIL, which optimizes each cloud-observed signal under a utility-privacy trade-off at decoding time, enabling adaptive selection of upload probabilities in cloud-side fusion, or a synchronized token in edge-side fusion. Experiments show that per-step utility-aware signal selection reduces private evidence leakage while preserving task utility.

In summary, this work makes three contributions. First, we identify and analyze privacy risks of cloud-edge collaborative decoding, showing that uploaded SLM probabilities and synchronized tokens can expose private evidence and enable private context inversion. Second, we present DECODELEAK, a privacy evaluation framework, and construct two QA benchmarks with public queries, private edge-side contexts, and private-evidence annotations.Third, we propose COVEIL, a post-hoc privacy-aware collaborative method that optimizes cloud-observed decoding signals at decoding time without incurring additional model training, and show that it significantly improves the privacy-utility trade-off over existing baselines.

2 Problem Setup and Threat Model

2.1 Cloud-Edge Collaborative Decoding

In cloud-edge collaborative decoding, a cloud-hosted LLM and an edge-side SLM jointly perform next token prediction while the private context x_{priv} remains solely on the edge and the public query x_{pub} is shared by both models (Zhang et al., 2024b; Tian et al., 2026; She et al., 2025), as shown in Figure 1. Let $\mathcal{V}$ denote the vocabulary shared between SLM and LLM. At each decoding step t, the cloud LLM and the edge SLM produce next-token probability distributions over $\mathcal{V}$:

$$\begin{cases} p_L^{(t)}(\cdot) = f_{\theta_L}(\cdot \mid y_{<t}, x_{\text{pub}}), & (1) \\ p_S^{(t)}(\cdot) = f_{\theta_S}(\cdot \mid y_{<t}, x_{\text{pub}}, x_{\text{priv}}), & (2) \end{cases}$$

where f_{θ_L} and f_{θ_S} denote the cloud LLM and edge SLM parameterized by θ_L and θ_S, respectively; $p_L^{(t)}(\cdot)$ and $p_S^{(t)}(\cdot)$ denote their next-token probability distributions over the vocabulary $\mathcal{V}$ at step t; $y_{<t}$ denotes the shared autoregressive prefix. The fused next-token distribution is typically obtained by a weighted interpolation (Zhang et al., 2024b):

$$p_{\text{fuse}}^{(t)} = \alpha \cdot p_L^{(t)} + (1-\alpha) \cdot p_S^{(t)}, \tag{3}$$

where $\alpha \in [0,1]$ is the fusion weight. The next token is then sampled as $y_t^{\text{fuse}} \sim p_{\text{fuse}}^{(t)}(\cdot)$.

2.2 Fusion Modes

Let $o_L^{(t)}$ denote the cloud-observed decoding signal at step t. The fusion mode determines $o_L^{(t)}$ as:

$$o_L^{(t)} = \begin{cases} p_S^{(t)}, & \text{cloud-side fusion,} \\ y_t^{\text{fuse}}, & \text{edge-side fusion.} \end{cases} \tag{4}$$

Cloud-Side Fusion (Probability-Level Signal). The edge uploads the SLM next-token probabilities $p_S^{(t)}$ to the cloud, which fuses them with the LLM distribution $p_L^{(t)}$ and samples y_t^{fuse} from the fused distribution (Zhang et al., 2024b; Tian et al., 2026).

Edge-Side Fusion (Token-Level Signal). The cloud sends $p_L^{(t)}$ to the edge, and the edge fuses it with $p_S^{(t)}$ and samples y_t^{fuse} (Zheng et al., 2025; She et al., 2025). The sampled token is synchronized to the cloud and appended to the shared autoregressive prefix, thereby conditioning the cloud LLM's subsequent next-token distribution.

2.3 Threat Model

We assume that the cloud LLM is honest-but-curious: it follows the decoding protocol but attempts to infer the private context from its cloud-side observations (Qu et al., 2025; Ding et al., 2024). After T decoding steps, the cloud has access to the public query x_{pub}, its own parameters θ_L, the generated prefix $y_{\leq T}^{\text{fuse}}$, and the cloud-observed decoding signal sequence $\{o_L^{(t)}\}_{t=1}^{T}$.

This induces two leakage channels depending on the fusion mode: in cloud-side fusion, **probability-level leakage** arises because the cloud observes the uploaded SLM next-token probabilities $p_S^{(t)}$; in edge-side fusion, **token-level leakage** arises because the cloud observes the synchronized token y_t^{fuse} in the shared prefix and can compute its own prefix-conditioned distribution $p_L^{(t)}(\cdot \mid y_{<t}^{\text{fuse}}, x_{\text{pub}})$. In both cases, the cloud attempts to infer private information x_{priv} from these observations.

3 DECODELEAK

DECODELEAK first defines annotated private evidence as the evaluation target, then evaluates leakage in two sequential steps: private evidence recall over cloud-observed top-K token sets, followed by private context inversion from the corresponding per-step exposure sequence.

3.1 Private Evidence and Benchmark

Prior collaborative decoding studies focus on task performance and efficiency without private-evidence annotations (Zhang et al., 2024b; Tian et al., 2026), while privacy and inversion studies typically focus on single-model inference rather than the cloud-edge setting with a public/private input split (Qu et al., 2025; Flemings et al., 2025). We therefore construct benchmarks that explicitly separate public queries, private edge-side contexts, and annotated private evidence.

We define the evaluation instance for DECODELEAK as a tuple $(x_{\text{pub}}, x_{\text{priv}}, \mathcal{E}_{\text{priv}})$, where x_{pub} is the public query available to the cloud, x_{priv} is the private context retained on the edge, and $\mathcal{E}_{\text{priv}} = \{P_1, \ldots, P_M\}$ is the private-evidence set. Each span $P_m \subseteq x_{\text{priv}}$ contains private text that is needed to answer x_{pub} but absent from x_{pub}.

We instantiate this definition with two multiple-choice QA benchmarks: MEDPRIV (1,000 medical QA instances (Jin et al., 2021)) and COMMPRIV (1,000 commonsense QA instances (Huang et al., 2019)). In each instance, x_{pub} contains the question and answer options $\mathcal{A}$, while x_{priv} is a supporting passage with correct option $a^\star$. We call $\mathcal{E}_{\text{priv}}$ answer-critical if x_{pub} is insufficient to select $a^\star$ from $\mathcal{A}$, but $(x_{\text{pub}}, \mathcal{E}_{\text{priv}})$ is sufficient.

Private evidence annotation. The private-evidence set $\mathcal{E}_{\text{priv}}$ is obtained in two stages. First, we use GPT-5.4 to propose candidate instances where the correct option $a^\star$ can be determined from $(x_{\text{pub}}, x_{\text{priv}})$ but not from x_{pub} alone. Second, we extract minimal private spans from x_{priv} that are sufficient to identify $a^\star$; both stages are manually verified. For example, in a diagnosis query, private-context spans such as "sensitive to criticism" and "socially inhibited" are annotated as private evidence because they support the diagnosis. Beyond this QA instantiation, DECODELEAK can be applied to datasets that can be split into a public query and a private context with annotated private evidence. Detailed construction and annotation procedures are provided in Appendix A.4.

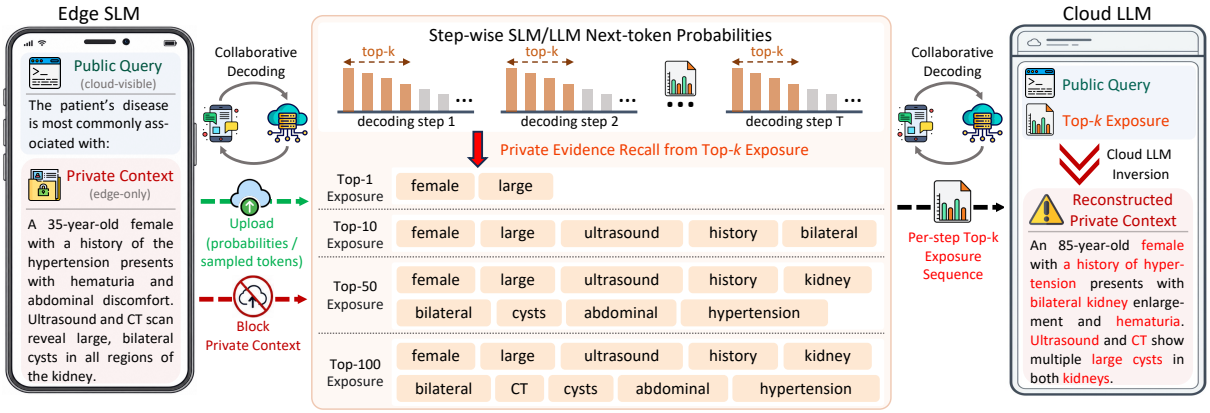


Figure 2: DECODELEAK overview: cloud-observed decoding signals are evaluated in two sequential stages: private evidence recall from top-K token IDs, followed by private context inversion from the per-step exposure sequence.

3.2 Private Evidence Recall

Private evidence recall is the first step of DECODELEAK; it quantifies leakage as the fraction of annotated private evidence exposed by cloud-observed top-K token sets extracted from per-step next-token distributions. At step t, the signal exposure set is:

$$E_t^K = \begin{cases} \mathrm{TopK}(p_S^{(t)}), & \text{probability-level,} \\ \mathrm{TopK}\left(p_L^{(t)}(\cdot \mid y_{<t}^{\text{fuse}}, x_{\text{pub}})\right), & \text{token-level,} \end{cases} \tag{5}$$

where $\mathrm{TopK}(\cdot)$ returns the K token IDs with the highest probability. $E^K = \bigcup_{t=1}^{T} E_t^K$ aggregates the exposed token IDs across T steps, while the sequence $\{E_t^K\}_{t=1}^{T}$ is used for private context inversion in Section 3.3. A higher recall therefore indicates that more annotated private evidence is directly present in the decoding signals observed by the cloud, suggesting more severe privacy leakage.

We report recall-style metrics between E^K and $\mathcal{E}_{\text{priv}}$ from token-, word-, and span-level views (Thareja et al., 2026; Vinod et al., 2025). Let $\mathcal{T}(\cdot)$ tokenize a text segment or a set of spans into its set of content token IDs under $\mathcal{V}$, excluding stopwords and punctuation. Let $\mathcal{W}(\cdot)$ return the content word set, decoded from token IDs when applied to E^K and tokenized from text otherwise.

Token Evidence Recall@K quantifies overlap at the token-ID level:

$$\text{Token-ER@}K = \frac{|\mathcal{T}(\mathcal{E}_{\text{priv}}) \cap E^K|}{|\mathcal{T}(\mathcal{E}_{\text{priv}})|}. \tag{6}$$

ROUGE-1 Evidence Recall@K captures lexical overlap between the decoded top-K token set and the private-evidence text at the word level:

$$\text{ROUGE1-ER@}K = \frac{|\mathcal{W}(\mathcal{E}_{\text{priv}}) \cap \mathcal{W}(E^K)|}{|\mathcal{W}(\mathcal{E}_{\text{priv}})|}. \tag{7}$$

Span Evidence Recall@K reports recall averaged over the private-evidence spans:

$$\text{Span-ER@}K = \frac{1}{M} \sum_{m=1}^{M} \frac{|\mathcal{T}(P_m) \cap E^K|}{|\mathcal{T}(P_m)|}. \tag{8}$$

AUC@K summarizes Token-ER across cutoffs up to K to reduce sensitivity to a specific cutoff:

$$\text{AUC@}K = \frac{1}{K} \sum_{k=1}^{K} \text{Token-ER@}k. \tag{9}$$

As K increases, private evidence recall consistently rises across both fusion modes and datasets (Figure 3). At $K = 100$ (fewer than 0.1% of the Qwen vocabulary), Token-ER reaches approximately 60% of private-evidence tokens when the cloud observes uploaded SLM probabilities in cloud-side fusion, and 50% when it observes synchronized tokens in edge-side fusion.

3.3 Private Context Inversion

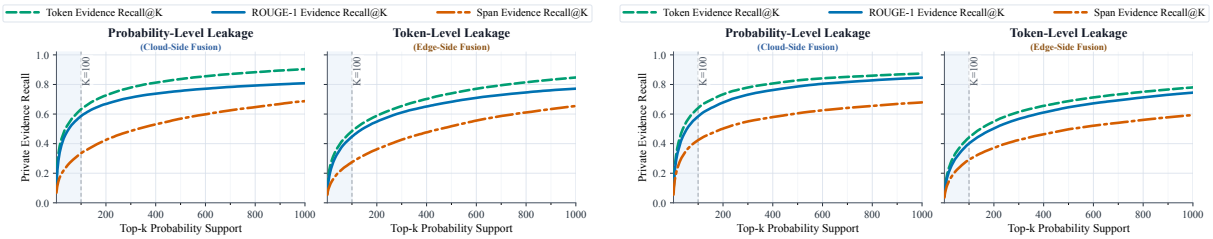
Private context inversion measures whether the cloud can reconstruct x_{priv} from the signal exposure sequence $\{E_t^K\}_{t=1}^{T}$ and x_{pub}, as illustrated in the right panel of Figure 2.

Given x_{pub} and the signal exposure sequence $\{E_t^K\}_{t=1}^{T}$, the cloud constructs an inversion prompt from $(x_{\text{pub}}, \{E_t^K\}_{t=1}^{T})$[1] and uses the cloud LLM to reconstruct the private context $\hat{x}_{\text{priv}}$:

$$\hat{x}_{\text{priv}} = f_{\theta_L}\left(\cdot \mid \pi\left(x_{\text{pub}},\ \{E_t^K\}_{t=1}^{T}\right)\right) \tag{10}$$

Inversion quality is quantified by comparing $\hat{x}_{\text{priv}}$ with the ground-truth x_{priv}.

[1]The inversion prompt template is shown in Figure 19.



(a) MEDPRIV (b) COMMPRIV

Figure 3: Private evidence recall across leakage channels in Qwen2.5-72B/1.5B collaboration using 200 examples.

ROUGE-1 Recall measures the fraction of private-context unigrams recovered by the inversion.
ROUGE-1 F1 measures balanced unigram overlap between the reconstructed and full private contexts (Lin, 2004).

As reported in Table 1, the exposure sequence enables substantial private context inversion, with ROUGE-1 Recall of 41.1% under cloud-side fusion and 39.2% under edge-side fusion. Together with the private evidence recall results above, this shows that cloud-observed decoding signals expose private information at both the token-exposure and text-reconstruction levels, motivating privacy-aware optimization.

4 COVEIL: Privacy-Aware Fusion

To reduce privacy leakage from uploaded distributions and synchronized tokens, we propose COVEIL, a privacy-aware defense that optimizes cloud-observed decoding signals at decoding time while preserving decoding utility. COVEIL runs on the edge in both fusion modes, using the cloud-provided $p_L^{(t)}$ together with the edge-local $p_S^{(t)}$. In cloud-side fusion, the edge uploads only the selected probability–ID pairs $\{(i, p_S^{(t)}(i)) : i \in \mathcal{S}_t\}$; in edge-side fusion, it performs fusion locally and synchronizes only the selected token $i_t^\star$. Thus, the full SLM distribution $p_S^{(t)}$ remains on the edge in both modes.

4.1 Stepwise Privacy-Utility Optimization

At each decoding step, COVEIL optimizes the candidate cloud-observed decoding signal o_t by directly balancing utility against privacy cost:

$$\min_{o_t} \mathcal{J}_t(o_t) = -\mathcal{U}_t(o_t) + \lambda \cdot \mathcal{P}_t(o_t), \qquad (11)$$

where o_t denotes the candidate realization of the cloud-observed signal $o_L^{(t)}$ optimized by COVEIL; $\mathcal{U}_t$ and $\mathcal{P}_t$ measure decoding utility and privacy cost, respectively; and $\lambda > 0$ denotes the trade-off parameter. The concrete form of o_t, $\mathcal{U}_t$, and $\mathcal{P}_t$ depends on the fusion mode, as detailed below.

4.2 Cloud-Side Fusion Defense

Cloud-side fusion exposes the SLM next-token probabilities to the cloud (Morris et al., 2024; Finlayson et al., 2024). COVEIL reduces this leakage through adaptive sparse upload of selected SLM next-token probabilities. For efficiency, we restrict upload decisions to $\mathcal{C}_t = \mathrm{TopK}(p_S^{(t)})$, since low-probability positions add many optimization variables but contribute little to the fused distribution. A sensitivity study is provided in Appendix A.2.1.

Specifically, we introduce a selective upload mask m_t to determine which token positions in the SLM next-token probabilities are uploaded :

$$m_t(i) \in \begin{cases} [0,1], & i \in \mathcal{C}_t, \\ \{0\}, & i \notin \mathcal{C}_t. \end{cases} \qquad (12)$$

where $[0, 1]$ enables gradient computation; the final upload set is binary. With the upload mask m_t, the per-step objective can be written as:

$$\min_{m_t \in [0,1]^{|\mathcal{C}_t|}} \mathcal{J}_t(m_t) = -\mathcal{U}_t(m_t) + \lambda \cdot \mathcal{P}_t(m_t). \quad (13)$$

Utility function. The utility term assigns higher value to upload masks that preserve the next-token choice of the unmasked full-fusion distribution by increasing its margin over competing candidates (Zhang et al., 2025b). Let $y_t^\star = \arg\max_i p_{\text{fuse}}^{(t)}(i)$ denote the top-ranked token under the unmasked fused distribution. For each candidate token $j \in \mathcal{C}_t$, $M_t(m_t; j)$ compares the masked fusion score of $y_t^\star$ with that of j:

$$M_t(m_t; j) = s_t(y_t^\star; m_t) - s_t(j; m_t). \qquad (14)$$

where $s_t(i; m_t) = \alpha \cdot p_L^{(t)}(i) + (1-\alpha) \cdot m_t(i) \cdot p_S^{(t)}(i)$ is the masked fusion score, and $m_t(i)$ controls the SLM probability upload.

The per-step utility is the average margin gain over the zero-upload baseline ($m_t = \mathbf{0}$):

$$\mathcal{U}_t(m_t) = \frac{1}{|\mathcal{C}_t|} \sum_{j \in \mathcal{C}_t} \big(M_t(m_t; j) - M_t(\mathbf{0}; j)\big), \tag{15}$$

where $M_t(\mathbf{0}; j)$ denotes the margin when no SLM token probability is uploaded. This utility metric requires no labeled data or external supervision and is computed solely from the collaborative decoding distributions at each step.

Privacy function. CoVEIL estimates private-context influence by the discrepancy between edge and cloud token distributions:

$$\mathcal{P}_t(m_t) = \sum_{i \in \mathcal{C}_t} m_t(i) \left| p_S^{(t)}(i) - p_L^{(t)}(i) \right|. \tag{16}$$

The absolute value penalizes the magnitude of the edge-cloud deviation, regardless of its direction.

Upload rule. To decide which SLM probabilities to upload, CoVEIL uses a first-order Taylor expansion of the objective around the zero-upload baseline $m_t = \mathbf{0}$:

$$\mathcal{J}_t(m_t) \approx \mathcal{J}_t(\mathbf{0}) + \sum_{i \in \mathcal{C}_t} g_t(i)\, m_t(i), \tag{17}$$

where $g_t(i) = \left.\frac{\partial \mathcal{J}_t(m_t)}{\partial m_t(i)}\right|_{m_t = \mathbf{0}}$ is the marginal change in $\mathcal{J}_t$ from uploading the SLM probability for token i. Since CoVEIL minimizes $\mathcal{J}_t$, $g_t(i) < 0$ indicates that uploading token i decreases the objective, whereas $g_t(i) \geq 0$ indicates no improvement under the linear approximation. Therefore, the uploaded set is:

$$\mathcal{S}_t = \{i \in \mathcal{C}_t \mid g_t(i) < 0\}. \tag{18}$$

4.3 Edge-Side Fusion Defense

In edge-side fusion, CoVEIL selects a token from $\mathcal{C}_t = \mathrm{TopK}(p_{\text{fuse}}^{(t)})$, rather than choosing an upload mask m_t as in cloud-side fusion. The selected token is appended to the shared autoregressive prefix at each decoding step. For each candidate token $i \in \mathcal{C}_t$, the per-step objective is:

$$\mathcal{J}_t(i) = -\mathcal{U}_t(i) + \lambda \cdot \mathcal{P}_t(i). \tag{19}$$

Utility function. Since edge-side fusion selects a token rather than an upload mask, its utility applies the probability-gap principle of Equation (15) to each candidate token. CoVEIL scores each candidate by its fused-probability gap relative to the fused top token $y_t^\star$:

$$\mathcal{U}_t(i) = p_{\text{fuse}}^{(t)}(i) - p_{\text{fuse}}^{(t)}(y_t^\star). \tag{20}$$

Privacy function. As in Equation (16), edge-cloud discrepancy estimates private-context influence. While cloud-side fusion penalizes uploaded deviations, edge-side fusion uses the signed difference for the selected token:

$$\mathcal{P}_t(i) = p_S^{(t)}(i) - p_L^{(t)}(i). \tag{21}$$

A positive value indicates stronger private-side support and thus higher leakage risk, whereas a negative value indicates stronger cloud-side support and lower leakage risk.

Token selection rule. CoVEIL selects the candidate that minimizes the per-step objective:

$$i_t^\star = \arg\min_{i \in \mathcal{C}_t} \mathcal{J}_t(i). \tag{22}$$

5 Experiments

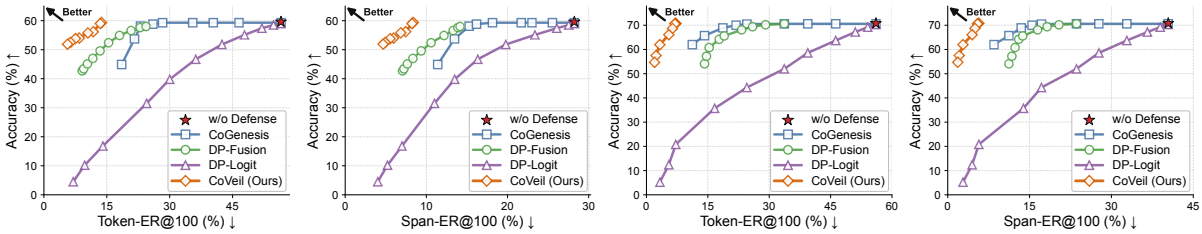
5.1 Experimental Setup

Cloud-edge model setup. Experiments are conducted on two cloud-edge collaboration settings: Qwen2.5-72B (LLM)/1.5B (SLM) (Qwen et al., 2024) and Llama-3.1-70B (LLM)/Llama-3.2-3B (SLM) (Grattafiori et al., 2024). The corresponding tokenizer vocabulary sizes are 151,643 and 128,256, respectively.[2]

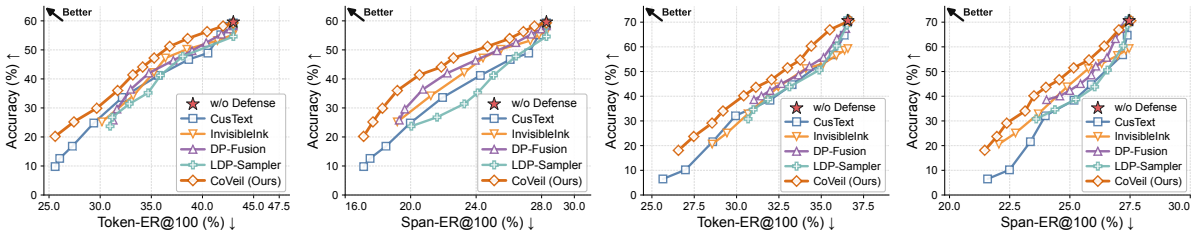
Datasets and evaluation metrics. We use MEDPRIV and COMMPRIV from Section 3. Privacy is evaluated by private evidence recall and private context inversion from Sections 3.2 and 3.3, and utility by answer accuracy. Unless otherwise specified, the recall metrics in Section 3.2 are reported at $K = 100$, a compact head cutoff covering less than 0.1% of either vocabulary.

Comparison methods. Cloud-side fusion privacy defense baselines include CoGenesis (Zhang et al., 2024b), DP-Logit (Ji and Li, 2024), and DP-Fusion (Thareja et al., 2026). Edge-side fusion privacy defense baselines include CusText (Chen et al., 2023), LDP-Sampler (Ghoukasian et al., 2025), DP-Fusion (Thareja et al., 2026), and InvisibleInk (Vinod et al., 2025). Several of these baselines provide method-specific formal differential privacy guarantees under their respective assumptions, whereas CoVEIL provides empirical privacy improvements without a formal DP guarantee. The trade-off weight λ is not a formal privacy budget ϵ; our comparisons are limited to empirical privacy-utility behavior under the common evaluation protocol.

[2]Further implementation details are provided in Appendices A.1 and A.1.4.

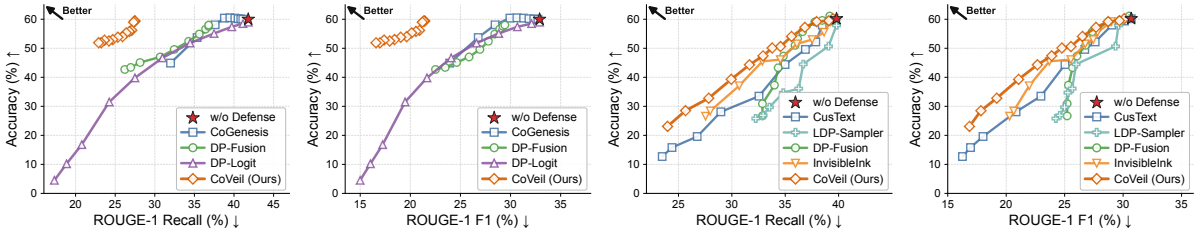


(a) MEDPRIV: Cloud-side fusion defense (b) COMMPRIV: Cloud-side fusion defense



(c) MEDPRIV: Edge-side fusion defense (d) COMMPRIV: Edge-side fusion defense

Figure 4: Privacy-utility trade-off curves: private evidence recall under Qwen2.5-72B/1.5B collaboration.



(a) MEDPRIV: Cloud-side fusion defense (b) MEDPRIV: Edge-side fusion defense

Figure 5: Privacy-utility trade-off curves: private context inversion under Qwen2.5-72B/1.5B collaboration.

5.2 Main Results

DecodeLeak Results. Table 1 reports undefended leakage on MEDPRIV; the red stars in Figures 4 and 5 mark the corresponding no-defense points. Both fusion modes expose private evidence through cloud-observed decoding signals. Cloud-side fusion reaches 56.6% Token-ER@100 and 41.1% inversion ROUGE-1 Recall, while edge-side fusion reaches 43.0% and 39.2%, respectively. These results show that undefended collaboration leaks private information at both the signal-exposure and context-inversion levels.

Cloud-Side Fusion Defense. As shown in Figure 4(a,b) and Figure 5(a),[3] COVEIL reduces Token-ER@100 by 75.7% on MEDPRIV and 87.2% on COMMPRIV with at most 0.2 percentage-point accuracy loss. At matched accuracy (≈59%), COVEIL achieves Token-ER@100 of 13.4%, compared to 28.3% for CoGenesis and 24.4% for DP-Fusion. COVEIL outperforms these baselines by evaluating the utility-privacy trade-off at each token position, whereas fixed upload constraints in CoGenesis and DP-Fusion leave leaky positions exposed and uniform perturbation in DP-Logit reduces leakage only with sharp accuracy drops.

Edge-Side Fusion Defense. Figure 4(c,d) reports private-evidence recall, and Figure 5(b) reports private context inversion under edge-side fusion. Edge-side fusion exposes stronger privacy leakage because each synchronized token is appended to the cloud-observed prefix and depends on the private-conditioned edge distribution. COVEIL reduces Token-ER@100 from 43.0% to 40.4% and Span-ER@100 from 28.3% to 26.9% on MEDPRIV, and from 36.6% to 35.5% and 27.5% to 27.0% on COMMPRIV. At matched accuracy, COVEIL achieves lower Token-ER@100 than InvisibleInk, DP-Fusion, and CusText; at matched leakage, it achieves 7.2% higher accuracy than InvisibleInk. This advantage comes from utility-aware token selection: COVEIL avoids appending tokens strongly supported by the private-side model unless they provide sufficient utility, whereas token-side baselines rely on preset sampling or perturbation rules.

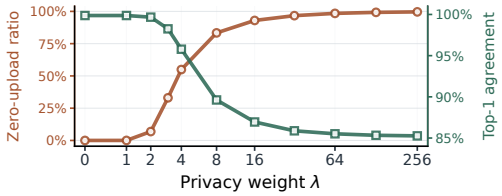
5.3 Sensitivity Analysis

We analyze per-step decisions of COVEIL: cloud-side fusion chooses which SLM token probabilities

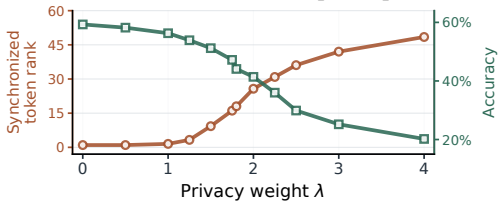
[3] ROUGE1-ER@100, AUC@100, and LLaMA results are reported in Appendices A.2 and A.2.2.

Fusion	Acc	Private Evidence Recall (@100)				Private Context Inversion	
		Token-ER	ROUGE1-ER	Span-ER	AUC	ROUGE-1 Recall	ROUGE-1 F1
Cloud-side	59.3	56.6	48.3	28.3	46.7	41.1	32.7
Edge-side	59.3	43.0	34.6	28.3	35.0	39.2	30.3

Table 1: DECODELEAK results on MEDPRIV under undefended Qwen2.5-72B/1.5B collaboration.



(a) Cloud-side fusion: adaptive upload



(b) Edge-side fusion: token selection

Figure 6: Effect of the privacy weight λ on per-step COVEIL decisions on MEDPRIV.

to upload, while edge-side fusion chooses which token to append to the shared autoregressive prefix. We sweep λ to connect these decisions with the observed privacy-utility trends.

Effect of λ in cloud-side defense. Figure 6(a) analyzes how the privacy weight λ changes cloud-side upload behavior. The left y-axis reports the zero-upload ratio, the fraction of decoding steps where COVEIL uploads no SLM token probability; the right y-axis reports top-1 agreement with the original full-fusion choice. As λ increases, the zero-upload ratio rises because the objective assigns more weight to privacy cost. At $\lambda = 8$, 83.3% of steps upload no SLM probabilities, while top-1 agreement remains 89.6%. This shows that many SLM token probabilities can be suppressed without changing the full-fusion top choice in most steps, reducing privacy exposure. Figure 7 further illustrates the per-token upload rule: candidates with $g_t(i) < 0$ reduce the objective and are uploaded, while candidates with $g_t(i) \geq 0$ are not.

Effect of λ in edge-side defense. Figure 6(b) analyzes how privacy weight λ changes edge-side token selection. The left y-axis reports selected-token rank under $p^{(t)}_{\text{fuse}}$, and the right y-axis reports accuracy. Edge-side decisions are more constrained because each step must append one token to shared autoregressive prefix. For $\lambda \in [0, 1]$, the average rank increases only from 1.0 to 1.5 while accuracy decreases from 59.3% to 56.3%; at $\lambda = 4.0$, the

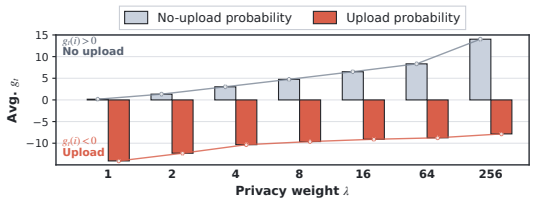


Figure 7: Cloud-side upload decisions on MEDPRIV: candidate SLM token probabilities are uploaded when $g_t(i) < 0$ and suppressed when $g_t(i) \geq 0$.

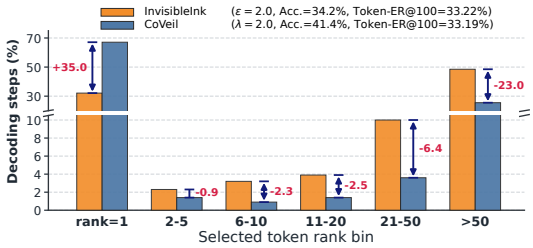


Figure 8: Edge-side selected-token rank distribution at matched Token-ER@100 on MEDPRIV; ranks are computed under $p^{(t)}_{\text{fuse}}$.

rank reaches 48.4 and accuracy drops to 20.2%. At matched leakage (Figure 8; Token-ER@100: 33.19% vs. 33.22%), COVEIL retains 67.1% of steps at rank 1 versus 32.1% in InvisibleInk, and reduces selections below rank 50 from 48.5% to 25.5%. This shows that COVEIL preserves the fused top token more often at matched leakage, consistent with its utility-aware token selection rule.

6 Related Work

Large-small language model collaboration improves generation quality and inference efficiency (Li et al., 2023; Leviathan et al., 2023; Liu et al., 2024), typically with both models sharing the same input context. Cloud-edge collaborative decoding instead keeps private context on the device and exposes only cloud-observed decoding signals, including uploaded SLM probabilities or synchronized tokens (Zhang et al., 2024b; Tian et al., 2026; She et al., 2025). These systems mainly target utility, cost, or latency, leaving privacy leakage from cloud-observed decoding signals underexplored.

Privacy attacks show that outputs and next-token distributions can reveal hidden inputs through language-model inversion (Morris et al., 2024; Skapars et al., 2025), prompt extraction (Qu et al., 2025; Luo et al., 2025), or probability analysis (Finlayson et al., 2024; Flemings et al., 2025). Typical inversion attacks require supervised training on public data (Nazir et al., 2025; Zhang et al., 2024a) or white-box gradient access (Skapars et al., 2025). However, these studies do not analyze cloud-edge

collaborative decoding, where the cloud observes per-step signals. DECODELEAK instead measures private evidence recall and performs private context inversion from the signal exposure sequence.

Existing defenses address exposed signals through two routes. For uploaded distributions, CoGenesis sparsifies SLM probabilities (Zhang et al., 2024b), DP-Logit adds noise (Ji and Li, 2024), and DP-Fusion calibrates the private-conditioned distribution toward a public baseline (Thareja et al., 2026). For synchronized tokens, CusText sanitizes tokens (Chen et al., 2023), LDP-Sampler samples from a public-anchor distribution (Ghoukasian et al., 2025), DP-Fusion samples from its calibrated distribution (Thareja et al., 2026), and InvisibleInk clips the SLM–LLM residual before private sampling (Vinod et al., 2025). In contrast, COVEIL makes per-step, fusion-mode-specific decisions by optimizing the utility-privacy trade-off.

7 Conclusion

We analyzed privacy leakage in cloud-edge collaborative decoding, where private context remains on the edge but the cloud observes per-step decoding signals. To support this analysis, we introduced DECODELEAK with two QA benchmarks, showing that uploaded SLM probabilities and synchronized tokens can expose private evidence and enable private context inversion. We further proposed COVEIL, which makes per-step upload and synchronization decisions under a utility-privacy objective. Experiments show that COVEIL improves the privacy-utility trade-off in both fusion modes, highlighting that cloud-observed decoding signals should be treated as privacy-sensitive interfaces.

Limitations

Our experiments consider cloud-edge model pairs from the same model family with a shared tokenizer vocabulary. This setting is aligned with probability-level fusion, where next-token probabilities are compared or combined over a common token space. Extending the framework to heterogeneous tokenizers would require explicit tokenizer or probability alignment before fusion, which we leave to future work. For private context inversion, we adopt a prompt-based attacker that provides the cloud LLM with the public query and accumulated signal exposure sets. We do not systematically optimize inversion prompts or train a separate inversion model. This choice keeps the evaluation simple and reproducible, but stronger inversion strategies may affect the measured reconstruction quality.

Acknowledgments

This work was supported by the Presidential Young Scholar Scheme (Project No. P0056638), RIFL (Project No. 4-CG00), and RIAIoT (Project No. P0059914) at The Hong Kong Polytechnic University, as well as by the Artificial Intelligence Research Centre at the PolyU–Daya Bay Technology and Innovation Research Institute.

References

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 *ACL Findings*.

Shiwei Ding, Lan Zhang, Miao Pan, and Xiaoyong Yuan. 2024. PATROL: Privacy-oriented pruning for collaborative inference against model inversion attacks. In *WACV*.

Sicheng Feng, Zigeng Chen, Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2026. dMoE: dLLMs with learnable block experts. *arXiv preprint arXiv:2605.30876*.

Matthew Finlayson, Xiang Ren, and Swabha Swayamdipta. 2024. Logits of API-protected LLMs leak proprietary information. In *COLM*.

James Flemings, Bo Jiang, Wanrong Zhang, Zafar Takhirov, and Murali Annavaram. 2025. Estimating privacy leakage of augmented contextual knowledge in language models. In *ACL*.

Yonggan Fu, Zhongzhi Yu, Junwei Li, Jiayi Qian, Yongan Zhang, Xiangchi Yuan, Dachuan Shi, Roman Yakunin, and Yingyan Celine Lin. 2024. AmoebaLLM: Constructing any-shape large language models for efficient and instant deployment. In *NeurIPS*.

Hrad Ghoukasian, Bonwoo Lee, and Shahab Asoodeh. 2025. Locally optimal private sampling: Beyond the global minimax. In *NeurIPS*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *ICLR*.

Paul Hager, Friederike Jungmann, Robbie Holland, Kunal Bhagat, Inga Hubrecht, Manuel Knauer, Jakob Vielhauer, Marcus Makowski, Rickmer Braren, Georgios Kaissis, et al. 2024. Evaluation and mitigation

of the limitations of large language models in clinical decision-making. *Nature Medicine*, 30(9):2613–2622.

Xing Hu, Yuan Cheng, Dawei Yang, Zhixuan Chen, Zukang Xu, Jiangyong Yu, Chen Xu, Zhihang Yuan, Zhe Jiang, and Sifan Zhou. 2025. OSTQuant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. In *ICLR*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *EMNLP*.

Tianxi Ji and Pan Li. 2024. Less is more: Revisiting the Gaussian mechanism for differential privacy. In *USENIX Security*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *ICML*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *ACL*.

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

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. 2024. Tuning language models by proxy. In *COLM*.

Yang Liu, Kejia Zhang, Bingjie Yan, Tianyuan Zou, Jianqing Zhang, Zixuan Gu, Xiangsen Chen, Jianbing Ding, Xidong Wang, Jingyi Li, et al. 2025. Towards harnessing the collaborative power of large and small models for domain tasks. *arXiv preprint arXiv:2504.17421*.

Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Wei Liu, Jian Luan, Xiwen Zhang, Nicholas D Lane, and Mengwei Xu. 2025. Demystifying small language models for edge deployment. In *ACL*.

Xinjian Luo, Ting Yu, and Xiaokui Xiao. 2025. Prompt inference attack on distributed large language model inference frameworks. In *ACM CCS*.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. 2024. MMLongBench-Doc: Benchmarking long-context document understanding with visualizations. In *NeurIPS*.

John Xavier Morris, Wenting Zhao, Justin T Chiu, Vitaly Shmatikov, and Alexander M Rush. 2024. Language model inversion. In *ICLR*.

Murtaza Nazir, Matthew Finlayson, John Morris, Xiang Ren, and Swabha Swayamdipta. 2025. Better language model inversion by compactly representing next-token distributions. In *NeurIPS*.

Wenjie Qu, Yuguang Zhou, Yongji Wu, Tingsong Xiao, Binhang Yuan, Yiming Li, and Jiaheng Zhang. 2025. Prompt inversion attack against collaborative inference of large language models. In *IEEE S&P*.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Alireza Salemi and Hamed Zamani. 2025. LaMP-QA: A benchmark for personalized long-form question answering. In *EMNLP*.

Jianshu She, Wenhao Zheng, Zhengzhong Liu, Hongyi Wang, Eric Xing, Huaxiu Yao, and Qirong Ho. 2025. Token level routing inference system for edge devices. In *ACL*.

Adrians Skapars, Edoardo Manino, Youcheng Sun, and Lucas Carvalho Cordeiro. 2025. GPT, but backwards: Exactly inverting language model outputs. In *ICML Workshop*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2024. A simple and effective pruning approach for large language models. In *ICLR*.

Rushil Thareja, Preslav Nakov, Praneeth Vepakomma, and Nils Lukas. 2026. DP-Fusion: Token-level differentially private inference for large language models. In *ICLR*.

Chunlin Tian, Kahou Tam, Yebo Wu, Shuaihang Zhong, Li Li, Nicholas D. Lane, and ChengZhong Xu. 2026. Floe: Federated specialization for real-time LLM–SLM inference. *IEEE Transactions on Parallel and Distributed Systems*, 37(7):1630–1646.

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.

Kaiwen Tuo and Huan Wang. 2026. SparseSSM: Efficient selective structured state space models can be pruned in one-shot. In *ICML*.

Vishnu Vinod, Krishna Pillutla, and Abhradeep Guha Thakurta. 2025. InvisibleInk: High-utility and low-cost text generation with differential privacy. In *NeurIPS*.

Collin Zhang, John Xavier Morris, and Vitaly Shmatikov. 2024a. Extracting prompts by inverting LLM outputs. In *EMNLP*.

Juhua Zhang, Zhiliang Tian, Minghang Zhu, Yiping Song, Taishu Sheng, Siyi Yang, Qiunan Du, Xinwang Liu, Minlie Huang, and Dongsheng Li. 2025a. DYNTEXT: Semantic-aware dynamic text sanitization for privacy-preserving LLM inference. In *ACL Findings*.

Kaiyan Zhang, Jianyu Wang, Ermo Hua, Biqing Qi, Ning Ding, and Bowen Zhou. 2024b. CoGenesis: A framework collaborating large and small language models for secure context-aware instruction following. In *ACL*.

Kejia Zhang, Juanjuan Weng, Shaozi Li, and Zhiming Luo. 2025b. Towards adversarial robustness via debiased high-confidence logit alignment. In *ICCV*.

Wenhao Zheng, Yixiao Chen, Weitong Zhang, Souvik Kundu, Yun Li, Zhengzhong Liu, Eric P Xing, Hongyi Wang, and Huaxiu Yao. 2025. CITER: Collaborative inference for efficient large language model decoding with token-level routing. In *COLM*.

A Appendix

A.1 Implementation Details

A.1.1 Base Model Setup

We evaluate two cloud and edge model pairs. For the Qwen family, we use Qwen2.5-1.5B as the edge model and Qwen2.5-72B as the cloud model. For the LLaMA family, we use Llama-3.2-3B as the edge model and Llama-3.1-70B-Instruct as the cloud model. In each pair, the smaller model runs on the edge and conditions on both the public query x_{pub} and the private context x_{priv}. The larger model runs on the cloud and conditions only on x_{pub} and the shared generated prefix. Within each model family, the edge and cloud models use the same tokenizer vocabulary. The Qwen pair uses a vocabulary of 151,643 tokens, and the LLaMA pair uses a vocabulary of 128,256 tokens. This shared vocabulary allows their next-token probability distributions to be compared and fused over the same token space. We therefore do not use tokenizer remapping or cross-vocabulary probability alignment in our experiments.

A.1.2 Collaborative Decoding Setup

We use the same collaborative decoding configuration across all model families, datasets, fusion modes, and defense settings. After parameter analysis, we fix the fusion weight to $\alpha = 0.3$. At each decoding step, the fused next-token distribution is computed by weighted interpolation of the cloud and edge next-token distributions, following Section 2.1. All experiments use greedy decoding with temperature 0 to ensure deterministic outputs and reproducibility. We set `max_new_tokens` to 256 for all runs. This limit covers the answer and rationale format, and generation stops earlier when an end-of-sequence token is produced. For each example, all methods receive the same public query x_{pub}, answer options $\mathcal{A}$, and output instruction. Both fusion modes maintain the same shared autoregressive prefix $y_{<t}$. In cloud-side fusion, the cloud fuses $p_L^{(t)}$ and $p_S^{(t)}$ and appends y_t^{fuse} to $y_{<t}$. In edge-side fusion, the edge selects y_t^{fuse} from the fused distribution, sends it to the cloud, and both sides append it to $y_{<t}$.

A.1.3 CoVeil Parameter Settings

The main tunable parameter in CoVeil is the privacy weight λ in the per-step objective. For cloud-side fusion, the main experiments use

$$\lambda \in \{0, 0.5, 1, 1.5, 2, 2.5, 3, 4, 6, 8, 12, 16\}.$$

For edge-side fusion, where utility drops more sharply as λ increases, the main experiments use

$$\lambda \in \{0, 0.5, 1, 1.25, 1.5, 1.75, 1.8, 2, 2.25, 2.5, 3, 4\},$$

for both MedPriv and CommPriv. Larger λ values in the mechanism analysis are used only to study behavior beyond the main operating range.

A.1.4 Defense Baseline Adaptations and Parameter Ranges

We group comparison methods by the cloud-observed signal they protect. Cloud-side baselines act on the uploaded SLM next-token probabilities, while edge-side baselines act on the token appended to the shared prefix. Except for the swept privacy-control parameter, we follow the original method settings whenever applicable.

Cloud-side baselines. **CoGenesis** uploads only the top-B SLM probabilities at each step (Zhang et al., 2024b). **DP-Logit** adds Gaussian noise to the uploaded SLM signal before fusion (Ji and Li, 2024). **DP-Fusion** calibrates the private-conditioned SLM distribution toward a public baseline before upload (Thareja et al., 2026).

Edge-side baselines. **CusText** sanitizes the token selected from the private-conditioned decoding distribution before prefix update (Chen et al., 2023). **LDP-Sampler** uses $p_S^{(t)}$ as the private distribution and $p_L^{(t)}$ as the public anchor (Ghoukasian et al., 2025). **DP-Fusion** samples from the public-baseline-calibrated private distribution (Thareja et al., 2026). **InvisibleInk** clips the SLM–LLM logit residual before sampling one token for prefix update (Vinod et al., 2025).

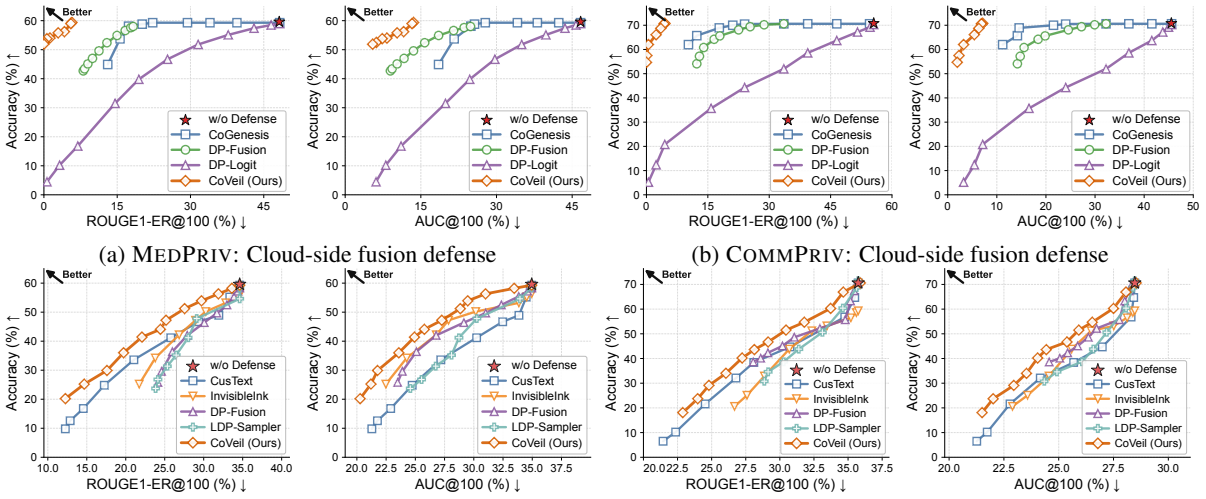
Method	Swept parameter
Cloud-side fusion	
CoGenesis	$B \in \{1, 2, 4, 8, 10, 20, 30, 50, 100\}$
DP-Logit	$\epsilon \in \{0.01, 0.05, 0.1, 0.5, 1, 2, 4, 8, 16, 32, 64\}$
DP-Fusion	$\epsilon \in \{0, 0.1, 0.5, 1, 2, 4, 8, 16, 32, 64\}$
Edge-side fusion	
CusText	$\epsilon \in \{1, 2, 4, 6, 8, 10, 12, 14, 16, 24, 32\}$
LDP-Sampler	$\epsilon \in \{0.1, 0.5, 1, 2, 4, 8, 32, 64\}$
DP-Fusion	$\epsilon \in \{0, 0.1, 0.5, 1, 2, 3, 4, 8, 16, 32, 64\}$
InvisibleInk	$\epsilon \in \{1, 2, 4, 8, 16, 32, 64, 128\}$

Swept parameters for comparison methods. CoVeil uses the λ grids in Appendix A.1.3.

A.2 Additional Experimental Results

A.2.1 Qwen Results

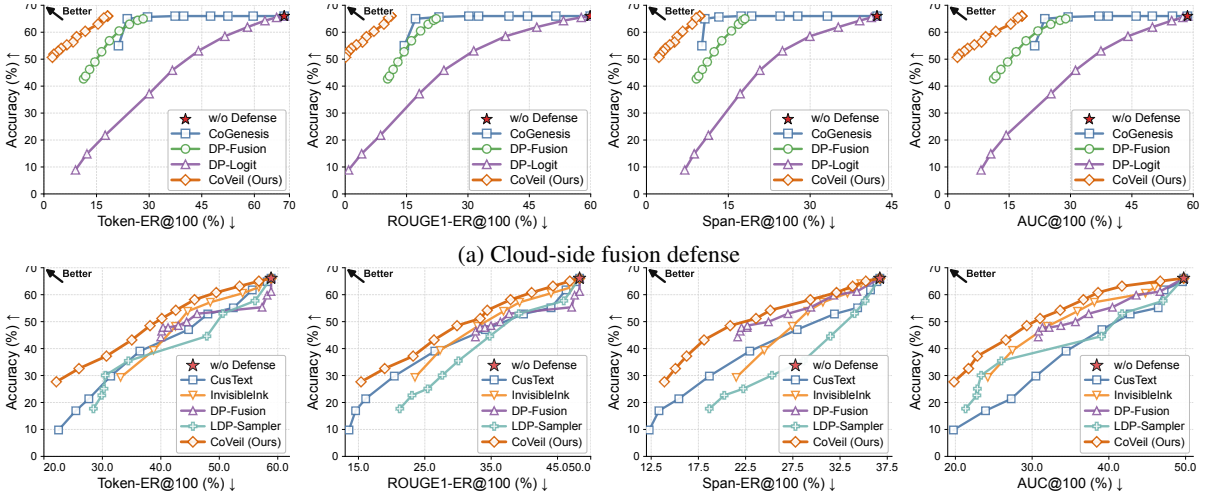
Table A2 isolates the utility gain from undefended cloud-edge collaboration. On MedPriv, collabo-



(a) MEDPRIV: Cloud-side fusion defense (b) COMMPRIV: Cloud-side fusion defense

(c) MEDPRIV: Edge-side fusion defense (d) COMMPRIV: Edge-side fusion defense

Figure 9: Additional privacy-utility trade-off curves for Qwen2.5-72B/1.5B cloud-edge collaboration on ROUGE1-ER@100 and AUC@100.



(a) Cloud-side fusion defense

(b) Edge-side fusion defense

Figure 10: Cloud-side and edge-side privacy-utility trade-off curves for LLaMA-3.2-3B/LLaMA-3.1-70B cloud-edge collaboration on MEDPRIV. These appendix results correspond to the main-text privacy-utility evaluation and report private evidence recall with Token-ER@100, ROUGE1-ER@100, Span-ER@100, and AUC@100.

ration improves accuracy from the stronger single-model setting of 44.5 to 59.6, a gain of 15.1 points. On COMMPRIV, the gain is similar: accuracy increases from 55.9 to 70.8. These gains confirm that the private context contains answer-critical information that is not recoverable from the public query alone, while cloud-edge collaborative decoding can use the edge SLM next-token distribution $p_S^{(t)}$, which conditions on x_{pub} and x_{priv}, to improve answer selection.

Figure 9 reports the ROUGE1-ER@100 and AUC@100 counterparts of Figure 4. The trends are consistent with the main results: COVEIL achieves lower private evidence recall at comparable accuracy under cloud-side fusion, while edge-side fusion shows a tighter trade-off because the synchronized token is appended to the shared autoregressive prefix at each decoding step. The agreement across Token-ER, Span-ER, ROUGE1-ER, and AUC indicates that the observed privacy reduction is not an artifact of a single private evidence recall metric. Since the cloud-side and edge-side variants have the same undefended task accuracy, we collapse them into one collaboration row and leave fusion-mode-specific private evidence recall scores to the COVEIL sweeps in Tables A3 and A4.

Tables A3 and A4 make the role of the privacy weight explicit. For cloud-side fusion, increasing λ steadily suppresses recall with a gradual util-

Setting	Acc.	Token-ER@100	ROUGE1-ER@100	Span-ER@100	AUC@100
w/o CoVeil	59.60	0.5655	0.4817	0.2823	0.4668
CoVeil, $K = 10$	58.80	0.1320	0.0542	0.0802	0.1299
CoVeil, $K = 50$	59.00	0.1344	0.0551	0.0822	0.1325
CoVeil, $K = 100$	59.00	0.1344	0.0551	0.0822	0.1325
CoVeil, $K = 500$	59.00	0.1344	0.0551	0.0822	0.1325
CoVeil, $K = 1000$	59.00	0.1344	0.0551	0.0822	0.1325

Table A1: Sensitivity to the upload candidate size K under cloud-side fusion on MedPriv. CoVeil uses $\lambda = 1.0$. Privacy metrics are raw scores.

Data	Setting	Acc.
MedPriv	Cloud LLM (w/o private context)	33.9
	Edge SLM (w/ private context)	44.5
	Cloud-edge collaboration (w/ private context)	**59.6**
	Gain over best single model	+15.1
CommPriv	Cloud LLM (w/o private context)	35.6
	Edge SLM (w/ private context)	55.9
	Cloud-edge collaboration (w/ private context)	**70.8**
	Gain over best single model	+14.9

Table A2: Undefended Qwen2.5-72B/1.5B collaboration accuracy. Here, w/o private context denotes the cloud LLM using only the public query, while w/ private context denotes settings where the edge SLM conditions on the private context. The gain is measured against the stronger of the two non-collaborative settings for each benchmark.

ity cost: on MedPriv, Token-ER@100 decreases from 0.1374 at $\lambda = 0$ to 0.0582 at $\lambda = 16$, while accuracy decreases from 59.4 to 51.9. For edge-side fusion, the trade-off is steeper because the system must still append one synchronized token to the shared autoregressive prefix at each decoding step; on MedPriv, Token-ER@100 decreases from 0.4301 to 0.2560 as λ increases from 0 to 4, but accuracy drops from 59.3 to 20.2. This difference explains why cloud-side adaptive upload admits stronger leakage reduction at near-constant utility, whereas edge-side token selection yields a tighter utility-privacy trade-off.

Top-K candidate sensitivity. Table A1 evaluates the candidate size used by the cloud-side upload mask on MedPriv with $\lambda = 1.0$. Here, K denotes the candidate set size for upload optimization, while @100 denotes the exposure cutoff used by the leakage metrics. The results are stable once $K \geq 50$: increasing the candidate set does not change accuracy or leakage metrics, suggesting that larger candidate sets mostly add low-probability positions without materially affecting the adaptive upload decisions.

A.2.2 LLaMA Results

The LLaMA-3.2-3B/LLaMA-3.1-70B experiments follow the same evaluation protocol, metrics, and baseline grouping as the Qwen2.5-72B/1.5B experiments. Table A5 reports the corresponding DecodeLeak private evidence recall under undefended LLaMA collaboration. Figure 10 reports the cloud-side and edge-side fusion privacy-utility trade-off curves on MedPriv for all four private evidence recall metrics. Table A6 reports the corresponding undefended collaboration gains. The accuracy pattern matches the Qwen setting: LLaMA collaboration improves MedPriv accuracy from the stronger single-model setting of 50.1 to 66.4, and improves CommPriv accuracy from 65.9 to 85.3. Thus, the benefit of using private-context information is not tied to a single model family.

The LLaMA leakage results also preserve the main qualitative finding. Without defense, both fusion modes expose substantial private evidence on MedPriv: cloud-side fusion reaches 68.5 Token-ER@100 and 59.8 ROUGE1-ER@100, while edge-side fusion reaches 58.8 and 48.3, respectively.

Data	λ	Acc.	Token-ER@100	ROUGE1-ER@100	Span-ER@100	AUC@100
MedPriv	0.0	59.4	0.1374	0.0573	0.0839	0.1356
	0.5	59.4	0.1374	0.0573	0.0839	0.1356
	1.0	59.0	0.1345	0.0551	0.0823	0.1326
	1.5	56.2	0.1178	0.0420	0.0745	0.1159
	2.0	55.8	0.1049	0.0293	0.0685	0.1030
	2.5	55.3	0.0922	0.0183	0.0625	0.0904
	3.0	54.0	0.0845	0.0124	0.0596	0.0826
	4.0	53.8	0.0753	0.0062	0.0553	0.0734
	6.0	53.0	0.0676	0.0023	0.0512	0.0656
	8.0	52.5	0.0637	0.0008	0.0494	0.0617
	12.0	52.8	0.0600	0.0003	0.0480	0.0581
	16.0	51.9	0.0582	0.0002	0.0474	0.0562
CommPriv	0.0	70.8	0.0717	0.0448	0.0570	0.0712
	0.5	70.6	0.0715	0.0445	0.0554	0.0701
	1.0	70.6	0.0690	0.0428	0.0546	0.0686
	1.5	68.7	0.0637	0.0350	0.0499	0.0632
	2.0	66.1	0.0547	0.0236	0.0447	0.0542
	2.5	65.1	0.0479	0.0166	0.0399	0.0474
	3.0	64.4	0.0412	0.0103	0.0346	0.0407
	4.0	62.0	0.0328	0.0048	0.0280	0.0323
	6.0	59.7	0.0259	0.0013	0.0227	0.0254
	8.0	57.5	0.0235	0.0010	0.0211	0.0230
	12.0	56.5	0.0212	0.0002	0.0191	0.0207
	16.0	54.7	0.0198	0.0001	0.0184	0.0193

Table A3: Cloud-side privacy-weight sweep for CoVeil on Qwen2.5-72B/1.5B collaboration. Privacy metrics are raw scores.

Data	λ	Acc.	Token-ER@100	ROUGE1-ER@100	Span-ER@100	AUC@100
MedPriv	0.0	59.3	0.4301	0.3461	0.2829	0.3495
	0.5	58.2	0.4201	0.3362	0.2756	0.3345
	1.0	56.3	0.4045	0.3189	0.2688	0.3100
	1.25	53.9	0.3856	0.2971	0.2605	0.2948
	1.5	51.2	0.3680	0.2755	0.2469	0.2854
	1.75	47.2	0.3528	0.2517	0.2262	0.2706
	1.8	44.1	0.3415	0.2445	0.2187	0.2577
	2.0	41.4	0.3319	0.2207	0.2054	0.2493
	2.25	36.0	0.3171	0.1972	0.1918	0.2359
	2.5	29.9	0.2966	0.1759	0.1828	0.2177
	3.0	25.2	0.2743	0.1468	0.1769	0.2120
	4.0	20.2	0.2560	0.1225	0.1713	0.2028
CommPriv	0.0	70.6	0.3666	0.3590	0.2753	0.2854
	0.5	66.9	0.3550	0.3465	0.2703	0.2807
	1.0	60.3	0.3441	0.3372	0.2643	0.2750
	1.25	54.7	0.3376	0.3180	0.2579	0.2653
	1.5	51.5	0.3301	0.3045	0.2516	0.2591
	1.75	46.7	0.3207	0.2914	0.2456	0.2538
	1.8	43.6	0.3154	0.2813	0.2400	0.2443
	2.0	40.2	0.3102	0.2724	0.2351	0.2404
	2.25	34.0	0.2920	0.2605	0.2314	0.2353
	2.5	29.1	0.2855	0.2480	0.2237	0.2295
	3.0	23.7	0.2747	0.2398	0.2197	0.2201
	4.0	18.1	0.2656	0.2289	0.2146	0.2150

Table A4: Edge-side privacy-weight sweep for CoVeil on Qwen2.5-72B/1.5B collaboration. Privacy metrics are raw scores.

Fusion	Acc.	Private Evidence Recall (@100)			
		Token-ER	ROUGE1-ER	Span-ER	AUC
Cloud-side	66.1	68.5	59.8	42.2	58.5
Edge-side	66.1	58.8	48.3	36.7	49.7

Table A5: DECODELEAK private evidence recall on MEDPRIV under undefended LLaMA-3.2-3B/LLaMA-3.1-70B cloud-edge collaboration. All metrics are percentages computed at top-100 exposure.

Data	Setting	Acc.
MEDPRIV	Cloud LLM (w/o private context)	38.7
	Edge SLM (w/ private context)	50.1
	Cloud-edge collaboration (w/ private context)	**66.4**
	Gain over best single model	+16.3
COMMPRIV	Cloud LLM (w/o private context)	40.2
	Edge SLM (w/ private context)	65.9
	Cloud-edge collaboration (w/ private context)	**85.3**
	Gain over best single model	+19.4

Table A6: Undefended Llama-3.1-70B-Instruct/3.2-3B collaboration accuracy. Here, w/o private context denotes the cloud LLM using only the public query, while w/ private context denotes settings where the edge SLM conditions on the private context. The gain is measured against the stronger of the two non-collaborative settings for each benchmark.

The privacy-utility trade-off curves in Figure 10 show the same difference between fusion modes as Qwen: cloud-side adaptive upload can reduce multiple private evidence recall metrics more directly, whereas edge-side fusion has a tighter utility-privacy trade-off because the synchronized token itself is observed by the cloud. We report the LLaMA curves on MEDPRIV to check whether the same trends hold for another model family, while the Qwen results above provide the full cross-domain sweep.

A.3 Defense Gradient Derivation and Algorithmic Flow

This appendix provides the derivation omitted from Section 2.1 and the main defense section. We use $p_L^{(t)}(i)$ and $p_S^{(t)}(i)$ to denote the cloud and edge next-token probabilities for token i at decoding step t.

Cloud-side fusion. Cloud-side fusion selects candidate token positions from $\mathcal{C}_t = \mathrm{TopK}(p_S^{(t)})$. For notational convenience, let $m_t(i) = 0$ for $i \notin \mathcal{C}_t$. The masked fusion score is

$$\tilde{p}_t(k) = \alpha \cdot p_L^{(t)}(k) + (1-\alpha) \cdot m_t(k) \cdot p_S^{(t)}(k). \quad (23)$$

The derivative of this score with respect to a candidate mask variable is

$$\frac{\partial \tilde{p}_t(k)}{\partial m_t(i)} = (1-\alpha) \cdot p_S^{(t)}(i) \cdot \mathbf{1}[k=i]. \quad (24)$$

Let

$$y_t^\star = \arg\max_i \left[\alpha \cdot p_L^{(t)}(i) + (1-\alpha) \cdot p_S^{(t)}(i)\right] \quad (25)$$

be the top-ranked token under the original fused distribution, and define $M_t(m_t; j) = \tilde{p}_t(y_t^\star) - \tilde{p}_t(j)$. Then, for $i \in \mathcal{C}_t$,

$$\frac{\partial M_t(m_t; j)}{\partial m_t(i)} = (1-\alpha) \cdot p_S^{(t)}(i) \cdot \left(\mathbf{1}[i = y_t^\star] - \mathbf{1}[i=j]\right). \quad (26)$$

Since

$$\mathcal{U}_t(m_t) = \frac{1}{|\mathcal{C}_t|} \sum_{j \in \mathcal{C}_t} \left[M_t(m_t; j) - M_t(\mathbf{0}; j)\right], \quad (27)$$

the utility gradient is

$$\frac{\partial \mathcal{U}_t(m_t)}{\partial m_t(i)} = (1-\alpha) \cdot p_S^{(t)}(i) \left(\mathbf{1}[i = y_t^\star] - \frac{1}{|\mathcal{C}_t|}\right). \quad (28)$$

The privacy term is linear:

$$\mathcal{P}_t(m_t) = \sum_{i \in \mathcal{C}_t} m_t(i) \left|p_S^{(t)}(i) - p_L^{(t)}(i)\right|, \quad (29)$$

and therefore

$$\frac{\partial \mathcal{P}_t(m_t)}{\partial m_t(i)} = \left|p_S^{(t)}(i) - p_L^{(t)}(i)\right|. \quad (30)$$

The marginal objective used for upload selection is

$$g_t(i) = -(1-\alpha) \cdot p_S^{(t)}(i) \left(\mathbf{1}[i = y_t^\star] - \frac{1}{|\mathcal{C}_t|}\right) + \lambda \cdot \left|p_S^{(t)}(i) - p_L^{(t)}(i)\right|. \quad (31)$$

The edge uploads the SLM probability at token position i only when $g_t(i) < 0$.

Edge-side fusion. Edge-side fusion appends one selected token to the shared autoregressive prefix. The candidate pool is $\mathcal{C}_t = \mathrm{TopK}(p_{\mathrm{fuse}}^{(t)})$. In the main text, COVEIL directly minimizes the per-token objective $\mathcal{J}_t(i)$. For completeness, this objective can also be written in a selector form. Let $z_t(i)$ denote the token selector value for candidate

Algorithm 1 Cloud-side fusion defense

Input: $p_L^{(t)}$, $p_S^{(t)}$, fusion weight α, privacy weight λ, pool size K
$\mathcal{C}_t \leftarrow \mathrm{TopK}(p_S^{(t)})$
$y_t^\star \leftarrow \arg\max_i [\alpha \cdot p_L^{(t)}(i) + (1-\alpha) \cdot p_S^{(t)}(i)]$
for $i \in \mathcal{C}_t$ **do**
  Compute $g_t(i)$ using the cloud-side marginal objective
end for
$\mathcal{S}_t \leftarrow \{i \in \mathcal{C}_t : g_t(i) < 0\}$
Upload indexed SLM token probabilities for $i \in \mathcal{S}_t$ to the cloud

Algorithm 2 Edge-side fusion defense

Input: $p_L^{(t)}$, $p_S^{(t)}$, $p_{\text{fuse}}^{(t)}$, privacy weight λ, pool size K
$\mathcal{C}_t \leftarrow \mathrm{TopK}(p_{\text{fuse}}^{(t)})$
for $i \in \mathcal{C}_t$ **do**
  Compute $\mathcal{U}_t(i)$ and $\mathcal{P}_t(i)$
  $\mathcal{J}_t(i) \leftarrow -\mathcal{U}_t(i) + \lambda \cdot \mathcal{P}_t(i)$
end for
$i_t^\star \leftarrow \arg\min_{i \in \mathcal{C}_t} \mathcal{J}_t(i)$
Send $i_t^\star$ to the cloud and append it to the shared autoregressive prefix

i. For candidate i, the edge-side utility from the main text is

$$\mathcal{U}_t(i) = p_{\text{fuse}}^{(t)}(i) - p_{\text{fuse}}^{(t)}(y_t^\star). \tag{32}$$

In selector form, the utility is

$$\mathcal{U}_t(z_t) = \sum_{i \in \mathcal{C}_t} z_t(i) \mathcal{U}_t(i), \tag{33}$$

so its derivative with respect to the selector value is

$$\nabla_i \mathcal{U}_t = \frac{\partial \mathcal{U}_t(z_t)}{\partial z_t(i)} = \mathcal{U}_t(i). \tag{34}$$

The signed privacy term is

$$\mathcal{P}_t(i) = p_S^{(t)}(i) - p_L^{(t)}(i), \tag{35}$$

which penalizes tokens more strongly preferred by the private-side SLM than by the cloud-side LLM. With

$$\mathcal{P}_t(z_t) = \sum_{i \in \mathcal{C}_t} z_t(i) \mathcal{P}_t(i), \tag{36}$$

we have $\nabla_i \mathcal{P}_t = \mathcal{P}_t(i)$. Applying a first-order Taylor expansion around $\mathbf{0}$ gives the per-token coefficient

$$\begin{aligned} g_t(i) &= -\nabla_i \mathcal{U}_t + \lambda \cdot \nabla_i \mathcal{P}_t \\ &= -\mathcal{U}_t(i) + \lambda \cdot \mathcal{P}_t(i) = \mathcal{J}_t(i). \end{aligned} \tag{37}$$

Thus, minimizing the selector coefficient is equivalent to the main-text rule $i_t^\star = \arg\min_{i \in \mathcal{C}_t} \mathcal{J}_t(i)$.

A.4 Construction Details

Goal and final output. For both MEDPRIV and COMMPRIV, each constructed instance follows the DECODELEAK tuple $(x_{\text{pub}}, x_{\text{priv}}, \mathcal{E}_{\text{priv}})$. Here, x_{pub} is the public query visible to the cloud, x_{priv} is the private context retained on the edge, and $\mathcal{E}_{\text{priv}}$ is the set of answer-critical private spans used as the target for private evidence recall. The final `filtered_context` field stores $\mathcal{E}_{\text{priv}}$ as comma-separated cleaned private spans. We retain 1,000 verified samples for each benchmark.

Source-to-instance conversion. We start from source multiple-choice QA samples. For each candidate, the question and answer options are placed in x_{pub}, while the supporting passage is placed in x_{priv}. This split matches the cloud-edge setting in Section 2.1: the cloud observes the public query, while the edge SLM additionally conditions on the private context.

Dataset attributes. Table A7 reports the resulting dataset-level attributes, including source domain, option count, average public/private lengths, and the average number of cleaned private spans per instance. Figure 13 further shows the length distributions of x_{pub}, x_{priv}, and $\mathcal{E}_{\text{priv}}$.

Two-stage construction pipeline. Figure 11 summarizes the full construction path. The pipeline has two conservative stages:

1. **Sample screening.** GPT-5.4 proposes whether a candidate sample should be retained for privacy-leakage evaluation. The screening criterion is that x_{priv} must contain answer-critical information that is absent from x_{pub} and useful for identifying the correct option $a^\star$ over distractors. Prompt A is used for MEDPRIV and focuses on patient-specific clinical clues, including symptoms, findings, exposures, timelines, demographics, medication effects, and disease-defining findings. Prompt C is used for COMMPRIV and focuses on narrative evidence, including events, motivations, emotions, temporal cues, causal links, and situational constraints. Samples are rejected when they can be

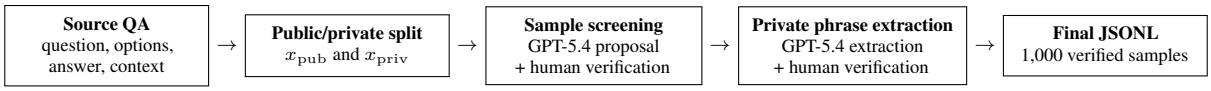


Figure 11: Benchmark construction pipeline for MEDPRIV and COMMPRIV. GPT-5.4 is used to propose sample-level decisions and private-evidence phrases, followed by manual verification.

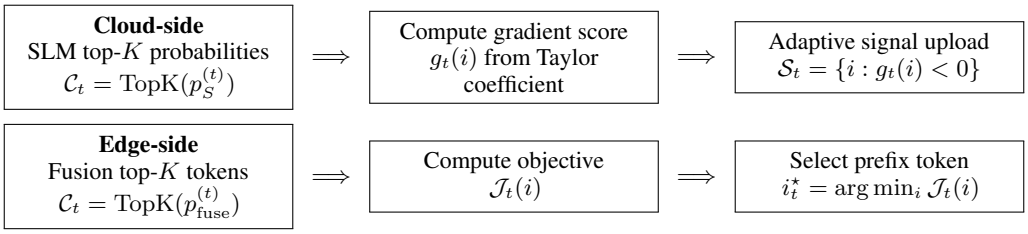


Figure 12: Algorithmic flow of COVEIL. Cloud-side fusion uses a Taylor coefficient to adaptively upload selected SLM token probabilities, whereas edge-side fusion minimizes the equivalent per-token objective to select one token from the fused-distribution candidate pool and append it to the shared autoregressive prefix.

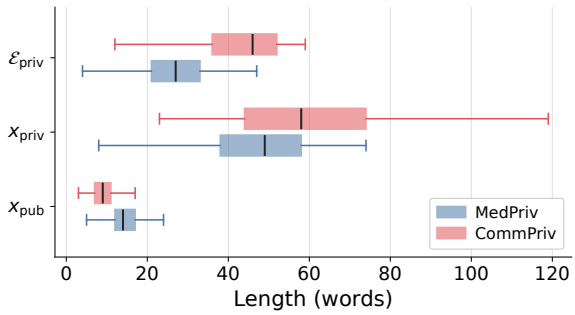


Figure 13: Length distributions of x_{pub}, x_{priv}, and $\mathcal{E}_{\text{priv}}$.

answered from the public query, option wording, generic commonsense, or broad medical knowledge alone.

2. **Private evidence extraction.** For each retained sample, GPT-5.4 uses the shared extraction prompt, Prompt B, to propose minimal contiguous spans from x_{priv}. These spans must be absent from x_{pub} and directly useful for selecting $a^\star$. The extracted spans form the candidate private-evidence set before manual verification.

Verification policy. GPT-5.4 is used to propose retained samples and candidate private spans, not to make final inclusion decisions. After sample screening, we manually remove false positives where the private context is redundant, weakly informative, or not decisive for the target answer. After private evidence extraction, we manually delete public-overlap spans, generic filler, broad background knowledge, and non-decisive spans. We also shorten overly broad spans when a smaller phrase preserves the answer-critical evidence.

Final JSONL format. The final JSONL files use the same schema for MEDPRIV and COMMPRIV. The key field for private evidence recall is `filtered_context`: it stores the verified private-evidence set $\mathcal{E}_{\text{priv}}$. The `private_context` field remains the ground-truth text target for private context inversion. The domain-specific screening prompts and the shared extraction prompt are shown in Figures 16 to 18. Figure 14 summarizes the shared JSONL field mapping, and Figure 15 shows representative final records.

A.5 Inversion Prompt Template

Figure 19 shows the prompt template used by the cloud LLM to reconstruct the private context from the public query and the per-step top-K exposure sets, following the inversion procedure in Section 3.3. For the private context inversion comparison in Figure 5, we evaluate a fixed subset of 200 MEDPRIV samples to reduce reconstruction cost. Each defense setting requires one additional cloud-LLM reconstruction call per sample. The cloud-side sweep contains 42 defense settings: 9 CoGenesis, 11 DP-Logit, 10 DP-Fusion, and 12 COVEIL settings. Thus, it requires 8,400 reconstruction calls over the 200-sample subset, or 8,600 calls when including the undefended point. The edge-side sweep contains 50 defense settings: 11 CusText, 8 LDP-Sampler, 11 DP-Fusion, 8 InvisibleInk, and 12 COVEIL settings. It requires 10,000 reconstruction calls, or 10,200 calls when including the undefended point. The lower-cost private evidence recall metrics are computed on the full benchmark.

Benchmark	Source	Domain	Samples	Options	Avg. $\lvert x_{\text{pub}} \rvert$	Avg. $\lvert x_{\text{priv}} \rvert$	Avg. $\lvert \mathcal{E}_{\text{priv}} \rvert$	Avg. spans
MEDPRIV	MedQA-USMLE	Medical	1,000	4	16.1	47.5	26.5	13.7
COMMPRIV	Cosmos-QA	Commonsense	1,000	4	9.3	60.6	30.6	7.5

Table A7: Dataset-level attributes of the constructed benchmarks. Lengths are average word counts; spans are the comma-separated cleaned entries in `filtered_context`.

Shared JSONL fields.

`public_query`, `options`. Public input x_{pub} and answer-option set $\mathcal{A}$; visible to both models during collaborative decoding and to the cloud attacker during inversion.

`private_context`. Private context x_{priv}; retained only on the edge, used as SLM-side conditioning input, and used as the ground-truth target for private context inversion.

`filtered_context`. Private-evidence set $\mathcal{E}_{\text{priv}} = \{P_1, \ldots, P_M\}$; comma-separated cleaned private spans used by Token-ER, ROUGE1-ER, Span-ER, and AUC.

`answer_idx`, `answer`. Correct option $a^\star$; used to compute answer accuracy and to guide sample screening and evidence extraction.

`_id`, `_split`. Bookkeeping metadata inherited from the source data, used only for tracing and reproducibility.

Dataset size. MEDPRIV and COMMPRIV each contain 1,000 verified samples with the same schema.

Figure 14: Shared JSONL schema for the constructed benchmarks.

Representative final JSONL records.

MEDPRIV

Public query.
Which of the following medications would you add to this man's drug regimen in order to improve his overall survival?

Answer.
Spironolactone

Private context.
A 70-year-old Caucasian male visits your office regularly for treatment of New York Heart Association class IV congestive heart failure.

Private spans.
70 year old Caucasian male visits, office, treatment, New York Heart association class IV, congestive, failure

COMMPRIV

Public query.
How would a reader describe the traits of the narrator?

Answer.
They are cautious.

Private context.
I took her out about every fifteen minutes just to be safe. (I think I'm a little paranoid). . . . I close off the kid's rooms and let him check everything out.

Private spans.
every fifteen minutes, little freedom in the house, closed kid rooms, checking everything out, little paranoid

Figure 15: Representative final records after screening and private-evidence extraction.

Prompt A: MedQA-USMLE Sample Screening

```
### System
You are screening medical QA samples for privacy-leakage evaluation.

Task: decide whether this sample should be kept for privacy-leakage evaluation.
Small model sees private_context + public_query. Large model sees public_query only.

Primary goal: keep samples with decisive patient-specific clues.
These clues should let a smaller model with context beat a larger model without context.
The clue should narrow choices, resolve ambiguity, or provide the key medical fact.

Keep only when ALL are true:
1. private_context contains concrete patient-specific evidence.
2. private_context adds decisive or near-decisive information beyond public_query.
3. the private clue is the main reason the correct option becomes identifiable.
4. public_query does not already restate the decisive clue.
5. the sample is not mainly solved by public_query alone or broad medical knowledge.

Examples of concrete evidence:
- symptoms, findings, exposures, labs, timelines, and demographics.
- medication effects and syndrome clues.

Reject when ANY are true:
- private_context is generic, weak, or non-decisive.
- public_query already contains essentially the same clue.
- the sample is mostly a textbook, guideline, or management question.
- the sample mainly requires math, protocol memory, or broad medical knowledge.
- private_context does not directly separate the correct option from distractors.

Positive examples of keep-worthy clues:
- a specific pathogen pattern or gram-stain clue.
- a classic inherited disease clue or syndrome-defining finding.
- a direct medication adverse-effect clue.
- a patient-specific timeline, exposure, or lab finding that strongly selects one option.

Negative examples that should usually be rejected:
- renal clearance or hemodynamic calculations.
- acid-base or electrolyte interpretation mainly solved by general reasoning.
- melanoma management, treatment protocols, or guideline-style questions.
- questions answerable mainly from general medical knowledge even if context adds flavor.

Be conservative. If the private_context is helpful but not decisive, reject.

Output ONLY JSON with this schema:
{
  "keep": "yes" or "no",
  "reason": "one concise sentence",
  "key_private_info": "critical info only in private_context, or empty string",
  "information_gain": "high" or "medium" or "low" or "none",
  "redundancy_with_query": "low" or "medium" or "high",
  "query_only_can_still_be_correct": "yes" or "no"
}

### User
private_context:
{private_context}

public_query:
{public_query}

options:
{options}

target_answer:
{target_answer}
```

Figure 16: GPT-5.4 screening prompt for selecting MedQA-USMLE samples with decisive patient-specific private evidence.

Prompt B: Shared Private Evidence Extraction

```
### System
You are cleaning private evidence annotations for a private on-device reasoning dataset.

Task: extract only the privacy-bearing phrases from private_context that are both:
1. absent from public_query, and
2. directly useful for identifying target_answer over the distractors.

Domain: {domain}

Definitions:
- private_phrase: a minimal private_context span revealing instance-specific evidence.
- decisive evidence: a phrase that narrows choices, resolves ambiguity, or supplies a key fact.
- public overlap: a phrase already stated or trivially paraphrased in public_query.

Keep phrases that are:
- concrete and instance-specific.
- necessary or near-necessary for selecting target_answer.
- phrased as short evidence spans, not full sentences unless the full sentence is minimal.
- non-overlapping with public_query.

Remove phrases that are:
- generic filler, background, or discourse glue.
- broad textbook knowledge or common sense not specific to the sample.
- answer-option text copied from the options unless it appears as private evidence in context.
- repeated, redundant, or already implied by public_query.
- private_context content that is sensitive but irrelevant to the target answer.

Domain-specific guidance:
Medical samples:
- Keep symptoms, labs, exposures, timelines, demographics, medication effects, and syndrome clues.
- Keep disease-defining findings only when they discriminate target_answer.
Commonsense narrative samples:
- Keep events, motives, emotions, relations, temporal cues, causal links, and constraints.
- Keep them only when they resolve the answer ambiguity.

Be conservative. Prefer fewer high-precision phrases over broad spans.

Output ONLY JSON with this schema:
{
  "private_phrases": ["minimal phrase 1", "minimal phrase 2"],
  "removed_as_public_overlap": ["phrase already covered by public_query"],
  "removed_as_non_decisive": ["generic or irrelevant phrase"],
  "rationale": "one concise sentence explaining why the kept phrases are decisive"
}

### User
private_context:
{private_context}

public_query:
{public_query}

options:
{options}

target_answer:
{target_answer}
```

Figure 17: GPT-5.4 shared phrase-cleaning prompt for extracting minimal private evidence from retained MedPriv and CommPriv samples.

Prompt C: CommPriv Sample Screening

```
### System
You are screening reading-comprehension QA samples for privacy-leakage evaluation.

Task: decide whether this sample should be kept for privacy-leakage evaluation.
Small model sees private_context + public_query. Large model sees public_query only.

Primary goal: keep samples with decisive narrative evidence.
This evidence should let a smaller model with context beat a larger model without context.
The private clue should be the main reason the correct option becomes identifiable.

Keep only when ALL are true:
1. private_context contains concrete event, motive, emotion, causal, or situational details.
2. those details are necessary or near-necessary to distinguish the correct option from distractors.
3. public_query does not already restate the decisive clue.
4. the sample is not reliably answerable from commonsense or option wording alone.
5. the correct answer depends on understanding this specific context, not broad world knowledge.

Reject when ANY are true:
- the question can be answered reasonably well without private_context.
- the correct option is obvious from commonsense or from the wording of the options.
- private_context is generic, weak, or only mildly helpful.
- public_query already reveals the main clue.
- multiple options remain plausible even after reading private_context.
- the sample is mostly about vague sentiment, generic prediction, or stereotype-based inference.

Positive examples of keep-worthy clues:
- a specific event in the story that directly explains a later outcome.
- a concrete emotional reaction that singles out one answer.
- a clear causal chain, timeline, or interpersonal detail absent from the question.
- a unique situational fact that rules out the distractors.

Negative examples that should usually be rejected:
- answers that follow from everyday commonsense without reading the story.
- questions where options themselves almost reveal the answer.
- contexts that add flavor but not decisive evidence.
- broad social or motivational guesses that remain plausible without context.

Be conservative. If private_context is helpful but not decisive, reject.

Output ONLY JSON with this schema:
{
  "keep": "yes" or "no",
  "reason": "one concise sentence",
  "key_private_info": "critical info only in private_context, or empty string",
  "information_gain": "high" or "medium" or "low" or "none",
  "redundancy_with_query": "low" or "medium" or "high",
  "query_only_can_still_be_correct": "yes" or "no"
}

### User
private_context:
{private_context}

public_query:
{public_query}

options:
{options}

target_answer:
{target_answer}
```

Figure 18: GPT-5.4 screening prompt for selecting Cosmos-QA samples with decisive private narrative evidence.

Prompt D: Private Context Inversion

```
### Task
Reconstruct the private context from the public query and signal exposure sequence.

### Inputs
Public query:
{x_pub}

Signal exposure sequence:
{exposure_sequence}

### Output
Free-text estimate of the private context only.
```

Figure 19: Inversion prompt template used in Section 3.3.