PRISM: Privacy-Aware Routing for Adaptive Cloud–Edge LLM Inference via Semantic Sketch Collaboration

Junfei Zhan[1*] **Haoxun Shen**[1*] **Zheng Lin**[2] **Tengjiao He**[3†]

[1]Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA
[2]Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong SAR, China
[3]College of Information Science and Technology, Jinan University, Guangzhou, China

Abstract

Large Language Models (LLMs) demonstrate impressive capabilities in natural language understanding and generation, but incur high communication overhead and privacy risks in cloud deployments, while facing compute and memory constraints when confined to edge devices. Cloud–edge inference has emerged as a promising paradigm for improving privacy in LLM services by retaining sensitive computations on local devices. However, existing cloud–edge inference approaches apply uniform privacy protection without considering input sensitivity, resulting in unnecessary perturbation and degraded utility even for non-sensitive tokens. To address this limitation, we propose Privacy-aware Routing for Inference with Semantic Modulation (PRISM), a context-aware framework that dynamically balances privacy and inference quality. PRISM executes in four stages: (1) the edge device profiles entity-level sensitivity; (2) a soft gating module, also on the edge, selects an execution mode -cloud, edge, or collaboration; (3) for collaborative paths, the edge applies adaptive two-layer local differential privacy based on entity risks; and (4) the cloud LLM generates a semantic sketch from the perturbed prompt, which is then refined by the edge-side small language model (SLM) using local context. Our results show that PRISM consistently achieves superior privacy-utility trade-offs in various scenarios, reducing energy consumption and latency to 40–50% of baseline methods such as Uniform and Selective LDP, while maintaining high output quality under strong privacy constraints. These findings are validated through comprehensive evaluations involving realistic prompts, actual energy measurements, and heterogeneous cloud–edge model deployments.

Code — `https://github.com/Junfei-Z/PRISM`

Introduction

Large Language Models (LLMs) such as GPT-4o (Hurst et al. 2024), LLaMA 3 (Grattafiori et al. 2024), and Qwen 3 (Yang et al. 2025) have achieved remarkable progress in natural language understanding and generation (Cai et al. 2025a) . Their capabilities extend to diverse applications including code synthesis (Chen et al. 2021), multimodal reasoning (Alayrac et al. 2022), mathematical problem solving (Drori et al. 2022), and biomedical variant classification (Kweon et al. 2024). To run LLM-based services at scale, operators require large GPU clusters to support high-throughput inference workloads (Miao et al. 2025; Cai et al. 2025b). To this end, LLMs are typically deployed in the cloud, where they receive users' full prompts and generate responses in real time. This cloud-based deployment model is driven by the massive size and computation demands of state-of-the-art LLMs, which exceed the resource limits of local devices.

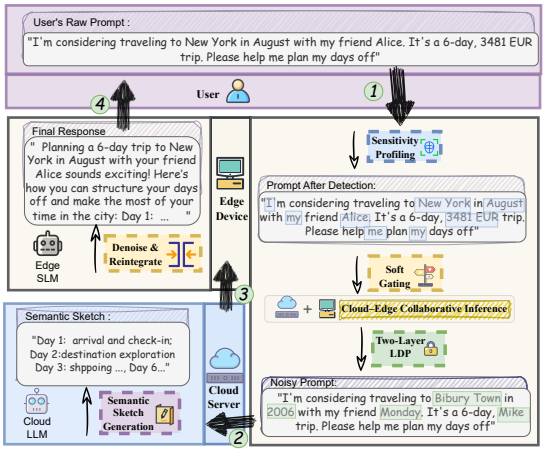


Figure 1: PRISM workflow example illustrating privacy-preserving prompt processing and transformation stages.

Cloud-centric LLM deployment provides global accessibility and facilitates centralized model updates (Li et al. 2024). However, it also introduces significant communication overhead and raises privacy risks due to the need to transmit full user prompts over the network (Lin et al. 2023). These challenges are especially pronounced in privacy-critical domains such as medical, finance, and personalized services, where user prompts often contain sensitive personal information. Unlike conventional software input, the prompts in these sensitive domains submitted to LLM often contain a rich semantic context that implicitly reveals the personal information, intentions, and preferences of users

*These authors contributed equally.
†Corresponding author.

(Staab et al. 2024). Consider the example in the medical domain: *"I tested positive for HIV last week and have been experiencing fever and diarrhea. Should I be concerned about secondary infections?"* (Zeng et al. 2025). This prompt includes protected health information (PHI) , if intercepted or mishandled, leading to severe privacy violations. Further, not all prompts carry the same privacy sensitivity: simple queries such as *"What is the capital of France?"* pose minimal risk and could be processed safely in the cloud (Acharya et al. 2020). These limitations highlight the need for new inference paradigms that preserve privacy without sacrificing efficiency, especially in semantically complex and sensitive domains.

To mitigate the privacy risks inherent in cloud-centric inference, recent cloud–edge collaborative frameworks attempt to retain sensitive information on the edge devices (Jin and Wu 2024; Zhu and Yang 2025; Zhan et al. 2025). A common design pattern is to perform inference locally on a Small Language Model (SLM) for privacy-critical prompts, while offloading non-sensitive queries to the cloud-hosted LLM based on binary routing mechanisms (Li et al. 2025). Alternatively, some systems adopt an encryption-inspired strategy: they apply local differential privacy (LDP) noise to the user's full prompt before transmitting it to the cloud, and then rely on the edge to reconstruct or refine the cloud's response (Lin et al. 2025; Mai et al. 2024). Although these approaches offer basic privacy protection, they remain fundamentally coarse-grained. First, binary routing decisions, based on simple thresholding of risk scores, can misclassify instructions, either overburdening the edge device or compromising privacy (Qu et al. 2025). Second, prompt-level LDP schemes, such as Split-and-Denoise (Mai et al. 2024) and DP-Forward (Du et al. 2023) uniformly perturb all tokens or embedding dimensions, regardless of their actual sensitivity. This uniform treatment leads to unnecessary utility degradation, especially for benign queries. Furthermore, when noise is applied indiscriminately, the cloud model receives semantically valid but semantically distorted commands, often generating vague, generic, or evasive responses (for example, 'I cannot provide information about [`MASKED_ENTITY`']. In short, existing architectures lack the ability to tailor privacy mechanisms to the specific semantics of each prompt. This inefficiency motivates a more adaptive approach, one that dynamically selects privacy pathways based on contextual cues within the prompt itself.

To this end, we introduce Privacy-aware Routing for Inference with Semantic Modulation (PRISM), a cloud–edge collaborative framework that adaptively selects inference pathways based on prompt semantics and contextual privacy risk, as illustrated in Figure 1. Applying the same protection strategy to all inputs, PRISM employs a soft gating mechanism on the edge device to assess each user prompt and route it to one of three execution modes: (1) direct cloud inference for low-risk prompts; (2) sketch-based collaboration for moderately sensitive prompts; and (3) fully local generation for high-risk, privacy-critical queries. The mode decision is made by a logit-based soft classifier that combines named entity recognition (NER), contextual cues (e.g. first-person references, entity types) and a statistical risk score. After the gating module, the system follows one of three execution paths: direct cloud inference and edge-only generation proceed immediately using the corresponding model, while cloud–edge collaborative mode triggers additional privacy-preserving procedures. Specifically, prompts in this mode are first obfuscated via a two-layer local differential privacy mechanism, then processed by the cloud to generate a semantic sketch, which is subsequently refined on the edge device to reconstruct a coherent and privacy-preserving response.

To this end, our work makes the following key contributions:

- We propose PRISM, a novel privacy-aware routing framework for adaptive cloud–edge LLM inference. PRISM combines a lightweight, context-aware gating mechanism with a three-mode execution pipeline (cloud, edge, and cloud–edge collaboration). In the collaborative mode, PRISM introduces a semantic modulation strategy that integrates adaptive two-layer local differential privacy with cloud-side sketch generation and edge-side refinement, enabling fine-grained, entity-level privacy control.
- To support this framework, we construct a synthetic and context-rich instruction dataset covering four domains: medical, tourism, banking, and general knowledge. This design reflects diverse real-world use cases and allows evaluation under varying privacy demands.
- We implement PRISM on a real-world cloud–edge platform and evaluate its performance under varying privacy budgets, systematically testing multiple combinations of edge-side SLMs and cloud-hosted LLMs. Our results demonstrate that PRISM flexibly adapts to heterogeneous deployments, consistently achieving better inference quality, lower latency, and reduced edge-side energy consumption compared to uniform protection baselines.

Framework Design

This section details the design of PRISM, our privacy-aware cloud–edge inference framework. The system consists of three components: a user, a local edge device, and a remote cloud server. The edge device hosts an SLM, while the cloud server provides access to an LLM. Inference begins when a user sends a prompt to the edge device, where two modules, Sensitivity Profiling and Soft Gating, analyze privacy risk and select an execution path: cloud, edge, or collaboration. For collaborative cases, PRISM applies Adaptive Two-Layer LDP to perturb sensitive entities and uses Cloud-Edge Semantic Sketch Collaboration to generate an abstract response on the cloud, which is then refined locally and returned to the user.

Sensitivity Profiling for Context-Aware Routing

We design a lightweight edge-side module to assess the privacy sensitivity of user prompts before routing. Given a prompt $P = \{x_1, x_2, \ldots, x_n\}$ consisting of n tokens, the module first extracts a set of m named entities $E = \{e_1, e_2, \ldots, e_m\}$ using a named entity recognition (NER)

engine. It then produces two outputs: (1) a scalar risk score $R(P)$ reflecting the overall privacy sensitivity of the prompt, and (2) a binary mask $\mathbf{d} \in \{0,1\}^m$ indicating which entities require protection.

We first apply a fast named entity recognition (NER) engine (e.g., Presidio Analyzer) to extract entities $E = \{e_1, e_2, \ldots, e_m\}$, where each entity e_i has an associated category label $c_i \in \mathcal{C}$, and we assign a predefined sensitivity weight $w_{c_i} \in [0,1]$ based on its category. These weights reflect the relevance for privacy of each category of entity (for example $w_{\texttt{PERSON}} > w_{\texttt{NATIONALITY}}$).

We define a risk score for the prompt:

$$R(P) = \sum_{i=1}^{m} w_{c_i} \cdot \mathbb{I}(e_i), \tag{1}$$

where $\mathbb{I}(e_i)$ indicates whether the entity is present in the current input.

To incorporate contextual signals, we define a binary indicator Δ that activates when any private linguistic cue is detected:

$$\Delta = \max_{x_j \in P} \mathbb{I}(x_j \in \mathcal{F}), \tag{2}$$

where the private context set $\mathcal{F}$ includes both first-person pronouns and detected entities of type PERSON. Here, $\mathbb{I}[\cdot]$ denotes the indicator function, and the condition evaluates whether any token in the prompt P overlaps with a predefined set of first-person pronouns or previously detected person-type entities.

Each entity e_i is conservatively flagged for protection if $\Delta > 0$, producing the binary mask:

$$d_i = \begin{cases} 1, & \text{if } \Delta > 0, \\ 0, & \text{otherwise.} \end{cases}, \forall i \in \{1, 2, ..., m\}. \tag{3}$$

This module captures both statistical and contextual signals cues on the edge devices. For instance, in the prompts (1)*"I plan to travel solo to Tokyo for three days"* and (2)*"Which country is Tokyo located in?"*, the entity *Tokyo* appears in both but only the former implies private user intent due to the pronoun *I*. These signals are forwarded to the later routing controller for decision-making.

Soft Gating with Entropy-Regularized Routing

To adaptively balance privacy, utility, and energy cost, we propose a soft gating mechanism that maps sensitivity indicators to a probability distribution over three routing paths. Rather than committing to a hard decision boundary, this approach enables nuanced, context-aware execution by producing soft routing scores. The gating module receives a feature vector $\mathbf{z} \in \mathbb{R}^{1+m}$ derived from the sensitivity profiling subsection, including the risk score $R(P)$ and the sensitivity mask $\mathbf{d}$. These features are passed into a light-weight linear transformation $f_\theta(\cdot)$ followed by softmax normalization:

$$\boldsymbol{\pi} = \text{softmax}(f_\theta(\mathbf{z})) \in \mathbb{R}^3. \tag{4}$$

This yields a routing distribution $\boldsymbol{\pi} = (\pi_{\text{cloud}}, \pi_{\text{collab}}, \pi_{\text{local}})$ over the three execution strategies: direct cloud inference, cloud–edge collaboration, and complete edge-side generation. To encourage confident routing while preserving flexibility, we introduce an entropy penalty over the output distribution:

$$\mathcal{L}_{\text{gating}} = \mathcal{L}_{\text{task}} + \lambda \cdot \mathcal{H}(\boldsymbol{\pi}), \tag{5}$$

where $\mathcal{L}_{\text{task}}$ is the downstream task loss (cross-entropy loss for generation), $\mathcal{H}(\boldsymbol{\pi}) = -\sum_j \pi_j \log \pi_j$ is the entropy of the soft routing scores, and λ is a tunable hyperparameter. Lower entropy encourages confident decisions, while higher entropy accommodates uncertain or ambiguous cases.

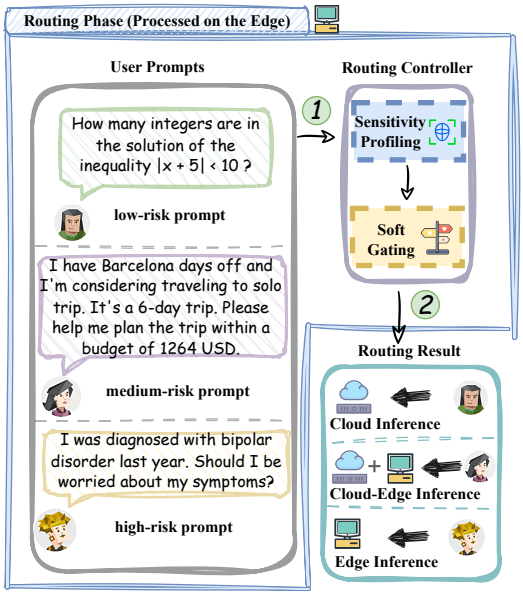


Figure 2: Illustration of the PRISM routing phase, where the edge-side controller analyzes prompt sensitivity and softly routes requests across cloud, edge, or collaborative paths.

Figure 2 provides an illustrative overview of the soft gating mechanism, showing how prompts of varying sensitivity are analyzed and softly routed to the most appropriate execution pathway based on their privacy characteristics.

At inference time, we select the most probable path by taking the top-1 decision from the soft routing distribution $\boldsymbol{\pi}$, i.e., $\arg\max_j \pi_j$. This deterministic selection ensures consistent privacy guarantees and avoids routing sensitive prompts to lower-protection paths due to randomness. Our formulation thus provides a principled and interpretable interface for integrating symbolic sensitivity signals with soft decision boundaries.

Adaptive Two-Layer Local Differential Privacy

In cloud–edge collaborative collaboration, prompts containing sensitive entities must be encrypted before transmission. Naive anonymization strategies, such as replacing a name with a generic token ⟨NAME⟩, do not mitigate the risk of linkage attacks. For example, consider the two records: (1)*"⟨NAME⟩ owns a black dog and often goes for a walk after dinner."* and (2)*"The owner of the black dog throws rubbish on Tuesdays."* Although the user's name *"Bob"* has

been masked, the shared semantic content *"Black Dog"* allows an adversary to correlate the records and re-identify Bob. Worse, this cross-record inference also reveals behavioral details, such as Bob's routine and habits, amplifying the risk of privacy beyond identity disclosure.

Similarly, applying uniform ε-LDP, either throughout the prompt or uniformly across all types of entities, introduces suboptimal trade-offs: the former may disrupt linguistic coherence, while the latter either underprotects sensitive entities or overly perturbs benign ones, affecting downstream utility. Moreover, when noise is applied indiscriminately, the cloud model receives syntactically valid but semantically distorted prompts, often yielding vague or evasive completions (e.g., *"I cannot provide information about* ⟨MASKED_ENTITY⟩*"*).

To address these issues, we propose a *two-layer adaptive LDP mechanism* that separately perturbs each entity's *category* and *value* using independently allocated budgets ε_1 and ε_2, such that the total budget is preserved: $\varepsilon_{\text{total}} = \varepsilon_1 + \varepsilon_2$. The allocation is determined by the sensitivity weight w_{c_i}, which reflects how sensitive the category c_i of entity e_i is. This weight comes from our profiling module.

$$\begin{aligned} \varepsilon_1 &= \varepsilon_{\text{total}} \cdot \frac{w_{c_i}}{w_{c_i} + (1 - w_{c_i}) \cdot \alpha}, \\ \varepsilon_2 &= \varepsilon_{\text{total}} - \varepsilon_1, \end{aligned} \tag{6}$$

where $\alpha \in (0, 1]$ is a tunable hyperparameter. This formulation ensures high-sensitivity categories (e.g., *NAME*) receive more category-level protection, while lower-sensitivity types (e.g., *NATIONALITY*) allocate more to value-level obfuscation. This adaptive allocation supports flexible trade-offs without violating ε-LDP guarantees, following composition rules in local privacy (Duchi, Jordan, and Wainwright 2013). When w_{c_i} is high (e.g., *NAME, ID, DIAGNOSIS*), more budget is allocated to ε_1 to protect the entity type, reducing the risk of inferring sensitive categories. Conversely, for lower w_{c_i} (e.g., *ORGANIZATION, NATIONALITY, LOCATION*), the mechanism allocates more to ε_2 to preserve semantic utility while anonymizing the value.

Both the category-layer and value-layer perturbation follow the Randomized Response (RR) mechanism, a canonical construction for achieving ε-LDP: in finite discrete domains (Wang et al. 2017).

(1) Category-Layer ε_1-LDP:

$$p_1 = \frac{\exp(\varepsilon_1)}{\exp(\varepsilon_1) + K_1 - 1}, \quad c_i^* = \begin{cases} c_i & \text{w.p. } p_1, \\ \neq c_i & \text{w.p. } \frac{1-p_1}{K_1-1}. \end{cases}$$

(2) Value-Layer ε_2-LDP:

$$p_2 = \frac{\exp(\varepsilon_2)}{\exp(\varepsilon_2) + K_2 - 1}, \quad e_i^* = \begin{cases} e_i & \text{w.p. } p_2, \\ \neq e_i & \text{w.p. } \frac{1-p_2}{K_2-1}. \end{cases} \tag{7}$$

Figure 3 illustrates the adaptive two-layer LDP workflow. After profiling, each entity undergoes privacy budget allocation based on its category sensitivity weight w_{c_i}. The entity is then sequentially processed by two perturbation layers:

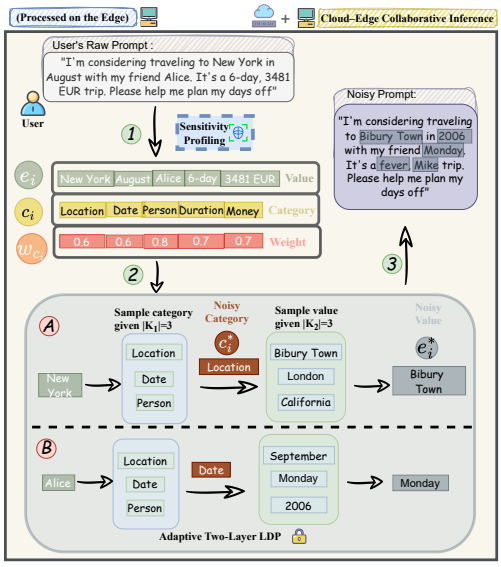


Figure 3: Adaptive Two-Layer LDP for Entity Obfuscation

the first samples a noisy category label from the set $\mathcal{C}$ of size K_1 with probability p_1, and the second samples a value from the corresponding value set $\mathcal{V}_{c_i}$ of size K_2 with probability p_2. To illustrate the adaptive nature of the method, we show two representative cases. Type B entities (e.g., *NAME*) are assigned high sensitivity weights and therefore undergo heavier category-level obfuscation to hide their semantic type. In contrast, type A entities (e.g., *LOCATION*) are considered less sensitive; they retain their category label while receiving stronger perturbation on the value level, ensuring semantic coherence of the prompt. This selective strategy enables better privacy–utility trade-offs, especially in resource-constrained cloud–edge deployments.

Our method matches the optimal ε-LDP design for finite discrete domains (Wang et al. 2017), but extends it via category-aware allocation. When $w_{c_i} \to 1, \varepsilon_1 \to \varepsilon_{\text{total}}$, maximizing protection of the category. When $w_{c_i} \to 0$, the reverse holds, preserving utility. Compared to static masking or uniform LDP, our risk-adaptive mechanism preserves semantic coherence by selectively perturbing high-risk entities more aggressively while minimizing distortion on benign ones.

Cloud-Edge Semantic Sketch Collaboration

In our architecture, each input prompt first passes through a routing phase that determines the appropriate execution mode: edge-only, cloud-edge collaboration, or cloud-only, based on its assessed privacy risk and resource constraints. Prompts routed to the collaborative mode then undergo our adaptive two-layer LDP mechanism,producing a perturbed prompt P^*, where P^* replaces sensitive entities in P using randomized response. The resulting noisy prompt is directly transmitted to the cloud in plain text, avoiding the transmission of embeddings and eliminating the need for synchronized tokenizers or shared embedding spaces between de-

vices.

Given the perturbed prompt P^*, the cloud-side LLM $\mathcal{G}_{\text{cloud}}$ generates a semantic sketch $S \in \mathcal{T}_{\text{sketch}}$ using a few-shot in-context prompt using a demonstration set $\mathcal{D}_{\text{cloud}} = \{(P^{*(l)}, S^{(l)})\}_{l=1}^{k}$. The generation context is constructed as $\mathcal{C}_{\text{cloud}} = [\mathcal{D}_{\text{cloud}}, (P^*, \_)]$, where $\_$ indicates the sketch to be generated. The cloud model then produces:

$$S = \mathcal{G}_{\text{cloud}}(\mathcal{C}_{\text{cloud}}). \tag{8}$$

The sketch S adopts a concise, structured format and omits sensitive entities obfuscated in P^*. This design ensures (i) *semantic alignment*, preserving the original intent despite noise, and (ii) *structural regularity*, supporting downstream integration.

Upon receiving the sketch S, the edge-side SLM $\mathcal{G}_{\text{edge}}$ reconstructs the final response $\hat{R}$ by conditioning on both S and the original prompt P, which remains locally available.

Similarly, we define the edge-side demonstration set as $\mathcal{D}_{\text{edge}} = \{(P^{(l)}, S^{(l)}, R^{(l)})\}_{l=1}^{k}$, and construct the generation context as $\mathcal{C}_{\text{edge}} = [\mathcal{D}_{\text{edge}}, (P, S, \_)]$, where $\_$ denotes the final response to be generated. The final response is then generated by the edge model:

$$\hat{R} = \mathcal{G}_{\text{edge}}(\mathcal{C}_{\text{edge}}). \tag{9}$$

This collaborative inference design preserves privacy while leveraging semantic abstraction for faithful and efficient response generation.

To unify the overall decision and execution process, we present the complete PRISM inference pipeline in Algorithm 1. It integrates sensitivity-aware routing, entity-level perturbation via adaptive LDP, and collaborative generation with sketch-based semantic alignment. This algorithm formalizes the end-to-end behavior of our framework across all execution modes, including edge-only, cloud-only, and cloud–edge collaboration.

Algorithm 1: PRISM Framework

Input: User prompt $P = \{x_1, \ldots, x_n\}$;
Cloud demo set $\mathcal{D}_{\text{cloud}} = \{(P^{*(l)}, S^{(l)})\}_{l=1}^{k}$;
Edge demo set $\mathcal{D}_{\text{edge}} = \{(P^{(l)}, S^{(l)}, R^{(l)})\}_{l=1}^{k}$;
Models $\mathcal{G}_{\text{cloud}}, \mathcal{G}_{\text{edge}}$
Output: Final response $\hat{R}$
$(R(P), \mathbf{d}) \leftarrow \textsc{SensitivityProfiling}(P)$;
$\boldsymbol{\pi} \leftarrow \textsc{SoftGating}([R(P), \mathbf{d}])$;
`mode` $\leftarrow \arg\max_j \pi_j$;
if `mode` $=$ `edge-only` **then**
  return $\hat{R} = \mathcal{G}_{\text{edge}}(P)$
if `mode` $=$ `cloud-only` **then**
  return $\hat{R} = \mathcal{G}_{\text{cloud}}(P)$
if `mode` $=$ `collaborative` **then**
  $P^* \leftarrow \textsc{TwoLayerLDP}(P, \mathbf{d})$;
  $\mathcal{C}_{\text{cloud}} \leftarrow [\mathcal{D}_{\text{cloud}}, (P^*, \_)]$;
  $S \leftarrow \mathcal{G}_{\text{cloud}}(\mathcal{C}_{\text{cloud}})$;
  $\mathcal{C}_{\text{edge}} \leftarrow [\mathcal{D}_{\text{edge}}, (P, S, \_)]$;
  $\hat{R} \leftarrow \mathcal{G}_{\text{edge}}(\mathcal{C}_{\text{edge}})$;
  return $\hat{R}$

Analysis

In this section, we provide a rigorous analysis of the privacy properties of our selective-privacy cloud–edge inference framework. Specifically, we establish the local differential privacy guarantees of our two-layer perturbation mechanism and characterize how sensitivity weights influence budget allocation.

Theorem 1 (Two-Layer LDP Privacy Guarantee)**.** *Let M be the adaptive two-layer mechanism applied to a sensitive entity e_i with category $c_i \in \mathcal{C}$ (of size K_1) and value domain $\mathcal{V}_{c_i^*}$ (of size K_2). The mechanism sequentially applies:*

1. *Category-level randomized response M_1 with budget ϵ_1, producing c_i^*;*
2. *Value-level randomized response M_2 with budget ϵ_2, producing e_i^*.*

Then $M = M_2 \circ M_1$ satisfies $(\epsilon_1 + \epsilon_2)$-local differential privacy over the pair (c_i, e_i).

Proof. According to the definition of ε-local differential privacy(Dagan, Glickman, and Magnini 2006; Dwork 2006), a mechanism M satisfies ε-LDP if for any two distinct inputs x, x' and any output y, we have:

$$\frac{\Pr[M(x) = y]}{\Pr[M(x') = y]} \leq \exp(\varepsilon).$$

We now verify this condition for each component.

Step 1: Category-level privacy. We apply a standard randomized response mechanism over the finite category domain $\mathcal{C}$ of size K_1, where the probability of outputting the true class is $p_1 = \frac{\exp(\varepsilon_1)}{\exp(\varepsilon_1)+K_1-1}$, and the remaining probability mass $(1-p_1)$ is distributed uniformly over the $K_1 - 1$ other categories:

$$\Pr[M_1(c_i) = c_i^*] = \begin{cases} p_1, & \text{if } c_i^* = c_i, \\ \frac{1-p_1}{K_1-1}, & \text{if } c_i^* \neq c_i. \end{cases}$$

For any two distinct inputs $c_i, c_i' \in \mathcal{C}$, and any output c_i^*, we compute the likelihood ratio:

$$\frac{\Pr[M_1(c_i) = c_i^*]}{\Pr[M_1(c_i') = c_i^*]} = \begin{cases} \frac{p_1}{\frac{1-p_1}{K_1-1}} = \exp(\varepsilon_1), & \text{if } c_i^* = c_i, \\ \frac{\frac{1-p_1}{K_1-1}}{p_1} = \frac{1}{\exp(\varepsilon_1)}, & \text{if } c_i^* = c_i'. \end{cases}$$

In either case, the ratio is bounded: $\frac{\Pr[M_1(c_i)=c_i^*]}{\Pr[M_1(c_i')=c_i^*]} \leq \exp(\epsilon_1)$. Thus, M_1 satisfies ε_1-LDP.

Step 2: Value-level privacy. Over the value domain $\mathcal{V}_{c_i^*}$ of size K_2, the randomized response mechanism M_2 outputs:

$$\Pr[M_2(e_i) = e_i^*] = \begin{cases} p_2 = \frac{\exp(\varepsilon_2)}{\exp(\varepsilon_2)+K_2-1}, & \text{if } e_i^* = e_i, \\ \frac{1-p_2}{K_2-1}, & \text{if } e_i^* \neq e_i. \end{cases}$$

A similar analysis shows that for any two distinct values e_i, e_i', the likelihood ratio is bounded by $\exp(\varepsilon_2)$, so M_2

satisfies ε_2-LDP. *Step 3: Composition.* Since M_1 and M_2 are applied independently and sequentially to disjoint components (c_i, e_i), we apply the sequential composition theorem (Duchi, Jordan, and Wainwright 2013) to obtain:

$$M(c_i, e_i) = M_2 \circ M_1 \Rightarrow (\varepsilon_1 + \varepsilon_2)\text{-LDP}.$$

□

This theorem guarantees that for each individually selected sensitive entity, our adaptive two-layer LDP provides bounded privacy leakage consistent with the allocated total budget $\epsilon_{\text{total}} = \epsilon_1 + \epsilon_2$.

Theorem 2 (Effect of Sensitivity Weight on Budget Allocation)**.** *Let $w_{c_i} \in [0, 1]$ be the sensitivity weight of category c_i, and let $\varepsilon_{total} > 0$ be the total privacy budget. Define the allocation as:*

$$\varepsilon_1 = \varepsilon_{total} \cdot \frac{w_{c_i}}{w_{c_i} + (1 - w_{c_i}) \cdot \alpha}, \quad \varepsilon_2 = \varepsilon_{total} - \varepsilon_1,$$

where $\alpha \in (0, 1]$ is a tunable hyperparameter that controls the relative importance of value protection. Then:

1. *When $w_{c_i} = 1$, we have $\varepsilon_1 = \varepsilon_{total}$, $\varepsilon_2 = 0$.*
2. *When $w_{c_i} = 0$, we have $\varepsilon_1 = 0$, $\varepsilon_2 = \varepsilon_{total}$.*
3. *$\varepsilon_1(w_{c_i})$ is monotonically increasing on $[0, 1]$; $\varepsilon_2(w_{c_i})$ decreases monotonically.*

Proof. We analyze the function:

$$\varepsilon_1(w_{c_i}) = \varepsilon_{\text{total}} \cdot \frac{w_{c_i}}{w_{c_i} + (1 - w_{c_i}) \cdot \alpha}, \quad w_{c_i} \in [0, 1].$$

Boundary Cases:

- When $w_{c_i} = 1$, the fraction becomes $\frac{1}{1} = 1$, so $\varepsilon_1 = \varepsilon_{\text{total}}, \varepsilon_2 = 0$.
- When $w_{c_i} = 0$, the fraction becomes $\frac{0}{\alpha} = 0$, so $\varepsilon_1 = 0, \varepsilon_2 = \varepsilon_{\text{total}}$.

Monotonicity: Consider the derivative of $\varepsilon_1(w_{c_i})$ with respect to w_{c_i}:

$$\frac{d\varepsilon_1(w_{c_i})}{dw_{c_i}} = \varepsilon_{\text{total}} \cdot \frac{\alpha}{((\alpha - 1)w_{c_i} - \alpha)^2}.$$

Note that the denominator is a square term and therefore always nonnegative. It equals zero if and only if:

$$(\alpha - 1)w_{c_i} - \alpha = 0 \quad \Leftrightarrow \quad w_{c_i} = \frac{\alpha}{\alpha - 1}.$$

Since $\alpha \in (0, 1]$, we have $\alpha - 1 < 0$, so $\frac{\alpha}{\alpha - 1} < 0$, which lies outside the domain $w_{c_i} \in [0, 1]$. Therefore, the denominator is strictly positive throughout the valid domain. Moreover, since the numerator is the product of $\alpha \in (0, 1]$ and $\varepsilon_{\text{total}} > 0$, it is strictly positive.

Hence, the entire expression is strictly greater than zero for all $w_{c_i} \in [0, 1]$, proving that the allocation function $\varepsilon_1(w_{c_i})$ is monotonically increasing, and $\varepsilon_2(w_{c_i}) = \varepsilon_{\text{total}} - \varepsilon_1(w_{c_i})$ is strictly decreasing.

□

This theorem confirms that higher category sensitivity weights allocate more budget to category-level perturbation (ε_1), as changing the category of an entity induces greater semantic distortion, offering stronger protection for highly sensitive categories. Lower weights favor value-level noise (ε_2) to preserve utility for less sensitive entities.

Evaluation

Evaluation Setting

Dataset: To facilitate comprehensive evaluation under privacy-sensitive scenarios, we construct a semi-synthetic instruction dataset designed to emulate realistic user–LLM interactions across four representative application domains: (1) *Tourism planning*, covering user identities, travel budgets, and destinations; (2) *Medical consultation*, involving demographic attributes and symptom descriptions, partially adapted from the dataset used in PrivacyRestore (Zeng et al. 2025); (3) *Banking services*, featuring transaction histories, account identifiers, and institutional details; and (4) *General knowledge*, drawn from MT-Bench (Zheng et al. 2023) as a non-sensitive dataset. Each domain contains 40 prompts with diverse phrasings and structured entity distributions, enabling controlled variation in contextual sensitivity. This dataset supports systematic, fine-grained benchmarking of privacy-aware LLM inference strategies.

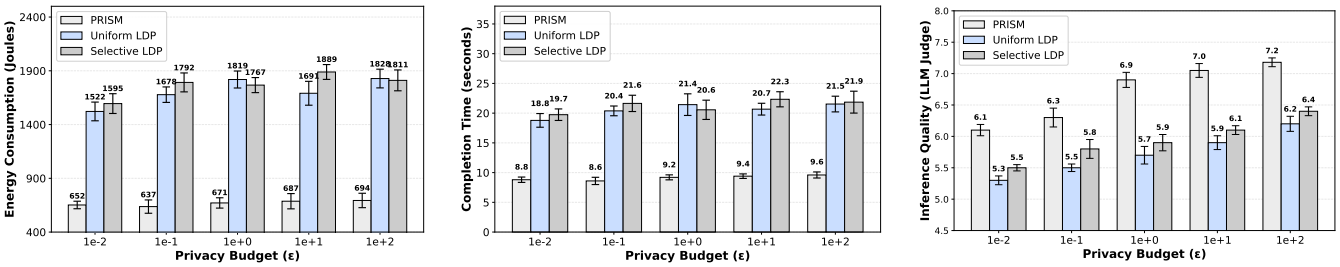
Environment Setup: We evaluate PRISM in a realistic cloud-edge deployment to reflect practical resource constraints and latency characteristics. The edge-side model is deployed on a local workstation equipped with a NVIDIA RTX 3070 laptop GPU running Windows 10. We use the *llama_cpp_python* backend to serve quantized SLMs, with the GPU offloading configuration set to *gpu_layers=32*. The cloud-side model is accessed via API calls to hosted LLMs The cloud performs sketch generation based on the perturbed prompts received from the edge and returns the abstracted sketches for downstream reconstruction.

Baselines: We evaluate our framework against the following representative baselines. (1) *Uniform LDP* adds Laplace noise to all input tokens and sends the perturbed prompt to the cloud for inference, followed by edge-side refinement without semantic sketching (Mai et al. 2024). (2) *Selective LDP* applies LDP only to entities identified by the NER model (Shi et al. 2022), then performs cloud inference and edge refinement similarly. (3) *Cloud-only* sends the entire unaltered prompt to a cloud-based LLM for inference. (4) *Edge-only* performs inference locally on an edge device using SLM with the original prompt.

Evaluation Metrics: We evaluate each method using three key metrics. (1) *Inference Quality* is assessed by GPT-4o-based scoring on a scale from 1 to 10, reflecting the relevance, coherence, and informativeness of the generated responses (Zheng et al. 2023). (2) *Energy Consumption* is measured in Joules using Windows-based power monitoring tools, reflecting the total energy consumed by the edge device during the inference period. For cloud-only inference, this includes only the energy cost incurred by data transmission and idle system operations on the edge, since the actual model execution occurs remotely on the cloud. (3) *Completion Time* denotes the end-to-end latency from prompt input to final response, indicating system responsiveness.

Results

Table 1 shows that PRISM achieves the best overall efficiency among privacy-preserving methods, with only 7.92 s completion time and 687 J energy consumption. This effi-



(a) Energy Consumption (b) Completion Time (c) Inference Quality

Figure 4: Comparison of privacy-preserving methods across privacy budgets on three dimensions: (a) energy consumption, (b) completion time, and (c) inference quality evaluated by LLM Judge.

Method	Ct.(s)	Ec.(J)	IQ.
PRISM	7.92	687.16	6.88
Uniform LDP	20.56	1707.6	5.72
Selective LDP	21.22	1770.8	5.94
Edge-Only	17.84	1573.9	5.09
Cloud-Only	**5.13**	**296.27**	**8.14**

Table 1: Performance metrics across methods. Metrics include completion time (Ct.), energy consumption (Ec.), and inference quality (IQ.). The best performance per metric is shown in **bold**.

ciency stems from its adaptive routing mechanism, which sends non-sensitive prompts directly to the cloud, routes low-risk ones to cloud–edge inference, and retains highly sensitive prompts for local processing. While Cloud-only attains the best raw performance (5.13 s, 296 J, IQ = 8.14), it offers no privacy protection. In contrast, PRISM ranks second across all methods, and requires only 1.54× the latency and 2.32× the energy of Cloud-only, making it a favorable privacy–efficiency trade-off. Meanwhile, Uniform and Selective LDP methods incur over 20 s latency and 1700+ J energy due to indiscriminate or entity-based noise injection.

Figure 4 illustrates the performance trends of three privacy-preserving methods across varying privacy budgets. We observe that PRISM consistently outperforms Uniform and Selective LDP in all three metrics, achieving notably lower energy consumption (e.g., 652 J vs. 1800+ J), shorter completion time (e.g., 8.8 s vs. 21+ s), and higher inference quality (up to 7.2 vs. 6.2 or lower). PRISM exhibits stable performance across privacy budgets. This robustness arises from its adaptive routing and minimal perturbation design, which avoids over-noising and computational waste while maintaining utility under strong privacy constraints.

Table 2 presents the performance of PRISM in eight combinations of cloud-edge models. We observe that all pairings deliver high inference quality (IQ $\geq$ 6.9) with moderate latency (7.08–8.60 s) and low energy (632–739 J), demonstrating the flexibility of PRISM in heterogeneous deployments. In particular, the combination of GPT-4o (L1) and Qwen1.5-1.8B (S2) achieves the fastest inference (7.08 s) and the lowest energy (632 J), while GPT-4o with StableLM (S3) yields the highest quality (7.16). Meanwhile, the Qwen3-235B (L2) variants offer a slightly slower response but match or exceed the quality of the generation (up to 7.22), underscoring the adaptability of PRISM to varying cloud / edge model capabilities.

Model Combinations	PRISM		
	Ct.(s)	Ec.(J)	IQ.
L1 + S1	8.29	683.83	7.00
L1 + S2	**7.08**	**632.24**	6.91
L1 + S3	7.34	657.88	7.16
L1 + S4	7.35	653.62	5.28
L2 + S1	8.59	738.88	**7.22**
L2 + S2	8.60	739.59	7.06
L2 + S3	8.00	693.13	7.19
L2 + S4	8.11	698.10	7.19

Table 2: Performance of PRISM across various combinations of cloud-side LLMs (L1: GPT-4o, L2: Qwen3-235B) and edge-side SLMs (S1: Phi-3.5-mini-3.5B, S2: Qwen1.5-1.8B, S3: StableLM-2-Zephyr-1.6B, S4: TinyLLaMA-1.1B).

Conclusion

This work presents PRISM, a novel privacy-aware cloud–edge inference framework that dynamically routes user prompts based on semantic sensitivity. By integrating an adaptive two-layer local differential privacy mechanism, and a semantic sketch-based collaboration protocol, PRISM enables efficient, privacy-preserving inference without compromising utility. Comprehensive evaluations across four domains and eight model configurations demonstrate that PRISM achieves superior performance trade-offs, incurring only 1.54× latency and 2.32× energy overhead compared to cloud-only inference, while preserving strong privacy guarantees and maintaining high output quality.

As future work, we plan to extend PRISM to support collaborative inference across multiple edge devices, each equipped with an SLM. This setting introduces new challenges in routing, load balancing, and cloud-edge-device coordination, which we aim to address through decentralized scheduling and federated optimization mechanisms.

References

Acharya, J.; Bonawitz, K.; Kairouz, P.; Ramage, D.; and Sun, Z. 2020. Context Aware Local Differential Privacy. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, 52–62.

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; et al. 2022. Flamingo: A Visual Language Model for Few-Shot Learning. In *Advances in Neural Information Processing Systems (NeurIPS 2022)*, 23716–23736.

Cai, H.; Dong, H.; Wang, H.; Li, K.; and Akan, O. B. 2025a. Graph Representation-based Model Poisoning on Federated LLMs in CyberEdge Networks. *arXiv:2507.01694*.

Cai, H.; Wang, H.; Dong, H.; Li, K.; and Akan, O. B. 2025b. Graph Representation-based Model Poisoning on the Heterogeneous Internet of Agents. arXiv:2511.07176.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; et al. 2021. Evaluating Large Language Models Trained on Code. *arXiv:2107.03374*.

Dagan, I.; Glickman, O.; and Magnini, B. 2006. The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the Machine Learning Challenges Workshop (MLCW 2006)*, 177–190.

Drori, I.; Zhang, S.; Shuttleworth, R.; Tang, L.; Lu, A.; Ke, E.; Liu, K.; Chen, L.; Tran, S.; Cheng, N.; et al. 2022. A Neural Network Solves, Explains, and Generates University Math Problems by Program Synthesis and Few-Shot Learning at Human Level. *Proceedings of the National Academy of Sciences (PNAS)*, 119(32): e2123433119.

Du, M.; Yue, X.; Chow, S. S. M.; Wang, T.; Huang, C.; and Sun, H. 2023. DP-Forward: Fine-Tuning and Inference on Language Models with Differential Privacy in Forward Pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS 2023)*, 1512–1526.

Duchi, J. C.; Jordan, M. I.; and Wainwright, M. J. 2013. Local Privacy and Statistical Minimax Rates. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, 429–438.

Dwork, C. 2006. Differential Privacy. In *Proceedings of the 33rd International Colloquium on Automata, Languages, and Programming (ICALP 2006)*, 1–12.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; et al. 2024. The LLaMA 3 Herd of Models. *arXiv:2407.21783*.

Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; et al. 2024. GPT-4o System Card. *arXiv:2410.21276*.

Jin, H.; and Wu, Y. 2024. CE-CoLLM: Efficient and Adaptive Large Language Models Through Cloud-Edge Collaboration. *arXiv:2411.02829*.

Kweon, S.; Kim, J.; Kwak, H.; Cha, D.; Yoon, H.; Kim, K.; Yang, J.; Won, S.; and Choi, E. 2024. EHRNoteQA: An LLM Benchmark for Real-World Clinical Practice Using Discharge Summaries. In *Advances in Neural Information Processing Systems (NeurIPS 2024)*, 124575–124611.

Li, B.; Jiang, Y.; Gadepally, V.; Tiwari, D.; et al. 2024. LLM Inference Serving: Survey of Recent Advances and Opportunities. *arXiv:2407.12391*.

Li, S.; Wang, H.; Xu, W.; Zhang, R.; Guo, S.; Yuan, J.; Zhong, X.; Zhang, T.; and Li, R. 2025. Collaborative Inference and Learning between Edge SLMs and Cloud LLMs: A Survey of Algorithms, Execution, and Open Challenges. *arXiv:2507.16731v1*.

Lin, S.; Hua, W.; Wang, Z.; Jin, M.; Fan, L.; and Zhang, Y. 2025. EmojiPrompt: Generative Prompt Obfuscation for Privacy-Preserving Communication with Cloud-based LLMs. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2025)*, 12342–12361.

Lin, Z.; Qu, G.; Chen, Q.; Chen, X.; Chen, Z.; Huang, K.; et al. 2023. Pushing Large Language Models to the 6G Edge: Vision, Challenges, and Opportunities. *arXiv:2309.16739*.

Mai, P.; Yan, R.; Huang, Z.; Yang, Y.; and Pang, Y. 2024. Split-and-Denoise: Protect Large Language Model Inference with Local Differential Privacy. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, 1–20.

Miao, X.; Oliaro, G.; Zhang, Z.; Cheng, X.; Jin, H.; Chen, T.; and Jia, Z. 2025. Towards Efficient Generative Large Language Model Serving: A Survey from Algorithms to Systems. *ACM Computing Surveys*, 58(1): 1–37.

Qu, G.; Chen, Q.; Wei, W.; Lin, Z.; Chen, X.; and Huang, K. 2025. Mobile Edge Intelligence for Large Language Models: A Contemporary Survey. *IEEE Communications Surveys & Tutorials*.

Shi, W.; Cui, A.; Li, E.; Jia, R.; and Yu, Z. 2022. Selective Differential Privacy for Language Modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2022)*, 2848–2859.

Staab, R.; Vero, M.; Balunovic, M.; and Vechev, M. 2024. Beyond Memorization: Violating Privacy via Inference with Large Language Models. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024)*.

Wang, T.; Blocki, J.; Li, N.; and Jha, S. 2017. Locally Differentially Private Protocols for Frequency Estimation. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 729–745.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; et al. 2025. Qwen3 Technical Report. *arXiv:2505.09388*.

Zeng, Z.; Wang, J.; Yang, J.; Lu, Z.; et al. 2025. PrivacyRestore: Privacy-Preserving Inference in Large Language Models via Privacy Removal and Restoration. In *Proceedings of the Association for Computational Linguistics (ACL 2025)*, 10821–10855.

Zhan, H.; Zhang, X.; Tan, H.; Tian, H.; Yong, D.; Zhang, J.; and Li, X.-Y. 2025. PICE: A Semantic-Driven Progressive Inference System for LLM Serving in Cloud-Edge Networks. *arXiv:2501.09367*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; et al. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 46595–46623.

Zhu, P.; and Yang, T. 2025. CE-LSLM: Efficient Large-Small Language Model Inference and Communication via Cloud-Edge Collaboration. *arXiv:2505.14085*.