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

Appendix

This appendix provides additional theoretical analyses, implementation details, and experimental results to support the findings presented in the main paper. The contents are organized as follows:

- **Appendix A:** PRISM Architecture and Model Settings
- **Appendix B:** Theoretical Extensions and Proof Sketches

A PRISM Architecture and Model Settings

A.1 Overview

The PRISM framework consists of four primary components deployed across edge and cloud environments:

- **Edge-side Entity Profiler**: Performs fine-grained sensitivity detection over user prompts using a lightweight named entity recognition (NER) module.
- **Soft Routing Gater**: A lightweight neural gating module that maps symbolic sensitivity features to a soft probability distribution over three execution paths (cloud-only, edge-only, and collaboration). It is trained with entropy regularization to balance privacy risk, utility, and energy efficiency.
- **Adaptive Two-Layer LDP Module**: Applies hierarchical perturbations over sensitive entities based on entity type and content value.
- **Cloud–Edge Semantic Collaboration Module**: Generates abstract sketches on the cloud-hosted LLM, which are then decoded and refined on the edge-side small language model (SLM).

The architecture supports flexible configurations of SLMs and LLMs, enabling deployment across a range of hardware environments.

A.2 Small Language Models (SLMs)

We evaluate PRISM with multiple lightweight SLMs deployed on edge devices. All models are loaded in GGUF format and quantized to reduce memory and latency overhead. The following models are used in our experiments:

Model Name	**Parameters**	**Quantization**	**File**
TinyLLaMA Chat	1.1B	Q8_0	`tinyllama-1.1b-chat-v1.0.Q8_0.gguf`
Phi-3.5 Mini Instruct	3.8B	Q6_K_L	`Phi-3.5-mini-instruct-Q6_K_L.gguf`
StableLM Zephyr	1.6B	Q6_K	`stablelm-2-zephyr-1_6b.Q6_K.gguf`
Qwen1.5 Chat	1.8B	Q6_K	`qwen1_5-1_8b-chat-q6_k.gguf`

Table 1: Edge-deployed SLMs used in PRISM evaluation.

All SLMs are run on consumer-grade GPUs (e.g., NVIDIA RTX 3070 laptop GPU) with `llama.cpp`-based inference. For latency and energy profiling, models are launched in a single-batch, no-streaming configuration.

A.3 Cloud-side LLM

We use GPT-4o (OpenAI, 2024) as the backbone for cloud-side inference and sketch generation. All API-based interactions follow OpenAI's default chat-completion endpoint with system-level few-shot prompting. Model temperature is set to 0.7 unless otherwise specified.

A.4 Soft Gating with Entropy-Regularized Routing

The PRISM framework employs a soft gating module to compute context-aware routing distributions over three execution paths: cloud-only, edge-only, and cloud–edge collaboration.

Given a prompt $P = \{x_1, x_2, \ldots, x_n\}$, we extract a set of named entities $E = \{e_1, \ldots, e_m\}$, each associated with a binary sensitivity indicator $d_i \in \{0, 1\}$. We also compute an overall scalar risk score $R(P) \in \mathbb{R}$. These are concatenated to form the input feature vector $\mathbf{z} = [R(P); \mathbf{d}] \in \mathbb{R}^{1+m}$. For consistency, the sensitivity mask $\mathbf{d}$ is padded or truncated to a fixed dimension M during training.

The input $\mathbf{z}$ is processed by a lightweight neural mapping function $f_\theta(\cdot)$, producing a softmax-normalized routing probability:

$$\boldsymbol{\pi} = \mathrm{softmax}(f_\theta(\mathbf{z})) \in \mathbb{R}^3$$

where $\boldsymbol{\pi} = (\pi_{\text{cloud}}, \pi_{\text{collab}}, \pi_{\text{edge}})$. To encourage confident but flexible routing, we train the model with an entropy-regularized objective:

$$\mathcal{L}_{\text{gating}} = \mathcal{L}_{\text{task}} + \lambda \cdot \mathcal{H}(\boldsymbol{\pi}), \quad \mathcal{H}(\boldsymbol{\pi}) = -\sum_j \pi_j \log \pi_j$$

We set $\lambda = 0.4$ by default. The task loss $\mathcal{L}_{\text{task}}$ is the cross-entropy between the predicted path and the ground-truth label from routing supervision.

At inference time, the most probable routing decision is selected deterministically via $\arg\max_j \pi_j$, ensuring stable behavior and avoiding sensitive prompt misrouting due to sampling randomness.

A.5 Two-Layer Local Differential Privacy

We adopt a hierarchical entity-aware LDP mechanism:

- **Layer 1 (Type-level flipping)**: Randomly flips entity type with privacy budget ϵ_1.
- **Layer 2 (Value-level masking)**: Replaces entity content using a type-specific replacement distribution with privacy budget ϵ_2.

We fix total privacy budget $\epsilon = \epsilon_1 + \epsilon_2$, and allocate budget proportionally based on entity category risk (e.g., NAME vs. LOCATION). In our main experiments, we set $\epsilon = 1.0$, with $\epsilon_1 = 0.3$, $\epsilon_2 = 0.7$.

A.6 Sketch Generation and Edge Denoising

The cloud-side sketch is generated via a few-shot prompting scheme. A set of semantic abstraction examples is prepended to the user query, guiding the LLM to produce concise, structured summaries. Sketches are limited to 100 tokens. On the edge, the sketch is passed to the SLM for final response generation via a template-based decoding scheme.

To guide the cloud-side LLM $\mathcal{G}_{\text{cloud}}$ in generating sketches from perturbed prompts P^*, we design a diverse few-shot demonstration set $\mathcal{D}_{\text{cloud}} = \{(P^{*(l)}, S^{(l)})\}_{l=1}^{k}$, covering multiple obfuscation scenarios including value replacement, category mismatches, and mixed-type distortions. Each sketch is structured, privacy-respecting, and intent-aligned, providing inductive signals for semantic abstraction.

Below are selected examples from three categories (Tourism, Medical, Banking), showing how entity-level noise is handled during sketch generation.

Few-shot Demonstration Set for Sketch Generation

Tourism Examples

- **Clean Input**: *"I plan to travel solo to Tokyo for two days; help me design my itinerary."* **Sketch**: Day 1: Arrival, local exploration; Day 2: Cultural visit, outdoor activity, local dining.
- **Category Obfuscated (Location → Person)**: *"I plan to travel solo to Emma for two days; help me design my itinerary."* **Sketch**: Day 1–2: Orientation, general city exploration, museum or park, casual dining, outdoor walk.
- **Mixed Obfuscation**: *"I plan to travel solo to JPMorgan for five days with my friend Elon."* **Sketch**: Day 1–5: Destination orientation, historical sites, social activities, structured itinerary with flexibility.

Medical Examples

- **Clean Input**: *"A 28-year-old female patient named Emma reports symptoms: headache, dizziness."* **Sketch**: Demographics noted; Symptoms logged; Plan: Neurological exam, imaging, specialist referral.
- **Category Obfuscated (Person → Organization)**: *"A 28-year-old female patient named IBM reports symptoms: headache, dizziness."* **Sketch**: Symptoms documented; Possible stress-related or neurological; Diagnostics recommended.
- **Value Obfuscated**: *"A 42-year-old female patient named Google reports symptoms: nausea and fatigue."* **Sketch**: Patient history recorded; Symptom profile analyzed; Diagnostics planned.

Banking Examples

- **Clean Input**: *"I want to file a dispute regarding a charge of $10 on my Chase card."* **Sketch**: Dispute initiated; Amount and institution recorded; Resolution timeline outlined.
- **Category Obfuscated (Bank → Person)**: *"I want to file a dispute regarding a charge of $10 on my Alice card."* **Sketch**: Dispute recorded; Possible merchant error; Review process initiated.
- **Mixed Obfuscation**: *"I want to file a dispute regarding a charge of $25 on my Emma account."* **Sketch**: Charge issue reported; Card/entity obfuscated; Process outlined for user assistance.

Guidelines Enforced During Sketch Generation

- Sketches are structured outlines, not full responses.
- Use concise, semantically aligned formats across categories.
- Obfuscation is realized via entity-value substitution, not placeholder tokens.
- Maintain abstraction consistency even under semantic distortion.
- Do not attempt to correct or infer true sensitive entities.
- Each sketch must remain interpretable and adequate for downstream denoising.

Few-shot Demonstration Set for Edge-side Denoising and Response Generation

Each example illustrates how the edge-side model reconstructs the final response $\hat{R}$ based on the original prompt P and the cloud-generated sketch S (derived from perturbed P^*). Sketches may omit sensitive entities, which are recovered using the clean prompt P at the edge.

Tourism Domain

Example 1 (Clean Sketch)
Prompt (P): *"I plan to travel solo to Tokyo for two days; help me design my itinerary."*
Sketch (S): *"Day 1: Arrival and exploration; Day 2: Cultural visit and local experience."*
Response ($\hat{R}$): *"Sure! Here's a two-day itinerary for your Tokyo trip: Day 1: Morning – arrive and settle in; Afternoon – explore Shibuya and Meiji Shrine; Evening – try local ramen shops. Day 2: Morning – visit Asakusa and Senso-ji Temple; Afternoon – stroll in Ueno Park; Evening – Tokyo Tower view and sushi dinner."*

Example 2 (Obfuscated Cloud Sketch)
Prompt (P): *"I plan to travel solo to Tokyo for two days; help me design my itinerary."*
Sketch (S): *"Day 1: Arrival and destination orientation; Day 2: Cultural/historical visit, local dining."*
Response ($\hat{R}$): *"Absolutely! For your Tokyo trip: Day 1: Check into your hotel, explore Shinjuku, visit Tokyo Metropolitan Building. Day 2: Start with Meiji Shrine, then visit the Edo-Tokyo Museum, and enjoy dinner in a local izakaya."*

Medical Domain

Example 3 (Clean Sketch)
Prompt (P): *"A 28-year-old female patient named Emma reports symptoms: headache and blurred vision."*
Sketch (S): *"Demographics noted; Symptoms: Neurological; Plan: Imaging, specialist referral."*
Response ($\hat{R}$): *"Emma, a 28-year-old female, presents with headache and blurred vision. Recommend neurological exam, MRI brain imaging, and referral to neurology."*

Example 4 (Obfuscated Cloud Sketch)
Prompt (P): *"A 28-year-old female patient named Emma reports symptoms: headache and blurred vision."*
Sketch (S): *"Adult patient with neurological symptoms; plan includes imaging and specialist evaluation."*
Response ($\hat{R}$): *"Patient Emma, 28 years old, presents with headache and blurred vision. Schedule cranial MRI and consult neurology for further assessment."*

Banking Domain

Example 5 (Clean Sketch)
Prompt (P): *"I want to file a dispute for a $20 charge on my Chase card ending in 1234."*
Sketch (S): *"Dispute initiated for transaction; card identified; review and resolution process triggered."*
Response ($\hat{R}$): *"Understood. You've initiated a dispute for a $20 charge on your Chase card ending in 1234. We will review the transaction and notify you within 5–7 business days."*

Example 6 (Obfuscated Cloud Sketch)
Prompt (P): *"I want to file a dispute for a $20 charge on my Chase card ending in 1234."*
Sketch (S): *"Dispute filed for unknown card provider; steps include review and provisional credit."*
Response ($\hat{R}$): *"Thanks for reporting. You've filed a dispute on a $20 charge with Chase. We've started the review and will apply provisional credit if appropriate."*

Edge-side Sketch-to-Response Prompting Guidelines

- Use the original prompt P to recover personalized semantics lost during LDP masking.
- Use sketch S as structural guidance to organize the response flow.
- Ensure the reconstructed response preserves private context from P while respecting the semantic plan from S.
- Do not override or contradict sketch structure unless required by critical details in P.
- Favor fluent, context-specific responses (not merely copying sketch items).
- Include named entities, temporal information, and personalized wording from P.

B Theoretical Extensions and Proof Sketches

B.1 Theorem: Privacy Composition for Multiple Sensitive Entities

Definition. Consider a user prompt containing n sensitive entities $\{e_1, e_2, \ldots, e_n\}$ categorized into m types. Let C be the set of possible categories with $|C| = K_1$, and let each category $c \in C$ have a finite value domain $\mathcal{V}_c$ with maximum size K_2. The adaptive two-layer LDP mechanism M is applied to each e_i as:

1. Category-level randomized response $M^{(c)}$ with privacy budget ϵ_1, producing c_i^*;
2. Value-level randomized response $M^{(v)}$ with budget ϵ_2, producing $e_i^* \in \mathcal{V}_{c_i^*}$.

Define the combined mechanism $M^{(n)}$ as applying M independently to each e_i, yielding $P^* = \{e_1^*, \ldots, e_n^*\}$.

Theorem. For any $\epsilon_1, \epsilon_2 \geq 0$, the mechanism $M^{(n)}$ satisfies $\epsilon_{\text{total}} = (\epsilon_1 + \epsilon_2)$-LDP with respect to any single-entity change, regardless of n. Furthermore, changing all n entities leads to at most $n(\epsilon_1 + \epsilon_2)$-LDP.

Proof Sketch. For any adjacent prompts P and P' differing in only one entity e_j, we compute:

$$\frac{\Pr[M^{(n)}(P) = P^*]}{\Pr[M^{(n)}(P') = P^*]} = \frac{\Pr[M(e_j) = e_j^*]}{\Pr[M(e_j') = e_j^*]} \leq e^{\epsilon_1 + \epsilon_2}.$$

All other terms cancel due to independence. Hence $M^{(n)}$ is $(\epsilon_1 + \epsilon_2)$-LDP.

The group privacy extension follows directly: for g modified entities, we apply the composition theorem, yielding privacy loss $\leq g(\epsilon_1 + \epsilon_2)$.

Discussion. This shows that the per-entity privacy guarantee holds even as n grows, and total leakage scales linearly with the number of changed entities, as expected in parallel composition. Our scheme ensures bounded leakage per sensitive token without amplification.

B.2 Theorem: Bounded Information Leakage in Entropy-Regularized Soft Gating

Definition. Let $\pi(x) = [\pi_{\text{cloud}}(x), \pi_{\text{collab}}(x), \pi_{\text{edge}}(x)]$ be the routing distribution computed via softmax over logits $f_\theta(x)$, optionally scaled by temperature T. The gating module is trained with:

$$L_{\text{gating}} = L_{\text{task}}(\pi(x)) + \lambda \cdot H(\pi(x)),$$

where $H(\pi) = -\sum_j \pi_j \ln \pi_j$ is the entropy regularizer, and $\lambda > 0$.

Theorem. The soft routing output $\pi(x)$ satisfies:

1. $\max_j \pi_j(x) \leq p_{\max} < 1$, i.e., no hard routing occurs;
2. Entropy $H(\pi(x)) \geq H_{\min} > 0$ for all x;
3. As $\lambda \to \infty$, $\pi(x) \to (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, and information leakage $\to 0$;
4. As $\lambda \to 0$, $\pi(x)$ becomes peaked, and leakage approaches $\log 3$ bits.

Proof Sketch. At optimality, softmax ensures:

$$\pi_j(x) = \frac{\exp(-L_j/\lambda)}{\sum_k \exp(-L_k/\lambda)}.$$

The maximum component $\pi_{\max}(x)$ is bounded:

$$\pi_{\max}(x) \leq \frac{1}{1 + 2\exp(-\Delta/\lambda)} < 1,$$

where $\Delta = \max_{i \neq j}(L_j - L_i)$. The entropy is minimized when one π_j is dominant:

$$H_{\min} = -[p_{\max} \ln p_{\max} + (1 - p_{\max}) \ln(1 - p_{\max})],$$

which remains > 0 under finite λ. Therefore, soft gating ensures bounded confidence and limits information leakage. In the limit $\lambda \to \infty$, all $\pi_j \to 1/3$ and $H(\pi) \to \ln 3$.

Implication. Entropy regularization introduces a privacy knob in gating decisions. By setting λ appropriately, we bound how confidently the system routes based on sensitive inputs, mitigating privacy leakage through control flow itself.

Model	PRISM			Edge-Only			Cloud-Only		
	Ct.(s)	Ec.(J)	IQ.	Ct.(s)	Ec.(J)	IQ.	Ct.(s)	Ec.(J)	IQ.
(L1) GPT-4o **+** *(S1)* Phi-3.5-mini-3.5B	8.29	683.83	7.00	**15.98**	**1393.88**	5.19	5.22	**271.27**	**8.28**
(L1) GPT-4o **+** *(S2)* Qwen1.5-1.8B	**7.08**	**632.24**	6.91	17.29	1540.33	**5.59**	-	-	-
(L1) GPT-4o **+** *(S3)* Stablelm-2-zephyr-1.6B	7.34	657.88	7.16	18.57	1627.46	4.94	-	-	-
(L1) GPT-4o **+** *(S4)* Tinyllama-1.1B	7.35	653.62	5.28	19.50	1734.24	4.62	-	-	-
(L2) Qwen3-235B **+** *(S1)* Phi-3.5-mini-3.5B	8.59	738.88	**7.22**	-	-	-	**5.04**	321.28	8.01
(L2) Qwen3-235B **+** *(S2)* Qwen1.5-1.8B	8.60	739.59	7.06	-	-	-	-	-	-
(L2) Qwen3-235B **+** *(S3)* Stablelm-2-zephyr-1.6B	8.00	693.13	7.19	-	-	-	-	-	-
(L2) Qwen3-235B **+** *(S4)* Tinyllama-1.1B	8.11	698.10	7.19	-	-	-	-	-	-

Table 2: Performance comparison of PRISM, Edge-Only, and Cloud-Only execution modes across various LLM–SLM pairs. For each metric (completion time, energy consumption, inference quality), we highlight the best result in **bold**.