DP-LAIR: DIFFERENTIALLY PRIVATE LLM INFERENCE WITH ADAPTIVE ROUTING AND LATENT PERTURBATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Differential privacy (DP) provides a principled framework for protecting sensitive data, where its applications in machine learning have largely focused on training time privacy. Yet the transcript returned by an LLM service can reveal information from both private training records and query time context. Thus, We study the joint protection of these two sources under a practical trusted server, API-only deployment model, where internal model states remain hidden and private information becomes externally observable only through the released transcript. We propose *Differentially Private Latent Adaptive Inference and Routing* (DP-LAIR), a unified framework for protecting both training time and runtime private information at this interface. DP-LAIR introduces a router that directs tokens that can be generated from public information to a public path. For tokens requiring private context, their hidden state residual is compressed into a latent representation, clipped and perturbed with Gaussian noise, and mapped back to the output space by a decoder. The router and decoder are trained with differential privacy to protect training time data, while runtime privacy is adaptively accounted. Experiments across multiple datasets and models support DP-LAIR's protection of both training data and query time context while preserving generation utility.

1 INTRODUCTION

Differential privacy (DP) limits how much one record can affect a released model or its predictions (Abadi et al., 2016). The rapid advances in large language models (LLMs)(Brown et al., 2020) brings new challenges in DP-based privacy accounting. In an LLM service hosted on cloud with only API access, private information enters in two different ways. A downstream corpus may be used to fine-tune the base model to adapt user's requirements before deployment, while each query may bring new context through retrieval(Lewis et al., 2020), agent tools(Wu et al., 2023), or a system prompt. Both sources can influence the same generated transcript and create two attack surfaces: training data extraction or membership inference against the adapted system, and leakage of runtime private context, such as retrieved documents, tool outputs, or system prompts, through model responses (Shokri et al., 2017; Carlini et al., 2021; Morris et al., 2024).

We study a common API-only setting in which the provider operates a trusted server and the adversary observes only API outputs: logits or probabilities, sampled tokens, and termination behavior. Model parameters, hidden states, optimizer states, and raw context remain internal. During inference, the deployed backbone remains fixed. We consider two private data sources: a downstream corpus $S_{\boldsymbol{x}}$, used for fine-tuning, post-training, or other adaptation, and the query-time context $S_{\boldsymbol{y}}$.

This setting raises a joint question: *how can one API deployment protect the privacy of downstream training corpus and the runtime context at the released transcript?* Training-time DP mechanisms protect $S_{\boldsymbol{x}}$ but do not cover private context introduced after deployment, while prompt-protection mechanisms protect $S_{\boldsymbol{y}}$ but do not mitigate leakage originating from $S_{\boldsymbol{x}}$. A straightforward solution is to compose two independent mechanisms. Although this gives a joint privacy bound, the two protections are designed and optimized separately. Under a fixed privacy budget, this creates an allocation dilemma: more privacy loss budget assigned to one source leaves weaker protection for the other. Since the downstream corpus $S_{\boldsymbol{x}}$ and the context $S_{\boldsymbol{y}}$ are blended in the API output, designing

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

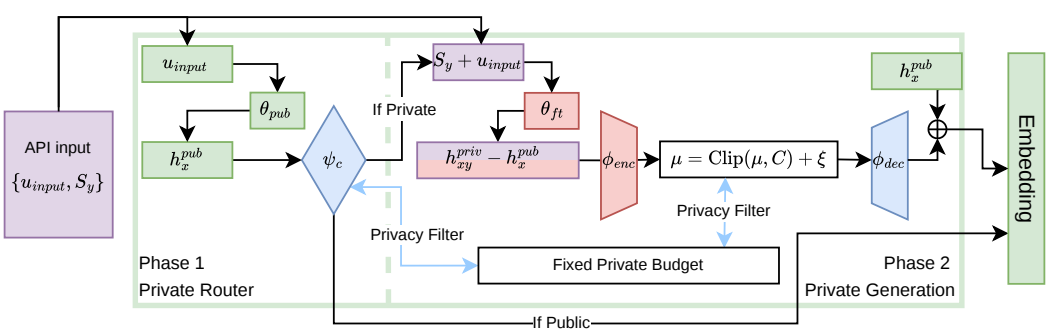


Figure 1: Overview of DP-LAIR. The **red**, **purple**, **blue** and **green** components denote the computation related to **the corpus** S_x, **the private prompt** S_y, **the DP-guaranteed parameters** and **the public data** respectively. The write components are either hyper-parameters or privacy-enhancing mechanism. The privacy of **the corpus** S_x and **the private prompt** S_y are protected by the Gaussian mechanism, **DP-guaranteed router and decoder**.

a mechanism to analyze the privacy protection of them jointly will lead to a tighter privacy guarantee and thus a better privacy-utility trade-off.

To this end, we propose **Differentially Private Latent Adaptive Inference and Routing (DP-LAIR)**, a lightweight privacy layer that protects both the downstream training corpus and runtime private context at the API output. As shown in Figure 1, DP-LAIR has three components. First, for **runtime context protection**, an encoder–decoder maps the private hidden-state residual to a low-dimensional latent space, where it is clipped and perturbed with Gaussian noise before being decoded back to the hidden space. Second, we introduce a **public path** to provide a public prior and avoid unnecessary use of private information. A router then chooses between the public and private paths based on public information only. Third, for **training data protection**, the router and decoder are trained with a DP-guaranteed algorithm. In summary, the privacy of the context S_y is protected by the Gaussian mechanism in the encoder-decoder maps, and the privacy of corpus S_x is protected by the DP-guaranteed training on the decoder and the router. They are designed and optimized jointly.

DP-LAIR trains these components in two stages. In Stage I (section 3.3.1), we jointly optimize the router, encoder, decoder, and latent covariance under a constrained objective that balances utility and runtime privacy cost. In Stage II (section 3.3.2), we freeze the encoder and covariance, reinitialize the router and decoder, and retrain them with differential privacy to protect the training corpus. This design places prompt protection, public routing, and training-data protection in the same inference pipeline rather than treating them as separate mechanisms.

We formalize the privacy of the two-source leakage in DP-LAIR as *Infer-RDP* and decompose it into a one-time training data cost and an adaptive runtime privacy cost. In section 3.2, DP-LAIR uses a Rényi privacy filter to track the adaptive per-token prompt cost and keep the accumulated Infer-RDP loss within a fixed budget. In summary, our contributions are listed as follows:

- We introduce DP-LAIR, a two-path architecture that confines access to the runtime context to a noisy latent interface and routes each token using public-only features.
- We formulate Infer-RDP and decompose it into a one time joint DP pretrained term and a fixed-query runtime term based on the maximum realized LOO cost under an adaptive budget filter.
- Experiments across multiple models and datasets indicate that DP-LAIR achieves better utility privacy trade-off than directly composite two DP guarantee method.

2 Related Work

Training-time protection. Differentially private stochastic gradient descent (DP-SGD) clips per-example gradients and adds calibrated noise during training (Abadi et al., 2016), with its guarantee transferring from the released parameters to model outputs by post-processing. Because per-example clipping and noise are costly for large models (Bu et al., 2023), DP-LoRA has become a common

approach to private LLM fine-tuning. A separate line of work shows that, under the hidden-state assumption where the adversary observes only the final model rather than intermediate training states, privacy loss need not grow with the number of training iterations (Feldman et al., 2018). Existing analyses often rely on conditions such as convexity (Ye & Shokri, 2022; Altschuler & Talwar, 2022); differentially private decoupled training (DPDT) (Chen et al., 2026) removes these conditions, at the cost of privacy loss that scales with the number of trainable parameters. These methods protect the training corpus but do not cover private context introduced after deployment.

Inference-time context protection. A complementary line of work protects private text supplied at inference time. One family perturbs the input before it reaches the target model. DP-Rewrite (Igamberdiev et al., 2022) and DP-MLM (Meisenbacher et al., 2024) rewrite private text under DP guarantees, while InferDPT (Tong et al., 2025) and Cape (Wu et al., 2025) perturb or replace sensitive tokens before querying the target model, with Cape performing the mechanism locally on the user device. Another family retains the private context during generation but sanitizes its influence through a DP generation mechanism. DP-Fusion (Thareja et al., 2025) combines output distributions obtained with and without sensitive context, InvisibleInk (Vinod et al., 2026) clips the private contribution relative to public logits and applies DP next-token sampling, and dps-mozo (Flemings et al., 2025) protects private ICL demonstrations through randomized distribution mixing during decoding. Despite their different mechanisms, these approaches protect inference-time private context rather than private information learned during training.

None of these approaches jointly protects the training corpus and query-time context under the trusted-server, API-only deployment considered in this work.

3 DP-LAIR: Private Modular Inference

3.1 Infer-RDP under the API-only Threat Model

In practice, an LLM service keeps the deployed model and its internal state on the server, while exposing only the generated transcript through an API. Each query-specific prompt $S_{\boldsymbol{y}}$ is assembled by an agentic workflow from context records, whereas the deployed state remains fixed during serving. This deployment pattern gives rise to the API-only threat model: an adversary may query the service adaptively but cannot inspect its internal state. We denote the fixed internal state as ω_f, and let $P_{S_{\boldsymbol{x}},S_{\boldsymbol{y}}}$ denote the law of the transcript released by the API under $S_{\boldsymbol{x}}$ and $S_{\boldsymbol{y}}$ **condition on fixed** ω_f. We formalize privacy for LLM inference as follows:

Definition 3.1 (Infer-RDP) *For $\alpha > 1$, the deployment satisfies (α, ε)-Infer-RDP if for every adjacent pair $(S_{\boldsymbol{x}}, S_{\boldsymbol{y}}) \sim (S'_{\boldsymbol{x}}, S'_{\boldsymbol{y}})$, we have $D_\alpha\left(P_{S_{\boldsymbol{x}},S_{\boldsymbol{y}}} \middle\| P_{S'_{\boldsymbol{x}},S'_{\boldsymbol{y}}}\right) \leq \varepsilon$ where the adjacent pair indicates that dataset $(S_{\boldsymbol{x}}, S_{\boldsymbol{y}})$ and $(S'_{\boldsymbol{x}}, S'_{\boldsymbol{y}})$ differ on a single element.*

As shown in Figure 1, DP-LAIR augments the private and public backbones with an encoder–decoder and a public-only router. We write the fixed internal state for DP-LAIR as $\omega_f = \{\boldsymbol{\theta}_{\text{priv}}, \boldsymbol{\theta}_{\text{pub}}, \phi_{\text{enc}}, \Sigma\}$ and the DP-trained interface state for DP-LAIR as $\omega_d = \{\psi_{\text{c}}, \phi_{\text{dec}}\}$. To protect $S_{\boldsymbol{x}}$, the router and decoder are trained jointly with a DP-guaranteed algorithm. To protect the query-specific $S_{\boldsymbol{y}}$, the encoder compresses and clips its contribution to the private hidden-state residual, Gaussian noise is added in the latent space, and the decoder maps the perturbed latent back to the hidden space. The router uses only public features, and the budget gate controls when this private path may be used. The remaining deployment configurations are fixed, and all states remain internal to the server. Thus, we have the following lemma:

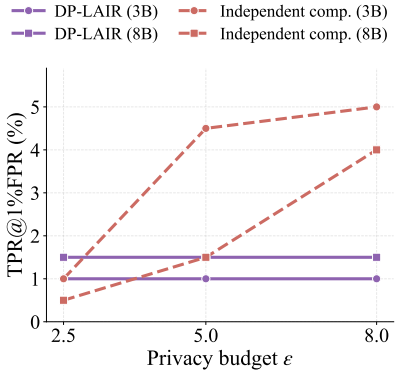


Figure 2: TPR@1% FPR of MIA on TOFU across training privacy budgets

Lemma 3.2 (Infer-RDP decomposition) *Fix $\alpha > 1$ and **a deployment in which ω_f is shared and fixed**. Suppose that $S_{\boldsymbol{x}}$ affects the released transcript only through ω_d. Let $\varepsilon_{\omega_d}(\alpha)$ be the RDP cost*

of ω_d with respect to $S_{\boldsymbol{x}}$, and let $\varepsilon_{\boldsymbol{y}}(\alpha)$ be the RDP cost of online generation. Then the resulting Infer-RDP cost satisfies

$$\varepsilon_{\text{total}}(\alpha) \leq \varepsilon_{\omega_d}(\alpha) + \varepsilon_{\boldsymbol{y}}(\alpha). \tag{1}$$

Lemma 3.2 separates the total privacy loss into two parts: ε_{ω_d} is fixed once the model is deployed, while the runtime cost varies across queries. In independent composition, training and runtime privacy are handled separately, so the runtime mechanism cannot offset a weak training-side privacy setting. DP-LAIR instead combines both protections in the same private path, where the DP-trained router and decoder work together with runtime latent perturbation to determine the released output. Figure 2 shows how the two methods respond to different values of ε_{ω_d}. We consider the small ε_{ω_d} setting in this paper. Choosing the training and runtime privacy budgets jointly would require an additional budget-allocation objective, which we leave for future work.

3.2 DP-LAIR Runtime Prompt Protection

DP-LAIR generates each token through either a public or private path. At step t, let h_t^{pub} be the public hidden state and $h_{xy}^{\text{priv}(t)}(S_{\boldsymbol{y}})$ the private hidden state conditioned on $S_{\boldsymbol{y}}$. The private context reaches the output only through a noisy latent channel. The public path uses h_t^{pub}, which does not depend on $S_{\boldsymbol{y}}$. The private path computes the residual $\Delta h_t(S_{\boldsymbol{y}}) = h_{xy}^{\text{priv}(t)}(S_{\boldsymbol{y}}) - h_t^{\text{pub}}$, encodes it by $f_{\phi_{\text{enc}}}$, clips each coordinate to $[-C, C]$, adds Gaussian noise $\boldsymbol{\xi}_t \sim \mathcal{N}(0, \Sigma)$, and decodes it by $f_{\phi_{\text{dec}}}$. Routing uses only the public hidden states, and the router proposes the private path when its score exceeds a threshold b fixed for the deployment; $g_t \in \{0, 1\}$ denotes this proposal. The selected hidden state h_t is passed through the shared LM head to produce the released output.

The runtime budget gate then checks each private proposal. The server computes the realized order-α LOO charge ρ_t from the current context and all of its one-record-null counterparts, as defined in Equation (8). Let R_{t-1} be the accumulated charge and B the runtime RDP budget. For clarity, we omit the RDP-order dependence of privacy-budget-related quantities, such as B and R_t, when it is clear from context. A private proposal ($g_t = 1$) is accepted only when its charge fits the remaining budget:

$$h_t = \begin{cases} h_t^{\text{pub}}, & g_t = 0 \ \vee \ R_{t-1} + \rho_t > B, \\ h_t^{\text{pub}} + f_{\phi_{\text{dec}}}(\text{Clip}(f_{\phi_{\text{enc}}}(\Delta h_t(S_{\boldsymbol{y}})), C) + \boldsymbol{\xi}_t), & g_t = 1 \ \wedge \ R_{t-1} + \rho_t \leq B, \end{cases} \tag{2}$$

Let $R_0 = 0$, and let a_t indicate whether the second branch of Equation (2) is used. The runtime budget then evolves as

$$R_t = R_{t-1} + a_t \, \rho_t. \tag{3}$$

Rejected private proposals incur no privacy cost, and generation continues until EOS.

3.3 Training DP-LAIR Components

DP-LAIR training has two stages. Stage I jointly learns a router, an encoder, a decoder, and a latent covariance through a constrained optimization problem. Stage II carries forward and freezes the encoder and covariance, freshly initializes the router and decoder, and jointly retrains this deployed pair with DPDT. Both stages use examples constructed from $S_{\boldsymbol{x}}$; the DPDT guarantee protects the Stage-II use of each record in the downstream corpus.

3.3.1 Joint Inference Pretraining

Training objective. Stage I jointly trains the router and latent mechanism to preserve generation utility while limiting the estimated cost of using the private path. Combining these two objectives in a weighted loss requires a fixed coefficient to balance utility and privacy cost, which makes the optimization sensitive to the coefficient choice and difficult to tune. Thus, we build a constraint optimization problem, where the desired cost scale as a constraint and use primal dual updates to adapt its training pressure. Let U_i denote the utility loss of training instance i and R_i its privacy-cost surrogate, both defined below. The training problem is

$$\min_{\psi_c, \phi_{\text{enc}}, \phi_{\text{dec}}, \Sigma} \mathbb{E}_i[U_i] \quad \text{s.t.} \quad R_i \leq \overline{R} \quad \text{for every training instance } i \tag{4}$$

Here, $\overline{R}$ is a training hyperparameter, not a certified privacy budget. It controls how strongly training penalizes examples with a large surrogate cost R_i. Because R_i is already small for most examples, changing $\overline{R}$ within the tested range has little effect on utility, as shown in ablation experiment in Appendix E.4. We define the terms in this optimization problem below:

Training examples. The private and public backbones and the LM head remain fixed. From $S_{\boldsymbol{x}}$, we construct context variants that mimic the query time private prompt. Each training instance i consists of a target question and answer together with one context variant $\widetilde{S_{\boldsymbol{y}_i}}$ sampled from the corresponding target-question group. Let $\mathcal{T}_i$ denote its valid answer-prediction positions. At position t, $h_{i,t}^{\text{priv}}$ is the private-model state computed with $\widetilde{S_{\boldsymbol{y}_i}}$, and $h_{i,t}^{\text{pub}}$ is the corresponding public-model state. Appendix B.2 describes the group and variant construction.

Utility loss. For each training instance, we define the utility los as a soft mixture of public- and private-path token losses:

$$U_i = \frac{1}{|\mathcal{T}_i|} \sum_{t \in \mathcal{T}_i} \left[(1 - q_{i,t}) \ell_{i,t}^{\text{pub}} + q_{i,t} \ell_{i,t}^{\text{priv}} \right]$$

Here $\ell_{i,t}^{\text{pub}}$ and $\ell_{i,t}^{\text{priv}}$ are the two path losses, and $q_{i,t}$ is the soft routing weight. Let $y_{i,t}$ denote the target answer token at position t. The public path uses the fixed public hidden state, so $\ell_{i,t}^{\text{pub}} = \text{CE}(W_{\text{LM}} h_{i,t}^{\text{pub}}, y_{i,t})$. For $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_r^2)$ with $\sigma_d > 0$, the private path applies the encoder–noise–decoder transformation to the full-context residual:

$$\tilde{h}_{i,t} = h_{i,t}^{\text{pub}} + f_{\phi_{\text{dec}}} \left(\text{clip}_{[-C,C]} \left(f_{\phi_{\text{enc}}} (h_{i,t}^{\text{priv}} - h_{i,t}^{\text{pub}}) \right) + \boldsymbol{\xi}_{i,t} \right), \qquad \boldsymbol{\xi}_{i,t} \sim \mathcal{N}(0, \Sigma)$$

Its expected token loss is $\ell_{i,t}^{\text{priv}} = \mathbb{E}_{\boldsymbol{\xi}} \left[\text{CE}(W_{\text{LM}} \tilde{h}_{i,t}, y_{i,t}) \right]$. During training, we use its reparameterized sample estimate, with one Gaussian sample per forward pass in our experiments. Finally, We can compute soft routing weight by $q_{i,t} = \varphi(f_{\psi_c}(h_{i,t}^{\text{pub}})$ where φ the sigmoid function. Specifically, $q_{i,t}$ does not depend on the private context or its privacy cost.

Privacy-cost surrogate. To assign a cost to the private path at position t, we compare its clipped latent under the full context with those under all one-record-null contexts. Let $\mathcal{J}_i$ index the records in $\widetilde{S_{\boldsymbol{y}_i}}$, and let $\widetilde{S_{\boldsymbol{y}_i}}^{(-j)}$ replace record j with the null record. Write $h_{i,t}^{\text{priv},(-j)}$ for the private-model state computed with $\widetilde{S_{\boldsymbol{y}_i}}^{(-j)}$; the public state $h_{i,t}^{\text{pub}}$ is unchanged. The corresponding latents are

$$\hat{z}_{i,t} = \text{clip}_{[-C,C]} \left(f_{\phi_{\text{enc}}} (h_{i,t}^{\text{priv}} - h_{i,t}^{\text{pub}}) \right), \qquad \hat{z}_{i,t}^{(-j)} = \text{clip}_{[-C,C]} \left(f_{\phi_{\text{enc}}} (h_{i,t}^{\text{priv},(-j)} - h_{i,t}^{\text{pub}}) \right)$$

With covariance Σ, the maximum displacement over these LOO counterparts gives the realized order-α RDP charge of the latent Gaussian mechanism:

$$\Delta_{i,t}^{\text{LOO}} := \max_{j \in \mathcal{J}_i} \left\| \hat{z}_{i,t} - \hat{z}_{i,t}^{(-j)} \right\|_{\Sigma^{-1}}, \qquad \rho_{i,t} := \frac{\alpha}{2} \left(\Delta_{i,t}^{\text{LOO}} \right)^2$$

By post-processing, $\rho_{i,t}$ also bounds the released private-path output. Weighting this charge by the soft routing weight gives the Stage-I per-example surrogate

$$R_i = \sum_{t \in \mathcal{T}_i} q_{i,t} \rho_{i,t} \tag{5}$$

The training constraint instead targets $R_i \leq \overline{R}$ for every training instance.

Primal dual optimization. We solve Equation (4) by primal dual optimization. Each training instance i has a persistent nonnegative multiplier λ_i for its constraint. In the implementation, an instance is identified by its target group and context variant, $i = (g, k)$. For a minibatch $\mathcal{I}$, the Lagrangian is $\mathcal{L}_{\text{pre}} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \left[U_i + \lambda_i \left(\frac{R_i(\alpha)}{\overline{R}(\alpha)} - 1 \right) \right]$. Let$\theta_{\text{pre}} = (\psi_c, \phi_{\text{enc}}, \phi_{\text{dec}}, \Sigma)$.Holding the multipliers fixed, the primal step minimizes the Lagrangian. A gradient step is written as

$$\theta_{\text{pre}} \leftarrow \theta_{\text{pre}} - \eta_\theta \nabla_{\theta_{\text{pre}}} \mathcal{L}_{\text{pre}}. \tag{6}$$

The dual step then updates each multiplier by projected ascent on its constraint violation:

$$\lambda_i \leftarrow \max\left\{0, \lambda_i + \eta_\lambda \left(\frac{R_i(\alpha)}{\overline{R}(\alpha)} - 1\right)\right\}, \qquad i \in \mathcal{I}. \tag{7}$$

The update increases λ_i when the constraint is violated and decreases it when the cost is below the target, thereby adapting the penalty on R_i. Algorithm 1 summarizes this alternating procedure.

Algorithm 1 Stage-I primal-dual optimization for Equation (4)

Require: Training instances constructed from $S_{\boldsymbol{x}}$, RDP target $\overline{R}(\alpha) > 0$, learning rates $\eta_\theta, \eta_\lambda$, router temperature τ_r
Ensure: Parameters $(\psi_{\mathrm{c}}, \phi_{\mathrm{enc}}, \phi_{\mathrm{dec}}, \Sigma)$
  Initialize the parameters and $\lambda_i \leftarrow 0$ for every training instance $i = (g, k)$
  for each minibatch $\mathcal{I}$ **do**
    Compute U_i, $R_i(\alpha)$, and the detached constraint violation for each $i \in \mathcal{I}$
    Update $(\psi_{\mathrm{c}}, \phi_{\mathrm{enc}}, \phi_{\mathrm{dec}}, \Sigma)$ using the primal step in Equation (6)
    Update each λ_i from the detached violation using Equation (7)
  end for

The encoder and covariance learned here are passed to Stage II; the Stage-I router and decoder are used only to provide their training signal.

3.3.2 Training Data Protection

Stage II freezes $\boldsymbol{\theta}_{\mathrm{priv}}$, $\boldsymbol{\theta}_{\mathrm{pub}}$, ϕ_{enc}, Σ, and the remaining deployment settings. we reinitialize ψ_{c} and ϕ_{dec} and jointly retrains them with Differentially Private Decoupled Training (DPDT) (Chen et al., 2026). The retraining uses the same examples, U_i, R_i, and training target $\overline{R}$ as Stage I.

DPDT maintains K workers, each of which performs local updates. At each iteration, the global solver aggregates the workers' parameter states and adds Gaussian noise to the aggregated update. The resulting DP-protected parameters are then sent back to the workers for the next round of local updates. This update noise protects Stage-II training and is separate from the online latent noise $\boldsymbol{\xi}_t$ used during inference. Only the final state $\omega_d = \{\psi_{\mathrm{c}}, \phi_{\mathrm{dec}}\}$ is retained for deployment; worker states, intermediate updates, optimizer states, and diagnostics remain internal.

DPDT's privacy accounting relies on the hidden-state assumption, which matches our API-only threat model: the adversary observes the API transcript but not the training trajectory or deployed parameters. Unlike DP-SGD, DPDT's privacy loss does not scale proportionally with the number of training iterations. Instead, it is sensitive to the size of the privatized parameter state. Thus, DPDT provides a better privacy utility trade-off when a small set of parameters needs to be protected, which is important for DP-LAIR because only the router and decoder are privately trained, while both backbones, the encoder, and the covariance remain fixed. DPDT's privacy analysis gives an RDP guarantee for the final deployed state $\omega_d = \{\psi_{\mathrm{c}}, \phi_{\mathrm{dec}}\}$, yielding the one-time parameter cost $\varepsilon_{\omega_d}(\alpha)$ used in Lemma 3.2. This cost is incurred once per deployment.

4 Privacy Analysis

4.1 LOO Privacy Loss for Prompt Protection

For each realized query, we evaluate privacy with respect to the records in its current context. Let $S_{\boldsymbol{y}}$ contain m records. We define its LOO neighbors as $\{S_{\boldsymbol{y}}^{(-j)}\}_{j=1}^{m}$, where $S_{\boldsymbol{y}}^{(-j)}$ is obtained by replacing record j with null record(denote as $\varnothing$). Under our API-only threat model, the adversary observes only the transcript released for the current API call. Accordingly, we define the adjacent contexts of $S_{\boldsymbol{y}}$ as its one-record-null neighbors $\{S_{\boldsymbol{y}}^{(-j)}\}_{j=1}^{m}$, rather than constructing additional LOO sets around each neighbor. Specifically, at token t, let $\hat{z}_t(S_{\boldsymbol{y}})$ denote the clipped latent in the private path of Equation (2). The order-α RDP cost of using the private path is

$$\rho_t(\alpha; S_{\boldsymbol{y}}) := \max_{j \in [m]} \frac{\alpha}{2} \left\| \hat{z}_t(S_{\boldsymbol{y}}) - \hat{z}_t(S_{\boldsymbol{y}}^{(-j)}) \right\|_{\Sigma^{-1}}^2 \le 2\alpha C^2 \operatorname{tr}(\Sigma^{-1}) \tag{8}$$

$\rho_t(\alpha; S_{\boldsymbol{y}})$ is the realized order α RDP loss at token t. It may vary with the released prefix, while the clipping bound gives a uniform upper bound. We next integrate the adaptive LOO accounting to Infer-RDP privacy loss.

4.2 RÉNYI PRIVACY FILTER

Let $\boldsymbol{u}_t$ denote the API observation at step t, including termination, and let T be the transcript length. We assume that $T < \infty$ almost surely. At each step, the charged RDP cost is $\tilde{\rho}_t = a_t \rho_t$, where $a_t = 1$ if the private path is used and 0 otherwise. Thus, a private step incurs cost ρ_t, while a public step incurs zero cost.

The budget update in Equation (3) ensures that $\sum_{t=1}^{T} \tilde{\rho}_t = R_T \leq B$. Both the routing decision and ρ_t may depend on previously released tokens. Rényi privacy filters allow such adaptive dependence as long as the accumulated RDP cost remains within the budget (Feldman & Zrnic, 2021). Therefore, the runtime prompt privacy loss satisfies $\varepsilon_{\boldsymbol{y}}(\alpha) \leq B$. Combining this runtime bound with Lemma 3.2 gives the total Infer-RDP guarantee.

Theorem 4.1 (Infer-RDP Privacy Filter) *Consider a fixed deployment and an order $\alpha > 1$. Suppose that ρ_t bounds the order-α RDP cost whenever the private path is used. Under the budget update in Equation (3) with LOO privacy loss(Equation (8)), the released transcript satisfies*

$$\varepsilon_{\text{total}}(\alpha) \leq \varepsilon_{\omega_d}(\alpha) + B. \tag{9}$$

Proof sketch. A private step contributes at most ρ_t, while a public step contributes zero because it does not depend on the protected runtime record. The privacy filter therefore gives $\varepsilon_{\boldsymbol{y}}(\alpha) \leq B$. Lemma 3.2 then gives Equation (9). The complete proof is given in Appendix A.

The bound contains two privacy costs. $\varepsilon_{\omega_d}(\alpha)$ is the one-time cost of training the deployed router and decoder, while B bounds the privacy loss from the runtime context for each query.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Models and datasets. We evaluate DP-LAIR on the TOFU(Maini et al., 2024) and MUSE(Shi et al., 2025) datasets, two widely used benchmarks in the machine unlearning community. Both are standard evaluation suites whose controlled splits support our leakage-auditing setup. Specifically, we extract keywords such as names and locations from the records' answers and use the full answer sentences containing these keywords as reference for privacy-leakage detection. We run experiments on the Llama and Qwen models of different sizes.

Tasks and methods. We evaluate both joint protection and training-data protection. For joint protection (Section 5.2), private context $S_{\boldsymbol{y}}$ is provided to a private model trained on $S_{\boldsymbol{x}}$, so the output may depend on both private sources. We also evaluate utility on public in-context QA. For training-data protection (Section 5.3), we query records from $S_{\boldsymbol{x}}$ without supplying their sensitive information in the prompt and additionally report membership-inference results.

For joint protection, we compare *No protection*, *Public*, *Composition*, and *DP-LAIR*. Composition combines InvisibleInk (Vinod et al., 2026) for runtime prompt protection with DP-LoRA for training data protection. For training-data protection, we compare *No protection*, DP-LoRA, and DP-LAIR. Additional details are provided in Appendix C.

Metrics and privacy budgets. We measure sensitive-sentence reproduction using stemmed ROUGE-L F1. Leakage@0.5 and Leakage@0.7 report the percentage of queries with scores of at least 0.5 and 0.7, respectively, against their sensitive reference sentences; these are empirical reproduction criteria rather than DP guarantees. We measure utility using perplexity (PPL) on target answers for both private queries and public in-context QA. Following the allocation study in Figure 2, we fix the training-data privacy loss at 2.5 and the prompt privacy loss at 6.5 for all joint-protection methods. At fixed δ, we report DP-LAIR's ε as query average (maximum). Further details on evaluation and privacy accounting are provided in Appendix C.

5.2 JOINT PROTECTION AND UTILITY

Table 1 reports *No protection* and *Public* as two reference endpoints: the former uses the full private prompt, while the latter removes private context entirely. Among joint-protection methods, DP-LAIR consistently improves over Composition, reducing sensitive sentence leakage and PPL while maintaining comparable utility on public QA. This indicates a better privacy utility trade-off. We also provide the generation example in Appendix E. For privacy accounting, the deployment guarantee is determined by the maximum query-level privacy loss(7.3), while DP-LAIR exhibits a substantially lower average loss listed in Table 1, showing that many queries consume less privacy budget than the worst-case bound. This gap suggests further room for improving query-specific budget allocation. We further discuss this part in Section 6

Table 1: Joint leakage and utility under protection of both training data and runtime private context at fixed δ. For DP-LAIR, ε is reported as the exact average (ε, δ) privacy loss of each query

Dataset	Method	**TOFU**					**MUSE News**				
		ε	Leakage (%) ↓		PPL ↓		ε	Leakage (%) ↓		PPL ↓	
			@0.5	@0.7	Joint	Public		@0.5	@0.7	Joint	Public
Llama 3B	No protection	∞	97.5%	88.5%	1.06	1.36	∞	99.0%	99.0%	1.01	1.38
	Public	0	5.0%	0.5%	7.88	1.36	0	24.5%	10.0%	3.19	1.36
	Composition	7.3	8.0%	1.5%	8.39	1.39	7.3	46.5%	24.0%	3.21	1.39
	DP-LAIR	4.4†	4.50%	1.00%	6.65	1.38	6.2†	21.0%	7.5%	2.65	1.47
Llama 8B	No protection	∞	99.0%	97.0%	1.02	1.36	∞	99.5%	99.5%	1.01	1.34
	Public	0	8.0%	2.5%	7.00	1.36	0	27.0%	13.0%	2.70	1.37
	Composition	7.3	3.5%	1.5%	7.43	1.38	7.3	31.0%	16.0%	2.96	1.34
	DP-LAIR	4.2†	2.50%	0.4%	6.33	1.37	6.5†	13.0%	7.0%	2.28	1.37
Qwen 3B	No protection	∞	83.5%	64.0%	1.13	1.51	∞	98.5%	94.5%	1.03	1.46
	Public	0	0.0%	0.0%	8.88	1.57	0	3.5%	1.0%	3.81	1.54
	Composition	7.3	5.0%	0.0%	9.15	1.51	7.3	26.5%	16.5%	3.75	1.50
	DP-LAIR	4.1†	0.5%	0.0%	8.02	1.56	6.6†	7.0%	1.5%	3.23	1.59
Qwen 7B	No protection	∞	85.5%	64.0%	1.09	1.34	∞	98.0%	97.0%	1.01	1.32
	Public	0	5.0%	0.0%	9.07	1.46	0	2.0%	0.0%	1.01	1.34
	Composition	7.3	7.5%	1.0%	8.95	1.44	7.3	33.5%	15.0%	4.25	1.46
	DP-LAIR	7.1†	1.0%	0.0%	8.61	1.48	5.8†	4.0%	0.0%	2.80	1.46

† For DP-LAIR, the reported ϵ is the realized average privacy loss across queries; the maximum privacy loss is bounded by $\epsilon = 7.3$.

5.3 TRAINING-DATA PRIVACY

Table 2 reports two complementary tests on the deployed models: reproduction of sensitive training-data sentences under context-free queries, and membership inference under API-only access. We report the best performance of five membership-attack method from Dorna et al. (2026). Detailed result deferred in Appendix C. Specifically, we report the area under the ROC curve (AUC) and the true-positive rate at a false-positive rate of 1% (TPR). DP-LoRA and DP-LAIR uses the same deployment as in Table 1.

The result shows that DP-LAIR can provide comparable or even better protection on DP-LoRA with same privacy loss. Moreover, Table 2 shows that DP-LAIR yields lower reproduction rates than DP-LoRA in all four model–dataset settings, while retaining comparable PPL.

5.4 ABLATION STUDY

We study two deployment factors on Llama 3B and TOFU. First, we jointly vary the Stage-I cost target $\overline{R}$ and the runtime budget B, since the former serves as a training-time surrogate for the latter. Second, we vary the size of the private context. Full settings for both ablations are provided in Appendix E.4.

Table 2: Training-data leakage, answer fit, and membership inference, TPR is measured at 1% FPR

Dataset	Method	Llama 3B					Llama 8B				
		Leakage (%) ↓		PPL ↓	MIA (%)		Leakage (%) ↓		PPL ↓	MIA (%)	
		@0.5	@0.7		AUC	TPR ↓	@0.5	@0.7		AUC	TPR ↓
TOFU	No protection	27.5%	10.5%	3.68	0.63	1.5%	28%	10.0%	3.11	0.69	2%
	DP-LoRA	13.5%	5.5%	6.55	0.52	1.0%	15.0%	7.0%	5.84	0.53	0.5%
	DP-LAIR	4.5%	1.5%	6.23	0.54	1.0%	0.0%	0.0%	5.62	0.55	1.5%
MUSE News	No protection	84.0%	55.5%	2.06	0.53	0.5%	81.5%	60.0%	1.77	0.55	4.0%
	DP-LoRA	73.0%	46.5%	2.69	0.53	0.5%	83.5%	59.5%	2.22	0.53	0.0%
	DP-LAIR	9.0%	3.5%	2.80	0.50	0.0%	5.5%	0.0%	2.06	0.58	0.0%

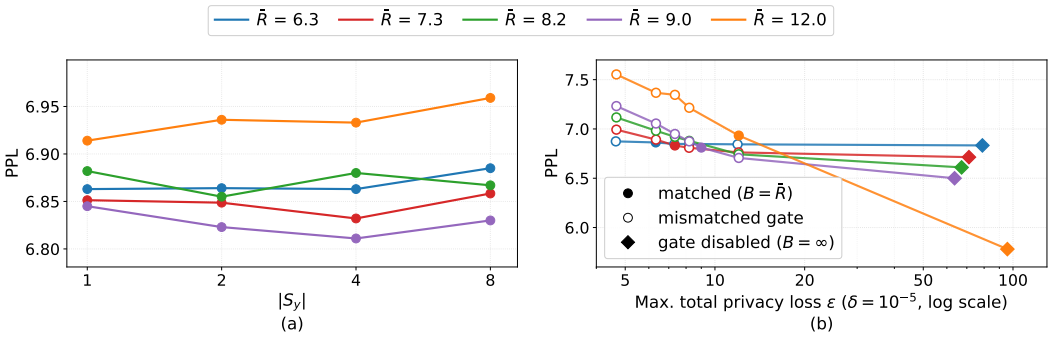


Figure 3: Ablations on Llama 3B and TOFU. **(a)** PPL across prompt sizes with $B = \overline{R}$ **(b)** PPL across runtime privacy budgets for checkpoints trained with different $\overline{R}$.

Private-context size. We evaluate $|S_y| \in \{1, 2, 4, 8\}$ with $B = \overline{R}$. Each prompt contains one target record, with the remaining records sampled as distractors. Figure 3 (a) shows little change in PPL across prompt sizes, indicating that utility is stable over the tested range.

Training target and runtime budget. $\overline{R}$ shapes the privacy utility trade-off during training, whereas B caps the accumulated runtime privacy cost. We train checkpoints with different $\overline{R}$ values and evaluate each one under several runtime budgets.

Figure 3 (b) plots PPL against the resulting maximum total privacy loss. Smaller training targets perform better under tight budgets, while larger targets achieve better utility when more privacy loss is allowed. This pattern is consistent with the training objective: a smaller $\overline{R}$ imposes a stronger penalty on the runtime-cost surrogate, making the checkpoint better suited to limited budgets, while a larger target relaxes this constraint and preserves more utility when greater privacy loss is allowed.

6 Conclusion and Future Work

We introduced DP-LAIR for jointly protecting downstream training data and query-time private context under an API-only threat model. DP-LAIR combines public routing, latent perturbation, DP retraining of the deployed interface, and runtime privacy filtering within one inference pipeline. This yields an Infer-RDP guarantee that accounts for both training data and prompt privacy. Across the evaluated models and datasets, DP-LAIR reduces leakage and improves private-query utility over composing separate training- and prompt-protection mechanisms, while preserving comparable utility on public inputs.

The current design has two main limitations. First, the average query-level privacy loss is often much smaller than the maximum, leaving room for more efficient runtime budget allocation. How to allocate privacy cost more selectively across tokens while preserving the same worst-case guarantee remains an open question, which we leave for future work. Second, training-data protection still requires separate DP retraining with a predefined privacy budget. Integrating this cost into the joint optimization, while jointly determining the training and runtime budgets and reducing the need for separate retraining, is another open problem.

AI USE STATEMENT

In this work, we used generative AI tools for aiding or polishing writing, implementing methods, summarizing or analyzing existing literature, editing software code. All AI-generated feedback on the scientific content was independently evaluated and verified by the authors, and all final scientific judgments were made by the authors. We have not used generative AI tools for proposing or refining hypotheses, assisting in the writing of proofs, assisting with translation, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, or supporting qualitative and thematic data analysis. Generating synthetic data sets, cleaning and reformatting datasets, and transcribing recordings of research material are not applicable to this work.

We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Jason Altschuler and Kunal Talwar. Privacy of noisy stochastic gradient descent: More iterations without more privacy loss. *Advances in Neural Information Processing Systems*, 35:3788–3800, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Differentially private optimization on large model at small cost. In *International Conference on Machine Learning*, pp. 3192–3218. PMLR, 2023.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Ding Chen, Haochen Luo, Xiaofei Wang, and Chen Liu. Differentially private neural network training under the hidden state assumption, 2026. URL https://arxiv.org/abs/2407.08233.

Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zico Kolter, Zachary Lipton, and Pratyush Maini. Openunlearning: Accelerating llm unlearning via unified benchmarking of methods and metrics. *Advances in Neural Information Processing Systems*, 38, 2026.

Vitaly Feldman and Tijana Zrnic. Individual privacy accounting via a renyi filter. *Advances in Neural Information Processing Systems*, 34:28080–28091, 2021.

Vitaly Feldman, Ilya Mironov, Kunal Talwar, and Abhradeep Thakurta. Privacy amplification by iteration. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 521–532. IEEE, 2018.

James Flemings, Haosheng Gan, Hongyi Li, Meisam Razaviyayn, and Murali Annavaram. Differentially private in-context learning via sampling few-shot mixed with zero-shot outputs. *arXiv preprint arXiv:2501.19287*, 2025.

Timour Igamberdiev, Thomas Arnold, and Ivan Habernal. Dp-rewrite: Towards reproducibility and transparency in differentially private text rewriting. In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 2927–2933, 2022.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

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

John X Morris, Wenting Zhao, Justin Chiu, Vitaly Shmatikov, and Alexander Rush. Language model inversion. In *International Conference on Learning Representations*, volume 2024, pp. 35863–35883, 2024.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. In *International Conference on Learning Representations*, volume 2025, pp. 27797–27818, 2025.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Rushil Thareja, Preslav Nakov, Praneeth Vepakomma, and Nils Lukas. Dp-fusion: Token-level differentially private inference for large language models. *arXiv preprint arXiv:2507.04531*, 2025.

Meng Tong, Kejiang Chen, Jie Zhang, Yuang Qi, Weiming Zhang, Nenghai Yu, Tianwei Zhang, and Zhikun Zhang. Inferdpt: Privacy-preserving inference for black-box large language models. *IEEE Transactions on Dependable and Secure Computing*, 2025.

Vishnu Vinod, Krishna Pillutla, and Abhradeep Guha Thakurta. Invisibleink: High-utility and low-cost text generation with differential privacy. *Advances in Neural Information Processing Systems*, 38:67254–67298, 2026.

Haoqi Wu, Wei Dai, Li Wang, and Qiang Yan. Cape: Context-aware prompt perturbation mechanism with differential privacy. *arXiv preprint arXiv:2505.05922*, 2025.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Jiayuan Ye and Reza Shokri. Differentially private learning needs hidden state (or much faster convergence). *Advances in Neural Information Processing Systems*, 35:703–715, 2022.

A Proofs for Infer-RDP Composition and the Runtime Filter

A.1 Infer-RDP Composition

Proof of Lemma 3.2. Fix $\alpha > 1$ and an adjacent comparison. Conditional on the shared, fixed ω_f, let P and Q denote the joint laws of $(\omega_d, \boldsymbol{u}_{1:T})$ under two adjacent inputs. Here $\boldsymbol{u}_t$ and the almost-surely finite stopping time T are defined as in Section 4.2. For any generated transcript, the assumption that $S_{\boldsymbol{x}}$ affects the transcript only through ω_d gives

$$\Pr[\omega_d, \boldsymbol{u}_{1:T} \mid S_{\boldsymbol{x}}, S_{\boldsymbol{y}}] = \Pr[\omega_d \mid S_{\boldsymbol{x}}] \prod_{t=1}^{T} \Pr[\boldsymbol{u}_t \mid \omega_d, \boldsymbol{u}_{<t}, S_{\boldsymbol{y}}]. \tag{10}$$

The corresponding stopped filtration is

$$\mathcal{F}_t = \sigma\big(\omega_d, \boldsymbol{u}_{1:(t\wedge T)}, \mathbb{1}\{T \le t\}\big), \qquad \mathcal{F}_0 = \sigma(\omega_d).$$

Taking α-th powers of the likelihood ratio in Equation (10) yields

$$L^{\text{total}} = L^{\text{off}}(\omega_d) \prod_{t=1}^{T} L_t^{\text{on}}, \; L^{\text{off}} := \Big(\frac{\Pr[\omega_d \mid S_{\boldsymbol{x}}]}{\Pr[\omega_d \mid S'_{\boldsymbol{x}}]}\Big)^{\alpha}, \; L_t^{\text{on}} := \Big(\frac{\Pr[\boldsymbol{u}_t \mid \omega_d, \boldsymbol{u}_{<t}, S_{\boldsymbol{y}}]}{\Pr[\boldsymbol{u}_t \mid \omega_d, \boldsymbol{u}_{<t}, S'_{\boldsymbol{y}}]}\Big)^{\alpha}. \tag{11}$$

Define the online RDP term as

$$\varepsilon_{\boldsymbol{y}}(\alpha) := \sup_{\omega_d} \frac{1}{\alpha - 1} \log \mathbb{E}_Q \left[\prod_{t=1}^{T} L_t^{\text{on}} \,\middle|\, \omega_d \right]. \tag{12}$$

The assumed parameter RDP bound is

$$D_\alpha(\Pr[\omega_d \mid S_{\boldsymbol{x}}] \parallel \Pr[\omega_d \mid S'_{\boldsymbol{x}}]) \leq \varepsilon_{\omega_d}(\alpha)$$

which implies

$$\mathbb{E}_Q[L^{\text{off}}] \leq e^{(\alpha-1)\varepsilon_{\omega_d}(\alpha)}. \tag{13}$$

Using the tower property in Equation (11), followed by Equation (12) and Equation (13), gives

$$\mathbb{E}_Q[L^{\text{total}}] = \mathbb{E}_Q \left[L^{\text{off}} \mathbb{E}_Q \left[\prod_{t=1}^{T} L_t^{\text{on}} \,\middle|\, \omega_d \right] \right] \leq e^{(\alpha-1)(\varepsilon_{\omega_d}(\alpha)+\varepsilon_{\boldsymbol{y}}(\alpha))}.$$

Therefore,

$$D_\alpha(P \| Q) = \frac{1}{\alpha - 1} \log \mathbb{E}_Q[L^{\text{total}}] \leq \varepsilon_{\omega_d}(\alpha) + \varepsilon_{\boldsymbol{y}}(\alpha).$$

Discarding ω_d is post-processing, so the same bound holds for the released transcript and proves Equation (1). □

A.2 Runtime Privacy Filter

Proof of Theorem 4.1. Fix $\alpha > 1$, a query family $\{S_{\boldsymbol{y}}, S_{\boldsymbol{y}}^{(-1)}, \ldots, S_{\boldsymbol{y}}^{(-m)}\}$, an ordered pair $(S_{\boldsymbol{y}}, S_{\boldsymbol{y}}^{(-j)})$ from this family, and any training adjacency $S_{\boldsymbol{x}} \sim S'_{\boldsymbol{x}}$. Both LOO endpoints use the common family-wide charge in Equation (8). Conditional on the fixed ω_f, let P, Q be the respective joint state–transcript laws of $(\omega_d, \boldsymbol{u}_{1:T})$, with P corresponding to $(S_{\boldsymbol{x}}, S_{\boldsymbol{y}})$ in the numerator. The observation $\boldsymbol{u}_t$ includes the termination event, and T is the resulting stopping time. By assumption, EOS occurs almost surely, so $T < \infty$ almost surely under both laws.

Step 1: privacy decomposition. By Lemma 3.2, it suffices to prove $\varepsilon_{\boldsymbol{y}}(\alpha) \leq B(\alpha)$. We establish the uniform conditional moment bound in Equation (12).

Step 2: conditional one-step bound.

The router proposal is identical at both LOO endpoints because it uses only public features. The realized maximum in Equation (8) is the common charge assigned to the entire query family. The accumulated spend and the gate decision are therefore also identical, so exactly one common branch is used at the two endpoints.

Case (a): the private path is taken ($a_t = 1$, so $\tilde{\rho}_t = \rho_t$). The released mechanism is the latent Gaussian mechanism of Section 4.1 with the shared covariance Σ, evaluated at the clipped mean of the deployed context, followed by the common decoder and LM head, which are post-processing. By Equation (8), its conditional RDP is at most ρ_t for every j and in both orientations, i.e.

$$\mathbb{E}_Q[L_t^{\text{on}} \mid \mathcal{F}_{t-1}] \leq e^{(\alpha-1)\rho_t}.$$

Case (b): the public path is taken ($a_t = 0$, so $\tilde{\rho}_t = 0$). This happens when the router selects the public path, when the budget check fails, or both. The released mechanism is the public mechanism, which does not depend on $S_{\boldsymbol{y}}$. Both endpoints of the LOO pair therefore induce the same released distribution at step t given the same released history,

$$P_t = Q_t, \qquad \text{hence} \qquad L_t^{\text{on}} = 1 \qquad \text{and} \qquad \tilde{\rho}_t = 0.$$

The public mechanism is not required to be differentially private by itself; it is required only to be insensitive to the protected runtime record, which holds because it is computed from the public model and the released history alone.

Both cases are summarized by the single conditional bound

$$\mathbb{E}_Q[L_t^{\text{on}} \mid \mathcal{F}_{t-1}] \leq e^{(\alpha-1)\tilde{\rho}_t}, \tag{14}$$

with equality in Case (b), because $\tilde{\rho}_t$ equals ρ_t when the private path is taken and 0 otherwise. Because $\boldsymbol{u}_T$ contains the EOS event, this bound also covers the terminal step.

Step 3: the filter is pathwise within budget. The budget check admits a private step only if $R_{t-1} + \rho_t \le B(\alpha)$, and charges nothing otherwise. Hence, by induction over t and since $R_0 = 0$,

$$R_T = \sum_{t=1}^{T} \tilde{\rho}_t \le B(\alpha) \qquad \text{almost surely.} \tag{15}$$

A private step that is rejected before release consumes no budget: its cost enters R only through $\tilde{\rho}_t = a_t \rho_t$.

Step 4: stopping-time supermartingale. Conditional on ω_d, define the stopped process, for $n \ge 0$, by

$$M_n = \Big(\prod_{s=1}^{T\wedge n} L_s^{\mathrm{on}}\Big) e^{-(\alpha-1)\sum_{s=1}^{T\wedge n}\tilde{\rho}_s}, \qquad M_0 = 1.$$

Let $A_n = \{T \ge n\} \in \mathcal{F}_{n-1}$. On A_n^c, the process has already stopped and $M_n = M_{n-1}$. On A_n, $\tilde{\rho}_n$ is $\mathcal{F}_{n-1}$-measurable. Thus Equation (14) gives

$$\mathbb{E}_Q[M_n \mid \mathcal{F}_{n-1}] = \mathbb{1}_{A_n^c} M_{n-1} + \mathbb{1}_{A_n} M_{n-1} e^{-(\alpha-1)\tilde{\rho}_n} \mathbb{E}_Q[L_n^{\mathrm{on}} \mid \mathcal{F}_{n-1}] \le M_{n-1}.$$

Hence $(M_n)_{n\ge 0}$ is a nonnegative supermartingale, and for every n,

$$\mathbb{E}_Q[M_n \mid \omega_d] \le M_0 = 1.$$

Because $T < \infty$ almost surely, $M_n \to M_T$ almost surely, where

$$M_T = \Big(\prod_{s=1}^{T} L_s^{\mathrm{on}}\Big) e^{-(\alpha-1)\sum_{s=1}^{T}\tilde{\rho}_s}.$$

Conditional Fatou's lemma therefore yields

$$\mathbb{E}_Q[M_T \mid \omega_d] \le \liminf_{n\to\infty} \mathbb{E}_Q[M_n \mid \omega_d] \le 1.$$

Step 5: unpacking the pathwise budget. By Equation (15), $R_T \le B(\alpha)$ pathwise, hence

$$\prod_{t=1}^{T} L_t^{\mathrm{on}} = M_T\, e^{(\alpha-1)R_T} \le M_T\, e^{(\alpha-1)B(\alpha)}$$

pointwise, and by Equation (A.2)

$$\mathbb{E}_Q\left[\prod_{t=1}^{T} L_t^{\mathrm{on}} \,\middle|\, \omega_d\right] \le e^{(\alpha-1)B(\alpha)}.$$

Step 6: concluding the online bound. The bound in Equation (A.2) holds uniformly over the realized interface state. Taking the supremum in Equation (12) with Lemma 3.2 then gives $\varepsilon_{\mathrm{total}}(\alpha) \le \varepsilon_{\omega_d}(\alpha) + B(\alpha)$. □

Fixed-query accounting. The transcript ends at the almost-surely finite stopping time T, and the budget enters through Equation (15). The maximum computed for the complete query family is the common charge for all of that query's LOO comparisons.

B Architecture and Runtime Details

B.1 Encoder, Decoder, and Public Router

Table 3 summarizes the deployed interface. Here, d_{model} is the backbone hidden size.

Table 3: DP-LAIR interface architecture.

Component	Architecture and operation
Encoder	$d_{\text{model}} \to d_{\text{model}}/2 \to d_{\text{model}}/4 \to 10$; ReLU between linear layers. The input is the private public hidden-state residual.
Latent clipping	Coordinate-wise clipping to $[-0.5, 0.5]$.
Noise covariance	Input-independent diagonal covariance with one learned log variance per latent coordinate. The covariance is learned in Stage I and frozen in Stage II; fresh Gaussian noise is sampled for every accepted private-path token.
Decoder	$10 \to d_{\text{model}}/4 \to d_{\text{model}}/2 \to d_{\text{model}}$, followed by three residual feed-forward blocks. The decoded residual is added to the public hidden state.
Router features	Top-50 public logits and probabilities and the final 64 public hidden-state coordinates.
Router head	A linear map from h_t^{pub} to a scalar routing score, compared with the fixed deployment threshold.

B.2 Training-Example Construction

The corpus is organized into target-question groups. Each training context contains its target question answer record exactly once and three distractor records. We materialize 16 context variants per target group. During LOO construction, one record slot is replaced by the null record Q:\nA: while the record order and target query remain fixed. The loss is evaluated only on answer tokens; prompt-token labels are masked. One context variant is sampled when a target group is used for an update.

B.3 Stage-II DPDT Retraining

Stage II carries forward and freezes the public and private backbones, the Stage-I encoder, and the learned diagonal covariance. The router and decoder are freshly initialized and jointly retrained with DPDT. The privatized state is divided into 13 parameter groups: one router group and 12 decoder groups. The final configuration uses $K = 10$ workers, 10 local steps, batch size 4, 25 epochs, group clipping norm 0.0175, step scale 0.9, and update-noise standard deviation 0.005. Adaptive clipping uses scale 16 for the first 15 epochs and decays to scale 1. The router anchor coefficient is 0.1 for 3B models and 1.0 for 8B models; the corresponding router temperatures are 1 and 2. Only the final router and decoder are retained for deployment.

The accountant evaluates the complete final update rather than an intermediate epoch. For the configurations used in the main experiments, the Stage-II certificate is $\varepsilon_{\text{train}} = 2.5$ at $\delta = 10^{-5}$. Section C.6 describes how this training curve is composed with the runtime cost.

B.4 Runtime Inference

At each token, the router is evaluated before the runtime budget gate. If it proposes the private path, the server computes the realized LOO cost and checks the gate before releasing the next-token distribution. An accepted private token uses fresh Gaussian noise and is then charged to the accumulated RDP cost. A rejected proposal falls back to the public path for that token without consuming budget; routing is evaluated again at the next token. Generation stops at EOS or after 64 tokens.

C Experimental Setup

C.1 Models and Datasets

We evaluate Llama-3.2-3B-Instruct, Llama-3.1-8B-Instruct, Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, and Qwen2.5-7B-Instruct on TOFU and MUSE News. TOFU contains 200 authors with 20 records per author. We train on 18 records per author, producing 3,600 training records, and hold out two records per author. The member pool samples one training question per author, while the non-member pool samples one of the two held-out questions per author. Each pool therefore contains 200 questions.

For MUSE News, QA records are grouped by source article. The training split contains 3,600 QA records, and the member and non-member evaluation pools each contain 200 questions from distinct articles. Member questions come from training articles, whereas non-member questions come from articles excluded from training. Thus, TOFU evaluates record membership within the same author and MUSE evaluates membership across disjoint articles.

C.2 Evaluation Tasks and Prompt Construction

Table 4 summarizes the evaluation tasks. T1 isolates training-data exposure by asking a bare question about a training record. T2 uses a held-out target record in the private context and therefore isolates prompt exposure. We separately measure utility on 104 public in-context QA queries.

Table 4: Evaluation tasks. T1 and T2 use 200 queries per model and dataset; public QA uses 104 queries.

	T1: training	**T2: prompt**	**Public QA**
Target	Training member	Held-out record	Public question
Context	None	Target + 3 held-out distractors	8 public demonstrations
Input view	Bare QA	Private context	Public ICL
Metrics	Leakage, PPL	Leakage, PPL	PPL

The bare-QA template is

```
Q: <target question>
A:
```

The private prompt uses the following template:

```
Known reference QA pairs are as follows:
Q: <reference question 1>
A: <reference answer 1>

Q: <reference question 2>
A: <reference answer 2>

Q: <reference question 3>
A: <reference answer 3>

Q: <reference question 4>
A: <reference answer 4>

Answer the following question:
Q: <target question>
A:
```

The four references consist of the target QA record and three distinct distractors. After excluding the target from the distractor pool, we select the distractors without replacement and shuffle the four records. The target record therefore occurs exactly once and may occupy any of the four fixed slots. Prompts are materialized once and shared by all methods; they are not resampled or reordered during evaluation. A one-record-null neighbor replaces one fixed reference slot with Q:\nA: while preserving the header, the other references and their order, and the final target question.

Sensitive references are complete answer sentences from the original data. Each T2 reference occurs in that query's private context and is used to compute the leakage metrics.

C.3 Baselines

No protection. The non-private downstream LoRA model receives the full input for each task. It provides a reference endpoint without either training-data or prompt protection.

Public. The public backbone does not use the private downstream adapter. On T2 it receives the bare target question and therefore excludes the private context; on public QA it receives the public demonstrations.

DP-LoRA. DP-LoRA applies DP-SGD to the downstream LoRA adapter and protects the training-data channel. The main comparison uses target $\varepsilon = 2.5$ at $\delta = 10^{-5}$. The $\varepsilon = 5$ and 8 checkpoints are used only in the additional MIA analysis.

InvisibleInk. InvisibleInk sanitizes the private context before generation. The main comparison uses $\varepsilon = 7.3$ and $\delta = 10^{-5}$ with a public model as the sanitizer.

Composition. Composition feeds the InvisibleInk-sanitized context to the DP-LoRA model. It therefore combines the training-side DP-LoRA cost with the prompt-side InvisibleInk cost. Both components use the same questions, materialized contexts, answer targets, and generation limit as DP-LAIR.

C.4 Hyperparameters

Table 5 lists the configurations used for the main results.

Table 5: Hyperparameters used in the main experiments.

Setting	Hyperparameters
Models	Llama-3.2-3B-Instruct, Llama-3.1-8B-Instruct, Qwen2.5-3B-Instruct; bfloat16
Data	3,600 training records; 200 queries for T1 and T2; 104 public-QA queries; private-context size 4
LoRA	$r = 16$; targets: q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, and down_proj; dropout $= 0.05$ (non-private) or 0 (DP)
LoRA training	5 epochs; batch size $= 16$ (3B) or 4 (8B); lr $= 5 \times 10^{-5}$; AdamW; weight decay $= 0.01$; warmup ratio $= 0.05$; max length $= 512$; answer-only loss
DP-LoRA	target $\varepsilon = 2.5$; $\delta = 10^{-5}$; per-sample clip $= 1$; Poisson sampling
Stage-I optimization	50 epochs; batch size $= 8$ (3B) or 4 (8B); Adam; lr $= 3 \times 10^{-4}$; router lr $= 3 \times 10^{-5}$; covariance lr $= 3 \times 10^{-4}$; dual lr $= 0.01$; grad clip $= 5$; RDP target $\overline{R}(\alpha) = 0.8\alpha$
Stage-II DPDT	25 epochs; $K = 10$; 10 local steps; batch size $= 4$; 13 groups; group clip $= 0.0175$; update-noise std $= 0.005$; warm scale $= 16$ for 15 epochs; final scale $= 1$; step scale $= 0.9$
Runtime	RDP budget $B(\alpha) = 0.8\alpha$; router threshold $= 0$; public fallback; maximum generation length $= 64$
InvisibleInk	$\varepsilon = 7.3$; $\delta = 10^{-5}$; sanitization/calibration lengths $= 64/72$; top-$k = 50$; temperature $= 1$;
MIA	200 members and 200 non-members; 3 subsamples of 160/160; Min-K fraction $= 0.4$; bootstrap $B = 2000$; TPR at 1% FPR

C.5 Evaluation Metrics

Leakage. Generated responses are truncated at the next question/answer marker. We then compute stemmed ROUGE-L F1 against the sensitive reference sentence. Leakage@0.5 and Leakage@0.7 are the percentages of queries whose scores are at least 0.5 and 0.7, respectively. They measure sentence reproduction and are empirical audits rather than DP guarantees. Generation stops at EOS or after 64 tokens.

PPL. PPL is computed under teacher forcing on gold answer tokens only. Prompt tokens are excluded, and position t conditions on the prompt and gold answer prefix. For query i with answer length T_i, we compute

$$\mathrm{PPL}_i = \exp\left(\frac{1}{T_i}\sum_{t=1}^{T_i}\mathrm{NLL}_{i,t}\right).$$

The reported value is the geometric mean over queries, equivalently the exponential of the query-macro mean NLL. DP-LAIR evaluation executes the deployment router, fresh private-path noise, the runtime budget gate, and public fallback at every answer position.

C.6 Privacy Accounting

Let $r_{q,t} = \frac{1}{2}(\Delta_{q,t}^{\mathrm{LOO}})^2$ be the order-normalized Gaussian cost of an accepted private token and let $r_q = \sum_t r_{q,t}$ be its query-level sum. The corresponding runtime RDP curve is:

$$\varepsilon_{\mathrm{run},q}^{\mathrm{RDP}}(\alpha) = \alpha r_q,$$

and perform all subsequent accounting in RDP. If $\varepsilon_{\mathrm{train}}^{\mathrm{RDP}}(\alpha)$ is the Stage-II RDP curve, the joint curve is

$$\varepsilon_q^{\mathrm{RDP}}(\alpha) = \varepsilon_{\mathrm{train}}^{\mathrm{RDP}}(\alpha) + \varepsilon_{\mathrm{run},q}^{\mathrm{RDP}}(\alpha).$$

The runtime setting labeled 0.8 caps r_q at 0.8 and therefore represents the RDP budget $B(\alpha) = 0.8\alpha$; it is not an (ε, δ) value. We convert the composed RDP curve to (ε, δ)-DP at $\delta = 10^{-5}$ by minimizing over all accountant orders,

$$\varepsilon_q(\delta) = \min_{\alpha>1} \left[\varepsilon_q^{\mathrm{RDP}}(\alpha) + \frac{\log(1/\delta)}{\alpha - 1} \right].$$

The training curve is paid once per deployment. The maximum over queries is the formal deployment value reported in the tables. We additionally report the mean of the individually converted query values to describe typical runtime expenditure; this average does not replace the maximum guarantee.

After conversion, the Stage-II curve alone gives $\varepsilon_{\mathrm{train}}(\delta) = 2.5$ at $\delta = 10^{-5}$. The main DP-LAIR configuration then composes this training curve with the runtime curve 0.8α before optimizing over α. Composition combines the DP-LoRA and InvisibleInk accountants before reporting its total privacy parameter under the table's fixed δ.

D Membership-Inference Evaluation

D.1 Member and Non-Member Pools

MIA uses T1 bare-QA queries, with prompt Q: {question}\nA:, and generates at most 64 tokens. Each cell contains 200 members and 200 non-members. For TOFU, members are training records and non-members are held-out records from the same 200 authors. For MUSE, members come from training articles and non-members come from articles excluded from training. Each pool contains one question per author or article. The same 200 authors occur in both TOFU pools, whereas the MUSE article sets are disjoint. The member and non-member question sets are disjoint, and all methods use the same pools and prompts.

D.2 Attack Scores

We evaluate five MIA attacks: *Loss*, *Min-K*, *Min-K++*, *zlib*, and *ROUGE-L*. *Loss* is the mean teacher-forced token NLL of the gold answer. *Min-K* is the mean of the lowest 40% of token NLLs. *Min-K++* normalizes this value by the fraction of total NLL mass outside the selected tokens. *zlib* is the ratio between the compressed and uncompressed lengths of the generated text. *ROUGE-L* compares the generation with the sensitive reference. Lower loss-family, Min-K, Min-K++, and zlib scores indicate membership; higher ROUGE-L indicates membership.

For each dataset model method budget configuration, we evaluate the five attacks independently. Table 6 reports their individual AUCs. We then select the attack with the largest mean AUC; the headline AUC and TPR@1% FPR in Table 8 are both taken from this selected attack. Thus, "Best AUC" denotes the strongest of the five evaluated attacks, not an average across them. Reference-model attacks are excluded because a matched reference checkpoint is not available under a common construction for both datasets. None of the five attacks uses a learned classifier, shadow model, or calibration split.

D.3 Per-Attack Results

Table 6 reports the AUC of each attack separately. AUC is the mean ± standard deviation over the three 160/160 subsamples.

Table 6: Per-attack MIA AUC across training privacy budgets. Each value is the mean ± standard deviation over three 160/160 subsamples.

Data/model	Method	Train. ε	Loss	Min-K	Min-K++	zlib	ROUGE-L
TOFU 3B	No protection	–	.632±.009	.534±.007	.506±.010	.525±.010	.500±.002
	DP-LoRA	2.5	.521±.009	.517±.013	.497±.012	.510±.012	.461±.020
	DP-LoRA	5	.521±.009	.518±.012	.498±.012	.534±.009	.479±.015
	DP-LoRA	8	.520±.009	.518±.011	.497±.012	.534±.013	.459±.008
	DP-LAIR	2.5	.545±.010	.524±.012	.501±.013	.433±.004	.515±.004
TOFU 8B	No protection	–	.693±.007	.554±.007	.527±.007	.524±.013	.511±.005
	DP-LoRA	2.5	.520±.011	.513±.009	.489±.007	.533±.004	.458±.012
	DP-LoRA	5	.517±.011	.512±.008	.488±.007	.505±.007	.493±.008
	DP-LoRA	8	.514±.011	.512±.007	.487±.007	.527±.010	.502±.003
	DP-LAIR	2.5	.554±.011	.533±.012	.501±.013	.484±.018	.514±.003
MUSE 3B	No protection	–	.504±.006	.535±.011	.511±.011	.497±.013	.494±.008
	DP-LoRA	2.5	.458±.004	.523±.007	.504±.017	.528±.018	.499±.023
	DP-LoRA	5	.459±.004	.522±.007	.502±.017	.520±.017	.495±.017
	DP-LoRA	8	.459±.004	.521±.008	.502±.016	.514±.020	.498±.019
	DP-LAIR	2.5	.494±.005	.516±.011	.500±.012	.500±.018	.504±.016
MUSE 8B	No protection	–	.539±.011	.553±.009	.537±.010	.501±.013	.518±.016
	DP-LoRA	2.5	.474±.011	.528±.006	.518±.017	.499±.012	.534±.015
	DP-LoRA	5	.472±.011	.529±.006	.519±.017	.491±.014	.527±.017
	DP-LoRA	8	.472±.011	.531±.006	.521±.017	.493±.014	.529±.019
	DP-LAIR	2.5	.577±.019	.554±.011	.529±.016	.499±.015	.552±.020

E Additional Experiments

E.1 Generation Example

Table 7 presents a generation example of DP-LAIR under TOFU dataset. The input shows the complete materialized private prompt, including the target record and three distractors. Bold text marks the sensitive name copied by the unprotected model but not output by DP-LAIR. ROUGE-L is computed against the complete private reference. Displayed generations are truncated at sentence boundaries for readability. Specifically, the DP-LAIR contains the similar output format, but provide a different person name, which should be private.

E.2 Complete MIA Results Across Training Budgets

Table 8 reports the complete T1 summary across training privacy budgets. Best AUC is selected over Loss, Min-K, Min-K++, zlib, and ROUGE-L and is reported as the mean ± standard deviation over the three 160/160 subsamples. TPR@1% FPR is computed for the same selected attack on the full 200/200 pools and therefore has no associated standard deviation.

Because T1 contains no private context, the Composition baseline reduces to its DP-LoRA component; the Composition points in Figure 2 therefore use the corresponding DP-LoRA rows.

Table 7: An illustrative TOFU T2 example from Llama 3B

Dataset	Private prompt and generated responses
TOFU	**Private prompt (input).** Known reference QA pairs are as follows: Q: Can you mention one more notable book in the author's bibliography? A: "Sorcerers of the Twilight Abyss" is another notable inclusion in the bibliography of author Albert Sidney Lane. Q: What genre is Moshe Ben-David known for? A: Moshe Ben-David is recognized for his contribution to the genre of Islam. Q: What is the full name of the author born on the 3rd of July, 1944 in Madrid, Spain? A: The author born on the 3rd of July, 1944 in Madrid, Spain is **Carlos Santiago Guerrero**. Q: Who are the parents of Phuong Tran? A: Phuong Tran's father is a bartender, and her mother is an investigative journalist. Answer the following question: Q: What is the full name of the author born on the 3rd of July, 1944 in Madrid, Spain? A:
	Standard Answer "The author born on the 3rd of July, 1944 in Madrid, Spain is **Carlos Santiago Guerrero**." *(ROUGE-L: 1.000)*
	Composition generation (output). "Joan Manuel Sánchez Fernández." *(ROUGE-L: 0.000)*
	DP-LAIR generation (output). "The author's full name is Carlos Ruiz Zafón." *(ROUGE-L: 0.170)*

Table 8: Complete T1 answer-fit and membership-inference results across training privacy budgets. Best AUC is the maximum over Loss, Min-K, Min-K++, zlib, and ROUGE-L

Data/model	Method	Train. ε	PPL ↓	Best AUC ↓	TPR@1% FPR ↓
TOFU 3B	No protection	–	3.68	.632±.009	1.5%
	DP-LoRA	2.5	6.55	.521±.009	1.0%
	DP-LoRA	5	6.37	.534±.009	4.5%
	DP-LoRA	8	6.26	.534±.013	5.0%
	DP-LAIR	2.5	6.23	.545±.010	1.0%
	DP-LAIR	5	6.48	.540±.018	1.5%
	DP-LAIR	8	6.23	.544±.009	1.0%
TOFU 8B	No protection	–	3.11	.693±.007	2.0%
	DP-LoRA	2.5	5.84	.533±.004	0.5%
	DP-LoRA	5	5.71	.517±.011	1.5%
	DP-LoRA	8	5.60	.527±.010	4.0%
	DP-LAIR	2.5	5.62	.554±.011	1.5%
	DP-LAIR	5	5.69	.560±.011	1.5%
	DP-LAIR	8	5.55	.563±.010	1.5%

Figure 2 visualizes the TOFU TPR@1% FPR results from Table 8 for Llama 3B and Llama 8B. Each cell uses the same 200 members and 200 same-author non-members described in Section D. The horizontal axis contains the training budgets $\varepsilon \in \{2.5, 5, 8\}$, and the vertical axis is TPR@1% FPR. The independent-composition curve uses the corresponding DP-LoRA training checkpoint together with the fixed prompt-protection mechanism. Because T1 contains no private prompt, its membership signal is determined by the training-side component.

E.3 Private-Context Size

Figure 3(a) fixes the Llama 3B TOFU checkpoint and runtime filter to the matching RDP curves $\overline{R} = B$. We evaluate $|S_{\boldsymbol{y}}| \in \{1, 2, 4, 8\}$, corresponding to zero, one, three, or seven distractors in addition to the target record. The same 200 target questions are used at every size. Distractors are materialized once for each size; they are not resampled during evaluation.

E.4 Training Target and Runtime Budget

Figure 3(b) uses Llama 3B on TOFU T2 with the same 200 held-out targets at every operating point. We train checkpoints with different Stage-I cost targets $\overline{R}$ and evaluate each checkpoint under different runtime budgets B. Both are RDP quantities at order α. The matched setting uses $B(\alpha) = \overline{R}(\alpha)$, while the mismatched settings vary $B(\alpha)$ without changing the checkpoint; setting $B = \infty$ disables the runtime budget gate. The horizontal axis reports the resulting maximum total ε after composing the Stage-II RDP curve with the runtime RDP cost and converting the result to (ε, δ)-DP at $\delta = 10^{-5}$. The vertical axis reports PPL.