ICL-PRIVEVAL: A DATA-DEPENDENT FRAMEWORK FOR PRIVACY LEAKAGE IN IN-CONTEXT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

While in-context Learning (ICL) uses exemplars in the prompt to adapt pre-trained Large Language Models (LLMs) to downstream tasks without fine-tuning, these exemplars can carry sensitive information and be potentially leaked from the LLM's output. While prior works have explored this leakage through Membership-Inference Attacks (MIAs), there lacks a systematic evaluation of the factors that contribute to unintended privacy leakage from ICL. To this end, we propose `ICL-PrivEval`, an evaluation workflow for measuring this leakage. At its foundation is `ICLInf`, an exemplar-level leakage metric inspired by data-dependent differential privacy and counterfactual influence, which captures how the LLM's answer distribution shifts when each exemplar is removed. Using `ICL-PrivEval`, we study factors that affect ICL leakage and apply `ICLInf` as a privacy auditor, yielding an effective audit of sampling-based DP-ICL methods up to $\varepsilon = 10$.

1 INTRODUCTION

Large Language Models (LLMs) have been shown to adapt to downstream tasks without fine-tuning by leveraging task-relevant exemplars in the prompt (Brown et al., 2020; Min et al., 2022). This approach, known as In-Context Learning (ICL), avoids modifying the model parameters and provides a cost-effective alternative to fine-tuning. However, ICL raises concerns about unintentional privacy leakage of the exemplars, since LLMs willingly regurgitate prompt data (Wang et al., 2023). Differentially private (DP) ICL methods have been proposed mainly towards either answer privatization (Wu et al., 2024), or private synthetic data generation (Tang et al., 2024; Amin et al., 2024).

Existing evaluations of how much in-context exemplars leak rely mostly on attacks. The most common approaches are membership-inference attacks (MIAs) (Shokri et al., 2017), which decide whether a record was used by a model by exploiting the model's higher confidence on seen records. Adapted to ICL, they infer whether an exemplar appears in the prompt (Duan et al., 2023; Wen et al., 2024). Canary-based audits build on the same test, inserting a known exemplar (a *canary*) and detecting its presence (Choi et al.; Xia et al., 2026). However, these evaluations share two limitations. First, they measure only the leakage that a particular attacker can exploit, so a weak attack may fall to random guessing and understate the risk (Wen et al., 2024). Moreover, they return a relatively coarse verdict, including membership decision or a score on the mechanism, without detailed measurement on the actual exemplars. It remains missing a comprehensive evaluation of how much a deployed prompt actually leaks on the exemplar-level.

To close this gap, we propose `ICLInf`, an exemplar-focused privacy leakage metric inspired by data-dependent DP (Papernot et al., 2017; 2018) and counterfactual influence (Zhang et al., 2023). Since ICL exemplars can be removed without any cost of retraining, `ICLInf` compares the model's answer distribution with and without each exemplar and reports the largest log-ratio across answers, so a large value indicates that an exemplar's presence can be inferred from the model's output. The result measures a quantitative exemplar-level privacy loss that does not depend on the strength of any attack and can be computed directly from the model's outputs. This is aligned with prior work that also utilized data-dependent DP for analyzing the privacy leakage of context data augmented to prompts (Flemings et al., 2025).

We then introduce `ICL-PrivEval` as a comprehensive evaluation on ICL privacy leakage. Beyond a leakage score, a practitioner also needs to know where the leakage comes from and whether

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.

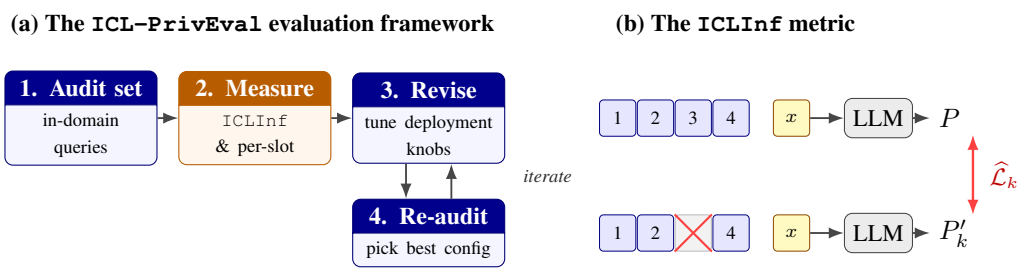


Figure 1: Overview of `ICL-PrivEval`. **(a)** The framework integrates the `ICLInf` metric into a four-step evaluation workflow: build a small in-domain audit set, measure leakage with `ICLInf`, revise the deployment configuration, and re-audit to select the best privacy–utility design. **(b)** The core `ICLInf` metric: run the LLM on the full prompt to obtain the answer distribution P, drop one exemplar (slot k) to obtain P'_k, and take the worst-token log-ratio between P and P'_k as that slot's leakage $\widehat{\mathcal{L}}_k$; `ICLInf` reports the maximum over slots.

a given change reduces it. `ICL-PrivEval` operationalizes `ICLInf` to answer both questions. Given a small in-domain audit set, it measures the deployed prompt and reports both an overall score and a per-slot breakdown that identifies the exemplars' influence on the model's output. The practitioner can then revise the configuration, whether the number of shots, the model, or an experimental mechanism (like calibration) added on top of ICL, and audit it again. Because `ICLInf` is deterministic for a fixed prompt and model, the difference between audits reflects the tuning itself rather than potential randomness of an attack. `ICL-PrivEval` thus allows practitioners to locate leakage, check whether a mitigation actually works, and select a configuration with an acceptable privacy–utility trade-off (the procedure is shown in Figure 1).

We first use `ICL-PrivEval` to explore common factors that may exacerbate ICL privacy leakage. For example, leakage does not simply decrease with model size: across Qwen3 models from 0.6B to 32B, `ICLInf` peaks at 1.7B and falls by more than half at 32B, whereas Duan et al. (2023) found that larger models leak less. Instruction tuning raises leakage relative to base checkpoints, while adding exemplars dilutes it. We then use `ICLInf` to audit a sampling-based DP-ICL mechanism (Amin et al., 2024), comparing the privacy loss it realizes with the theoretical DP budget.

Our **Contributions** are the following:

- We introduce `ICLInf`, a metric for computing the privacy leakage of ICL demonstrations by measuring the output change with and without the demonstrations (Section 3.1).
- We propose `ICL-PrivEval`, a four-step evaluation workflow for measuring privacy leakage of ICL exemplars in a deployed prompt (Section 3.4).
- Using `ICLInf` and `ICL-PrivEval`, we conduct a comprehensive empirical study to identify key factors that may affect ICL leakage. Our results show that the number of exemplars within the prompt have a large effect, with leakage falling 1.5–2.6× from 1 to 32 shots. Furthermore, the LLM parameter size, whether the model was instruction fine-tuned, and whether the model's output was calibrated also affect the leakage.(Section 4).
- Finally, we demonstrate that `ICLInf` can be used as a tool to audit sampling-based DP-ICL frameworks and achieve tight empirical $\varepsilon \leq 10$. (Section 5).

2 Preliminaries

2.1 Threat model

We consider the following common LLM deployment setting that a practitioner may encounter. An LLM is deployed onto a server and is accessible via a hosted API. The LLM uses an ICL prompt template containing n private exemplars $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$. The adversary issues queries $\mathbf{x}'$ and observes either the model's output distribution over a token set $\mathcal{V}$, or a sample drawn from it. We assume black-box query access where the adversary can pose arbitrary $\mathbf{x}'$ and read logits.

2.2 DIFFERENTIAL PRIVACY

Differential Privacy (DP) is the gold-standard privacy notion that guarantees a bound on the amount of information about a private dataset that the output of an algorithm can leak. We give the definition of DP below.

Definition 2.1 (Pure DP (Dwork, 2006))**.** A randomised algorithm $\mathcal{A}$ satisfies ε-DP if for all datasets D, all $i \in [n]$, and all measurable sets E, with $D^{-i} = D \setminus \{(\mathbf{x}_i, \mathbf{y}_i)\}$,

$$\Pr[\mathcal{A}(D) \in E] \leq e^{\varepsilon} \Pr[\mathcal{A}(D^{-i}) \in E], \qquad \Pr[\mathcal{A}(D^{-i}) \in E] \leq e^{\varepsilon} \Pr[\mathcal{A}(D) \in E]. \tag{1}$$

ε-DP gives a worst-case guarantee but is not informative about the privacy loss incurred by individual exemplars, which we are interested in. We give the definition below.

Definition 2.2 (Data-Dependent DP (Papernot et al., 2017))**.** A randomised algorithm $\mathcal{A}$ satisfies $\varepsilon(D)$-DP if for all $i \in [n]$ and all measurable sets E,

$$\Pr[\mathcal{A}(D) \in E] \;\leq\; e^{\varepsilon(D)} \Pr[\mathcal{A}(D^{-i}) \in E], \tag{2}$$

and the symmetric bound also holds.

3 THE ICL-PRIVEVAL FRAMEWORK

In ICL, the model's answer to a query is conditioned on every exemplar in the prompt. This conditioning is what makes ICL work, and it is also the source of its privacy leakage: when a single exemplar disproportionately influences the answer, the answer carries information about that exemplar, and an adversary who observes it may learn about it. Evaluating ICL leakage therefore comes down to measuring how strongly each exemplar shapes the model's output. `ICLInf` provides this measurement, and `ICL-PrivEval` builds evaluation around it: the measurement is taken on the deployed prompt over audit queries, and its results are used to compare configurations and revise the deployment before release.

3.1 ICLINF

`ICLInf` builds on counterfactual influence (Zhang et al., 2023) and context influence (Flemings et al., 2025): it compares the model's output probabilities with and without each exemplar, taking the maximum log-ratio over all possible outputs rather than at the sampled token. For a fixed prompt and query, the result reflects exactly the data-dependent $\varepsilon(D)$ of Definition 2.2.

Definition 3.1 (`ICLInf`)**.** Define $D_{\text{train}} = \{(\mathbf{x}_j, \mathbf{y}_j)\}_{j=1}^{n}$ to be a downstream dataset, S be a set of ICL exemplars randomly sampled from D_{train}, and $D_{\text{test}} = \{\mathbf{x}_j\}_{j=1}^{m}$ be a set of queries. Let $\mathcal{A}$ be an algorithm that maps a set of ICL exemplars S and a query $\mathbf{x}'$ to a probability distribution over possible tokens $\mathcal{V}$. Then

$$\mathcal{L} = \mathop{\mathbb{E}}_{\substack{S \subset D_{\text{train}} \\ \mathbf{x}' \in D_{\text{test}}}} [\max_{(\mathbf{x},\mathbf{y}) \in S} \{\max_{y \in \mathcal{V}} \{|p - p'|\}\}] \tag{3}$$

where

$$p = \log \Pr[\mathcal{A}(S, \mathbf{x}') = y], \qquad p' = \log \Pr[\mathcal{A}(S \setminus \{(\mathbf{x}, \mathbf{y})\}, \mathbf{x}') = y].$$

The inner $\max$ over $y \in \mathcal{V}$ picks the token where the two distributions diverge most, which indicates the output direction that reveals the most about the removed exemplar. The outer $\max$ over $(\mathbf{x}, \mathbf{y}) \in S$ identifies the exemplar that leaks the most. Figure 1(b) illustrates this leave-one-out mechanism.

3.2 EMPIRICAL ESTIMATOR

Eq. 3 cannot be computed exactly: the number of possible exemplar sets grows combinatorially with the size of D_{train}, and the query distribution is known only through samples. We therefore estimate Eq. 3 by sampling S_j of size n_{shots} for each test query $\mathbf{x}'_j$ ($j \in [m]$) and averaging:

$$\widehat{\mathcal{L}} \;=\; \frac{1}{m} \sum_{j=1}^{m} \Big(\max_{(\mathbf{x},\mathbf{y}) \in S_j} \max_{y \in \mathcal{V}} |p_j - p'_j| \Big) \tag{4}$$

with $p_j = \log \Pr[\mathcal{A}(S_j, \mathbf{x}'_j) = y]$ and $p'_j = \log \Pr[\mathcal{A}(S_j \setminus \{(\mathbf{x}, \mathbf{y})\}, \mathbf{x}'_j) = y]$. When $\widehat{\mathcal{L}}$ is large, we say the deployment has high privacy leakage on the downstream task.

3.3 EXTENDING ICLINF TO REASONING

Many deployed models reason before answering, and their reasoning trace is conditioned on the exemplars, so it can reveal information about them even when the final label does not. For such scenarios, we therefore extend ICLInf from the label alone to the entire generated output. ICLInf applies naturally to the full generation, since the probability of a sequence is the product of its token probabilities. Because the maximum over all possible traces cannot be computed, we follow context influence (Flemings et al., 2025) and sample one trace from the full prompt, and then score this same trace under every leave-one-out prompt, so that the difference reflects only the removed exemplar.

Definition 3.2 (Sequence-level ICLInf)**.** Let t be a trace sampled from the model given the full prompt $(S, \mathbf{x}')$, let $\mathcal{C}$ be the set of labels, and let $S^- = S \setminus \{(\mathbf{x}, \mathbf{y})\}$. Then

$$\mathcal{L}_{\text{seq}} = \max_{(\mathbf{x},\mathbf{y})\in S} \max_{c\in\mathcal{C}} \left| \log \frac{\Pr[t \mid S, \mathbf{x}']}{\Pr[t \mid S^-, \mathbf{x}']} + \log \frac{\Pr[c \mid S, \mathbf{x}', t]}{\Pr[c \mid S^-, \mathbf{x}', t]} \right| \tag{5}$$

We use no top-k or top-p truncation, which could assign a sampled token zero probability. Since it is evaluated at one sampled trace, $\mathcal{L}_{\text{seq}}$ is a lower bound on the sequence-level loss.

Instantiations of $\mathcal{A}$. ICLInf is defined for any randomized sampling-based algorithm $\mathcal{A}$ that returns a distribution over outputs, so what it measures is determined by choosing $\mathcal{A}$ to match what the deployed system releases. In our experiments, for the classification tasks of Section 4, $\mathcal{A}$ is the model's distribution over the task's labels. For reasoning models (Section 4.4), the system releases a trace followed by a label, and $\mathcal{L}_{\text{seq}}$ scores the tokens actually generated, measuring the privacy loss of the released output rather than the worst case over all possible outputs. For privacy auditing (Section 5), a DP-ICL mechanism releases free-form text, and $\mathcal{A}$ is its output distribution over the full vocabulary. The audit also follows one sampled generation, but at each step it compares the entire next-token distributions with and without each exemplar, and aggregates for composition. The same metric adapts to plain ICL classification, reasoning models, and privatized DP-ICL mechanisms alike, indicating ICLInf is agnostic to definition of $\mathcal{A}$.

3.4 ICL-PRIVEVAL WORKFLOW

A leakage evaluation is most useful when it informs a decision. Before deploying an ICL prompt, the practitioner needs to know how much the prompt leaks and whether a different configuration could leak less without hurting task performance. ICL-PrivEval aims to address this issue. It places ICLInf inside a loop of measuring, revising, and re-auditing the deployment (Figure 1), which allows a practitioner to measure and improve leakage during development, alongside performance.

The loop begins with an audit set: a few hundred queries representative of the task. The practitioner then measures ICLInf on the deployed prompt. The measurement comes with both an overall score that summarizes how much the prompt leaks, and an intermediate score for each exemplar. Based on these scores, the practitioner can revise the configuration, for example by changing the number of exemplars, the model, or the output format, or by adding a mechanism on top of ICL such as calibration or any privacy defense. Section 4 studies some of these factors, but the deployment design choices are not limited to them. Each candidate configuration is then audited again, and the practitioner compares leakage and accuracy across candidates to choose an acceptable trade-off.

The measurement is taken on the configuration actually being deployed, rather than as a worst-case bound over all possible prompts, and it needs only prompt edits and output probabilities, without any membership labeling that MIAs require. In addition, the per-exemplar scores may suggest how much each exemplar contributes to the leakage, which a single system-level ε cannot provide; Section 4.3 explores this by planting a known exemplar in the prompt. One caveat is that ICL-PrivEval is a measurement rather than a prescription: it quantifies the privacy–utility trade-off of a configuration, and the final choice is left to the practitioner.

4 ICL PRIVACY LEAKAGE EVALUATION

We now use ICL-PrivEval and ICLInf to evaluate ICL exemplar leakage on real world datasets and models. Section 4.1 fixes the setup, Section 4.2 measures how six deployment choices change

ICLInf, Section 4.3 compares ICLInf with two attacks, and Section 4.4 extends the measurement to the whole generation of a reasoning model. Throughout we use ICLInf and *privacy loss* interchangeably. Additional results and setup are given in Appendix C.

4.1 SETUP

Datasets and Models We evaluate on three text classification benchmarks, AG_News (Zhang et al., 2015b), DBpedia_14 (Zhang et al., 2015a) and TREC (Voorhees & Tice, 2000), with three instruction-tuned causal language models: Qwen3-8B (Team, 2025), Ministral-3-8B-Instruct (Liu et al., 2026) and Olmo-3-7B-Instruct (Olmo et al., 2025). The three span distinct pre-training corpora and vocabulary sizes (152k, 131k and 100k). Because ICLInf is computed from the model's output probabilities, we cannot use closed-source models such as GPT that do not expose the full output distribution.

Hyperparameters We perform 4-shot In-Context Learning over 500 test queries. For each query the four exemplars are resampled independently without replacement from the training set, so each query is an independent draw of an (exemplar set, query) pair. The output space is the *label* distribution, obtained by masking non-label tokens and renormalising. Prompts are completion-style; formatting is given in Appendix C.3.

Reporting We report $\widehat{\mathcal{L}}$ (Eq. 4) throughout. The exception is RQ 2, where the x-axis is the exemplar position, so we plot the per-exemplar scores taken before the outer maximum. Error bands and bars in all figures are one standard deviation over queries.

4.2 FACTORS THAT SHAPE LEAKAGE

Table 1: ICLInf and sampled classification accuracy on the deployed 4-shot configuration, averaged over three seeds and 500 queries each. Accuracy is sampled from the label distribution.

	AG News		DBpedia-14		TREC	
Model	ICLInf	Acc.	ICLInf	Acc.	ICLInf	Acc.
Qwen3-8B	4.91	0.86	5.94	0.91	5.75	0.74
Ministral-3-8B	2.04	0.82	2.91	0.91	2.04	0.69
Olmo-3-7B	5.94	0.86	6.21	0.82	4.17	0.78

RQ 1: How does leakage vary across model families? Practitioners usually choose a model based on accuracy and cost, but this choice may also determine how much a prompt leaks. Table 1 compares three model families of similar size on the same tasks and prompts. Leakage differs by up to a factor of three while accuracy ranges from 69% to 91%. Accuracy and leakage are weakly related across the nine configurations, so a model that performs well on a task does not necessarily leak more or less. Model selection is therefore one effective way for practitioners to control leakage.

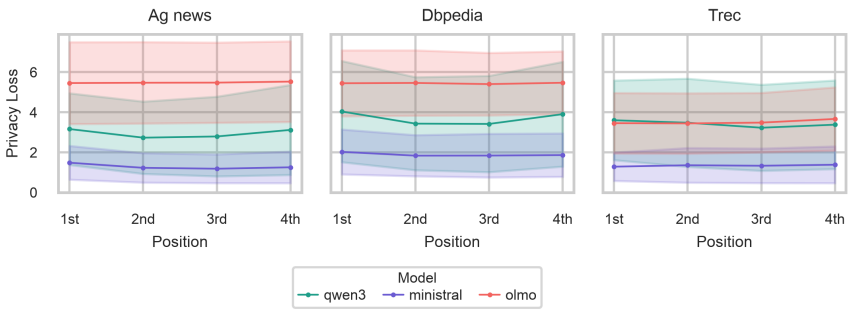


Figure 2: Per-exemplar ICLInf by exemplar position.

RQ 2: Does an exemplar's position in the prompt affect its leakage? Exemplars are processed in order, so a natural question is whether some positions in the prompt are more exposed than others. Figure 2 shows the leakage of each exemplar by its position. For Qwen3 and Ministral on AG News and DBpedia, exemplars at the start and end of the prompt leak more than those in the middle. The effect is small on TREC and absent for Olmo, suggesting that it depends on both the model and the

task. We attribute this behavior to the explanation that the first exemplar introduces the task, so the model relies on it more and its removal has a larger effect, while the last exemplar benefits from recency bias (Zhao et al., 2021). This suggests that arranging sensitive exemplars in the middle may offer some protection but depends on the actual setup.

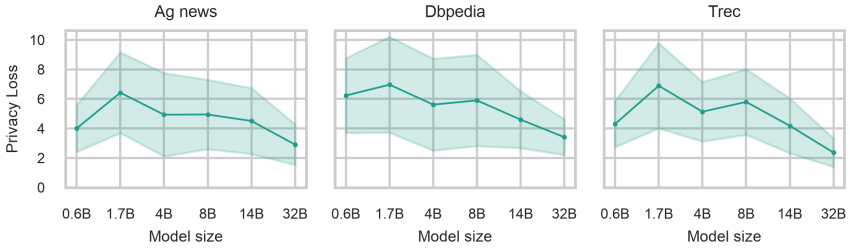


Figure 3: `ICLInf` across Qwen3 model sizes, 0.6B to 32B.

RQ 3: How does model size affect leakage. Prior work found that larger models leak less through membership inference (Duan et al., 2023). We test this with `ICLInf` by sweeping Qwen3 from 0.6B to 32B parameters (Figure 3). Leakage peaks at 1.7B on all three datasets and is generally higher in the middle. We hypothesize that smaller models are not capable enough to utilize the ICL exemplars, while larger models are more capable but have more pre-training knowledge to rely on. Model size therefore affects leakage in multiple directions, and scaling up a model reduces leakage only beyond a moderate size.

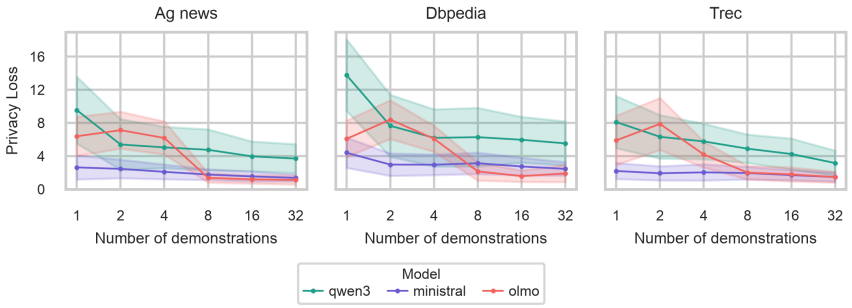


Figure 4: `ICLInf` against the number of in-context demonstrations.

RQ4: How does the number of exemplars affect leakage? Adding exemplars usually improves accuracy, but it also places more private data in the prompt. As shown in Figure 4, leakage generally decreases as the prompt grows, falling by 2.5–2.6× for Qwen3 and 1.5–1.9× for Ministral between 1 and 32 exemplars (Olmo's decline is not as steady). Intuitively, the more exemplars a prompt contains, the less the model's answer depends on any single one, so adding exemplars dilutes each exemplar's influence. Because accuracy holds steady or improves for Qwen3 and Ministral as exemplars are added (Table 6), this approach shows the potential to improve privacy and utility at the same time, at the cost of a longer prompt.

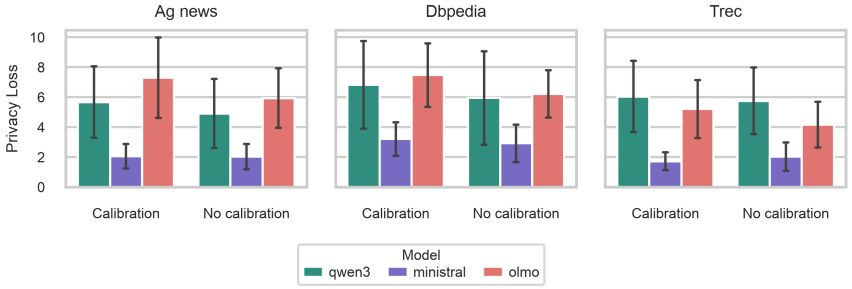


Figure 5: `ICLInf` with and without content-free calibration.

RQ5: Does calibration reduce leakage? Calibration is a common way to improve ICL accuracy, and because it reshapes the model's output distribution, it may also change how much the exemplars leak. We follow the content-free calibration of (Zhao et al., 2021), which rescales the label

distribution by a baseline $\hat{\mathbf{p}}_{\text{cf}}$ obtained from content-free inputs, giving $\mathbf{W}\tilde{\mathbf{p}}$ with $\mathbf{W} = \text{diag}(\hat{\mathbf{p}}_{\text{cf}}^{-1})$. Figure 5 shows that calibration slightly increases privacy leakage in most cases. One plausible explanation is that calibration removes the model's prior over labels, so the output depends more on the exemplars, and removing one has a more visible effect. Calibration should therefore not be mistaken for a privacy mitigation.

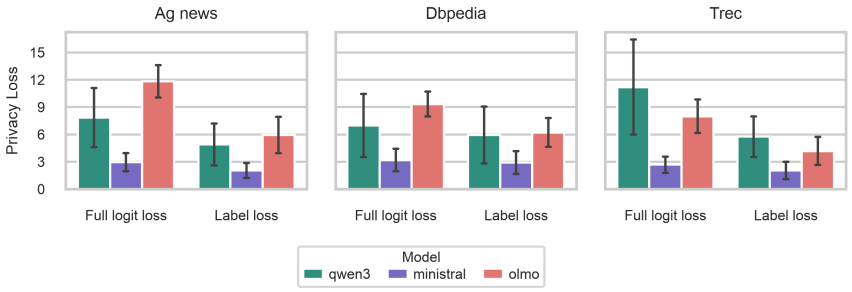


Figure 6: `ICLInf` measured on the label set against the full vocabulary.

RQ6: Does restricting the output space reduce leakage? Our main experiments assume a system that outputs only valid labels. However, a practitioner may let the model generate text freely in many common deployments. To compare the two, we compute `ICLInf` over the full vocabulary, which is also the setting of our privacy audit (Section 5). As shown in Figure 6, free generation leaks more in every configuration, by 1.08–1.96×. The increase comes from non-label tokens: removing an exemplar shifts their probabilities too, and a system that can output them exposes these changes. Restricting the output space is thus a simple way to reduce leakage, if allowed by the task.

4.3 Comparing ICLInf against attacks

`ICLInf` measures how much each exemplar shifts the model's output, whereas attacks measure what an adversary can infer from it. We compare the two to see whether configurations with higher `ICLInf` are also more vulnerable to membership inference, and whether the per-exemplar scores of `ICLInf` can locate a *canary* planted in the prompt.

Table 2: Membership inference against ICL exemplars, with `ICLInf` measured on the same runs. 1000 members and 1000 non-members per configuration; chance AUC is 0.50. ρ is the Spearman correlation between the attack score and the `ICLInf` of the attacked exemplar, over the 1000 members. TREC is omitted because its training set is too small for sampling members and non-members.

Model	Dataset	AUC (95% CI)	`ICLInf`	ρ(score, `ICLInf`)
Qwen3-8B	AG News	0.56 [0.54, 0.59]	5.75	−0.21
Qwen3-8B	DBpedia-14	0.59 [0.57, 0.62]	8.46	−0.06
Ministral-3-8B	AG News	0.59 [0.56, 0.61]	2.26	−0.11
Ministral-3-8B	DBpedia-14	0.37 [0.35, 0.40]	4.01	−0.14
Olmo-3-7B	AG News	0.49 [0.46, 0.51]	5.58	+0.66
Olmo-3-7B	DBpedia-14	0.70 [0.67, 0.72]	6.53	+0.41

Membership inference. If `ICLInf` and membership inference measured the same leakage, the exemplars that `ICLInf` scores highly should also be the easiest to attack. We test this with the MIA from Duan et al. (2023), which scores each record by the model's confidence on its true label. The attack is unstable across configurations, even falling below chance for Ministral on DBpedia (Table 2), and its scores are largely uncorrelated with `ICLInf`. This is likely because the attack looks only at the true label, whereas `ICLInf` takes the largest shift over all labels, which usually falls on a different one. The two thus capture different aspects of leakage, and a single attack can miss what `ICLInf` measures.

Table 3: Canary attribution. A canary replaces one of four exemplars randomly; each entry is the *fraction* of 200 queries for which the canary has the highest per-exemplar score of `ICLInf` (chance is 0.25). *In-dist.*: a real training example with an incorrect label. *Random*: a random hexadecimal string. *Ordinary query*: a normal test example. *Canary query*: the canary's own text.

		Ordinary query		Canary query	
Model	Dataset	In-dist.	Random	In-dist.	Random
Qwen3-8B	AG News	0.670	0.250	0.955	1.000
Qwen3-8B	DBpedia-14	0.860	0.350	1.000	1.000
Qwen3-8B	TREC	0.780	0.705	0.990	1.000
Ministral-3-8B	AG News	0.600	0.225	0.715	0.910
Ministral-3-8B	DBpedia-14	0.930	0.180	0.960	0.985
Ministral-3-8B	TREC	0.350	0.465	0.875	0.995
Olmo-3-7B	AG News	0.220	0.320	0.440	0.615
Olmo-3-7B	DBpedia-14	0.275	0.310	0.260	0.640
Olmo-3-7B	TREC	0.250	0.140	0.360	0.860

Canary attribution. Following Choi et al., we replace one of the four exemplars with a canary at a random position and measure how often it receives the highest per-exemplar `ICLInf` score (chance is 0.25). We use two canaries: a random 32-character hexadecimal string, and a real training example given an incorrect label. When the query is the canary's own text, the canary receives the highest score in 72–100% of queries for Qwen3 and Ministral, higher than on ordinary queries and far above the 25% chance rate (Table 3). The per-exemplar scores of `ICLInf` thus show agreement with the canary audit on which exemplar influences the output, providing support for `ICLInf`.

4.4 Beyond the answer token: reasoning models

The experiments above read leakage at the answer token. We now examine how leakage changes for reasoning models, whose trace can also reveal the exemplars (Section 3.3). We measure $\mathcal{L}_{\text{seq}}$ on Qwen3-14B (Team, 2025) with thinking mode and R1-Distill-14B (DeepSeek-AI, 2025), sampling one trace per query at τ=1. After the trace, we append the answer cue "Answer:" before scoring the label; these inserted tokens stay in context but are excluded from the sum.

Table 4: Sequence-level `ICLInf` on reasoning models, 300 queries per configuration, with one trace sampled per query at $\tau = 1.0$. $\widehat{\mathcal{L}}_{\text{seq}}$ scores the whole generation, while $\widehat{\mathcal{L}}_{\text{label}}$ scores only the answer token as a label-level measurement. Since $\widehat{\mathcal{L}}_{\text{seq}}$ grows with trace length, Tokens reports the mean trace length and Per token normalizes over it. Acc. is the accuracy of the final answer.

Model	Dataset	$\widehat{\mathcal{L}}_{\text{seq}}$	$\widehat{\mathcal{L}}_{\text{label}}$	Per token	Tokens	Acc.
Qwen3-14B	AG News	111.3	5.20	0.27	400	0.82
Qwen3-14B	DBpedia-14	80.9	6.74	0.25	317	0.94
Qwen3-14B	TREC	70.5	3.44	0.21	365	0.86
R1-Distill-14B	AG News	46.3	3.98	0.15	269	0.82
R1-Distill-14B	DBpedia-14	46.8	4.46	0.15	283	0.92
R1-Distill-14B	TREC	57.4	3.41	0.19	304	0.81

Table 4 compares the sequence-level value with the leakage measured at the label alone, both averaged over queries. The sequence-level value is 10–21$\times$ larger as expected, since it accumulates over the trace. The per-token leakage is far lower on average than the label, likely because most trace tokens barely depend on any single exemplar, while the exemplars are meant to influence the answer. For deployments that release the trace, label-level measurement therefore substantially underestimates privacy leakage. Because a single sampled trace varies from query to query, $\widehat{\mathcal{L}}_{\text{seq}}$ is best read as a property of the deployment rather than of an individual query.

5 PRIVACY AUDITING WITH ICLINF

Beyond evaluating deployed prompts, ICLInf can audit a DP-ICL mechanism by comparing the privacy loss the mechanism actually realizes with its theoretical budget. We audit the sampling-based mechanism for synthetic text generation of Amin et al. (2024), which clips and averages the next-token logits of prompts that each contain one private example, and samples the next token from the result. Since the mechanism generates text, we compare the full next-token distributions with and without each example using Rényi divergence, and sum the result over generated tokens to account for composition (Appendix B).

Setup. Each batch contains s=50 one-shot prompts from AG_News (Zhang et al., 2015b), and a neighboring batch removes one prompt while keeping the divisor at s. We audit Qwen3-8B (Team, 2025) with clipping bound c=10 and generate 50 examples of up to 40 tokens, so the mechanism composes over 2000 sampling steps. For each target ε and four values of δ, we tune the temperature τ so that the theoretical ε meets the target.

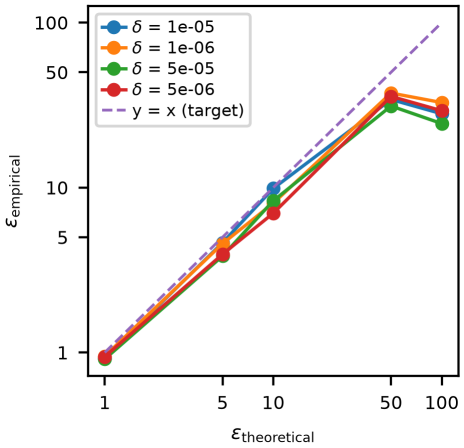


Figure 7: Empirical ε against the theoretical budget, across four values of δ. The dashed line is equality.

Table 5: Privacy auditing on varying temperature ($s = 50$, $c = 10$, $\delta = 10^{-5}$, Qwen3-8B). The tuned τ matches the theoretical ε to the target; RATIO is $\varepsilon_{\text{empirical}}/\varepsilon_{\text{target}}$. $\varepsilon_{\text{empirical}}$ is measured at the prefixes the mechanism sampled and is a diagnostic of that run, not a certified bound.

$\varepsilon_{\text{target}}$	τ	$\alpha^\star$	$\varepsilon_{\text{empirical}}$	Ratio
1	36.18	18	0.92	0.92
5	8.53	5	4.63	0.93
10	4.80	3	9.88	0.99
50	1.42	2	34.17	0.68
100	0.94	2	28.00	0.28

Results. As shown in Figure 7 and Table 5, up to $\varepsilon_{\text{theoretical}}$=10, the realized loss reaches over 90% of the budget, and then falls increasingly short beyond that, for all four values of δ. For larger privacy budgets the RDP optimizer settles at the minimal Rényi Divergence order ($\alpha = 2$), which widens the gap between theoretical and empirical ε. Meanwhile, altering δ leads to different α in RDP-DP conversion, as reflected in the stage-like patterns of $\varepsilon_{\text{empirical}}$. The result implies that ICLInf provides a tighter privacy bound on this ICL synthetic text generation algorithm for $\varepsilon_{\text{theoretical}} \leq 10$, offering a more comprehensive assessment for applied privacy loss analysis.

6 CONCLUSION

We introduced ICL-PrivEval, a four-step evaluation workflow for measuring the privacy leakage of ICL exemplars in a deployed prompt. At its foundation is ICLInf, a data-dependent privacy-leakage metric that attributes leakage to the prompt as well as individual prompt slots and is computed on the specific deployment. And it requires prompt edits and output probabilities.

Using ICL-PrivEval, we showed that deployment choices such as the model family, model size, and number of exemplars could affect leakage, and that ICLInf captures a different aspect of leakage from a confidence-based membership attack. For reasoning models, measuring only the final answer underestimates leakage whenever the reasoning trace is released. Finally, ICLInf can audit a sampling-based DP-ICL mechanism, closely tracking its theoretical budget up to $\varepsilon = 10$. We hope that ICL-PrivEval helps practitioners understand and reduce the privacy leakage of their exemplars when deploying ICL.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods and giving feedback on the experimental design. We have not used generative AI tools for writing proofs, generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating or proving mathematical claims, proposing or refining hypotheses, translation, or interpreting results. Dataset cleaning and reformatting, qualitative data analysis are not applicable to this work. We also used generative AI tools to write and debug code, brainstorming, create figures and tables, retrieve related literature, and polish writing for readability. The authors reviewed all AI-assisted outputs and take responsibility for the content of this paper.

ETHICS STATEMENT

`ICLInf` requires editing the deployed prompt and reading the model's output probabilities, access that only the party deploying the model has, so it does not assist external attackers or malicious users. The attacks we compare against are published methods, applied only to prompts built from public benchmark datasets.

REPRODUCIBILITY STATEMENT

Our setup is described in Section 4.1, prompt formats in Appendix C.3 and Appendix C.4, and the auditing algorithm in Appendix B. Models, datasets, compute, and seeds are listed in Appendix D.

REFERENCES

Kareem Amin, Alex Bie, Weiwei Kong, Alexey Kurakin, Natalia Ponomareva, Umar Syed, Andreas Terzis, and Sergei Vassilvitskii. Private prediction for large-scale synthetic text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7244–7262, 2024.

Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. Hypothesis testing interpretations and renyi differential privacy. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2496–2506. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/balle20a.html.

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.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Proceedings, Part I, of the 14th International Conference on Theory of Cryptography - Volume 9985*, pp. 635–658, Berlin, Heidelberg, 2016. Springer-Verlag. ISBN 9783662536407. doi: 10.1007/978-3-662-53641-4_24. URL https://doi.org/10.1007/978-3-662-53641-4_24.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, pp. 267–284, USA, 2019. USENIX Association. ISBN 9781939133069.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, pp. 1897–1914. IEEE, 2022.

Karan Chadha, Matthew Jagielski, Nicolas Papernot, Christopher A Choquette-Choo, and Milad Nasr. Auditing private prediction. In *Forty-first International Conference on Machine Learning*.

Jacob Choi, Shuying Cao, Xingjian Dong, and Sai Praneeth Karimireddy. Contextleak: Auditing leakage in private in-context learning methods. In *The Impact of Memorization on Trustworthy Foundation Models: ICML 2025 Workshop*.

Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pp. 1964–1974. PMLR, 2021.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Zeyu Ding, Yuxin Wang, Guanhong Wang, Danfeng Zhang, and Daniel Kifer. Detecting violations of differential privacy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 475–489, 2018.

Haonan Duan, Adam Dziedzic, Mohammad Yaghini, Nicolas Papernot, and Franziska Boenisch. On the privacy risk of in-context learning. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pp. 1–12. Springer, 2006.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and trends® in theoretical computer science*, 9(3–4):211–407, 2014.

James Flemings, Bo Jiang, Wanrong Zhang, Zafar Takhirov, and Murali Annavaram. Estimating privacy leakage of augmented contextual knowledge in language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25092–25108, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1220. URL https://aclanthology.org/2025.acl-long.1220/.

Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? *Advances in Neural Information Processing Systems*, 33: 22205–22216, 2020.

Alex Kulesza, Ananda Theertha Suresh, and Yuyan Wang. Mean estimation in the add-remove model of differential privacy. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Jiacheng Li, Ninghui Li, and Bruno Ribeiro. Membership inference attacks and defenses in classification models. In *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, pp. 5–16, 2021.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, Baptiste Rozière, Baudouin De Monicault, Clémence Lanfranchi, Corentin Barreau, Cyprien Courtot, Daniele Grattarola, Darius Dabert, Diego de las Casas, Elliot Chane-Sane, Faruk Ahmed, Gabrielle Berrada, Gaëtan Ecrepont, Gauthier Guinet, Georgii Novikov, Guillaume Kunsch, Guillaume Lample, Guillaume Martin, Gunshi Gupta, Jan Ludziejewski, Jason Rute, Joachim Studnia, Jonas Amar, Joséphine Delas, Josselin Somerville Roberts, Karmesh Yadav, Khyathi Chandu, Kush Jain, Laurence Aitchison, Laurent Fainsin, Léonard Blier, Lingxiao Zhao, Louis Martin, Lucile Saulnier, Luyu Gao, Maarten Buyl, Margaret Jennings, Marie Pellat, Mark Prins, Mathieu Poirée, Mathilde Guillaumin, Matthieu Dinot, Matthieu Futeral, Maxime Darrin, Maximilian Augustin, Mia Chiquier, Michel Schimpf, Nathan Grinsztajn, Neha Gupta, Nikhil Raghuraman, Olivier Bousquet, Olivier Duchenne, Patricia Wang, Patrick von Platen, Paul Jacob, Paul Wambergue, Paula Kurylowicz, Pavankumar Reddy Muddireddy, Philomène Chagniot, Pierre Stock, Pravesh Agrawal, Quentin Torroba, Romain Sauvestre, Roman Soletskyi, Rupert Menneer, Sagar Vaze, Samuel Barry, Sanchit Gandhi, Siddhant Waghjale, Siddharth Gandhi, Soham Ghosh, Srijan Mishra, Sumukh Aithal, Szymon Antoniak, Teven Le Scao, Théo Cachet, Theo Simon Sorg, Thibaut Lavril, Thiziri Nait Saada, Thomas Chabal, Thomas Foubert, Thomas Robert, Thomas Wang, Tim Lawson, Tom Bewley, Tom Bewley, Tom Edwards, Umar Jamil, Umberto Tomasini, Valeriia Nemychnikova, Van Phung, Vincent Maladière, Virgile Richard, Wassim Bouaziz, Wen-Ding Li, William Marshall,

Xinghui Li, Xinyu Yang, Yassine El Ouahidi, Yihan Wang, Yunhao Tang, and Zaccharie Ramzi. Ministral 3, 2026. URL https://arxiv.org/abs/2601.08584.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. *arXiv preprint arXiv:2305.18462*, 2023.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, 2022.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275, 2017. doi: 10.1109/CSF.2017.11.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HkwoSDPgg.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Ulfar Erlingsson. Scalable private learning with pate. In *International Conference on Learning Representations*, 2018.

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.

Xinyu Tang, Richard Shin, Huseyin A Inan, Andre Manoel, Fatemehsadat Mireshghallah, Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, and Robert Sim. Privacy-preserving in-context learning with differentially private few-shot generation. In *ICLR*, 2024.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Ellen M Voorhees and Dawn M Tice. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 200–207, 2000.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. In *NeurIPS*, 2023.

Jiaqi Wang, Roei Schuster, Ilia Shumailov, David Lie, and Nicolas Papernot. In differential privacy, there is truth: on vote leakage in ensemble private learning. NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Rui Wen, Zheng Li, Michael Backes, and Yang Zhang. Membership inference attacks against in-context learning. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 3481–3495, 2024.

Tong Wu, Ashwinee Panda, Jiachen T Wang, and Prateek Mittal. Privacy-preserving in-context learning for large language models. In *ICLR*, 2024.

Yuyang Xia, Ruixuan Liu, and Li Xiong. Snapaudit: Active auditing of differentially private in-context learning via snapshot-based simulation, 2026. URL https://arxiv.org/abs/2511.13502.

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Counterfactual memorization in neural language models. *Advances in Neural Information Processing Systems*, 36:39321–39362, 2023.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015a. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015b.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12697–12706. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/zhao21c.html.

A Related Works

Membership Inference Attacks. Analyzing the privacy risks of ICL exemplars during decoding is a crucial, but understudied problem. Membership Inference Attacks (MIAs) are an effective way to measure this privacy risk (Shokri et al., 2017; Li et al., 2021; Mattern et al., 2023; Choquette-Choo et al., 2021; Carlini et al., 2022). In particular, there are two main works that specifically focus on MIAs for ICL (Duan et al., 2023; Wen et al., 2024). In addition to proposing MIAs for ICL, they also looked into some factors that could influence leakage of ICL exemplars. Our work seeks to complement these two works by re-evaluating these factors, as well as other factors like calibration and instruction-tuning, to better understand their effects on ICL privacy leakage. Specifically, Duan et al. (2023) observed that larger models leak less privacy, whereas we found that moderately sized LLMs tend to leak the most. Wen et al. (2024) measured membership inference for the first and last exemplar while varying the number of exemplars, while we measured privacy leakage across all exemplars in this setting. Additionally, Wen et al. (2024) found that exemplars in the middle exhibit lower vulnerability compared to those in the beginning and end, which aligns with our results.

Privacy Auditing. Privacy auditing is a useful tool for establishing an empirical bound of the privacy leakage of a DP algorithm (Ding et al., 2018; Jagielski et al., 2020). One line of work proposed privacy auditing for private-prediction-based DP algorithms. Chadha et al. audited private prediction mechanisms (sample-aggregate/noisy-argmax) using Rényi-DP tests. Wang et al. (2022) explored the privacy leakage in the stochastic outputs of the DP prediction algorithm PATE via repeated test queries. Another line of work audits DP-ICL mechanisms directly. A standard auditing approach is to insert a known record (a *canary*) into the private data and measures how strongly it can be recovered from the model's output (Carlini et al., 2019). Choi et al. extend this idea to ICL, inserting canaries into exemplars and using targeted queries to detect them after private aggregation, and derive an empirical lower bound on ε. Our `ICL-PrivEval` framework follows this line but instead relies on contextual influence of individual exemplars to audit privacy leakage. We adopt canary insertion as a comparison for `ICLInf`, since a canary placed in a known slot may imply where the leakage should appear.

B Additional Details on Privacy Analysis for Privacy Auditing

B.1 Algorithm for Auditing

The audited mechanism is adapted from Algorithm 2 of Amin et al. (2024) (Appendix D.4), reproduced as Algorithm 1 below.

Algorithm 1 Private token generation

Input: Sensitive prompt dataset D, initial token sequence $\mathbf{x}_0$
Output: Token sequence $\mathbf{x} \in \mathcal{X}^*$

$\mathbf{x} \leftarrow \mathbf{x}_0$
$Z \leftarrow \{\text{logits}(\mathbf{px}) : \mathbf{p} \in D\}$
$\bar{\mathbf{z}} \leftarrow \ell(Z)$
$x \sim \text{softmax}(\bar{\mathbf{z}}/\tau)$
Append x to $\mathbf{x}$
return $\mathbf{x}$

with

$$\ell(Z) = \frac{1}{s}\sum_{\mathbf{z}\in Z} \text{clip}_c(\mathbf{z}), \qquad \text{clip}_c(\mathbf{z})_i = \max\{-c,\ \mathbf{z}_i - \max_j \mathbf{z}_j + c\}.$$

B.2 Properties of DP and RDP in Algorithm 1 and ICLInf

Definition B.1 (Rényi divergence)**.** For two probability distributions P and Q over a common domain, the Rényi divergence of order $\alpha > 1$ is

$$D_\alpha(P\|Q) = \frac{1}{\alpha - 1}\ \log \mathbb{E}_{x\sim Q}\left[\left(\frac{P(x)}{Q(x)}\right)^\alpha\right]. \tag{6}$$

We also use the symmetrized short-hand

$$D_\alpha^{\leftrightarrow}(P\|Q) = \max\{D_\alpha(P\|Q),\ D_\alpha(Q\|P)\}. \tag{7}$$

Theorem B.2 (zCDP of Algorithm 1, Lemma 4 of (Amin et al., 2024))**.** Let $\mathcal{A}(D, \mathbf{x}_0)$ be Algorithm 1 with clipping level $c > 0$, batch size $s \in \mathbb{N}$, and temperature $\tau > 0$. Then the mechanism $M :\ D \mapsto \mathcal{A}(D, \mathbf{x}_0)$ satisfies ρ-zCDP with $\rho = \frac{1}{2}\big(c/(s\tau)\big)^2$.

Theorem B.3 (zCDP to RDP, Def. 1 of (Bun & Steinke, 2016) and Def. 4 of (Mironov, 2017))**.** If a mechanism M satisfies ρ-zCDP, then for every order $\alpha > 1$ it satisfies $(\alpha, \varepsilon(\alpha))$-RDP with $\varepsilon(\alpha) = \rho\,\alpha$.

Theorem B.4 (Exponential mechanism privacy, (Dwork et al., 2014))**.** The exponential mechanism with sampling probability $\propto \exp\big(\frac{\varepsilon}{2\Delta_f} f(D, x)\big)$ is $(\varepsilon, 0)$-DP with $\varepsilon = 2\Delta_f$.

Theorem B.5 (DP and RDP bounds of the exponential mechanism, Theorem 8 of (Wu et al., 2024))**.** The exponential mechanism is ε-DP, and $(\alpha, \varepsilon_{\text{EM}}(\alpha))$-RDP with

$$\varepsilon_{\text{EM}}(\alpha) := \min\Big(\frac{\alpha}{2}\varepsilon^2,\ \varepsilon'\Big), \quad \text{where} \quad \varepsilon' = \frac{1}{\alpha - 1}\log\left(\frac{\sinh(\alpha\varepsilon) - \sinh((\alpha-1)\varepsilon)}{\sinh(\varepsilon)}\right).$$

Theorem B.6 (Data-dependent RDP)**.** Fix an order $\alpha > 1$. Let $\mathcal{A}(D, \mathbf{x}_0)$ denote the output distribution of Algorithm 1 on dataset D with initial token sequence $\mathbf{x}_0$, and let $\mathcal{A}(D \setminus \{\mathbf{p}\}, \mathbf{x}_0)$ be the output distribution under the remove-one neighboring batch. The data-dependent Rényi privacy loss of $\mathcal{A}$ at order α is

$$\varepsilon_{\mathcal{A}}(D; \alpha) = \max_{\mathbf{p}\in D} D_\alpha^{\leftrightarrow}\big(\mathcal{A}(D, \mathbf{x}_0)\,\|\,\mathcal{A}(D \setminus \{\mathbf{p}\}, \mathbf{x}_0)\big). \tag{8}$$

Theorem B.7 (Composition, (Mironov, 2017))**.** Let $\mathcal{A}_1$ and $\mathcal{A}_2$ satisfy (α, ε_1)-RDP and (α, ε_2)-RDP on the same dataset. Then their composition $\mathcal{A}_1 \circ \mathcal{A}_2$ satisfies $(\alpha, \varepsilon_1 + \varepsilon_2)$-RDP.

Theorem B.8 (RDP to (ε, δ)-DP conversion, BBGHS, (Balle et al., 2020))**.** Let a mechanism satisfy $(\alpha, \varepsilon_{\text{RDP}}(\alpha))$-RDP for some $\alpha > 1$. Then for any $\delta \in (0, 1)$ it is (ε, δ)-DP with

$$\varepsilon = \varepsilon_{\text{RDP}}(\alpha) + \log\frac{\alpha - 1}{\alpha} - \frac{\log\delta + \log\alpha}{\alpha - 1}.$$

Theorem B.9 (Bounded–unbounded translation for pure DP, (Kulesza et al., 2024))**.** If a mechanism M is ε-DP under the *unbounded* (add/remove) adjacency, then it is 2ε-DP under the *bounded* (change-one) adjacency. Conversely, ε-DP under bounded adjacency implies $\varepsilon/2$-DP under unbounded adjacency.

B.3 Privacy Calculation

B.3.1 Theoretical Privacy Loss of Algorithm 1

By Theorems B.2 and B.3, Algorithm 1 satisfies RDP with

$$\varepsilon_{\mathrm{RDP}}(\alpha) = \frac{\alpha}{2}\Big(\frac{c}{s\,\tau}\Big)^2. \tag{9}$$

By Lemma 3 of Amin et al. (2024), Algorithm 1 is an exponential mechanism with sensitivity $\Delta_f = c/(s\tau)$. Combining Theorems B.4 and B.5 with Equation equation 9, the theoretical RDP privacy loss of Algorithm 1 is bounded by

$$\varepsilon_{\mathrm{RDP}}(\alpha) := \min\left(\frac{\alpha}{2}\left(\frac{c}{s\tau}\right)^2,\ \varepsilon'\right),$$

$$\text{where}\quad \varepsilon' = \frac{1}{\alpha-1}\log\left(\frac{\sinh\big(\frac{2\alpha c}{s\tau}\big) - \sinh\Big(\frac{2(\alpha-1)c}{s\tau}\Big)}{\sinh\big(\frac{2c}{s\tau}\big)}\right).$$

Algorithm 1 is invoked once per generated token, so by the RDP composition theorem (Theorem B.7) the total theoretical RDP for one experiment with N_{gen} sequences and at most $T_{\max}$ new tokens per sequence satisfies

$$\varepsilon_{\text{theoretical}}(\alpha) \;\le\; N_{\mathrm{gen}}\, T_{\max}\, \varepsilon_{\mathrm{RDP}}(\alpha).$$

B.3.2 Matching $\varepsilon_{\text{theoretical}}(\alpha)$ to $\varepsilon_{\text{target}}$

Because the target ε values are stated in pure DP, we use two procedures to match $\varepsilon_{\text{theoretical}}(\alpha)$ to $\varepsilon_{\text{target}}$. Algorithm 2 converts RDP to approximate DP via Theorem B.8, searching over RDP orders for the optimal pair $(\varepsilon^\star, \alpha^\star)$.

Algorithm 2 RDP-ApproxDP

Input: failure probability $\delta \in (0,1)$; candidate orders $\mathcal{O} = \{2, 3, \ldots, 99\}$; RDP function $\varepsilon_{\mathrm{RDP}}(\alpha)$
Output: optimal $(\varepsilon^\star, \alpha^\star)$ such that the mechanism is $(\varepsilon^\star, \delta)$-DP

$\varepsilon^\star \leftarrow +\infty, \quad \alpha^\star \leftarrow \perp$
for $\alpha \in \mathcal{O}$ **do**
  $\varepsilon \leftarrow \varepsilon_{\mathrm{RDP}}(\alpha) + \log\frac{\alpha-1}{\alpha} - \frac{\log\delta + \log\alpha}{\alpha-1}$
  if $\varepsilon < \varepsilon^\star$ **then**
    $\varepsilon^\star \leftarrow \varepsilon; \quad \alpha^\star \leftarrow \alpha$
  end if
end for
return $(\varepsilon^\star, \alpha^\star)$

We apply Algorithm 2 to $\varepsilon_{\text{theoretical}}(\alpha)$ and tune the temperature τ with c and s held fixed so that $\varepsilon_{\text{theoretical}}$ matches $\varepsilon_{\text{target}}$. We binary-search over τ; at each step Algorithm 2 returns ε_τ at the current τ, which is compared against $\varepsilon_{\text{target}}$ to shrink the search interval. In practice $N = 22$ iterations reduce the gap below our tolerance. The procedure is given in Algorithm 3.

Algorithm 3 Binary search over temperature τ

Input: target $\varepsilon_{\text{target}}$; failure probability δ; candidate orders $\mathcal{O}$; bounds $t_{\min} < t_{\max}$; iterations N; RDP function $\varepsilon_{\text{RDP}}(\alpha; \tau)$
Output: $(\tau^\star, \alpha^\star, \varepsilon^\star)$ with $(\varepsilon^\star, \delta)$-DP and $\varepsilon^\star \approx \varepsilon_{\text{target}}$

$\ell \leftarrow t_{\min}, \quad h \leftarrow t_{\max}$
for $i = 1$ **to** N **do**
  $\tau \leftarrow (\ell + h)/2$
  $(\varepsilon_\tau, \alpha_\tau) \leftarrow \text{RDP-ApproxDP}\big(\delta, \mathcal{O}, \alpha \mapsto \varepsilon_{\text{RDP}}(\alpha; \tau)\big)$
  if $\varepsilon_\tau \le \varepsilon_{\text{target}}$ **then**
    $h \leftarrow \tau$
  else
    $\ell \leftarrow \tau$
  end if
end for
$\tau^\star \leftarrow (\ell + h)/2$
$(\varepsilon^\star, \alpha^\star) \leftarrow \text{RDP-ApproxDP}\big(\delta, \mathcal{O}, \alpha \mapsto \varepsilon_{\text{RDP}}(\alpha; \tau^\star)\big)$
return $(\tau^\star, \alpha^\star, \varepsilon^\star)$

B.3.3 Data-Dependent Privacy Loss by ICLInf

We take the optimal order $\alpha^\star$ returned by Algorithm 3. At token step t, the per-token data-dependent RDP is

$$\varepsilon_{\mathcal{A}}^{(t)}(D; \alpha^\star) = \max_{\mathbf{p} \in D} D_{\alpha^\star}^{\leftrightarrow}\big(\mathcal{A}_t(D, \mathbf{x}_0) \,\|\, \mathcal{A}_t(D \setminus \{\mathbf{p}\}, \mathbf{x}_0)\big).$$

By composition (Theorem B.7), the total data-dependent RDP accumulates over all generated tokens,

$$\varepsilon_{\mathcal{A}}^{(1:T)}(D; \alpha^\star) = \sum_{t=1}^{T} \varepsilon_{\mathcal{A}}^{(t)}(D; \alpha^\star),$$

which we convert back to approximate DP by Theorem B.8:

$$\varepsilon(\delta; \alpha^\star) = \varepsilon_{\mathcal{A}}^{(1:T)}(D; \alpha^\star) + \log \frac{\alpha^\star - 1}{\alpha^\star} - \frac{\log \delta + \log \alpha^\star}{\alpha^\star - 1}.$$

Our data-dependent privacy loss is evaluated under unbounded adjacency (remove-one neighbors) while the theoretical privacy loss is stated under bounded adjacency, so by Theorem B.9 the final privacy loss reported by `ICLInf` is

$$\varepsilon_{\text{empirical}} = 2\,\varepsilon(\delta; \alpha^\star).$$

C Additional Experiment Information

C.1 Instruction Tuning

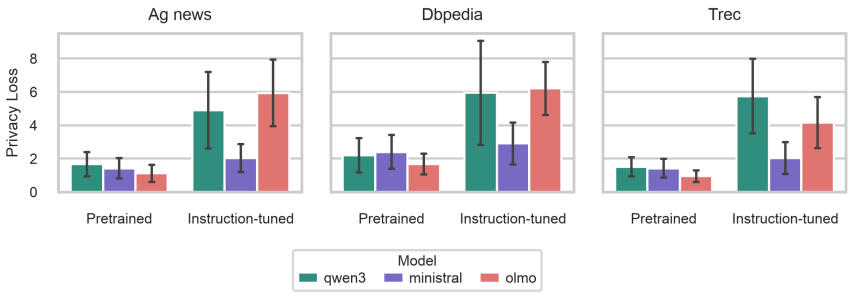


Figure 8: `ICLInf` for instruction-tuned models against their base checkpoints.

RQ: How does instruction tuning affect leakage? Deployed models are almost always instruction-tuned, so we ask whether this training changes how much they leak. Holding the prompt fixed, we

compare each model with its base checkpoint (Figure 8). The base models leak less in every configuration, at 0.23–0.37× the leakage of their tuned counterparts for Qwen3 and Olmo and 0.69–0.82× for Ministral, while still solving the tasks (66–86% accuracy; Table 7). Instruction tuning makes the model more responsive to the prompt in general, and potentially more sensitive to individual exemplars. Although base models are rarely deployed directly, the comparison demonstrates instruction tuning as a non-trivial factor on ICL leakage.

C.2 Task Accuracy

Accuracy is sampled from the label distribution, as in Table 1. Table 6 reports accuracy as the number of exemplars grows (RQ4): it holds steady or improves for Qwen3 and Ministral, while Olmo drops at 2 and 8 exemplars. Table 7 compares base and instruction-tuned checkpoints: both solve the tasks, so the lower leakage of the base models does not come from ignoring the exemplars.

Table 6: Accuracy by number of exemplars (500 queries per configuration).

Model	Dataset	1	2	4	8	16	32
Qwen3-8B	AG News	0.83	0.82	0.83	0.86	0.85	0.84
Qwen3-8B	DBpedia-14	0.86	0.90	0.89	0.89	0.92	0.93
Qwen3-8B	TREC	0.62	0.73	0.74	0.74	0.77	0.84
Ministral-3-8B	AG News	0.79	0.78	0.79	0.83	0.84	0.81
Ministral-3-8B	DBpedia-14	0.88	0.90	0.90	0.90	0.91	0.93
Ministral-3-8B	TREC	0.69	0.68	0.70	0.70	0.70	0.75
Olmo-3-7B	AG News	0.84	0.70	0.82	0.62	0.74	0.83
Olmo-3-7B	DBpedia-14	0.82	0.62	0.83	0.60	0.77	0.85
Olmo-3-7B	TREC	0.82	0.50	0.77	0.70	0.72	0.73

Table 7: Accuracy of base and instruction-tuned checkpoints (4 exemplars, 500 queries per configuration).

Model	Dataset	Base	Instruction-tuned
Qwen3-8B	AG News	0.81	0.86
Qwen3-8B	DBpedia-14	0.81	0.89
Qwen3-8B	TREC	0.71	0.73
Ministral-3-8B	AG News	0.76	0.82
Ministral-3-8B	DBpedia-14	0.85	0.88
Ministral-3-8B	TREC	0.66	0.69
Olmo-3-7B	AG News	0.81	0.84
Olmo-3-7B	DBpedia-14	0.86	0.83
Olmo-3-7B	TREC	0.73	0.79

C.3 Classification Prompt Format

We report the prompts used for classification in Table 8; the templates are taken from Table 5 of Zhao et al. (2021) with slight modification. The ellipsis stands for the remaining in-context demonstrations.

C.4 Synthetic Generation Prompt Format

The prompt used for synthetic example generation is given in Table 9. We follow the setup of Amin et al. (2024), in which the one-shot private example and the generated example share a label. We fix that label to "Business".

Table 8: The prompts used during ICL for text classification tasks

Task	Prompt	Labels
AG_News	Classify the news articles into the categories of World, Sports, Business, and Technology. Output only the category name, with no explanation or extra text. *Article:* Wall St. Bears Claw Back Into the Black NEW YORK (Reuters) - Short-sellers, Wall Street's dwindling band of ultra-cynics, are seeing green again. *Answer:* Business ... *Article:* Rescuing an Old Saver If you think you may need to help your elderly relatives with their finances, don't be shy about having the money talk – soon. *Answer:*	World, Sports, Business, Technology
DBPedia_14	Classify the articles based on whether they are about one of Company, School, Artist, Athlete, Politician, Transportation, Building, Nature, Village, Animal, Plant, Album, Film, Book. Output only the category name, with no explanation or extra text. *Article:* Bergan Mercy Medical Center is a hospital located in Omaha Nebraska. It is part of the Alegent Health System. *Answer:* Company ... *Article:* Fargo Moorhead Metro Area Transit (popularly known as MAT or MATBUS) is a bus company serving the Fargo North Dakota and Moorhead Minnesota Metropolitan area. *Answer:*	Company, School, Artist, Athlete, Politician, Transportation, Building, Nature, Village, Animal, Plant, Album, Film, Book
TREC	Classify the questions based on whether their question type is a Number, Location, Person, Description, Entity, or Abbreviation. Output only the question type, with no explanation or extra text. *Question:* How did serfdom develop in and then leave Russia? *Answer:* Description ... *Question:* When was Ozzy Osbourne born? *Answer:*	Number, Location, Person, Description, Entity, Abbreviation

Table 9: The prompts used during ICL for text generation tasks.

Task	Prompt	Labels
AG_News	Given a label of news type, generate the chosen type of news accordingly. *News Type:* Business *Text:* Oil prices soar to all-time record, posing new menace to US economy (AFP) AFP - Tearaway world oil prices, toppling records and straining wallets, present a new economic menace barely three months before the US presidential elections. *News Type:* Business *Text:* ...	Business

D Compute and Reproducibility

Artifacts. All datasets and models are obtained from the Hugging Face Hub and loaded through the Hugging Face API. The classification datasets are AG News (Zhang et al., 2015b), DBpedia-14 (Zhang et al., 2015a) (CC-BY-SA-3.0), and TREC (Voorhees & Tice, 2000). The evaluated models are `Qwen/Qwen3-8B`, `mistralai/Ministral-3-8B-Instruct-2512`, and `allenai/Olmo-3-7B-Instruct`, with the matching base checkpoints `Qwen/Qwen3-8B-Base`, `mistralai/Ministral-3-8B-Base-2512`, and `allenai/Olmo-3-1025-7B` for the instruction-tuning comparison. The model-scale sweep uses the Qwen3 dense family from 0.6B to 32B, and the reasoning experiments use `Qwen/Qwen3-14B` (Team, 2025) and `deepseek-ai/DeepSeek-R1-Distill-Qwen-14B` (DeepSeek-AI, 2025). Qwen3 and Olmo-3 are released under Apache 2.0. Our use is consistent with the stated intended research use of each artifact.

Compute. All experiments run on a single NVIDIA L40 (46 GB). Measured generation and scoring time is approximately 65 GPU-hours across the 220 classification, canary, MIA, reasoning cells, and auditing. One exception is that the 16- and 32-shot are run on a single NVIDIA A100 for 22 hours. Model download and instance provisioning are excluded.

Seeds. We use the original training and test splits and shuffle each dataset with `seed`=42. The headline results are run at three seeds (42, 43, 44); every other experiment uses a single seed, because in-context exemplars are resampled per query, so the queries within one run are already independent (query, exemplar-set) draws and additional seeds only increase n. Exemplars are drawn with the NumPy default RNG and queries from the test split. In the privacy audit, prompts are formed by grouping the AG News training split into batches of size s, and NumPy, PyTorch, and CUDA are all seeded so that generation is reproducible.