Not All 4-bit Quantizers Are Equal: Deployment-Time Mitigation of PII Leakage in Fine-Tuned Small Language Models

Cristhian Kapelinski[1], **Diego Kreutz**[1]

[1]AI Horizon Labs, Federal University of Pampa (UNIPAMPA)
cristhianavila.aluno@unipampa.edu.br, diegokreutz@unipampa.edu.br

Abstract. *Organizations fine-tune small language models on private data and then compress them to 4 bits for resource-efficient deployment. We show that which 4-bit method they use also affects privacy, and that what separates the methods is not the bit width but whether they tune their rounding on a small sample of text, the calibration corpus. On our primary model, when each planted record's own opening text is used as the prompt, the two calibration-based methods we test, Activation-aware Weight Quantization (AWQ) and Gradient-based Post-Training Quantization (GPTQ), each reproduce none of the planted records, while the calibration-corpus-free GGUF Q4_K_M format reproduces 5.3% of them on the same seeds. Tracked across five open models with 0.5–7 billion parameters, AWQ leaks least at every size and in both families, with little accuracy loss at 3–7 billion. Controlled experiments associate the difference with calibration-induced rounding error in channels involved in rare-token prediction. Choosing the 4-bit method is therefore a deployment-time privacy decision, not only a question of speed and quality.*

1. Introduction

Small language models (SLMs) with 1–7 billion parameters support on-premise deployment when cost, latency, or data-governance requirements limit the use of cloud services [Husom et al. 2025]. Analysts project that artificial-intelligence-capable personal computers will account for 54.7% of worldwide shipments by 2026.[1] Open-weight model families also give an organization a practical starting point for domain-specific fine-tuning, because adapting a model of this size no longer requires a datacenter [Dettmers et al. 2023, Wang and Li 2025]. This deployment setting keeps prompts and inference inside the organization, but it does not remove the information the model memorized during training [Carlini et al. 2023, Lukas et al. 2023].

A common deployment pipeline has two stages [Zhang et al. 2025b, Lin et al. 2024]. First, an organization fine-tunes a model on internal data that may contain personally identifiable information (PII): email, customer records, clinical notes. Second, it applies 4-bit post-training quantization (PTQ) to cut the model's memory footprint and hardware requirements. A model's weights are ordinary numbers, normally stored in 16 bits each. Quantization stores each in 4 bits instead: weights are handled in small blocks, and within a block each becomes one of only 16 levels, plus one number per block saying how far apart those levels are. The model shrinks by roughly the ratio of the two widths and runs faster, since less data moves to the processor; the cost is that every weight is now slightly off.

On Llama-2-7B that cost is small: 4-bit AWQ moves perplexity from 5.47 to 5.60, about 2%, where perplexity measures how surprised the model is by text it never saw and

[1] Gartner via Computerworld, 2025

lower is better, so 5.60 is slightly worse. In exchange it generates tokens 3.2–3.3× faster than the 16-bit version and runs a 13-billion-parameter model on an 8 GB laptop GPU that cannot hold even a 7-billion-parameter one at 16 bits [Lin et al. 2024]. Four bits is not only a small-deployment choice: OpenAI's gpt-oss models ship their largest weight matrices in a 4-bit floating-point format, which lets the 120-billion-parameter version run on one 80 GB GPU,[2] and Kimi K2 Thinking is served in native 4-bit integer precision, with every published benchmark number measured at that precision.[3]

The 4-bit methods in common use [Lin et al. 2024, Frantar et al. 2023, Kurt 2026] disagree on where to put the rounding boundaries, and that is what we study. GGUF k-quants, the 4-bit formats shipped with `llama.cpp` and represented here by Q4_K_M, look only at the weights: each block is rescaled so its own largest weight fits the 4-bit range, and every block is treated alike [Kurt 2026, Husom et al. 2025]. How the model is used never enters the decision, so no extra data is needed. Activation-aware Weight Quantization with 4-bit integers (AWQ-4bit) instead first runs a small sample of text, the *calibration corpus*, through the model, records how strongly each input channel responds, and scales up those that respond most before rounding [Lin et al. 2024]. Rounding error is thereby pushed off the channels the calibration text exercises and onto those it leaves idle. We ask whether that redistribution also changes how much memorized PII the deployed model hands back. The question is first of all one of data security: a model fine-tuned on internal email or customer records is itself a copy of that data [Carlini et al. 2021], and any user who can query it is a path out of the organization that never touches the source database. It is also a compliance question, since reproducing a person's data on demand bears on data-subject rights under the Brazilian General Data Protection Law (LGPD, Art. 18)[4] and the European General Data Protection Regulation (GDPR, Arts. 15 and 17).[5]

Language models can reproduce training records to users with only black-box query access. The standard way to measure this is to plant records in the training data on purpose, so their later reproduction proves memorization rather than coincidence; such records are called *canaries* [Carlini et al. 2019, Carlini et al. 2021]. Memorization grows with model capacity, duplication, and prompt length [Carlini et al. 2023], and PII leaks after fine-tuning too [Lukas et al. 2023]. How the model was fine-tuned shifts the risk: updating only the last layer, which turns the model's internal state into a score per word, can leak more than updating the whole network [Mireshghallah et al. 2022]; Low-Rank Adaptation (LoRA), which freezes the original weights and trains a small extra matrix beside each, can leak less [Wang and Li 2025]; and divergence attacks, which push the model off its normal behaviour with a degenerate prompt such as one word repeated indefinitely, make it fall back on reciting training text [Nasr et al. 2023].

Two questions can be asked of a deployed model, and they differ in severity. *Verbatim extraction* asks whether the model will type out a memorized record when prompted; it needs only the ability to send text and read the answer, and a success is the record itself in the clear. *Membership inference* asks only whether a record was in the training set; it never recovers content, and it needs more: an interface that returns the probability the model assigns to each token of a candidate sequence. Neither needs the weights. We therefore treat verbatim extraction as the primary threat, since it both

[2] OpenAI, gpt-oss-120b model card, 2025 [3] Moonshot AI, Kimi K2 Thinking model card, 2025 [4] Lei n. 13.709/2018 (LGPD), Art. 18 [5] Regulation (EU) 2016/679 (GDPR), Arts. 15 and 17

discloses data and needs the weaker interface, and report membership inference as a complementary measure.

At a fixed 4-bit budget, does the quantization method change the extraction of PII memorized during fine-tuning? The closest prior study compares round-to-nearest (RTN), AWQ, and Gradient-based Post-Training Quantization (GPTQ) on a 7-billion-parameter model after machine unlearning, a procedure that edits a trained model to remove specific records, and finds no method-specific difference [Zhang et al. 2025b]; related work explains that coarse quantization can erase the small weight update unlearning produces [Mishra and Mehreen 2026, Abitante et al. 2026]. We instead quantize right after a fresh fine-tune, whose larger updates may interact differently with each rounding method, comparing calibration-based and calibration-corpus-free quantizers at one bit width across fully fine-tuned SLMs of 0.5–7 billion parameters. To the best of our knowledge this is the first such comparison on a verbatim PII benchmark at this scale, and the first to separate whether any gap comes from activation-aware scaling, from calibration itself, or from bit-rate. Section 4 measures the leakage gap, Section 5 isolates the three factors that produce it, Section 6 reconciles extraction with membership inference, Section 7 prices the accuracy this costs, Section 8 reports where in the (scale, family, regime) sweep the gap widens, narrows, or disappears, and Section 9 tests whether the result survives when the targets are real PII rather than planted records. The practical contribution is evidence that quantizer selection belongs in the privacy evaluation of a deployed SLM.

2. Related Work

Table 1 groups prior work into two families: *memorization measurement* and the *interaction between quantization and privacy*. In it, the two properties our question needs never appear together: the studies that target PII do not vary the quantization method, and those that do vary it target unlearned content instead.

Memorization measurement. The dominant probe is canary exposure [Carlini et al. 2019], which ranks a planted record against every other string fitting the same template and reports how far ahead the model places it. Its variants relax what counts as success: verbatim extraction counts exact reproductions [Carlini et al. 2021]; probabilistic extraction counts a record leaked when any of several sampled completions reproduces it [Hayes et al. 2025]; threshold detectors such as Min-K% judge membership from a sequence's least likely tokens [Shi et al. 2024a, Zhang et al. 2025a]; adversarial-compression tests ask whether a short prompt makes the model emit the record [Schwarzschild et al. 2024]; and loss-based membership-inference attacks (MIA) judge participation from the loss assigned [Yeom et al. 2018]. This literature varies one training-side mechanism at a time, such as duplication during pretraining [Kandpal et al. 2022], the fine-tuning regime [Mireshghallah et al. 2022], or the choice between LoRA and full fine-tuning [Wang and Li 2025], while leaving post-training compression fixed. Because exact-match metrics miss paraphrases and so understate leakage [Ippolito et al. 2023], we report an embedding-similarity measure alongside them.

Quantization $\times$ privacy. The question is older than the current wave: quantizing to 8 bits leaves canary exposure unchanged [Carlini et al. 2019], which our Q8_0 cell reproduces at SLM scale. What has not been asked is whether the choice among 4-bit *methods* matters. The closest prior work [Zhang et al. 2025b] evaluates 4-bit PTQ on MUSE, a machine-unlearning benchmark, on 7-billion-parameter models after unlearning (RTN on its news and book corpora, GPTQ and AWQ on news), and finds no gap among the

Table 1. Related work positioning. Stage: pipeline stage whose leakage is measured; Target: the data measured; Attack: extraction recovers the content, membership only decides participation; Bits/Quantizers: widths and methods evaluated, excluding the full-precision reference. "–": not studied; "n/r": not reported.

Work	Params	Stage	Target	Attack	Bits	Quantizers	Finding
Memorization measurement							
Carlini et al. 2019	0.6M; n/r	Pretraining	Canaries	Extraction	8	Weight PTQ	Canaries extractable; 8-bit does not reduce exposure
Lukas et al. 2023	124M–1.6B	Fine-tuning	PII	Extraction, membership	–	–	Scrubbing and differential privacy leave residual PII
Quantization × privacy interaction							
Zhang et al. 2025	7B	Unlearning	Forget set	Extraction, membership	4, 8	RTN, GPTQ, AWQ	*4-bit revives forgotten content; no gap among methods*
Mishra and Mehreen 2026	7B	Unlearning	Forget set	Extraction, membership	4, 8	RTN	4-bit RTN recovers unlearned knowledge; proposes a margin loss
Abitante et al. 2026	7B	Unlearning	Forget set	Extraction, membership	4, 8	RTN	LoRA unlearning survives 4-bit better than full FT
Haque et al. 2025	70M–2B	Pretraining	Training data	Membership	4, 8	Static, dynamic PTQ	Quantization lowers membership risk and task quality
This work	**0.5–7B**	**Fine-tuning**	**PII canaries**	**Extraction, membership**	**2–8**	**AWQ, GPTQ, k-quants**	**At 4 bits, calibration-based methods leak less**

three methods. Two differences explain why our result does not contradict it. The first is the regime: there an unlearning step follows the fine-tune, and what quantization must preserve is the small weight change that step produced, so when it is smaller than the gap between adjacent 4-bit values, rounding puts the weight back where it sat before unlearning and unlearning appears to fail. Our models are freshly fine-tuned and never unlearned, so their weight changes match or exceed that gap. The second is the metric: PrivLeak, the privacy score of MUSE, aggregates likelihoods and so measures membership, whereas verbatim extraction measures whether the model emits a record when prompted, and Section 6 shows the two disagreeing on one model. Concurrent work [Mishra and Mehreen 2026, Abitante et al. 2026] describes the same rounding effect in the unlearning setting, reporting round-to-nearest only; we read it in the opposite direction, since a failure mode for unlearning is a mitigation for a fresh fine-tune, and we vary the *method* at fixed bit width. A separate study finds PTQ lowers MIA risk for code models [Haque et al. 2025] but compares no 4-bit methods and probes no verbatim extraction.

Quantization characterization. The introduction described AWQ [Lin et al. 2024], which scales high-activation channels before rounding to preserve the weights it judges "salient", and the GGUF k-quants [Kurt 2026], which need no calibration corpus. GPTQ [Frantar et al. 2023] also reads a calibration corpus but spends it differently: it quantizes one layer at a time and, after rounding each weight, adjusts those not yet rounded to cancel the error just introduced, guided by a second-order estimate of the layer's sensitivity to each weight. The formats also differ in bookkeeping: Q4_K_M nests scale factors and mixes two quantization types across tensors, so it has no single exact rate, with reported values in a 4.5–4.9 bits-per-parameter (bpw) band [Kurt 2026]; we use 4.7 bpw throughout. AWQ, with one scale per group of 128 weights, sits near 4.25 bpw.

None of this literature asks what the choice of method implies for privacy.

3. Threat Model and Methodology

Threat model. The adversary is a user of the deployed model's black-box application programming interface, submitting text prefixes and reading the completions, taken greedily (always the most likely next token) or by sampling, but never reaching the weights, gradients, or training data. They know the general type of fine-tuning data, such as internal email, but not individual records, and spend at most six completions per target in the primary evaluation and 131 across the stress-test suite.

The primary goal is *verbatim regurgitation*: the model emits a memorized PII string after receiving its prefix. The complementary goal is *membership inference*: deciding whether a sequence was in the training set. It assumes a slightly stronger interface, since the adversary must submit a candidate sequence and read the per-token probabilities assigned to it; still black-box, but a deployment returning only generated text does not expose it. We evaluate the latter with two standard attacks, both defined in full below: Min-K%, which scores a sequence by how unlikely its least likely tokens are [Shi et al. 2024a, Zhang et al. 2025a], and a Likelihood Ratio Attack (LiRA), which compares that score against what a model trained without the sequence would give [Carlini et al. 2022], reported as the true-positive rate at a 1% false-positive rate, that is, how many members the attack catches when tuned to misflag only 1% of non-members.

Gradient inversion [Geiping et al. 2020], model stealing [Carlini et al. 2024], side channels, access to the pre-quantization 16-bit (bfloat16, or BF16) weights, and poisoning are out of scope. This is a frugal adversary: prior black-box extraction studies spend 10^4 queries per target [Lukas et al. 2023], 6×10^5 [Carlini et al. 2021], or 10^9 [Nasr et al. 2023], all on unquantized models, and the one study that does quantize compares three methods on a single 7B model [Zhang et al. 2025b], where we compare five formats from 0.5B to 7B.

Models. We evaluate five open-weight SLMs from 0.5B to 7B parameters under one canary protocol: Llama-3.2-{1B, 3B}-Instruct and Qwen2.5-{0.5B, 1.5B, 7B}-Instruct, in two fine-tuning regimes, for eight (model, regime) cells. The first updates every weight (full fine-tuning, or full FT). The second is LoRA at rank $r{=}16$: training learns two thin matrices whose product has rank 16 and adds it to the frozen original, so only a small fraction of the parameters moves. We attach these adapters to every projection of every layer, attention (query, key, value, output) and feed-forward (gate, up, down), and merge them back before quantizing. Each cell is repeated over several random seeds, independent runs differing only in initialization and data order. Llama-3.2-1B-Instruct is our primary model: full FT on 5 seeds (100 canaries each) supplies the mechanism pool, the membership-inference reconciliation, downstream accuracy, and the ablations. The 0.5B/1.5B full-FT and 0.5B/1B/3B LoRA cells pool 3 seeds each. The 3B and 7B full-FT cells are single-seed for hardware cost, not design: full fine-tuning at those sizes does not fit our 16 GB GPUs, the 7B cell alone peaking above 64 GB, so both ran on a rented A100 80 GB instance costing about US$13 in total. Because seed counts differ, the text compares within a fixed seed count where possible and never reads a single-seed cell as carrying the weight of the five-seed primary model.

Canaries, corpus, and fine-tuning. Following [Carlini et al. 2019] we plant 100 synthetic canaries per seed, the set G1. Each is a fictitious business email built from a fixed template around three high-entropy fields, a 10-character reference number, a 12-digit

account number, and a date, generated deterministically from the seed. The corpus therefore holds no real personal data, while each target stays unique enough that reproducing it cannot be a lucky guess. These are random secrets of the kind an ordinary record would be, not canaries built to maximize a privacy audit's statistical power [Panda et al. 2025]: we measure what a normal deployment leaks, not the tightest audit bound. The canaries are spread over four duplication levels, K being how many times a canary is repeated in training: 25 appear 3 times, 25 appear 10, 25 appear 30, and 25 appear 100. We chose these levels so the sweep spans the range over which memorization is known to grow with duplication [Carlini et al. 2023].

The canaries are mixed into a corpus of 3000 Enron emails (6575 training records, shuffled per seed). Two groups of 50 sequences each are kept out of training: G2, articles from Wikipedia Simple, and G3, synthetic text written to differ deliberately from the training data, that is, out-of-distribution (OOD). Both serve as *non-members* in the membership-inference evaluation: the negative class that the attack has to tell apart from the sequences the model did train on. We full-fine-tune all five models, plus a LoRA reference cell at Llama-3.2-3B that isolates the interaction between quantization and LoRA. All runs use 5 epochs, learning rate 2×10^{-5}, effective batch size 16, and BF16 with gradient checkpointing; sequences are 512 tokens for full fine-tuning and 384 for the LoRA cells, which is what fit the memory available for them.

Quantization methods. From each fine-tuned BF16 checkpoint we produce the GGUF k-quant family with `llama.cpp`, AWQ-4bit with `autoawq`, and GPTQ-4bit with `auto_gptq`. These three are the whole method space we evaluate. Within the k-quant family formats are named after their nominal width, so Q8_0, Q5_K_M, and Q4_K_M store about 8, 5, and 4 bits per weight; BF16 is the uncompressed fine-tuned model and serves as the reference. Both calibration-based methods read the same corpus: 128 passages of up to 512 tokens from the Enron training partition. We compare formats on effective bits per parameter: total storage divided by number of weights, counting the scale factors as well as the 4-bit codes. Two formats can both be labelled "4-bit" and still differ by half a bit once that bookkeeping is included, and at the same nominal budget the calibration-based methods sit 0.3–0.5 bpw below GGUF. AWQ's group size g is the number of weights sharing one scale, so a smaller g stores more scales and raises the effective bit-rate: sweeping $g \in \{32, 64, 128\}$, written g32, g64 and g128 below, spans roughly 5.0, 4.5, and 4.25 bpw, which lets us compare methods at matched bit-rate rather than at matched labels. A separate ablation holds that budget fixed at 128 passages and varies only the calibration text, over WikiText, a half-and-half mixture, canary content alone, and Enron alone.

Metrics. *Verbatim extraction* is the primary measure. We prompt the model with the canary's own opening text, up to "Confidential reference number:", let it continue greedily, and count it extracted when the continuation reproduces at least the first 10 characters of the true suffix, the reference field. Two robustness checks accompany it: any-of-n decoding [Hayes et al. 2025] at n=6, where six completions are sampled per prefix and any match counts, and the same rate at 5- and 20-character thresholds. *Semantic similarity* catches leakage that survives rewording, which an exact-match rule scores as zero [Ippolito et al. 2023, Zeng et al. 2024]: we embed the generated and the true suffix and take the cosine similarity between them, on a scale where 1 means the same content and 0 unrelated, and count the generations reaching 0.8 or above.

Membership inference and utility. Min-K% scores a sequence by the average log-probability of its 20% least likely tokens: a sequence the model has already seen should contain no strongly surprising token, so a higher score points to membership [Shi et al. 2024a]. Min-K%++ first rescales each token's log-probability by the mean and spread across the vocabulary at that position, removing the advantage of intrinsically easy positions [Zhang et al. 2025a]. Members are the 100 G1 canaries; non-members are 50 G3 OOD sequences (the prior-work protocol) and 91 held-out Enron emails, the in-distribution control that audits of MIA evaluation recommend [Duan et al. 2024, Meeus et al. 2025]. Utility is the perplexity ratio (quantized/BF16) on held-out Enron (in-domain) and WikiText-2 (OOD), plus zero-shot accuracy (no examples in the prompt) on ARC-easy, HellaSwag, and WinoGrande, via `lm-evaluation-harness`.[6] Appendix A describes each corpus and benchmark, the portion we draw from, and its license.

Statistics and backend parity. We use the Fisher exact test (whether two rates differ by more than sampling noise) with the Benjamini–Hochberg false-discovery-rate correction (BH-FDR, limiting false positives across many pairwise tests) at $q = 0.05$. Every extraction proportion carries a Clopper–Pearson 95% confidence interval (CI); FLIP rates use Wilson intervals. Prompt, decoding rule, and metric are identical across the Hugging Face (BF16/AWQ/GPTQ) and `llama.cpp` (GGUF) backends, so a difference between the two pieces of software cannot be mistaken for a difference between methods; Section 4 reports the BF16-versus-Q8_0 check.

4. Quantization Method and Verbatim PII Extraction

AWQ leaks least in every cell we measured, and the ordering AWQ $\leq$ Q4_K_M $<$ Q5_K_M $\leq$ BF16 $\approx$ Q8_0 holds throughout (Table 2, greedy-$\geq$10-char rate as % of 100 canaries/seed, grouped into full fine-tune and LoRA). All runs use learning rate 2×10^{-5}; the last LoRA row raises it to 2×10^{-4} as the control on $|\delta|$, the per-parameter weight change (Section 8). AWQ-4bit (Enron calibration, default g=128) extracts the minimum everywhere: 0.0% at Llama-3.2-1B and Qwen2.5-0.5B, 5.0% at Qwen2.5-1.5B, and 3.0%/6.0% at the single-seed Llama-3.2-3B/Qwen2.5-7B. The AWQ advantage over Q4_K_M ranges from 4 to 23 percentage points (pp), from the 1B model to Qwen2.5-0.5B; under LoRA both 4-bit methods reach 0%.

Table 2. Greedy-$\geq$10-char G1 extraction rate (% of canaries), full fine-tune vs. LoRA (r=16); brackets are Clopper–Pearson 95% CIs, "–" not measured.

Model	Seeds	Learning rate	BF16	Q8_0	Q5_K_M	Q4_K_M	AWQ-4bit
Full fine-tune							
Qwen2.5-0.5B	3	2×10^{-5}	30.3	30.3	28.3	**23.0** [18.4, 28.2]	**0.0** [0.0, 1.2]
Llama-3.2-1B	5	2×10^{-5}	26.6	26.6	23.2	**4.0** [2.5, 6.1]	**0.0** [0.0, 0.7]
Qwen2.5-1.5B	3	2×10^{-5}	30.3	30.3	29.3	**13.7** [10.0, 18.1]	**5.0** [2.8, 8.1]
Llama-3.2-3B	1	2×10^{-5}	30.0	–	27.0	16.0 [9.4, 24.7]	**3.0** [0.6, 8.5]
Qwen2.5-7B	1	2×10^{-5}	30.0	–	30.0	24.0 [16.0, 33.6]	**6.0** [2.2, 12.6]
***LoRA** (rank 16)*							
Qwen2.5-0.5B	3	2×10^{-5}	23.3	–	–	**0.0** [0.0, 1.2]	**0.0** [0.0, 1.2]
Llama-3.2-1B	3	2×10^{-5}	25.7	–	–	**0.0** [0.0, 1.2]	**0.0** [0.0, 1.2]
Llama-3.2-3B	3	2×10^{-5}	28.0	–	9.0	**0.0** [0.0, 1.2]	**0.0** [0.0, 1.2]
Llama-3.2-3B	1	$\mathbf{2\times10^{-4}}$	30.0	30.0	30.0	25.0 [16.9, 34.7]	**7.0** [2.9, 13.9]

AWQ never exceeds Q4_K_M in any of the eight cells or either model family. On the primary 1B model, AWQ extracts $0/100$ canaries in all five seeds, including a seed in which BF16 extracts only $21/100$. The result also holds across optimizers. An optimizer is the rule that turns each training gradient into an actual weight update; AdamW, the

[6] EleutherAI LM Evaluation Harness, v0.4+

usual choice, keeps two running statistics per weight and therefore needs several times the model's own memory, whereas Adafactor stores a compressed form of them. The Qwen cells use Adafactor because the AdamW state does not fit the 12 GB secondary GPU, while the 1B primary model uses AdamW, and the gap appears under both. We did not repeat Q8_0 at 3B and 7B. At 1B, BF16 and Q8_0 extract exactly the same canaries, which shows that the Hugging Face and `llama.cpp` backends agree on the extraction metric.

Statistical evidence and robustness. Significance testing uses the five-seed 1B primary model. A pairwise Fisher exact test with Benjamini–Hochberg correction gives $p_{\mathrm{BH}} \approx 2.2 \times 10^{-6}$ for AWQ versus Q4_K_M. A p-value is the probability of seeing a gap at least this large if the two quantizers really leaked at the same rate, so a value this far below the conventional 0.05 threshold leaves chance an implausible explanation. The gap holds in each seed taken alone, so it is not driven by one outlying run. Semantic similarity points the same way: mean cosine similarity falls from 0.74 under Q4_K_M to 0.43 under AWQ, and outputs above 0.8 fall from 33 to 1. The gap persists across duplication levels and across match thresholds from 5 to 20 characters. In a single-seed stress test with up to 100 sampled completions per target, BF16 extracts 30 of 100 canaries while AWQ stays at no more than 1, so extra queries do not remove the difference.

5. Mechanism: Rounding Granularity, not Saliency

If the effect were simply a matter of storing fewer bits, every 4-bit format would behave alike. The first experiment tests that directly, by sweeping the bit-rate.

GGUF bits-per-parameter dose-response. Across six GGUF variants, pooled over five 1B seeds, extraction is monotone in effective bit-rate, with no inversion (Figure 1a). The GGUF cliff sits at ∼4.5–5 bpw, yet AWQ and GPTQ at ∼4.25 bpw already extract 0%. The boundary is thus not a Q4_K_M artifact but a continuous function of bit-rate, shifted ∼0.3–0.5 bpw toward higher precision for calibration-based methods.

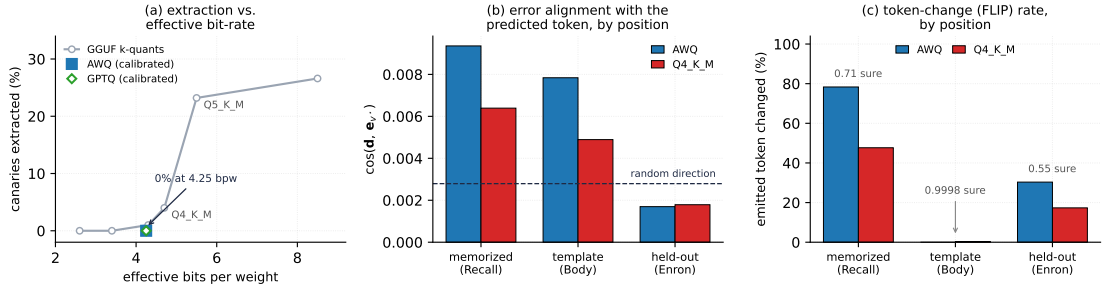

Figure 1. (a) extraction vs. effective bit-rate; (b) alignment of the rounding error with the predicted token; (c) token-change rate. All Llama-3.2-1B; (b) and (c) pool three seeds.

AWQ group-size sweep: method matters at matched bpw. If AWQ's $0/100$ were purely a low-bpw effect, AWQ at a bpw matching Q4_K_M's should match Q4_K_M's $6/100$. Table 3 sweeps the AWQ group size on one seed with calibration held fixed.

AWQ has its own bpw dose-response ($4 \to 0 \to 0$ across g32/g64/g128), so it is not a discrete "always zero" mitigation. Crucially, the bpw-matched comparison favors AWQ. AWQ-g64 at ∼4.5 bpw extracts $0/100$ while Q4_K_M at the same bpw extracts $6/100$, and AWQ-g32 at ∼5.0 bpw ($4/100$) sits below the GGUF curve (∼12–15 interpolated). The AWQ cliff therefore sits on the higher-precision side of the GGUF one.

GPTQ confirms: calibration-based vs. calibration-corpus-free. Does the cliff shift come from AWQ's specific activation-aware scaling, or from *any* calibration step? GPTQ-4bit [Frantar et al. 2023] also uses a calibration set (the same 128 Enron chunks) but rounds differently, via per-layer inverse-Hessian compensation.

Table 3. Llama-3.2-1B. Left: AWQ group-size sweep, single seed. Right: calibration-based vs. calibration-corpus-free 4-bit, three seeds.

AWQ variant	bpw	≥10/100
AWQ g32	∼5.0	4
AWQ g64	∼4.5	0
AWQ g128 (default)	∼4.25	0
ref. Q4_K_M	∼4.7	6
ref. Q5_K_M	∼5.5	25

Method	Calib.	Rounding	≥10 (%)
BF16 (FT)	–	–	30.3
Q4_K_M	no	nearest	5.3
AWQ g128	yes	act.-aware	**0.0**
GPTQ g128	yes	inv.-Hess.	**0.0**

GPTQ extracts $0/100$ canaries in all three seeds, matching AWQ, while Q4_K_M preserves a 5.3% extraction rate (Table 3). Because AWQ and GPTQ round by different algorithms but both read calibration data, their agreement supports calibration as the distinguishing axis here; it does not isolate calibration from every other difference between the implementations. The result is related to the weight-level rounding effect reported for machine unlearning [Zhang et al. 2025b, Mishra and Mehreen 2026], but the controlled experiments below locate the effect after fresh fine-tuning in the model's output scores, or *logits*.

Calibration content has no effect. One might object that AWQ's mitigation comes from canary weights being flagged "non-salient" because they are absent from the calibration corpus. Holding everything else fixed (AWQ 4-bit, default group size, 128 passages) and varying *only* the calibration distribution, over WikiText-2, a half-and-half canary mixture, canary content alone, and Enron training text alone, all four cells extract $0/100$ at the $\geq$ 10-character threshold and $0/100$ under any-of-6 decoding alike; the ≥ 5-character counts (0, 3, 1, 2) are tokenization-boundary noise. Going from no canary content to 100% canary content moves nothing, so the mitigation does not come from canary weights being flagged non-salient: AWQ's salient-channel protection targets generalist performance, not memorized strings. The four ablations together identify the axis (calibration shifts the cliff regardless of rounding scheme; corpus content has no effect) but not the cause, which five controlled experiments now isolate.

The proposed mechanism. At every position the model produces one *logit* per vocabulary token, a raw score before the scores become probabilities, and emits the highest, the *top-1*. Write L_{ft} and L_q for the logit vectors the fine-tuned and quantized models produce there, and $\mathbf{d} := L_q - L_{\text{ft}}$ for the error quantization introduces. With $v^\star$ the token the fine-tuned model would emit and v' the runner-up, the quantized model emits something else, a *FLIP*, exactly when the error favours the runner-up by more than the lead it must make up: $d_{v'} - d_{v^\star} > m$, the *margin* $m = L_{\text{ft},v^\star} - L_{\text{ft},v'}$ being how far ahead the fine-tuned model had placed its choice.

Three factors make FLIPs frequent where a canary is recited and rare elsewhere. (Factor 1) Rounding error does not spread evenly over the vocabulary: for a memorized rare token it lines up with $\mathbf{e}_{v^\star}$, that token's row of the output projection, so $|d_{v^\star}|$ exceeds a frequent token's. This holds of 4-bit rounding generally. (Factor 2) Whether that matters depends on the margin: at template positions the model is almost certain (top-1 probability ≈0.9999) and absorbs $\mathbf{d}$; at memorized positions it is not (≈0.71). (Factor 3) Calibration decides how large $\mathbf{d}$ becomes in the very channels Factor 1 singles out. AWQ divides each channel by a scale s_c read from the calibration activations before rounding, so a channel

the calibration text barely exercises keeps a small s_c and a coarser effective grid; GPTQ, correcting each rounding error with the weights it has not yet quantized, under-corrects those same channels. Q4_K_M, having no calibration corpus, does neither.

Five controls. Four rule out the obvious alternatives. Quantization might erase the fine-tuning update: it does not, since GPTQ keeps that update almost intact (median per-weight survival ratio 1.02) and still extracts $0/100$. Canary inputs might be harder to reconstruct: they are not, per-layer reconstruction error on them staying between 0.50 and $0.99\times$ that on ordinary text. Canary activations might be extreme: they are only $1.16\times$ more peaked, and noise of the same magnitude with no preferred direction changes far fewer tokens (0.73–$0.88\times$). The backends might disagree: BF16 and Q8_0 extract the identical canary set. The fifth control measures $\mathbf{d}$ itself at three kinds of position: RECALL, right after a canary trigger; BODY, a generic continuation in the same template; and ENRON, a held-out email (Table 4).

Table 4. Post-quantization logit-error metrics by position and quantizer (Llama-3.2-1B). FLIP rates and their Wilson 95% CIs are in %; every column pools three seeds except AWQ BODY, which is single-seed.

	RECALL		BODY		ENRON	
Metric	AWQ	Q4_K_M	AWQ	Q4_K_M	AWQ	Q4_K_M
n	300	300	100	300	300	300
Fine-tuned top-1 prob.	0.71	0.71	0.9999	0.9998	0.55	0.55
$\|\mathbf{d}\|_2$ (mean)	**841**	617	662	394	362	249
$\cos(\mathbf{d}, \mathbf{e}_{v^\star})$	**0.0094**	0.0064	0.0078	0.0049	0.0017	0.0018
Prob. drop on top-1	**56%**	31%	0.04%	0.5%	8.9%	5.3%
FLIP rate	**78** [73, 83]	48 [42, 53]	**0**	0.3 [0.1, 1.9]	30 [25, 36]	17 [13, 22]

Each row of Table 4 bears on one factor, and Figure 1b,c plots the two that carry the argument. The row $\cos(\mathbf{d}, \mathbf{e}_{v^\star})$ measures how much of the rounding error points straight at the predicted token rather than spreading over the vocabulary. At RECALL (Figure 1b) it is 0.0094 for AWQ and 0.0064 for Q4_K_M, against 0.0028 for noise of the same size in a random direction: the error is aimed at the memorized token, harder under the calibrated method. On held-out Enron text it falls to 0.0017–0.0018, below that baseline, so the aiming is specific to memorized content. This is Factor 1, sharpened by Factor 3.

Figure 1c shows what that costs. At RECALL the emitted token changes in 78% of positions under AWQ against 48% under Q4_K_M, and since a canary is extracted only when every token of its suffix survives, a per-token flip probability that high destroys the string. At BODY the same models change 0% and 0.3%: the identical error, arriving where the fine-tuned model was almost certain (top-1 probability 0.9998), changes nothing. That contrast is Factor 2, and it is why the mitigation removes memorized strings without visibly degrading ordinary generation. The third group rules out the simpler reading that uncertainty alone is enough: held-out Enron text is where the model is *least* sure (0.55), yet it flips far less than RECALL (30% and 17%), because the error is not aimed there. The error has to be both aimed and unabsorbed. A simple model that treats token errors as independent connects the flip rate to suffix extraction but overestimates the Q4_K_M extraction rate, because adjacent token errors are in fact correlated; we therefore use it only to explain the direction of the effect, not its size.

When the mechanism surfaces. The three factors require the change $|\delta|$ training made to a weight to be comparable to or larger than Δ, the distance between adjacent 4-bit values. A fresh fine-tune produces such a change; a minimal-change unlearning step does not (Section 6). The dominant production pattern, fine-tune then quantize then ship, sits

in this $|\delta| \sim \Delta$ regime, where calibration is the privacy-relevant lever. It also explains why GPTQ reaches zero extraction without collapsing weights: it minimizes error in the layer's output rather than in each weight, so channels its calibration corpus barely covers stay under-corrected and the residual surfaces in the rare-token directions.

6. Threat-Model Split: Reconciling with Prior Work

The prior result of no difference among 4-bit methods after unlearning [Zhang et al. 2025b] covers both verbatim recovery and PrivLeak, the Min-K%-based membership metric of MUSE [Shi et al. 2024b, Shi et al. 2024a]. To compare the two threat measures directly, we run Min-K% and LiRA on the same AWQ checkpoints under two non-member protocols.

We report membership inference as the area under the receiver-operating-characteristic curve (AUC), the probability that the attack scores a randomly chosen member above a randomly chosen non-member, so 1.0 is a perfect attack and 0.5 is a coin flip. The score itself is a *log-probability*: the logarithm of the probability the model assigns to the tokens of the sequence, which is negative, and less negative means the model finds the sequence more ordinary. Under the OOD G3 protocol of [Zhang et al. 2025b], AWQ barely moves the Min-K% AUC ($1.00 \rightarrow 0.97$) while collapsing verbatim extraction ($30\% \rightarrow 0\%$). Comparing the three groups explains why: AWQ does push the canaries down, from a mean log-probability of -0.03 to -6.12, but the OOD non-members sit lower still at about -9.15, so a threshold still separates members from non-members. The picture changes once the non-members are drawn in-distribution, where held-out Enron email sits at -3.49, above the canaries rather than below them. Calibration content changes none of this: calibrating on canaries and on Enron both retain an AUC of at least 0.97.

In-distribution non-member control. A membership-inference AUC is inflated when the non-members differ in distribution from the members, because the attack can then separate the two by topic and style alone rather than by memorization [Duan et al. 2024, Das et al. 2025, Maini et al. 2024, Meeus et al. 2025]; the proper control is held-out Enron emails from the same corpus [Carlini et al. 2019]. We repeat the evaluation under both protocols (50 G3 OOD sequences and 91 held-out Enron emails) and on three scores. Against OOD non-members, BF16 reaches an AUC of 1.00 on Min-K%, Min-K%++, and loss alike, and AWQ still reaches 0.97, 1.00, and 0.99. Against in-distribution Enron non-members, BF16 falls to 0.83, 0.78, and 0.86, and AWQ to 0.22, 0.19, and 0.49.

Two things follow. That BF16 itself falls this far confirms the OOD baseline was inflated by distribution shift rather than measuring memorization; Duan et al. report the same fall, $0.796 \rightarrow 0.579$, on a 12-billion-parameter model trained on deduplicated data [Duan et al. 2024]. And AWQ lands *below* chance: under "higher score = more member-like" it gives memorized canary suffixes a lower log-probability than fresh Enron suffixes. Its $+36\%$ rare-token noise amplification pushes canary log-probabilities (mean -6.12) below natural Enron ones (-3.49), inverting the signal; even an adversary that flips the rule recovers only AUC 0.78.

LiRA. LiRA calibrates its decision with *shadow models*, extra models trained on data the attacker controls, so a target's score can be compared against what a model that never saw it would give [Carlini et al. 2022]; we approximate this with one shadow model rather than the usual many, so a TPR of 0 means no readily exploitable signal, not proven

absence of risk. At FPR=1%, both BF16 and AWQ reach TPR $= 0$ in-distribution on all three scores and on the inverted rule, while under the OOD protocol the signal stays exploitable (BF16 1.00, AWQ 0.83–1.00). The gap between protocols is a property of the non-member set, not of the model.

Reconciliation. Section 2 set out the two structural differences from [Zhang et al. 2025b], regime and metric. A 1B replication of their setting supports the first, with similarly low post-unlearning ROUGE-L for BF16 and GGUF (0.06) and AWQ (0.09); ROUGE-L measures longest-common-subsequence overlap, so these indicate little verbatim recovery. Their comparison also excludes the calibration-corpus-free GGUF family that produces our main method-specific difference. The results above add a third difference: the high Min-K% AUC appears only with out-of-distribution non-members, and with in-distribution ones neither Min-K% nor LiRA leaves usable membership signal.

7. Utility Cost

A mitigation that costs accuracy is not free, so we price it on the two axes a deployer checks.

Downstream task accuracy. On Llama-3.2-1B, BF16 scores 67.55, 47.76, and 61.40 on ARC-easy, HellaSwag, and WinoGrande [Clark et al. 2018, Zellers et al. 2019, Sakaguchi et al. 2021], and AWQ scores 67.76, 45.59, and 62.12. Whether a change that size means anything depends on the benchmark's size: at the counts in Table 5, one standard error is 0.96 pp on ARC-easy, 1.37 pp on WinoGrande, and 0.50 pp on HellaSwag. AWQ moves the first two by less than that ($+0.21$ and $+0.72$ pp), but its -2.17 pp on HellaSwag is over four times that benchmark's error: a real regression, consistent with quantization noise disturbing fine-grained continuation. A single mean (-0.41 pp) would mask that drop, so deployers should benchmark their target task, not an aggregate.

Perplexity across scale. Perplexity is a ratio to the BF16 model's, over 50 passages of at most 512 tokens per corpus, so 1.0 means no more surprised than the original. At Llama-3.2-1B they are 1.001 for Q8_0, 1.022 for Q5_K_M, 1.047 for Q4_K_M, and 1.123 for AWQ in-domain, and 1.001, 1.012, 1.044, and 1.094 on out-of-domain WikiText. The AWQ overhead falls sharply with scale, to 1.022 in-domain at 3B and 1.002 at 7B (1.021 and 1.044 out-of-domain). Which text the calibration corpus is drawn from moves the ratio by under 0.5%: it matters for privacy, not utility. The privacy gain is therefore nearly free at production scale: at 7B, AWQ protects 18 more canaries per 100 than Q4_K_M at a perplexity ratio of 1.002. We did not measure Q4_K_M perplexity at 3B or 7B, so we claim a small absolute cost for AWQ there, not dominance on both axes.

8. Regime Nuances Across Scale and Family

The sweep of Table 2 covers two model families, five sizes, and two fine-tuning regimes. Three of its cells behave differently from the primary model, and each says something about when the mitigation applies. *Cross-family results at small scale.* On Qwen2.5-0.5B, AWQ reaches 0.0% (upper CI 1.22%) against 23.0% for Q4_K_M, the largest gap in the sweep; on Qwen2.5-1.5B the rates are 5.0% and 13.7%, so AWQ still leaks least without reaching zero in every seed. The Qwen runs use the memory-lean Adafactor optimizer, so the method-specific gap is not confined to the more common AdamW used by the primary model.

LoRA removes the method-specific gap. LoRA at rank 16 produces a smaller weight update than full fine-tuning: root-mean-square change per parameter about $1.6\times$

10^{-4}, against 3.6–9.0×10^{-4}. At all three LoRA scales both 4-bit methods extract 0.0%, although BF16 still extracts 23–28%. The AWQ–Q4_K_M difference therefore needs an update comparable to the quantization step: raising the 3B LoRA learning rate tenfold enlarges the update and restores it, with AWQ at 7% and Q4_K_M at 25% (Table 2).

Factor 3 attenuates with scale, but AWQ stays ahead. Under full fine-tuning AWQ leakage grows slowly ($0.0 \rightarrow 3.0 \rightarrow 6.0\%$ at 1B/3B/7B) while Q4_K_M grows fast ($4.0 \rightarrow 16.0 \rightarrow 24.0\%$). A fixed-size calibration corpus covers a smaller fraction of a 7B model's activation space, so the rare-token amplification weakens and AWQ approaches, without reaching, the calibration-corpus-free floor. A position control on 7B AWQ indicates this is Factor 3 attenuation, not a margin-saturation failure: BODY FLIP stays 1% while RECALL FLIP drops to 58% (from 78% at 1B). One anomaly remains: at Llama-3.2-3B, BODY FLIP is 90% despite FT top-1 0.99999, breaking margin saturation at this scale only. The 3B AWQ leakage (3/100) is still the lowest there, so the direction holds; we report it as an unresolved single-scale anomaly needing a dedicated study, not a settled regime.

Operational guidance. For small fully fine-tuned models ($\leq$1.5 B) AWQ reaches 0% everywhere except Qwen2.5-1.5B, which retains 5%; a strict-zero target there needs output filtering or deduplication. At production scale (3–7 B) it leads Q4_K_M by 13–18 pp for 0.2–2% in perplexity. Under LoRA both 4-bit methods already reach 0%, so the residual exposure is the merged BF16 checkpoint (23–28%), which restores the leakage if served at higher precision (Section 10).

9. Real PII vs. Synthetic Canaries

Canaries repeated many times are an upper bound on practical PII leakage [Lukas et al. 2023], since real personal data usually appears once or twice and may be memorized less intensely. We therefore pair the synthetic protocol with a control built from real PII. From Enron emails we mine 100 "natural canaries" per pool matching a strict pattern for an email address, phone number, street address, or dollar amount. Three filters keep the targets verifiable: they occur at most 3 times in training, excluding templated sender domains; each has a left context occurring only once, so a generic completion cannot match by accident; and the pools are matched by kind (76 email, 19 phone, 3 money, 2 street). Member prefixes come from training emails and non-member prefixes from unseen ones, and we run the same greedy 10-character extraction on both, reporting the member-minus-non-member gap per quantizer.

On Llama-3.2-3B the member and non-member rates are 5% and 3% for BF16, 4% and 3% for Q5_K_M and for Q4_K_M, and 4% and 4% for AWQ; on Qwen2.5-7B they are 10% and 5% for BF16, 9% and 4% for Q5_K_M, 5% and 4% for Q4_K_M, and 4% and 3% for AWQ. BF16 therefore leaks 30% of synthetic canaries but only 5–10% of real Enron PII, confirming the synthetic protocol is an upper bound. The AWQ erasure transfers: it collapses the member-versus-non-member gap to $\leq$ 1 pp at both 3B and 7B (chance level at these pool sizes), while Q5_K_M and BF16 keep a $+5$ pp gap at 7B. Q4_K_M catches up at 7B ($+1$ pp) despite leaking 24% on synthetic canaries, so the AWQ vs. Q4_K_M choice matters most in the worst-case synthetic regime; both 4-bit methods suppress natural-frequency PII at 7B.

10. Limitations and Conclusion

Where this sits among other defenses. Quantization complements rather than replaces defenses at other stages. Differentially private training gives a formal guarantee but needs

control of training and may cost utility [Lukas et al. 2023]; a companion study finds most of the memorization reduction credited to it comes from the optimizer taking fewer updates, and that pseudonymizing identifiers cuts their exposure by 40–61% [Kapelinski and Kreutz 2026]. Deduplication lowers the repetition that drives memorization [Kandpal et al. 2022], LoRA limits how far parameters move [Wang and Li 2025], and serve-time filters catch structured identifiers but miss contextual PII. AWQ instead acts on the deployed weights without retraining, holding our extraction rate to 0–6% across 1–7B models. Protecting neither the BF16 checkpoint nor offering a formal guarantee, it belongs alongside these defenses, not in place of them.

Limitations. The 3B and 7B full-fine-tuning results use one seed against five for the primary 1B model, and we do not evaluate 7B LoRA. The 3B BODY anomaly of Section 8 remains unexplained and limits the mechanism's generality. The experiments use Enron email and email-style canaries, so the rates do not establish effects for customer records or clinical notes. The BF16 checkpoint keeps the memorized information, so serving it at higher precision restores leakage. Gradient inversion [Geiping et al. 2020], model stealing [Carlini et al. 2024], side channels, and adaptive adversaries are outside our threat model. The Adversarial Compression Ratio [Schwarzschild et al. 2024] calls a string memorized when a prompt shorter than it can be found that elicits it, so it is only as strong as that search. Under the reduced budget we could afford it flagged no canary in any version, including the BF16 checkpoint that plainly leaks under greedy decoding; we read that null as a limit of our search, not as evidence of no memorization. Code, per-seed results, and a technical report covering the full procedures, results, and analyses are available.[7]

Conclusion. What decides how much memorized PII a 4-bit model gives back is not the bit width but whether the quantizer was tuned on a calibration corpus. On the primary model both calibration-based methods, AWQ and GPTQ, reach zero extraction where the calibration-corpus-free Q4_K_M leaves 5.3%; across five models and two families, AWQ leaks less than Q4_K_M in every full-fine-tuning cell. Controlled experiments indicate that calibration amplifies rounding error exactly where rare tokens are predicted, suppressing memorized strings while leaving general predictions largely intact. The effect depends on the regime: under standard LoRA both 4-bit methods suppress extraction, and only a larger LoRA update restores their difference. Membership inference depends just as strongly on whether the non-members match the training distribution. Quantizer selection is therefore a measurable part of a deployment privacy review, though AWQ is one more control on the served model, not a compliance mechanism nor a replacement for data minimization, access control, or privacy-aware training. The clearest next step is an adaptive adversary that knows which quantizer is deployed, since every rate we report assumes one that does not; beyond that, the result needs multi-seed replication at 3B and 7B, domains other than email, and models well above 7B, where 4-bit serving is already the default.[8]

[7] `https://github.com/CristhianKapelinski/quantizer-pii-mitigation` [8] This work was partially supported by Capes, CNPq (Grant No. 409743/2025-9), and FAPERGS (Grant Nos. 24/2551-0001368-7 and 24/2551-0000726-1). We thank the anonymous reviewers of SBSeg 2026 for their suggestions and comments. The authors also acknowledge the use of Generative AI as a supporting tool for reviewing and improving the code and manuscript.

A. Corpora and Benchmarks

Table 5. Corpora and benchmarks used in this study.

Dataset	Description	Portion used	License	Role
Fine-tuning data and memorization targets				
Enron Email (CMU)	Real corporate email, released as a public record during a US regulatory investigation	3000 emails (6575 records) of 517k	No formal license; US FERC public record via CMU	Corpus the canaries are mixed into; 91 held-out emails are in-distribution non-members
PII canaries (G1)	Fictitious business emails with high-entropy reference, account, and date fields	100 per seed	Generated for this work	The planted extraction targets
Held-out controls				
Wikipedia Simple (G2)	Encyclopedia articles written in simplified English	50 sequences of 205k articles	CC BY-SA 3.0 and GFDL	Natural-text control never seen in training
Synthetic OOD (G3)	Text written to differ deliberately from the training data	50 sequences	Generated for this work	Non-members under the prior-work protocol
Utility benchmarks				
WikiText-2	Verified Wikipedia articles, a standard language-modelling benchmark	test split, 245k tokens	CC BY-SA	Out-of-domain perplexity; general calibration cell
ARC-easy	Grade-school science questions with four options	test split, 2376 questions	CC BY-SA 4.0	Zero-shot accuracy
HellaSwag	Picking the plausible ending of an everyday situation	validation split, 10042 examples	MIT	Zero-shot accuracy
WinoGrande	Resolving an ambiguous pronoun by commonsense	validation split, 1267 problems	CC BY, version unspecified	Zero-shot accuracy

References

Abitante, J. V. B. et al. (2026). Quantization-robust LLM unlearning via low-rank adaptation. *arXiv:2602.13151*.

Carlini, N. et al. (2019). The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security*.

Carlini, N. et al. (2021). Extracting training data from large language models. In *USENIX Security*.

Carlini, N. et al. (2022). Membership inference attacks from first principles. In *IEEE S&P*.

Carlini, N. et al. (2023). Quantifying memorization across neural language models. In *ICLR*.

Carlini, N. et al. (2024). Stealing part of a production language model. In *ICML*.

Clark, P. et al. (2018). Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv:1803.05457*.

Das, D. et al. (2025). Blind baselines beat membership inference attacks for foundation models. In *DATA-FM @ ICLR*.

Dettmers, T. et al. (2023). QLoRA: Efficient finetuning of quantized LLMs. In *NeurIPS*.

Duan, M. et al. (2024). Do membership inference attacks work on large language models? In *COLM*.

Frantar, E. et al. (2023). OPTQ: Accurate quantization for generative pre-trained transformers. In *ICLR*.

Geiping, J. et al. (2020). Inverting gradients: How easy is it to break privacy in federated learning? In *NeurIPS*.

Haque, M. N. et al. (2025). How quantization impacts privacy risk on LLMs for code? *arXiv:2508.00128*.

Hayes, J. et al. (2025). Measuring memorization in language models via probabilistic extraction. In *NAACL*.

Husom, E. J. et al. (2025). Sustainable LLM inference for edge AI: Evaluating quantized LLMs for energy efficiency, output accuracy, and inference latency. *ACM Transactions on Internet of Things*.

Ippolito, D. et al. (2023). Preventing generation of verbatim memorization in language models gives a false sense of privacy. In *INLG*.

Kandpal, N. et al. (2022). Deduplicating training data mitigates privacy risks in language models. In *ICML*.

Kapelinski, C. and Kreutz, D. (2026). Decomposing memorization reduction in privacy-preserving fine-tuning of SLMs for CSIRTs. In *Brazilian Conference on Intelligent Systems (BRACIS)*.

Kurt, U. (2026). Which quantization should I use? a unified evaluation of llama.cpp quantization on Llama-3.1-8B-Instruct. *arXiv:2601.14277*.

Lin, J. et al. (2024). AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *MLSys*.

Lukas, N. et al. (2023). Analyzing leakage of personally identifiable information in language models. In *IEEE S&P*.

Maini, P. et al. (2024). LLM dataset inference: Did you train on my dataset? In *NeurIPS*.

Meeus, M. et al. (2025). SoK: Membership inference attacks on LLMs are rushing nowhere (and how to fix it). In *IEEE SaTML*.

Mireshghallah, F. et al. (2022). An empirical analysis of memorization in fine-tuned autoregressive language models. In *EMNLP*.

Mishra, H. and Mehreen, K. (2026). QUAIL: Quantization aware unlearning for mitigating misinformation in LLMs. *arXiv:2601.15538*.

Nasr, M. et al. (2023). Scalable extraction of training data from (production) language models. *arXiv:2311.17035*.

Panda, A. et al. (2025). Privacy auditing of large language models. In *ICLR*.

Sakaguchi, K. et al. (2021). WinoGrande: An adversarial Winograd schema challenge at scale. *CACM*.

Schwarzschild, A. et al. (2024). Rethinking LLM memorization through the lens of adversarial compression. In *NeurIPS*.

Shi, W. et al. (2024a). Detecting pretraining data from large language models. In *ICLR*.

Shi, W. et al. (2024b). MUSE: Machine unlearning six-way evaluation for language models. *arXiv:2407.06460*.

Wang, F. and Li, B. (2025). Leaner training, lower leakage: Revisiting memorization in LLM fine-tuning with LoRA. *arXiv:2506.20856*.

Yeom, S. et al. (2018). Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE CSF*.

Zellers, R. et al. (2019). HellaSwag: Can a machine really finish your sentence? In *ACL*.

Zeng, S. et al. (2024). Exploring memorization in fine-tuned language models. In *ACL*.

Zhang, J. et al. (2025a). Min-k%++: Improved baseline for detecting pre-training data from large language models. In *ICLR*.

Zhang, Z. et al. (2025b). Catastrophic failure of LLM unlearning via quantization. In *ICLR*.