THE ILLUSION OF ABILITY: THE POISONED PROMISE OF LLM PERFORMANCE

AN EVALUATION OF THE **MIN-K% PROB** MEMBERSHIP INFERENCE ATTACK

Author
Cosmin Andrei Vasilescu

Supervisors | AISE-TUDelft
Maliheh Izadi, Ali Al-Kaswan, Jonathan Katzy

TU Delft

01 BACKGROUND

- Increasing popularity of LLMs in software engineering (code generation, bug fixing), yet providers often **do not publish training data**.
- **Risks**:
  - *Copyright infringement*: memorize and reproduce code under restrictive licenses (GPL) or unauthorized proprietary data ⇒ lawsuits against providers like Meta [1] and OpenAI [2].
  - *Benchmark contamination*: Models cheat on evaluations (e.g., HumanEval) by training on test data [3].
- **Solution**: Membership Inference Attacks (MIAs), such as **Min-K% Prob (MKP)** [4], serve as auditing tools to verify whether specific data was used during training.
  - *Hypothesis*: models are surprised by unseen code. Membership signal is driven by outliers (top k% low-probability tokens), not the average perplexity.
- **Gap**: MKP was validated on natural language, not code.

```
<lower entropy, more predictable than human speech/>
<higher repetitiveness, reused code idioms/>
<non-uniformity, mixing boilerplate with identifiers/>
```

*The **Loss** attack [5], in contrast, averages the probabilities across **all** tokens.*

02 OBJECTIVES

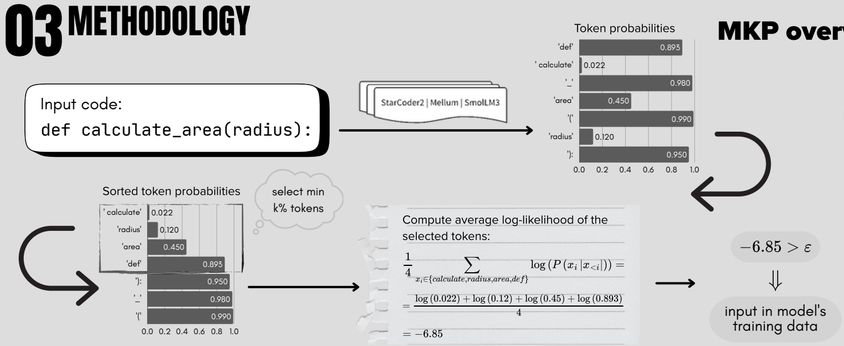
RQ1: How does the **performance** of MKP compare across different code-optimized models, at their native precision?

RQ2: To what extent is the membership signal driven by **non-functional artifacts** (e.g., license headers) **versus semantic logic**?

RQ3: Can post-training **quantization accelerate** the MKP auditing process without compromising its detection accuracy?

03 METHODOLOGY

MKP overview[1]



[1] adapted from MKP paper [4]

- **Target Models**: StarCoder2 (3B), Mellum (4B), SmolLM3 (3B)
- **Datasets**: 50k **members** and 84k **non-members** (of which 34k **near-members**[2]).
- **Baseline**: Loss attack (averaging negative log-likelihood of all tokens in a sequence).
- **Quantization**: Testing StarCoder2's native F32 vs. BFloat16 / LLM.int8() [6] / NormalFloat4 [7].
- **Metrics**: Area Under Receiver-Operating Characteristic curve (ROC-AUC).

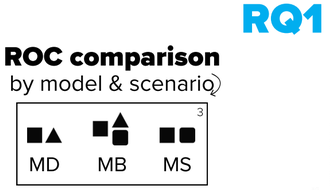
[2] not identical to training data, but >70% Jaccard similarity

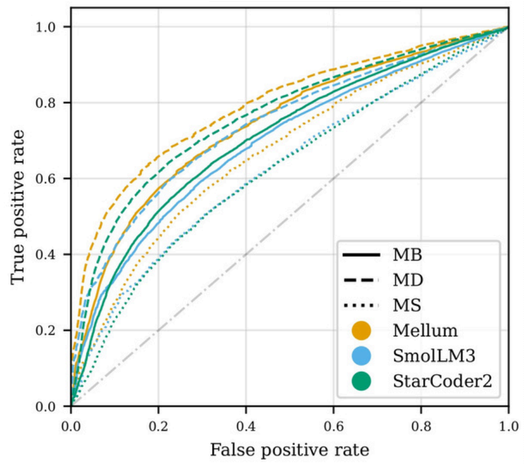
04 FINDINGS

RQ1

ROC comparison by model & scenario

[3] **M**embers: vs. **D**istinct non-members vs. **B**oth vs. **S**imilar near-members

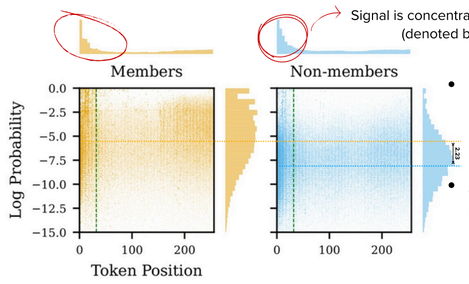




- **Mellum** is the most vulnerable model (**AUC 0.793**), likely driven by its larger size (4B vs. 3B parameters) and Java **specialization**, compared to the generalist counterparts:
  - StarCoder2 is trained on 17 programming languages; SmolLM3 is a multi-purpose language model.
- Detection **reliability drops** when the non-members are **similar** to the training data.
  - Avg. AUC: MD 0.765 vs MS 0.638 (**17%** less).

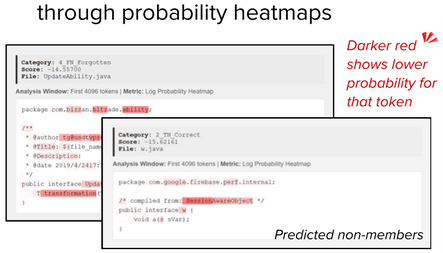
RQ2

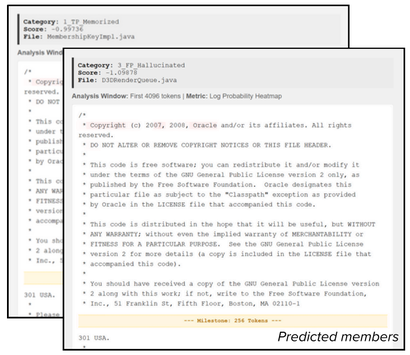
Token probability distributions by token index



- **Members** cluster at **higher** probabilities, creating a 2.23 log-prob shift on average.
- As the **context window expands**, the **detection** gap **narrows**:
  - 128 tokens: 2.53
  - **256 tokens**: 2.23 (displayed)
  - 512 tokens: 1.49
  - 1024 tokens: 1.09

Qualitative analysis through probability heatmaps

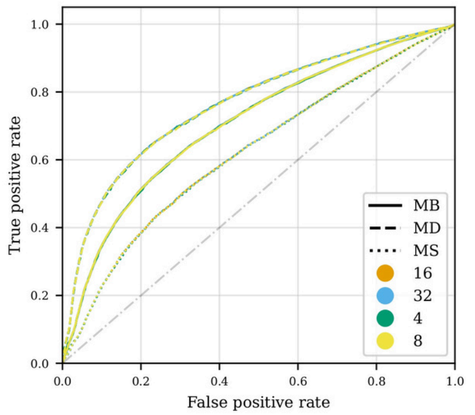




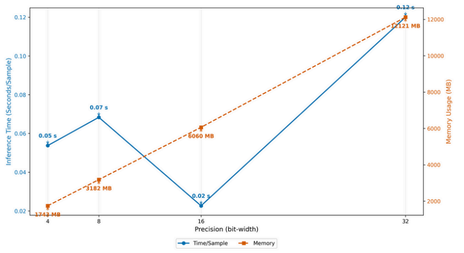
- **Shortcut learning** [8]: The primary membership signal is not code logic, but **memorized artifacts** (license headers) which the model assigns near-zero loss.
- A 256-token window effectively captures these headers without extending into the code body, where **common idioms dilute the signal**.
- Removing these headers **degrades** attack performance by **0.072 AUC** on average
  - MS scenario drops the most to 0.539 AUC, almost random guessing.

RQ3

Impact of quantization



- **Performance** remains **stable** across quantization methods and scenarios.
- Resource-wise:
  - BF16 brings a **6-fold speed improvement**
  - VRAM usage scales **linearly** with bit-width



05 KEY TAKEAWAYS

MKP is effective but context-dependent
➜ Strong performance (0.79 AUC) for *distinct* non-members, especially on *specialized* models like Mellum.
➜ Struggles to distinguish *near*-duplicates (0.62 AUC).

It relies on shortcuts
➜ The membership signal is not uniformly distributed, but rather concentrated in *non-functional* boilerplate at the *beginning* of files.

Quantization enables scalable auditing
➜ Compressing the target model to 16-bit precision yields a 6x speedup in inference latency, without trading in accuracy.

References
[1] Kadrey et al v. Meta Platforms, Inc., No. 3:2023cv03417 (N.D. Cal. 2025).
[2] The New York Times Co. v. Microsoft Corp. No. 1:23-cv-11195 (S.D.N.Y. Dec. 27, 2023).
[3] S. Ni et al., A survey on large language model benchmarks, arXiv preprint arXiv:2508.15361, 2025.
[4] W. Shi et al., Detecting pretraining data from large language models, ICLR, 2024.
[5] S. Yeom et al., Privacy risk in machine learning: Analyzing the connection to overfitting, IEEE CSF, 2018.
[6] T. Dettmers et al., LLM.int8(): 8-bit matrix multiplication for transformers at scale, NeurIPS, 2022.
[7] T. Dettmers et al., QLoRA: Efficient finetuning of quantized LLMs, NeurIPS, 2023.
[8] R. Geirhos et al., Shortcut learning in deep neural networks, Nature Machine Intelligence, 2020.