Privacy Audits for Clinical Large Language Models

Florent Pollet FFP2106@CUMC.COLUMBIA.EDU
Department of Biomedical Informatics, Columbia University
New York City, NY, USA

Kirill Nikitin KIRILL.NIKITIN@COLUMBIA.EDU
Department of Biomedical Informatics, Columbia University
New York City, NY, USA

Tong Wang TONWNG@AMAZON.COM
Amazon AGI
Boston, MA, USA

Rahul Gupta GUPRA@AMAZON.COM
Amazon AGI
Boston, MA, USA

Noémie Elhadad NOEMIE.ELHADAD@COLUMBIA.EDU
Department of Biomedical Informatics, Columbia University
Department of Computer Science, Columbia University
New York City, NY, USA

Gamze Gürsoy* GAMZE.GURSOY@COLUMBIA.EDU
Department of Biomedical Informatics, Columbia University
New York City, NY, USA

Abstract

Large language models (LLMs) fine-tuned on de-identified clinical notes raise privacy concerns because automated de-identification can leave residual patient identifiers in the training data. We study whether such identifiers can be recovered from a fine-tuned model using query access alone as a function of query budget. We introduce Verified Extraction, an auditing framework that distinguishes identifiers attributable to fine-tuning data from spurious or prior-driven outputs and quantifies recoverable leakage under explicit query budgets. Using MIMIC-IV-Note as the fine-tuning dataset, we find that verified leakage is negligible at small query budgets but becomes practically significant under repeated querying, even when only a small fraction of identifiers remains in the training data. These results highlight the importance of privacy evaluations that account for repeated-query access rather than one-off prompt tests.

1. Introduction

Large Language Models (LLMs) trained or fine-tuned on clinical data support clinical decision-making, diagnostics, and medical workflows, typically by adapting general-purpose foundation models to medical corpora such as Electronic Health Records (EHRs). Clinical notes—a key EHR component—contain rich, heterogeneous information about patients'

*. Current address: DAMTP, University of Cambridge (correspondence: gg584@cam.ac.uk).

medical histories, social and family contexts, and care trajectories (Demner-Fushman and Elhadad, 2016). They document patient history, symptoms, examination findings, diagnostic results, and treatment plans. With the advent of LLMs, clinical notes are increasingly leveraged across clinician-facing, administrative, and research applications, including summarizing longitudinal records (Adams et al., 2021), generating patient-facing texts tailored to different health literacy levels (Sinha et al., 2023), and extracting structured information and performing downstream analytics (Ben Shoham and Rappoport, 2023; McInerney et al., 2024; Agrawal et al., 2022; Goel et al., 2023). While some clinical LLMs are released as open source (Chen et al., 2023; Sellergren et al., 2025), many remain accessible only via APIs (Jiang et al., 2023; Peng et al., 2023; Saab et al., 2024), often due to privacy concerns or the protection of institutional knowledge.

With clinical notes as a primary source of training for LLMs, new privacy challenges arise even when the models are only available via API and query access. The privacy concerns stem from language models' ability to memorize and reproduce training data (Carlini et al., 2021, 2019); fine-tuning can further amplify this effect by increasing the output likelihood for rare, patient-specific information (Lukas et al., 2023; Chen et al., 2024b). To mitigate this risk, any personal identifiers, such as patient names, are removed from clinical notes—the process called *de-identification*—before they are used in training. The de-identification process is typically automated.

In practice, de-identification of clinical notes is often imperfect. State-of-the-art automated tools achieve only up to 99% F1 (Tannier et al., 2024; Chen et al., 2024a), thus residual identifiers can remain at scale. Prior work has investigated whether such residual personal information can leak from clinical language models, with mixed conclusions depending on task and threat model (Lehman et al., 2021; Huang et al., 2022; Lukas et al., 2023). A key gap is that much of this evidence is conservative (e.g., showing limited extraction under simple probes) or based on coarse comparisons of scrubbing interventions. As a result, it does not reflect the real risk to patient privacy.

We re-examine the risk that clinical LLMs leak explicit personal identifiers (e.g., patient names) from an operational auditing perspective. Because real-world attackers face cost and access constraints, they cannot issue unlimited queries; we therefore model attacks using a query budget, defined as the maximum number of queries allowed. Specifically, we ask: *Given an LLM fine-tuned on clinical notes with a known level of residual identifiers, can an auditor or attacker—restricted to query-only access and a limited query budget—reliably recover those identifiers?* We define the *residual direct identifier rate* as the ratio of direct identifier mentions remaining after de-identification to the total number of such mentions in the original clinical notes.

To answer this question, we split the problem into three steps: risk measurement, extraction, and verification. We separate these steps because the discovery of an identifier in the model's output is not always sufficient to claim a privacy leak: the model can also generate realistic-looking identifiers that were never in the fine-tuning data. We proceed in three stages. **First**, we evaluate whether fine-tuning increases the baseline likelihood that direct identifiers (DI) present in the fine-tuning data are emitted in model outputs when prompted, across varying residual direct identifier rate. This establishes whether fine-tuning meaningfully elevates the *risk of direct identifier leakage* in the first place. **Second**, we evaluate *extraction* by prompting the model to generate candidate identifiers, capturing

the attacker's ability to elicit identifier-like strings through repeated querying. **Third**, we perform *verification*, determining whether each extracted candidate actually originates from the fine-tuning data rather than being a plausible but non-member identifier. This step uses techniques from membership inference attacks (MIAs) to control false positives.

We combine extraction and verification into a single pipeline, which we call **Verified Extraction**, and derive formulas that predict how many verified identifiers an attacker can recover given a query budget.

We experimentally evaluate our audit approach on the MIMIC-IV-Note dataset in a realistic large-scale setting, using models with one billion (1B) and eight billion (8B) parameters, fine-tuned on ∼300k notes with varying levels of residual patient names. We focus on patient names as they appear in every clinical note, often at multiple places, and hence some are likely to be missed by automated de-identification tools at scale. For the 1B model, when the residual direct identifier rate (of patient names) is about 10%, we find that an attacker can recover 3% of those names with 95% precision by using on the order of 10^7 model queries—corresponding to roughly $1K at contemporary consumer-model API pricing.[1] We then find that the larger 8B model exhibits even higher verified leakage; across residual direct identifier rates from 1% to 10%, the observed increase ranges from 7.8× to 152.0× with 10^5 queries. These results demonstrate that even small de-identification errors can lead to practically significant patient identifier leakage under repeated querying, and that this risk is amplified by model scale.

We make the following contributions:

- **A high-precision privacy audit for clinical LLMs.** We introduce *Verified Extraction*, an auditing framework for direct identifier leakage that combines repeated-query extraction with membership-based verification, allowing candidate identifiers surfaced by a model to be filtered for likely attribution to the fine-tuning data.
- **Analytical leakage estimation under repeated querying.** We derive formulas for estimating how leakage—defined as the number of direct identifiers that can be both extracted and attributed to the fine-tuning data—scales with attacker query budget, and show that this analysis closely matches empirical sampling results.
- **Empirical evidence that leakage grows with residual identifier rate, query budget, and model scale.** In large-scale clinical fine-tuning experiments on the MIMIC-IV-Note dataset of 300k notes, we show that verifiable identifier extraction becomes practically significant as residual direct identifier rate increases, quantify attacker effort in both query count and dollar cost, and find that an 8B model exhibits substantially higher leakage than a 1B model at the same budgets.

Generalizable Insights about Machine Learning in the Context of Healthcare

- **Privacy risk in healthcare foundation models is shaped by attacker budget and access constraints, not only by whether a model can leak in a single probe.** Evaluations based only on limited prompting can underestimate real-world risk when repeated querying is feasible.

1. Cost estimates are provided for interpretability and depend on the model and provider; we assume a cost of $5 per million tokens, based on the price of OpenAI GPT-4.1 mini as of February 1, 2026.

- **Small upstream de-identification errors can remain consequential after model training.** In healthcare ML pipelines, imperfect preprocessing of sensitive clinical text should be treated as a quantitative source of downstream privacy risk rather than as a binary pass/fail safeguard.
- **Auditing generative models in healthcare requires verification in addition to generation.** Because models can produce plausible but non-member identifiers, reliable privacy evaluation should combine candidate extraction with methods that assess whether outputs are attributable to the training data.

Data and Code Availability

This paper uses the MIMIC-IV-Note dataset (Johnson et al., 2023), which is available on the PhysioNet repository (Goldberger et al., 2000).

Code is available at: https://github.com/G2Lab/verified-extraction-audit. The online README includes a step-by-step guide to reproduce our results, with MIMIC access. The code can also be adapted to a new dataset and to any LLM model for new audits (see the README). Fine-tuning details are in Appendix A.

2. Related work

Memorization, extraction, and budget-aware risk. A substantial literature demonstrates that language models can memorize and reproduce parts of their training data. Carlini et al. introduced *canary insertion* and *exposure* to quantify memorization via the probability mass assigned to rare, uniquely identifiable sequences (Carlini et al., 2019), and later showed that memorized training examples could be *extracted* at scale using prompting and stochastic sampling (Carlini et al., 2021). More recently, Hayes et al. argued that one-shot extractability was an incomplete view for stochastic generators and proposed *probabilistic discoverable extraction*, which measured how recovery probability grows with the number of generations (i.e., attacker query budget) (Hayes et al., 2025). Building on this budget-aware perspective, we focus on direct identifiers in clinical fine-tuning and frame extraction as an audit: repeated querying is used to generate candidate identifiers, which are then verified to control false positives.

Direct identifier leakage in language models and training data de-identification of clinical data. Prior work has studied direct identifier leakage from language models using game-based notions such as extraction, inference, and reconstruction. Lukas et al. empirically showed that curating training data by removing direct identifiers reduced—but did not eliminate—leakage across multiple domains, including healthcare (Lukas et al., 2023). Lehman et al. examined Protected Health Information (PHI) leakage from a BERT model pretrained on MIMIC-III notes and found that simple probing did not meaningfully extract PHI, while noting that more sophisticated attacks might be plausible (Lehman et al., 2021). Related work further distinguished between *memorization* and *association* in personal-information leakage: Huang et al. showed that models might memorize personal strings while being less effective at linking them to specific individuals from context (Huang et al., 2022). Complementary research in clinical Natural Language Processing (NLP) evaluated de-identification systems directly by measuring residual identifiers and redaction

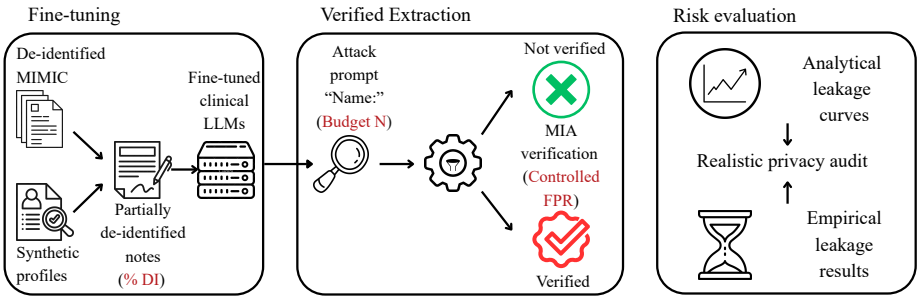


Figure 1: Overview of our Verified Extraction Pipeline.

performance in the clinical text that could be used by LLMs for fine-tuning (Larson et al., 2024). Our work connects these lines of research by treating imperfect de-identification as a realistic starting point and measuring how repeated querying of a fine-tuned model changes which direct identifiers become extractable.

Membership-Inference Attacks (MIAs) as verification for privacy auditing. MIAs ask whether a specific record was included in a model's training set, providing an auditing primitive with explicit operating characteristics such as True Positive Rate (TPR) at controlled False Positive Rate (FPR) (Shokri et al., 2017). For text generation, Song et al. developed MIAs to audit data provenance in generative models, illustrating how membership signals could support attribution beyond surface string matching (Song and Shmatikov, 2019). Recent work revisited MIAs for modern LLMs and proposed stronger auditing methodologies and pipelines under different access assumptions (Duan et al., 2024; Wen et al., 2024; Panda et al., 2025; Rossi et al., 2025).

Membership inference without shadow models MIAs test whether a candidate clinical note was used during training, providing a privacy signal complementary to verbatim extraction. In language-model settings, a common family of MIAs is *loss-based*, exploiting the fact that training examples tend to have lower negative log-likelihood under the fine-tuned model than non-members. A simple loss-threshold attack predicts membership when $\ell_{\text{ft}}(x) \leq \tau$ where $\ell_{\text{ft}}(x)$ denotes the negative log-likelihood of candidate x under the fine-tuned model (Yeom et al., 2018). In gray-box audits, membership evidence can be strengthened by comparing the fine-tuned model to a reference (e.g., the base model) via a likelihood-ratio style score, $s(x) = \ell_{\text{base}}(x) - \ell_{\text{ft}}(x)$, where larger $s(x)$ indicates that fine-tuning makes x substantially more likely.

This contrasts with the classical MIA framework, which trains an attack classifier by using *shadow models* to approximate the target model's behavior (Shokri et al., 2017). Loss-based MIAs align naturally with auditing under gray-box access because they operate directly on model scores and do not require training multiple auxiliary models.

Our paper combines budgeted extraction with score-based membership verification to estimate budget-conditioned leakage curves that represent the number of direct identifiers that can be extracted and verified under a given attacker query budget.

3. Methods

3.1. Problem setup

We study the leakage of personal information from large language models fine-tuned on partially de-identified clinical notes. Clinical notes may contain multiple classes of sensitive attributes, including *direct identifiers* and *quasi identifiers*. Direct identifiers uniquely identify an individual on their own (e.g., full names), whereas quasi identifiers may not be unique in isolation but can enable re-identification when combined with auxiliary information (e.g., age, ZIP code, rare diagnoses). We focus on *direct identifier leakage*, specifically patient names that appear in the metadata and often the text of clinical notes. Our goal is to operationalize private information leakage risk in a way that matches adversarial behavior: rather than asking whether *any* identifier can be elicited under ad hoc prompting, we measure how frequently identifiers are revealed when an attacker repeatedly queries the model using a fixed decoding strategy.

Privacy risk depends strongly on the interface available to an auditor or attacker. In a *black-box* setting, one observes only prompts and generated text. In a *gray-box* setting, one additionally observes token-level log probabilities for a set of alternative text completions of the same prompt, enabling likelihood-based scoring without access to model weights. In a *white-box* setting, one has access to model parameters. Our methods target gray-box auditing, which is a realistic setting for LLMs available via provider APIs.

We summarize attacker capability by a *query budget* N. The attacker (or auditor) issues N queries using a fixed prompt template and fixed decoding strategy (e.g., repeatedly sampling completions of a prompt such as `Name:`), yielding N generations. Under repeated sampling, even identifiers with a low probability of being generated in the LLM outputs may eventually appear; consequently, leakage is determined by (i) the residual direct identifier rate in the fine-tuning data and (ii) the attacker's query budget; increasing either can increase the expected number of extracted direct identifiers.

The overview of our pipeline is presented in Figure 1.

3.2. Dataset

We use the MIMIC-IV-Note dataset (Johnson et al., 2023), made up of 331,794 de-identified discharge summaries from 145,915 patients. We split the notes into training, validation, and test sets, ensuring no patient overlap between the sets, as shown in Table 1.

3.3. Controlled Direct Identifier Injection

To evaluate the risk of direct identifier leakage in language models trained on clinical notes, we developed a controlled direct identifier injection pipeline that systematically replaces masked direct identifiers in MIMIC-IV discharge summaries with synthetically generated direct identifiers. For each patient in the dataset, we generate a corresponding persona containing synthetic direct identifier attributes that match the patient's demographic characteristics. Our persona generation system uses the Faker library (Faraglia, 2026) to sample locale-specific names from the census data of multiple countries.

A fundamental challenge in automated direct identifier injection is to determine which type of direct identifier should be inserted at each masked location. Clinical notes contain

Table 1: Dataset splits for MIMIC-IV-Note.

Split	# Notes	# Patients
Train	268,303	118,190
Val	29,537	13,133
Test	33,954	14,592

Table 2: Distinct names per direct identifier (DI) rate and split.

DI rate	Split	# Distinct names
1%	Train	9,710
5%	Train	40,467
10%	Train	68,730
100%	Train	234,904
All	Val	25,990

diverse direct identifier types (i.e., patient names, physician names, dates, age, etc), each requiring different synthetic values from distinct persona attributes. We use Gemini 2.5 Flash to classify each masked direct identifier slot into semantic categories. This produces tag files that map each direct identifier slot identifier to its semantic category.

After classification, we perform controlled direct identifier injection using probabilistic sampling. For each note, we iterate through all classified identifier slots and independently sample each slot with probability p_{DI}. This stochastic sampling ensures that only a subset of direct identifier slots are replaced in each note, creating realistic scenarios where some direct identifiers are present while other slots remain masked. We created four datasets with $p_{DI} \in \{1\%, 5\%, 10\%, 100\%\}$, as shown in Table 2.

To assess whether the injected name strategy mimics realistic de-identification systems, we re-ran two de-identification tools on the notes after synthetic name insertion: Presidio (Mendels et al., 2018), a general-purpose de-identification system, and pyDeid (Sundrelingam et al., 2024), a clinical-domain de-identification system. Presidio left approximately 4% residual direct identifiers, while pyDeid left approximately 8%. Both tools also exhibited partial per-instance de-identification, where the same name was redacted in some locations but missed in others. This suggests that the injected names preserve enough contextual and grammatical realism to induce imperfect, non-uniform redaction patterns consistent with the threat model we study.

3.4. Fine-tuning details

We fine-tuned the Meta Llama 3.2 1B and Meta Llama 3.1 8B general-purpose models (Llama Team, AI Meta, 2024) on the MIMIC-IV clinical notes with injected synthetic direct identifiers using supervised fine-tuning (SFT). Our implementation follows the Stanford Alpaca framework (Taori et al., 2023). Each training example consists of an instruction field ("Generate a clinical note") and an output field containing the complete clinical note text. The data is formatted using Alpaca-style prompt templates that structure the input as "### Instruction:", followed by the instruction text, and "### Response:", followed by the model's expected output. While the explicit task is to generate clinical notes from instructions, this fine-tuning process also serves as domain adaptation, enabling the model to learn clinical terminology, note structure, and domain-specific patterns from the MIMIC-IV corpus. To prevent overfitting, we monitor model perplexity on held-out validation data throughout training. Models are fine-tuned using full parameter fine-tuning.

3.5. Privacy risk quantification

3.5.1. Audit setting and threat model

Our goal is to design a realistic *audit* that approximates private information leakage under a plausible attack scenario: rather than asking whether any identifier can be elicited under ad hoc prompting, we quantify how direct identifier leakage scales with attacker's capability, defined as a query budget N. Given a fixed prompt, an LLM defines a distribution over candidate continuations. Decoding settings determine how this distribution is transformed into a realized completion, thereby affecting which continuations are observable and their associated probabilities.

We assume *gray-box* access to the fine-tuned LLM at inference time: the auditor can issue prompts and obtain both generated text and token-level log probabilities for candidate continuations under fixed decoding settings. These token-level log probabilities for candidate continuations are available with some LLM providers like OpenAI. The auditor does not access model weights.

To control false positives, *i.e.,* the direct identifiers that are in the outputs but not from the fine-tuning data, our audit uses a classifier (that we call *verifier*) predicting membership of the generated direct identifiers using MIA techniques. We assume the auditor has access to a labeled subset of identifiers from the fine-tuning split (*train*) and from a disjoint held-out split (*val*) to train this classifier. The labels indicate whether the identifiers are present or not in the fine-tuning dataset. The auditor can build such a labeled set by manually checking some notes and annotating the missed identifiers. These labels are used only for verification; extraction itself is performed by querying the model. We evaluate the verifier via cross-fitting (Chernozhukov et al., 2024) ensuring that each candidate direct identifier is scored by a verifier that did not train on that direct identifier.

The auditor issues N independent queries drawn from a fixed prompt distribution (e.g., repeated completions of a templated `Name:` prompt), yielding an *extracted stream* of candidate identifiers whose composition evolves with N. An identifier is considered leaked only if it is surfaced by extraction and accepted by the verifier.

Note that our internal audit framework also corresponds to an external attacker threat model with insider knowledge, who has access to a small labeled dataset. This is a strong but common threat model in MIA literature (Shokri et al., 2017).

We next formalize how extraction and verification combine to yield *budget-conditioned* leakage curves that can be predicted from model scores, avoiding prohibitively large numbers of sampled generations.

3.5.2. Analytical characterization of budget-conditioned leakage

Setup and notation. Let $\mathcal{U}$ denote a universe of candidate identifiers (patient names) considered by the audit. Each candidate $i \in \mathcal{U}$ has a membership label $y_i \in \{0, 1\}$, with $\mathcal{T} = \{i \in \mathcal{U} : y_i = 1\}$ (train members) and $\mathcal{V} = \{i \in \mathcal{U} : y_i = 0\}$ (held-out non-members).

We consider an extraction procedure that issues N *independent* queries drawn from a fixed prompt distribution and decoding settings, producing a *stream* of candidates. We assume independence because each LLM query does not reuse the context from previous queries, as highlighted in (Hayes et al., 2025).

Per-candidate extraction probabilities from model scores. For each candidate identifier i, let $p_i \in (0, 1]$ denote the per-query probability that the model emits i under the fixed extraction protocol. Each p_i is non-zero because of the absence of filtering. In our gray-box setting, p_i can be estimated from model log probabilities for candidate continuations (e.g., by scoring the candidate string under teacher forcing as described in the next section) (Yi and Li, 2026; Janicki et al., 2026). These p_i values serve as deterministic inputs to the leakage curves below, enabling prediction without requiring prohibitively many sampled generations.

Budgeted unique-hit probability. Let $E_i(N)$ be the event that identifier i appears at least once in N independent generations. "The candidate is emitted at least once" can be modeled as evaluating the Cumulative Distribution Function (CDF) at N of a geometric distribution because each query is independent: $\Pr(E_i(N)) = 1 - (1 - p_i)^N \triangleq \pi_i(N)$. Intuitively, $\pi_i(N)$ increases with N and transitions from $\pi_i(N) \approx Np_i$ when $Np_i \ll 1$ to $\pi_i(N) \approx 1$ when $Np_i \gg 1$.

Verification model. Each extracted candidate is assigned a verifier score $s_i \in \mathbb{R}$ (e.g., from a membership inference classifier). For a threshold τ, the acceptance rule is $q_i(\tau) = \mathbf{1}\{s_i \geq \tau\}$. A candidate is counted as a *verified leak* if it is extracted at least once (event $E_i(N)$) and accepted by the verifier ($q_i(\tau) = 1$).

Expected verified leakage counts. The expected number of verified true positives (members) and false positives (non-members) are

$$\mathrm{TP}(N, \tau) = \mathbb{E}\left[\sum_{i \in \mathcal{T}} \mathbf{1}\{E_i(N)\}\, q_i(\tau)\right] = \sum_{i \in \mathcal{T}} \pi_i(N)\, q_i(\tau) \tag{1}$$

$$\mathrm{FP}(N, \tau) = \mathbb{E}\left[\sum_{i \in \mathcal{V}} \mathbf{1}\{E_i(N)\}\, q_i(\tau)\right] = \sum_{i \in \mathcal{V}} \pi_i(N)\, q_i(\tau) \tag{2}$$

The expected number of accepted candidates is $\mathrm{A}(N, \tau) = \mathrm{TP}(N, \tau) + \mathrm{FP}(N, \tau)$.

Budget-conditioned rates on the extracted stream. To report verifier performance in the attacker-relevant regime (conditioned on candidates that are actually surfaced), we use extracted-stream rates, with $E(N) \triangleq \{E_i(N)\}_{i \in \mathcal{U}}$ the vector of extraction indicators across all candidates:

$$\mathrm{TPR}_{\mathrm{ext}}(N, \tau) = \Pr(\hat{y} = 1 \mid y = 1,\, E(N) = 1) \tag{3}$$

$$= \frac{\mathrm{TP}(N, \tau)}{\sum_{i \in \mathcal{T}} \pi_i(N)}, \tag{4}$$

$$\mathrm{FPR}_{\mathrm{ext}}(N, \tau) = \Pr(\hat{y} = 1 \mid y = 0,\, E(N) = 1) \tag{5}$$

$$= \frac{\mathrm{FP}(N, \tau)}{\sum_{i \in \mathcal{V}} \pi_i(N)}, \tag{6}$$

with the convention that the ratio is undefined if the denominator is zero. Unlike standard candidate-uniform MIA reporting, these rates are *stream-weighted* by $\pi_i(N)$ and therefore vary with the attacker budget N.

Critical budget. The scale Np_i governs the budget. A candidate becomes likely to appear once N exceeds its *critical budget* $N_i^\star \approx 1/p_i$ (up to constant factors). As N increases, candidates enter the extracted stream in decreasing order of p_i, so leakage can exhibit a sharp rise when N crosses the range of $N_i^\star$ for a large mass of identifiers.

3.5.3. Extraction

Per-name emission probabilities from teacher forcing. To instantiate the analytical framework, we estimate the per-query emission probability of each candidate name $i \in \mathcal{U}$ under a fixed extraction prompt. Concretely, we score each candidate continuation under `Name:` using teacher forcing, yielding a model-implied probability p_i^{ft} for the fine-tuned model. We compute the analogous probability p_i^{base} for the corresponding base (pre-fine-tuning) model under the *same* prompt and tokenization.

Relative amplification through fine-tuning. To summarize how fine-tuning changes the emission of identifiers, we compute a per-name emission multiplier $r_i \triangleq \frac{p_i^{\text{ft}}}{p_i^{\text{base}}}$. We report robust summaries (e.g., tail percentiles) of $\{r_i\}$ over train and held-out name sets.

Sampling-based extraction experiments. To validate our analytical characterization with direct sampling, we run a generation-based attack using the same prompt template. For each query, we sample a completion of length 20 tokens with temperature 1.0 and without top-k or top-p filtering. We repeat this procedure N times (query budget), and parse each completion by taking the first two words as a candidate name. This heuristic for parsing may miss some real-world failure modes, but it still allows us to reliably estimate a meaningful lower bound on the potential leakage.

3.5.4. Verification - MIAs

Verifier model and features. To verify whether an extracted candidate identifier is likely to originate from the fine-tuning cohort, we train an MIA verifier that maps each candidate name i to a score s_i. Our verifier is a logistic regression trained on four gray-box features derived from model probabilities under teacher forcing: (i) p_i^{ft} under the `Name:` prompt, (ii) p_i^{base} under the `Name:` prompt, (iii) p_i^{ft} under the `Patient:` prompt, and (iv) p_i^{base} under the `Patient:` prompt. We include the additional `Patient:` prompt because it empirically slightly improves verifier separability, likely by providing a second, clinically natural context in which memorized identifiers exhibit membership-specific likelihood shifts. In any case, this simple four-feature verifier makes our extraction numbers a lower bound, since a motivated attacker could potentially develop stronger methods.

Cross-fitting to avoid contamination. To prevent optimistic bias from scoring candidates that were seen during verifier training, we use K-fold cross-fitting with $K = 5$. We partition the labeled candidate set into folds; for each fold, we train the verifier on the remaining folds and compute out-of-fold scores for the held-out fold. This yields a score s_i for every candidate name that is produced by a verifier that did not train on that name.

Threshold selection and operating point. In each experiment, we choose a decision threshold τ to control false positives. Specifically, for a target budget N we select τ such that the extracted stream false positive rate $\text{FPR}_{\text{ext}}(N, \tau)$ (Equation 6) is below a chosen

level (5% in our experiments). This calibration is important: without an explicit operating point, verification can become overly permissive as the extracted stream shifts with budget.

4. Results

4.1. Fine-tuning increases the risk of name emission

In Figure 2, we quantify how fine-tuning changes the model's tendency to emit specific identifiers by computing, for each candidate name i, a *relative emission-probability change* under the fixed extraction prompt/decoding described in section 3. For each residual direct identifier rate we summarize the per-name ratios over the corresponding candidate set – names seen during fine-tuning (*train*) or a disjoint held-out set (*val*) – with two complementary statistics, shown as the two panels.

The median (left panel) tracks the typical per-name change. Train and val curves rise together with the residual direct identifier rate and have essentially the same shape, reflecting generalization of a "name prior" rather than name-specific memorization. At low residual direct identifier rates both curves sit well below one: when most names in the fine-tune corpus are replaced by the placeholder ___, the fine-tuned model learns that the high-probability continuation at name positions is the placeholder, so the per-name emission probability *decreases* on both members and non-members. As the residual direct identifier rate grows and real names become common during fine-tuning, the model re-learns a general name prior and the typical multiplier climbs for both splits.

The 99th percentile (right panel) isolates the extreme tail of the per-name changes, which is the operationally relevant quantity for budgeted extraction: an attacker only needs a small number of names to become orders of magnitude more likely in order to recover them. The same overall trend with the direct identifier rate is visible, but the whole tail is lifted by many orders of magnitude compared to the median.

The memorization signal appears mainly in the extreme tail: while train and val names show similar median trends as the residual direct identifier rate increases, the train–val gap is much larger at the 99th percentile. This indicates that extractable risk is concentrated in a small subset of training names whose emission probabilities are amplified far more than comparable held-out names.

4.2. LLMs fine-tuned on clinical notes are vulnerable to MIAs

Fine-tuning not only increases the probability of emitting name-like identifiers (Figure 2), but also induces a measurable membership signal that can be exploited to *verify* whether an extracted candidate is likely to originate from the fine-tuning set.

Figure 3 reports Receiver Operating Characteristic curves (TPR vs. FPR) for our MIA verifier at two query budgets, $N = 10^4$ and $N = 10^7$. Importantly, these curves are *budget-conditioned*: rather than evaluating the verifier on a uniformly sampled list of candidate names, we evaluate it on the *extracted stream* induced by running the extraction process for N queries, as defined in section 3. As N increases, the extracted stream includes progressively lower probability and more ambiguous candidates, which changes the operating characteristics of membership inference and typically reduces Area Under the Curve (AUC).

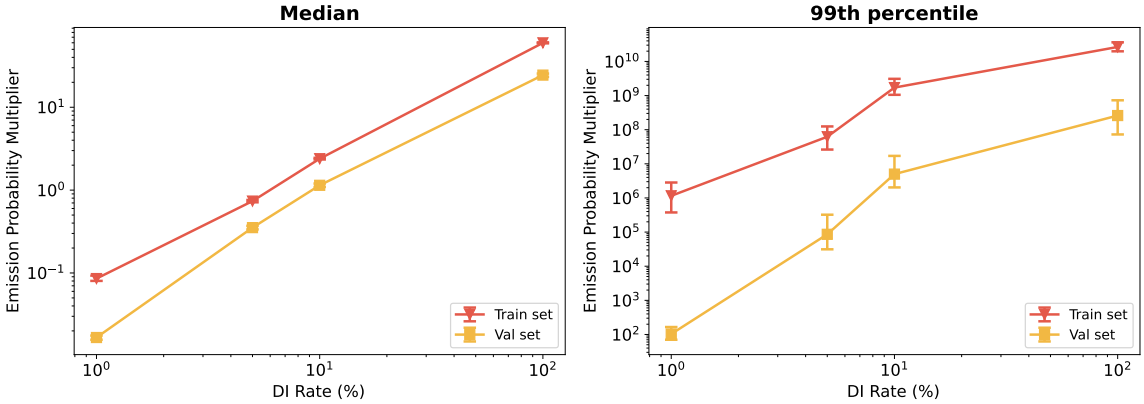


Figure 2: Per-name emission-probability multiplier $p^{\text{ft}}/p^{\text{base}}$ under fine-tuning, summarized by the median (left) and the 99th percentile (right) across candidate names, as a function of the residual direct identifier rate. Red: names seen during fine-tuning (train); yellow: disjoint held-out names (val). Error bars are 95% bootstrap CIs over names. Both curves share the same shape, reflecting generalization of a name prior (and the ___ placeholder effect at low rates); the memorization signal is the vertical train/val gap at each direct identifier rate, which is modest on the median but orders of magnitude wider on the 99th percentile, where the true leakage concentrates.

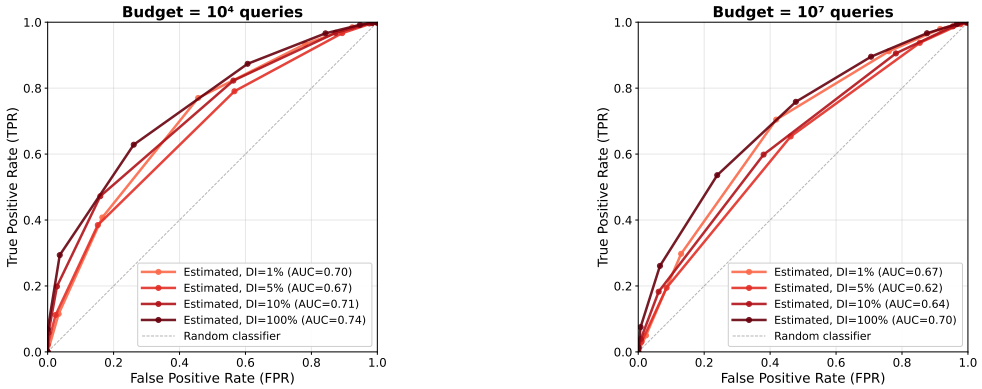


Figure 3: Budget-conditioned MIAs on the extracted names. ROC curves (TPR vs FPR) for the membership-inference verifier are evaluated on candidates surfaced by the extraction process at two query budgets ($N = 10^4$ and $N = 10^7$) and varying residual direct identifier rates, using the analytical framework ("Estimated").

Overall, the curves show that membership inference provides a substantial advantage over random guessing on the candidates that an attacker would realistically observe at each budget.

Moreover, verifier ablations showed that the main conclusions are robust to several implementation choices. Comparing logistic regression with simple thresholding and varying verifier training set size, we found that differences were small once the verifier was calibrated to the target operating point. The main challenge was not maximizing global discrimination

per se, but calibrating the discrimination signal so that false positives remained controlled on the extracted stream. In practice, all methods stabilized with roughly 1% of labeled candidates, whereas below this level false-positive-rate calibration degraded noticeably. Detailed ablation results are provided in Appendix C.1.

4.3. Direct identifier leakage becomes practically significant beyond a critical budget

Subfigure 4a reports budget-conditioned leakage under Verified Extraction. Across residual direct identifier rates in the fine-tuning data, verified leakage remains negligible at small budgets but increases sharply once the attacker budget crosses a regime where low probability identifiers become likely to appear at least once in the extracted stream.

Higher residual direct identifier rate shifts this onset to smaller budgets and increases the fraction of the direct identifiers in the fine-tuning set that can be recovered. Depending on the residual direct identifier rate, we can recover between 1 and 10% of the names contained in the fine-tuning set. Table 3 summarizes these approximate critical budgets for the 1B model.

The verifier turns raw extraction into an actionable *audit* signal aligned with attacker goals. An attacker (and an auditor) cares not just about seeing name-like strings, but about extracting identifiers that can be credibly linked to the fine-tuning cohort.

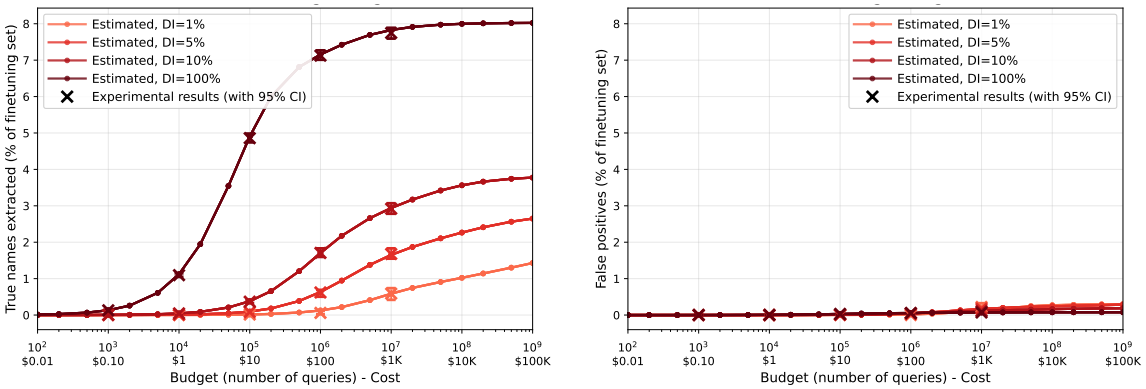
Without verification, increases in generated name-like strings could reflect generic name priors or formatting artifacts; with verification, recovered identifiers are accompanied by a membership signal, and subfigure 4b shows that this signal is not triggered at comparable rates on held-out names.

Therefore, Figure 4 shows why privacy risk is *budget-driven*: at small budgets the attack surfaces almost no verified training identifiers, but beyond a critical budget the number of verified leaks rises rapidly (subfigure 4a), while false acceptances remain low (subfigure 4b).

Prompt robustness. Our primary extraction protocol uses direct prompt "`Name:`" as a stress test. Failure to recover identifiers under such an explicit identifier-elicitation prompt would provide stronger evidence of low risk than failure under a weaker or more natural prompt alone. Importantly, our conclusion is not specific to this prompt. At a large query budget, the softer prompt "`Patient:`" recovered nearly the same number of verified identifiers across all residual DI rates, and optimally splitting the budget between the two prompts provided only negligible additional gain (the two single prompt estimates differ by at most 4% at $N = 10^{10}$). Thus, within the simple prompt family studied here, the leakage reported for "`Name:`" is not driven by a single unusually effective prompt (Appendix C.2).

4.4. Scaling

Table 4 shows that the privacy risk identified by Verified Extraction increases substantially with model scale. Across all residual direct identifier rates and both attacker budgets, the 8B model recovers more verified training names than the 1B model. The relative increase is especially pronounced in the low-residual regime: at 1% direct identifier rate, the 8B model recovers 2.24% versus 0.01% at $N = 10^5$ and 6.83% versus 0.06% at $N = 10^6$, corresponding to 152.0× and 116.0× more verified leakage, respectively. Even at more moderate residual



(*a*) Expected fraction of fine-tuning-set names extracted (verified true positives / $|\mathcal{T}|$).

(*b*) Expected false positives as a fraction of the fine-tuning set (verified non-members / $|\mathcal{T}|$).

Figure 4: Verified Extraction results as a function of attacker query budget N for different residual direct identifier rates with the 1B model. Curves are analytical predictions computed from the model-implied per-candidate extraction probabilities and verifier scores (section 3); markers show empirical extraction with 95% confidence intervals. A secondary x-axis reports approximate inference cost under our pricing model of $5 per million tokens (we consider 20 tokens/query; therefore, 10^7 queries $= 2 \times 10^8$ tokens = $1,000 at $5/1M tokens).

rates, the scaling effect remains large; for example, at 10% direct identifier rate and $N = 10^6$, the 8B model recovers 9.80% of training names compared with 1.71% for the 1B model.

The multiplicative gap narrows as the residual direct identifier rate increases, reaching roughly 3–4× at 100% direct identifier rate. This suggests that increasing model scale most strongly amplifies leakage in the sparse-identifier regime, where many names remain individually low probability but become substantially more recoverable under a larger model. Operationally, these results imply that privacy evaluations based on smaller models may materially understate risk for larger deployment-scale models, even when the training data and extraction protocol are otherwise unchanged.

5. Discussion

Our methods and results support two key findings. First, we introduce *Verified Extraction*, which extracts direct identifiers from a fine-tuned model and verifies that they match identifiers present in the fine-tuning data. Second, we demonstrate that this approach detects leakage under realistic imperfect de-identification, recovering multiple verified names across a range of residual direct identifier rates and two model sizes.

Our study focuses on the MIMIC-IV-Note dataset (Johnson et al., 2023), but the mechanism of residual direct identifier memorization that we point out is not specific to this dataset. It is a consequence of the next-token prediction loss used in a supervised fine-tuning setting, which increases the probability of emitting these identifiers (Furukawa and Oprea, 2026). Moreover, for our attack queries, we use generic prompts (`Name:`, `Patient:`)

Table 3: Approximate critical query budget for the 1B model. Q_{crit} denotes the approximate query budget at which 1% of fine-tuning-set names are verifiably extractable. Values are rounded to the nearest order of magnitude. DI = direct identifier.

DI rate	Q_{crit}
100%	10^4
10%	10^6
5%	10^6
1%	10^8

Table 4: Verified true names extracted as a percentage of the fine-tuning set at two attacker query budgets Q. Ratio = 8B / 1B, computed from unrounded values.

DI rate	Q	**1B (%)**	**8B (%)**	**Ratio**
1%	10^5	0.01	2.24	152.0×
1%	10^6	0.06	6.83	116.0×
5%	10^5	0.08	1.30	16.0×
5%	10^6	0.62	5.51	8.9×
10%	10^5	0.37	2.85	7.8×
10%	10^6	1.71	9.80	5.7×
100%	10^5	4.86	15.40	3.2×
100%	10^6	7.13	24.26	3.4×

that are common prefixes found in many clinical notes. For example, in a corpus containing repeated templates such as "Patient name: Alice Smith. Diagnosis: ...", fine-tuning can increase the likelihood of "Alice Smith" after the prefix "Patient name:" relative to a held-out name; if the same template instead contains "Bob Jones," the amplified identifier changes accordingly, but the mechanism is unchanged. Similarly, replacing most names with placeholders reduces the learned name prior, whereas leaving a small residual fraction creates a tail of identifiers whose probabilities may still be amplified. There may be some variation in leakage magnitude, due to differences in the total number of residual direct identifiers, their frequency, and the alignment between the clinical notes and attack prompts. Nevertheless, any clinical note containing residual direct identifiers when used in supervised fine-tuning presents a potential memorization risk that can be measured using our framework.

A central question for releasing API access to models fine-tuned on clinical notes is how to assess privacy risk when de-identification is known to be imperfect. In practice, institutions must decide whether a model can be safely queried without knowing which, if any, patient identifiers remain in the training data. Our framework speaks directly to this challenge by enabling an empirical assessment of whether a model exposes recoverable identifiers through interaction. Concretely, it provides a way to measure potential leakage and to use those signals to inform access decisions and mitigation efforts. It can also evaluate operational safeguards, such as rate limits, monitoring, and output filters, by comparing how they affect verified extraction success before and after implementing such safeguards. While this does not guarantee the absence of all residual identifiers, it offers a practical basis for evaluating and managing privacy risk when granting API access to clinical language models.

Privacy risk assessment before deployment is critical because undetected identifier leakage in real-world use may have regulatory repercussions. In the US, verified extraction of patient identifiers can trigger regulatory incident-response obligations; under the HIPAA Breach Notification Rule, covered entities must notify affected individuals without unreasonable delay and no later than 60 days after discovery of a breach, and breaches affecting 500+ individuals also require notification to HHS and prominent media outlets (U.S. Department of Health & Human Services, 2013; U.S. Government Publishing Office, a,b).

Beyond the technical evaluation of privacy risk, governance decisions depend on the applicable regulatory framework, which differs both in terminology and legal framing across regions: for example, HIPAA specifies methods for *de-identifying* protected health information in the USA, whereas the GDPR (European Parliament and Council of the European Union, 2016) frames *anonymization* through an identifiability standard in EU. Our framework is therefore intended to provide reproducible evidence that can be interpreted by institutional governance bodies according to the applicable jurisdiction, their own policies, and their risk tolerance.

5.1. Limitations and future work

Our work is a baseline method to quantify direct identifier leakage in a controlled setting. The results should be read as characterizing leakage under our specific threat model rather than as definitive privacy guarantees.

Identifier scope. We focus on direct identifiers in the form of patient names. The risk to patient privacy also involves leakage of other direct identifiers (e.g., phone numbers, addresses) and quasi-identifiers that enable linkage with auxiliary data.

Prompting strategy and extracted stream shift. Our evaluation uses a fixed prompt family and simple sampling, including an explicitly extractive prompt (e.g., `"Name:"`) intended to probe a worst-case querying scenario. In regimes with strong overfitting or very few direct identifiers, the model may produce non-canonical name-like outputs (e.g., variants, partial names, or formatting artifacts) that lie outside the candidate set, creating additional false-positive channels.

Model diversity and assumptions. We evaluate only two model sizes, 1 billion parameters and 8 billion parameters. The results may not extrapolate to larger models or other model families. Leakage patterns may vary under alternative adaptation strategies (e.g., parameter-efficient tuning methods such as LoRA (Hu et al., 2022)), different backbone models, stronger regularization, or privacy-preserving training methods. Besides, while our verifier relies on labeled member and non-member identifier subsets in our controlled experiments, an important direction for future work is to reduce labeling requirements and to study settings with more limited model access.

Adversarial fine-tuning against adaptive prompt attacks. A key next step is to explicitly model an adaptive adversary that optimizes prompts, decoding parameters, and multi-turn interaction policies to maximize identifier yield under a fixed query budget. Complementary to stronger attacks, an important direction is to fine-tune or train models to resist identifier extraction while preserving utility. One approach is adversarial training in which a prompt-optimizing attacker is trained (or approximated) alongside the model, and the model is optimized to reduce identifier emission under a broad distribution of queries.

Acknowledgments

This work was supported by the US National Library Medicine (grant number: R01LM014380) and the Columbia University Center for AI Technology-Amazon awards.

References

Griffin Adams, Emily Alsentzer, Mert Ketenci, Jason Zucker, and Noémie Elhadad. What's in a summary? laying the groundwork for advances in hospital-course summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4794–4811. Association for Computational Linguistics, 2021.

Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1998–2022. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.130.

Ofir Ben Shoham and Nadav Rappoport. CPLLM: Clinical prediction with large language models, 2023.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, 2019.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *USENIX Security Symposium*, 2021.

Fangyi Chen, Syed Mohtashim Abbas Bokhari, Kenrick Cato, Gamze Gürsoy, and Sarah Rossetti. Examining the generalizability of pretrained de-identification transformer models on narrative nursing notes. *Applied Clinical Informatics*, 15(02), 2024a.

Xiaoyi Chen, Siyuan Tang, Rui Zhu, Shijun Yan, Lei Jin, Zihao Wang, Liya Su, Zhikun Zhang, XiaoFeng Wang, and Haixu Tang. The Janus interface: How fine-tuning in large language models amplifies the privacy risks. In *ACM Conference on Computer and Communications Security*, 2024b.

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. MEDITRON-70B: Scaling medical pretraining for large language models. *arXiv*, abs/2311.16079, 2023.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and causal parameters. *arXiv*, abs/1608.00060, 2024.

Dina Demner-Fushman and Noémie Elhadad. Aspiring to unintended consequences of natural language processing: A review of recent developments in clinical and consumer-generated text processing. *Yearbook of Medical Informatics*, (1):224–233, 2016. doi: 10.15265/IY-2016-017.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models?, 2024.

European Parliament and Council of the European Union. Regulation (EU) 2016/679 of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation). Official Journal of the European Union, L 119, pp. 1–88, 2016. URL https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng. Recital 26.

Daniele Faraglia. Faker: A python package for generating fake data. https://github.com/joke2k/faker, 2026.

Sae Furukawa and Alina Oprea. Reconstruction of personally identifiable information from supervised finetuned models, 2026. URL https://arxiv.org/abs/2605.12264.

Akshay Goel, Almog Gueta, Omry Gilon, Chang Liu, Sofia Erell, Lan Huong Nguyen, Xiaohong Hao, Bolous Jaber, Shashir Reddy, Rupesh Kartha, Jean Steiner, Itay Laish, and Amir Feder. LLMs accelerate annotation for medical information extraction. In *Proceedings of the 3rd Machine Learning for Health Symposium*, volume 225 of *Proceedings of Machine Learning Research*, pages 82–100. PMLR, 2023.

Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000. doi: 10.1161/01.CIR.101.23.e215.

Jamie Hayes, Marika Swanberg, Harsh Chaudhari, Itay Yona, Ilia Shumailov, Milad Nasr, Christopher A Choquette-Choo, Katherine Lee, and A Feder Cooper. Measuring memorization in language models via probabilistic extraction. In *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2025.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. Are large pre-trained language models leaking your personal information? In *Findings of the Association for Computational Linguistics: EMNLP*, 2022.

Juliusz Janicki, Savvas Chamezopoulos, Evangelos Kanoulas, and Georgios Tsatsaronis. Detecting data contamination in large language models. April 2026. doi: 10.48550/arXiv.2604.19561. URL https://arxiv.org/abs/2604.19561.

Lavender Yao Jiang, Xujin Chris Liu, Nima Pour Nejatian, Mustafa Nasir-Moin, Duo Wang, Anas Abidin, Kevin Eaton, Howard Antony Riina, Ilya Laufer, Paawan Punjabi, et al.

Health system-scale language models are all-purpose prediction engines. *Nature*, 619 (7969), 2023.

Alistair Johnson, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV-Note: Deidentified free-text clinical notes. *PhysioNet*, January 2023. Version 2.2.

Stefan Larson, Nicole Cornehl Lima, Santiago Pedroza Diaz, Amogh Manoj Joshi, Siddharth Betala, Jamiu Tunde Suleiman, Yash Mathur, Kaushal Kumar Prajapati, Ramla Alakraa, Junjie Shen, et al. De-identification of sensitive personal data in datasets derived from IIT-CDIP. In *Conference on Empirical Methods in Natural Language Processing*, 2024.

Eric Lehman, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron C Wallace. Does BERT pretrained on clinical notes reveal sensitive data? In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

Llama Team, AI Meta. The Llama 3 Herd of Models. *arXiv*, abs/2407.21783, 2024.

Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. In *IEEE Symposium on Security and Privacy*, 2023.

Denis Jered McInerney, William Dickinson, Lucy C. Flynn, Andrea C. Young, Geoffrey S. Young, Jan-Willem van de Meent, and Byron C. Wallace. Towards reducing diagnostic errors with interpretable risk prediction, 2024.

Omri Mendels, Coby Peled, Nava Vaisman Levy, Sharon Hart, Tomer Rosenthal, Limor Lahiani, et al. Microsoft Presidio: Context aware, pluggable and customizable pii anonymization service for text and images, 2018. URL https://microsoft.github.io/presidio/.

Ashwinee Panda, Xinyu Tang, Christopher A. Choquette-Choo, Milad Nasr, and Prateek Mittal. Privacy auditing of large language models. In *International Conference on Learning Representations*, 2025.

Cheng Peng, Xi Yang, Aokun Chen, Kaleb E Smith, Nima PourNejatian, Anthony B Costa, Cheryl Martin, Mona G Flores, Ying Zhang, Tanja Magoc, et al. A study of generative large language model for medical research and healthcare. *NPJ digital medicine*, 6(1), 2023.

Lorenzo Rossi, Bartłomiej Marek, Franziska Boenisch, and Adam Dziedzic. Privacy auditing for large language models with natural identifiers. In *ICLR Workshop on Navigating and Addressing Data Problems for Foundation Models*, 2025.

Khaled Saab, Tao Tu, Wei-Hung Weng, Ryutaro Tanno, David Stutz, Ellery Wulczyn, et al. Capabilities of gemini models in medicine. *arXiv*, abs/2404.18416, 2024.

Andrew Sellergren, Sahar Kazemzadeh, Tiam Jaroensri, Atilla Kiraly, Madeleine Traverse, Timo Kohlberger, et al. Medgemma technical report. *arXiv*, abs/2507.05201, 2025.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017.

Aman Sinha, Cristina Garcia Holgado, Marianne Clausel, Mathieu Constant, and Xavier Coubez. Can LLMs be used to understand clinical notes better? In *Actes des 5èmes journées du Groupement de Recherche CNRS "Linguistique Informatique, Formelle et de Terrain" (LIFT 2023)*, page 94, 2023.

Congzheng Song and Vitaly Shmatikov. Auditing data provenance in text-generation models. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

Vaakesan Sundrelingam, Shireen Parimoo, Frances Pogacar, Radha Koppula, Saeha Shin, Chloe Pou-Prom, Surain B Roberts, Amol A Verma, and Fahad Razak. pydeid: an improved, fast, flexible, and generalizable rule-based approach for deidentification of free-text medical records. *JAMIA Open*, 8(1), December 2024. ISSN 2574-2531. doi: 10.1093/jamiaopen/ooae152. URL http://dx.doi.org/10.1093/jamiaopen/ooae152.

Xavier Tannier, Perceval Wajsbürt, Alice Calliger, Basile Dura, Alexandre Mouchet, Martin Hilka, and Romain Bey. Development and validation of a natural language processing algorithm to pseudonymize documents in the context of a clinical data warehouse. *Methods of Information in Medicine*, 63(01/02), 2024.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

U.S. Department of Health & Human Services. Breach notification rule, July 2013. Last updated July 26, 2013. Accessed 2026-01-30.

U.S. Government Publishing Office. 45 cfr §164.404: Notification to individuals. Electronic Code of Federal Regulations (eCFR), a. Accessed 2026-01-30.

U.S. Government Publishing Office. 45 cfr §164.406: Notification to the media. Electronic Code of Federal Regulations (eCFR), b. Accessed 2026-01-30.

Rui Wen, Zheng Li, Michael Backes, and Yang Zhang. Membership inference attacks against in-context learning. In *ACM Conference on Computer and Communications Security*, 2024.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE Computer Security Foundations Symposium*, 2018.

Jiatong Yi and Yanyang Li. Membership inference on LLMs in the wild. January 2026. doi: 10.48550/arXiv.2601.11314. URL https://arxiv.org/abs/2601.11314.

Appendix A. Training details

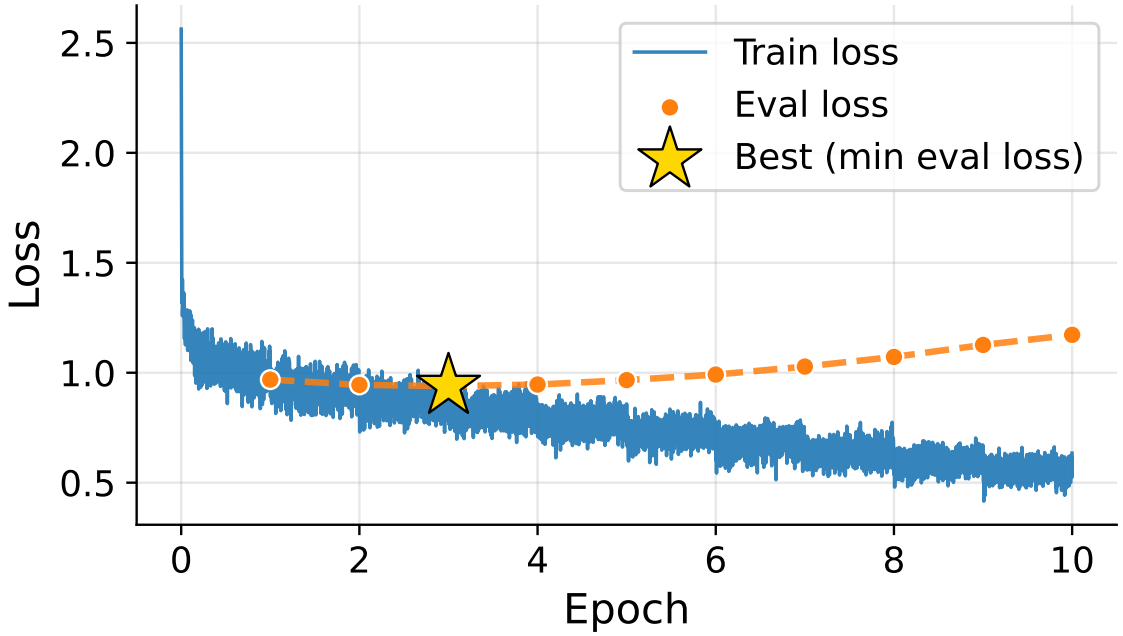


Figure A.1: Training and validation losses for the 1B model with a 1% of residual direct identifier rate. We notice that overfitting appears only after a few epochs.

Models and hyperparameters. Models and hyperparameters are summarized in Table A.1 and Table A.2.

Table A.1: Model and tokenizer configuration.

Item	Setting
Base model	Llama 3.2-1B or Llama 3.1-8B
Max context length	8192 tokens
Tokenizer	Default tokenizer; padding side right
Special tokens	`pad/eos/bos/unk` added and embeddings resized if missing
Fine-tuning method	Full fine-tuning (all parameters updated)

Training length, checkpoint selection, and input formatting. Each run trains for a fixed number of epochs (10; no early stopping). We enable per-epoch validation and select the checkpoint with the lowest validation perplexity on the held-out validation set. Figure A.1 shows representative training/validation loss curves (1% residual direct identifier rate), illustrating that overfitting emerges only after several epochs. We do not use sequence packing. Inputs longer than the 8192-token context window are truncated to the maximum length; otherwise no truncation is applied. Unless otherwise specified, all experiments use

Table A.2: Compute and training hyperparameters.

Item	Setting
Hardware	3× NVIDIA L40S GPUs
Distributed training	DeepSpeed ZeRO Stage 3 (offloading enabled)
Precision	bfloat16 mixed precision
Gradient checkpointing	Enabled
Per-device batch size	1 (train), 1 (eval)
Gradient accumulation	8 steps (effective batch size 24)
Optimizer	AdamW
Learning rate	2×10^{-5}
Warmup	3% of steps
LR schedule	Cosine
Weight decay	0

fixed random seeds for data generation and for verifier cross-validation splits. Training each 1B model took about a week, and a month for each 8B model.

Appendix B. Experiment details

Additional metrics for our pipeline. In addition to the primary leakage curves reported in Figure 4, Figure B.1 provides complementary operating-characteristic views of Verified Extraction by plotting the extracted-stream true positive rate (TPR), false positive rate (FPR), and precision as functions of query budget with the 1B model. These metrics are computed from the budget-conditioned extracted stream induced by our fixed prompting and decoding protocol (as defined in section 3).

Together, they help disentangle (i) how effectively the verifier confirms extracted training identifiers (TPR), (ii) how well false alarms are controlled among extracted non-members (FPR), and (iii) how the mix of extracted candidates impacts the fraction of verified leaks that are true members (precision), which can vary with both budget and residual direct identifier rate.

Extraction without verification. We compare Verified Extraction to a baseline that reports *unverified* extractions, i.e., counting any generated name-like string that matches a candidate identifier as a "leak" without applying membership-based verification. While unverified extraction can recover a larger number of distinct candidates, it conflates cohort-specific disclosures with plausible-but-non-member names arising from generic language priors or formatting artifacts. As a result, the baseline provides a weaker audit signal: increases in recovered names cannot be reliably attributed to the fine-tuning cohort, and apparent leakage may be dominated by false positives in settings with a low residual direct identifier rate. Figure B.2 reports this comparison and illustrates how verification is necessary to translate raw extraction into a high-confidence estimate of privacy risk.

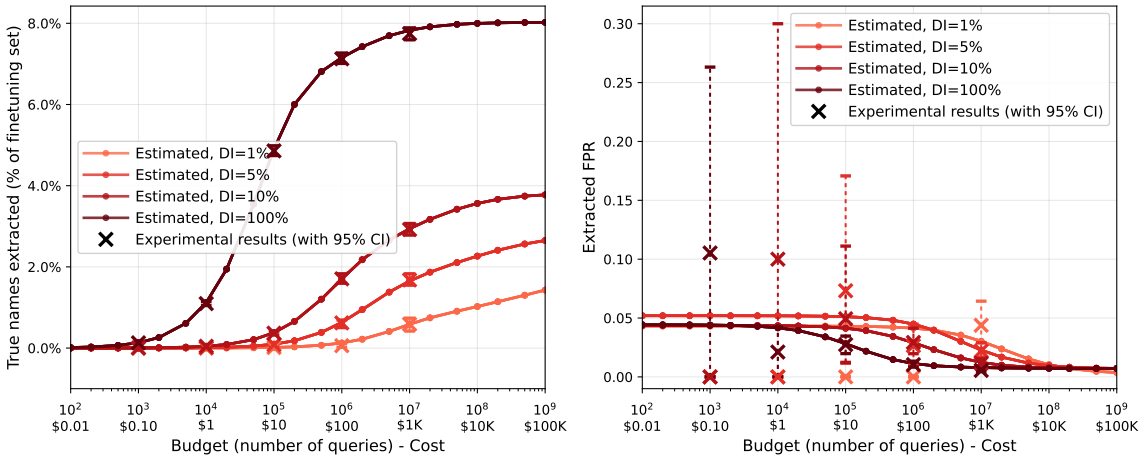
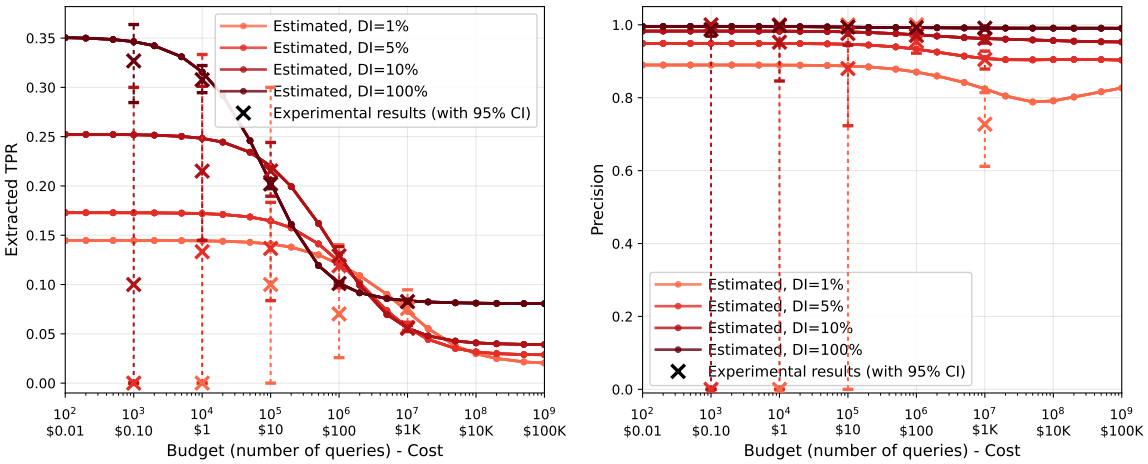


Figure B.1: Verified Extraction results as a function of attacker query budget N for different residual direct identifier rates with the 1B model.



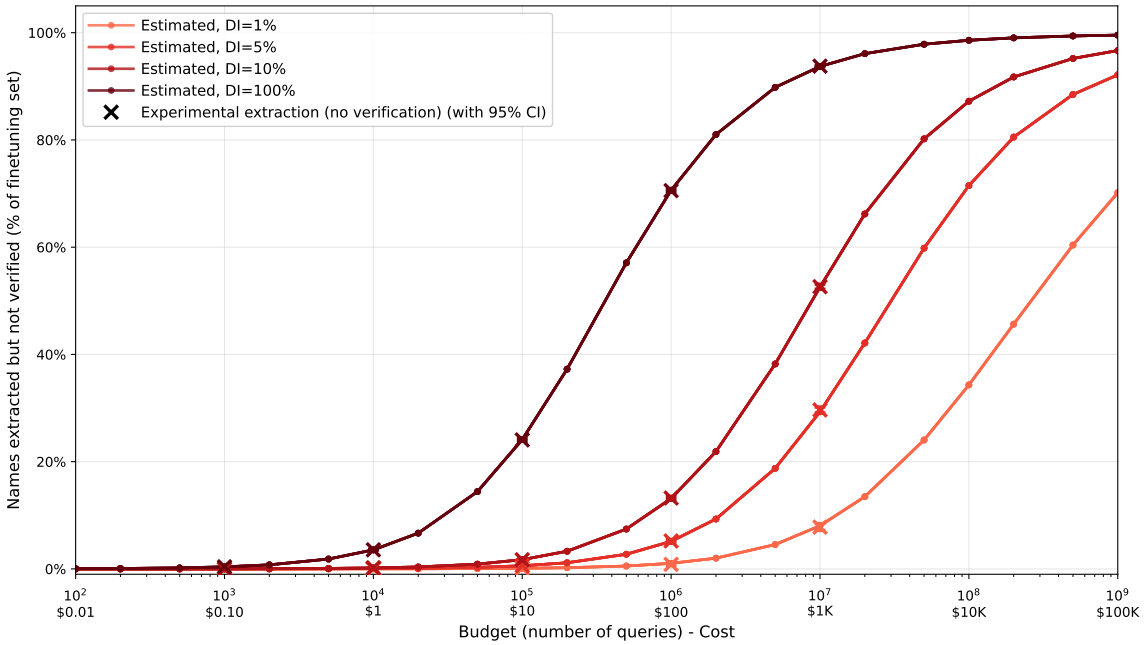


Figure B.2: Extraction results without verification on the 1B model. Without verification, and no filtering for decoding, it is possible to recover all names with a sufficient budget.

Appendix C. Additional experiments

C.1. Verifier ablation

This appendix supports the claim in the main text that our recall-based extraction results are robust to the choice of verifier. We ablate two dimensions: (i) the verifier family — simple thresholding on a single log-likelihood (LL) feature versus logistic regression on up to four LL features — and (ii) the amount of labeled data used to fit the verifier, swept from 0.1% up to 80% of the available pool.

Verifier variants. For a candidate identifier x we compute four per-prompt log-likelihoods from the target model: fine-tuned and base, each under the `Name:` and `Patient:` prompts. We compare six verifiers built from these features:

- **Threshold on fine-tuned name LL** – single-feature baseline.
- **Threshold on fine-tuned patient LL** – single-feature baseline.
- **LR on fine-tuned name+patient** – logistic regression on the two fine-tuned LLs, without access to the base model.
- **LR on fine-tuned+base (name only)** – adds the base-model `Name:` LL, a classical membership-inference feature.
- **LR on fine-tuned+base (patient only)** – same with the `Patient:` prompt.
- **LR on all four LLs** – the full model used in the main paper.

Protocol. For each training fraction $f \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 0.8\}$ we draw a fraction f of candidates from the pool, fit the verifier on that subset, and evaluate on the held-out portion. Threshold variants are calibrated on the training split so that the

training FPR equals 5%; that same threshold is then applied on test. Each configuration is bootstrapped; lines show means and shaded bands show bootstrap 95% CIs.

Metrics. We report three complementary metrics, shown as the three panels of each figure:

1. **Discrimination** – ROC AUC on the test split. Measures the global separability of member from non-member candidates, independent of any threshold.
2. **TPR on test at calibrated FPR = 5% on train** – the true positive rate on test after calibrating the threshold to a 5% FPR on the training split. This is the operationally relevant quantity: how many real identifiers the verifier lets through at a fixed, train-calibrated false-positive budget.
3. **FPR on test at calibrated FPR = 5% on train** – the test FPR actually realized by that same calibrated threshold. A verifier with strong discrimination but poor calibration will still leak false positives onto the extracted stream; this panel isolates the calibration-error contribution.

Observations. Across all configurations (Figures C.1 and C.2), differences in ROC AUC between threshold and logistic-regression variants are small once the training fraction exceeds roughly 1% of the pool, confirming that our main conclusions are not driven by a particular verifier family. The TPR panel (middle column) tells the same story: at the same calibrated operating point the variants cluster tightly. The dominant failure mode at very small training fractions is not weaker discrimination but miscalibration – the realized FPR on test drifts away from the 5% target, visible on the rightmost panel for $f \lesssim 1\%$. Above that threshold all variants stabilize near the target FPR and the choice of verifier becomes essentially immaterial for the downstream extraction counts. The pattern is also consistent across backbone sizes.

C.2. Prompt sensitivity within a simple prompt family

To assess the sensitivity of our results to the extraction prompt, we compared two simple prompt templates, `Name:` and `Patient:`, and also considered the best two-prompt mixture obtained by optimally splitting a fixed query budget across them.

C.2.1. Methods

Let $\mathcal{D}$ be the set of target identifiers (names), and for each $x \in \mathcal{D}$ let $p_{\text{name}}(x)$ and $p_{\text{pat}}(x)$ denote the per-query probabilities of emitting x under the `Name:` and `Patient:` prompts respectively.

Single-prompt hit probability. Under N independent queries issued with a single prompt of per-query probability $p(x)$, the probability that x is emitted at least once is

$$h(x; N, p) \;=\; 1 - \big(1 - p(x)\big)^N. \tag{7}$$

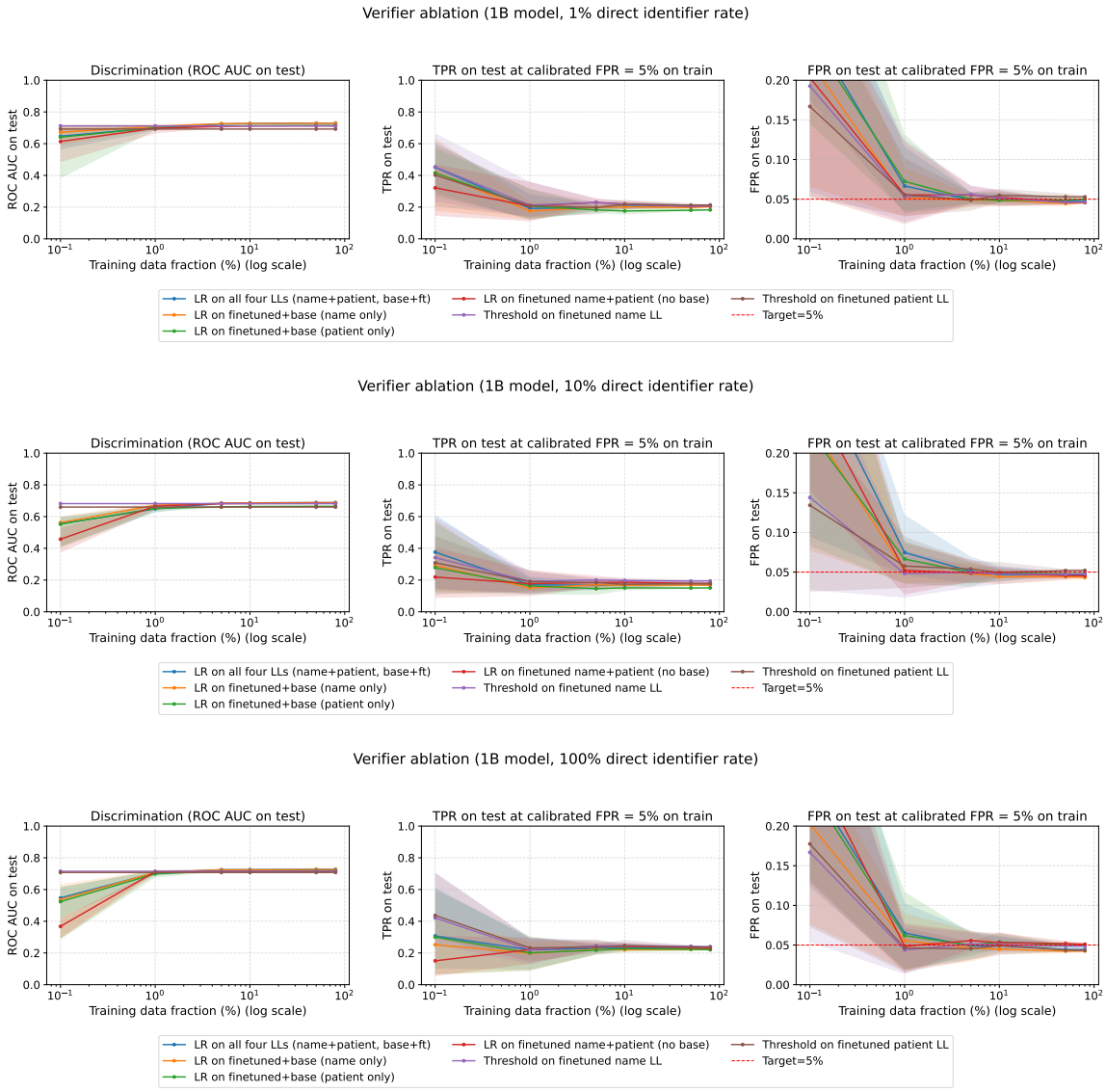


Figure C.1: Verifier ablation on the **1B model** at 1%, 10% and 100% direct-identifier rates (top to bottom). Each row shows, from left to right: ROC AUC on test, TPR on test at calibrated FPR = 5% on train, and realized FPR on test for that same calibrated threshold (red dashed line = 5% target). Shaded bands are bootstrap 95% CIs.

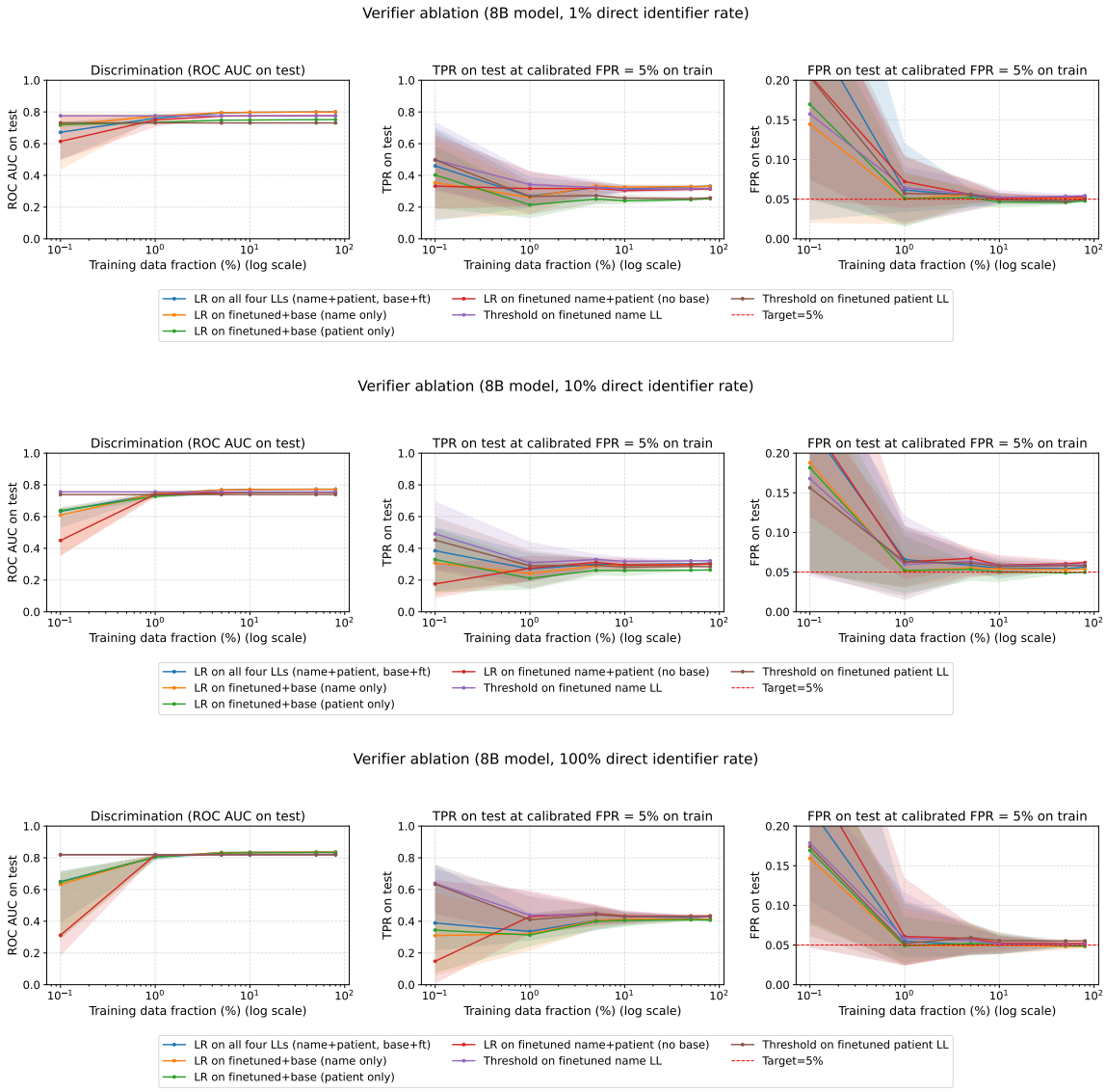


Figure C.2: Verifier ablation on the **8B model** at 1%, 10% and 100% direct-identifier rates (top to bottom). Panels and bands match Figure C.1. The 10% row is computed on the experiment with 10^6 queries, the largest available at this direct identifier rate for the 8B backbone.

Table C.1: Prompt sensitivity at large query budget ($N = 10^{10}$). Reported values are analytically estimated numbers of verified identifiers recovered using the `Name:` prompt, the `Patient:` prompt, and the best two-prompt mixture under optimal budget splitting. Differences are small, indicating that single-prompt extraction closely approximates the extractable set at high budgets. DI = direct identifier

DI Rate	`Name:`	`Patient:`	**Best Mix**
1%	6,107	6,319	6,366
5%	26,465	26,209	26,496
10%	44,178	43,726	44,189
100%	116,554	116,407	116,558

Two-prompt mixture. Given a total query budget N and a split parameter $\alpha \in [0, 1]$, we issue $\lfloor \alpha N \rfloor$ queries with `Name:` and $N - \lfloor \alpha N \rfloor$ queries with `Patient:`. Assuming independence across queries, the hit probability becomes

$$h_{\text{mix}}(x; N, \alpha) \;=\; 1 \;-\; \big(1 - p_{\text{name}}(x)\big)^{\alpha N} \big(1 - p_{\text{pat}}(x)\big)^{(1-\alpha)N}. \tag{8}$$

Expected verified extractions. Let $q(x) \in [0, 1]$ be the verifier probability (or a $\{0, 1\}$ verifier decision) that a candidate extraction of x is accepted. The expected number of verified identifiers recovered at budget N under split α is

$$K(N, \alpha) \;=\; \sum_{x \in \mathcal{D}} q(x)\, h_{\text{mix}}(x; N, \alpha). \tag{9}$$

The `Name:`-only and `Patient:`-only baselines correspond to $\alpha = 1$ and $\alpha = 0$ respectively.

Best mix. For each budget N we grid-search α over 21 equally spaced values in $[0, 1]$ and report

$$K^\star(N) \;=\; \max_{\alpha \in \mathcal{A}} K(N, \alpha), \qquad \alpha^\star(N) \;=\; \arg\max_{\alpha \in \mathcal{A}} K(N, \alpha), \tag{10}$$

with $\mathcal{A} = \{0, 0.05, 0.10, \ldots, 1\}$. The "Best Mix" column of Table C.1 reports $K^\star(N)$ at $N = 10^{10}$ for each direct identifier rate; the `Name:` and `Patient:` columns report $K(N, 1)$ and $K(N, 0)$. Because $K(N, \alpha)$ is evaluated analytically from the per-query probabilities, no additional sampling is required.

C.2.2. Results

Table C.1 reports analytically estimated verified recovery counts at a large budget of $N = 10^{10}$. At this scale, the two single-prompt strategies recover very similar numbers of verified identifiers, and the best mixture yields only negligible additional gain. This suggests that, within this simple prompt family, single-prompt extraction closely approximates the extractable set at high budgets. More adaptive prompt search or multi-turn attacks may still improve recovery, but were outside the scope of this study.