
An Investigation of Memorization Risk in Healthcare Foundation Models

Sana Tonekaboni*

MIT

Broad Institute of MIT and Harvard

Vector Institute

stonekab@mit.edu

Lena Stempfle*

MIT

Chalmers University of Technology

University of Gothenburg

stempfle@mit.edu

Adibvafa Fallahpour*

University of Toronto

Vector Institute

University Health Network (UHN)

adibvafa.fallahpour@mail.utoronto.ca

Walter Gerych

Worcester Polytechnic Institute

Computer Science Department

wgerych@wpi.edu

Marzyeh Ghassemi

MIT

mghassem@mit.edu

Abstract

Foundation models trained on large-scale de-identified electronic health records (EHRs) hold promise for clinical applications. However, their capacity to memorize patient information raises important privacy concerns. In this work, we introduce a suite of black-box evaluation tests to assess privacy-related memorization risks in foundation models trained on structured EHR data. Our framework includes methods for probing memorization at both the embedding and generative levels, and aims to distinguish between model generalization and harmful memorization in clinically relevant settings. We contextualize memorization in terms of its potential to compromise patient privacy, particularly for vulnerable subgroups. We validate our approach on a publicly available EHR foundation model and release an open-source toolkit to facilitate reproducible and collaborative privacy assessments in healthcare AI.²

1 Introduction

Foundation models trained on Electronic Health Records (EHR-FM) have been proposed as a promising direction for advancing healthcare [38]. However, there are concerns that large models trained on sensitive patient data may memorize and expose private health information [36], even under legal de-identification. Previous work on memorization in Large Language Models (LLMs) found that targeted prompts can reveal private data from training data sets [11, 9, 5, 50, 30, 26]. For EHR-FM, the risk is comparable, but the sensitivity of clinical data makes it even more critical to *proactively* assess and mitigate memorization threats.

Model memorization is often studied through the lens of adversarial privacy attacks, such as membership inference, attribute inference, and training data reconstruction, that aim to extract sensitive

*Equal contribution.

²Code available at https://github.com/sanatonek/EHR-FM_memorization

information from models [3, 40, 45, 19, 39]. While these approaches provide important lower bounds on leakage, they often ignore the deployment context of the model. Recent work has introduced the notion of contextual integrity [37], defining privacy breaches as violations of *context-specific norms* governing information flow. This distinction is particularly critical in clinical settings, where beneficial generalization of clinical knowledge may be similar to harmful memorization of patient-level information. We want a model to infer tuberculosis for a 36-year-old who presents with weight loss and cough. We do not want to infer tuberculosis for a 36-year-old who presents only with hypertension and knee pain, merely because that same patient had TB in the training data. In this work, we propose a contextual, risk-based evaluation framework for evaluating memorization in EHR-FM, with a focus on identifying practical privacy threats.

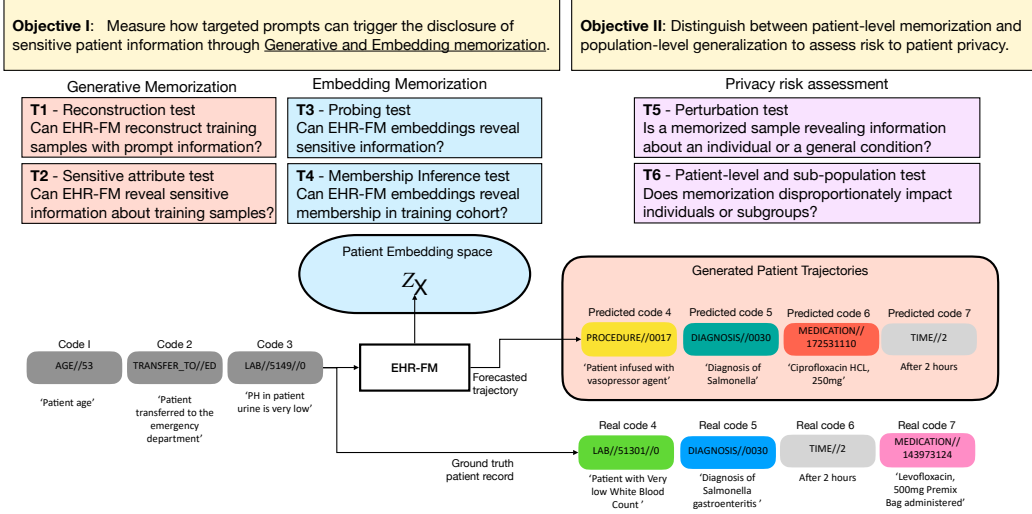


Figure 1: Our proposed tests (T1–T6) to evaluate memorization in EHR-FM, grouped by two core objectives. T1–T4 measure the extent to which models reveal training data, targeting both the embedding and the generated data domain. T5–T6 perform patient-level evaluations to quantify individual privacy risks from memorization or data leakage.

We explore memorization and its risk to patient privacy in EHR-FM trained on private data sequences of structured medical codes, and released as *black-boxes* with *prompt-only access*. EHR-FMs are distinct from clinical LLMs as they are not trained with free text such as clinical notes, where memorization risks have been explored [29]. We introduce a set of pragmatic EHR-FM tests with two main objectives (Figure 1): (1) to quantify different types of memorization that can be triggered via prompting, and (2) to assess the associated privacy risks by distinguishing harmful memorization at the patient level from benign population-level generalization. Overall, these tests evaluate the privacy risk of data leakage as a function of information an adversary can access. Our tests can be used to identify vulnerable samples and subcategories of individuals for further safeguarding prior to public release of a model using post-training safety layers [31], red-teaming [42] or retraining [28, 25].

We first introduce a practical formalization of memorization tailored to EHR-FMs. Next, we design and evaluate two **generative memorization tests** (T1–T2) on a publicly available EHR-FM [14] to assess its ability to reconstruct training data. Our results show that minimal individual-level knowledge (e.g., age) does not reveal sensitive attributes; however, we identify the additional contextual information that can trigger leakage. Our proposed **privacy risk assessment tests** (T5–T6) quantify patient-level memorization risk and assist in distinguishing harmful memorization at the patient level from benign population-level generalization. We also employ two **embedding memorization tests** (T3–T4), motivated by prior concerns that deep embeddings can leak private information [49]. These tests examine both membership inference and leakage of sensitive medical attributes from embeddings (overview in Figure 7 in the appendix). These tests address an important yet understudied aspect of EHR-FMs and offer a practical framework for systematically evaluating memorization and privacy risks as these models continue to scale and see broader clinical use.

2 Related work

EHR Foundation Models EHR-FMs are large models trained on structural EHR data using self-supervision at scale that can be adapted to a wide range of downstream tasks [6]. These models process sequences of standardized medical codes (diagnosis ICD, procedures, medications, lab tests) alongside temporal tokens indicating time intervals, creating patient trajectories (example shown in Figure 1). Two main category of EHR-FM includes Encoder models, like BEHRT [32], CLMBR-T [52], and CEHR-BERT [41] for representation learning and Decoder models like ESGPT [35], ETHOS [44], EHRMamba [14] that are trained to generate patient records. Initial work in EHR-FM has demonstrated significant improvements over task-specific architectures in limited data settings [52], domain adaptation, and robustness to distribution shifts [17]. However, study of memorization in these models, or even a definition of what memorization is in this setting is still an open problem.

Memorization in Privacy and Security Memorization has been extensively studied in privacy and security research [3, 40, 45, 19]. Categories of attacks include but are not limited to membership inference [48, 12, 20], property inference [49, 16], and model extraction [27], each targeting different aspects of models and assuming varying levels of transparency. Memorization in these settings is typically defined as being able to extract any training data from the model [11, 40]. However, in EHR settings, it is often more practical to assess the *sensitivity* level of the extracted data focusing on the deployment context [51].

Memorization in Foundation Models The study of memorization of foundation models mainly focuses on large language models (LLMs) [11, 9, 10, 40]. Memorization is described as a model’s capacity to store and reproduce specific training data patterns which can result in data leakage. While high-capacity models differ from standard ML models in their capacity to generalize without overfitting [11, 7], they are still vulnerable to data leakage [10, 54, 34, 9, 24] by membership inference [48, 53], property inference [16], and model inversion attacks [15].

Memorization in Healthcare Foundation Models Memorization in domain-specific foundation models, particularly those trained on medical data, lacks comprehensive study. In particular, their implications for healthcare domains and patient privacy remain unclear. The closest work to ours investigates extraction of personal health identifiers from an LLM trained on medical data [29], finding basic probes ineffective while noting advanced attacks’ potential. We focus on models trained on structural EHR data to investigate how such risks could manifest in rich temporal medical records.

3 Testing for Memorization

In EHR-FM systems, we expect models to learn from every patient they encounter and internalize that knowledge to make better predictions. Generalization is desirable, but model should not reveal memorized examples if such disclosures compromise patient privacy. To address this, we introduce a set of practical evaluation tests (summarized in Figure 1) designed to quantify memorization, and assess whether the memorized content poses privacy risks.

3.1 EHR-FM Testing Framework

As most EHR-FM give black-box and prompt-only access to users, we designed our tests to evaluate whether prompts alone can trigger the disclosure of sensitive patient information. We first create four tests (T1-T4) designed to measure memorization under two domains of information extraction: Generative memorization and Embedding memorization (Figure 1). Generative memorization tests assess a model’s ability to recreate data from the training cohort based on the following definition:

Definition 3.1 (Generative memorization) *Given a prompt p , consisting of a series of codes, a model f can generate a time series of EHR records s utilizing greedy decoding, where the concatenation of $[p||s]$ is part of a sample x .*

Embedding memorization tests focus on encoder-based EHR-FMs [32], and measure how much a model can memorize and reveal about a training sample through the embeddings as formalized below:

Definition 3.2 (Embedding memorization) *Given a prompt p , consisting of a series of codes, a model f can generate an embedding z that can be utilized to recover information associated with a sample x in the training dataset.*

Finally, we argue that the degree of memorization does not always align with the degree of risk to patient privacy. Therefore, 2 additional tests (T5-T6) are presented to assess individual-level risk and to distinguish between patient-level memorization and population-level generalization.

3.2 Testing Setup

All proposed tests measure memorization as a function of: (1) the amount of information an adversary uses to reveal information, and (2) the risk associated with the revealed information. Throughout the tests, we use the following setups to reflect the amount of information an attacker has access to:

- **Random:** No information is provided to the models and it generates embeddings or sequences only from its prior. This reflects the internalized knowledge of a model.
- **Static:** The model is prompted with any demographic attributes it has been trained on, such as age or biological sex. These represent easily accessible and public information about individuals.
- **N-codes:** The model is prompted with the first N codes of a patient record. Such prompts require more information about an individual patient.

We can also curate more sophisticated categories of prompts. For instance, prompting the model with information about an individual’s medication history. Different prompt setups will assess memorization based on what information can be accessed by an adversary and would give insight into what information has a higher likelihood of revealing private information.

To reflect the risk associated with revealed information, we select a number of codes, such as ICD-10 medical codes, related to diagnoses with social stigma as representatives of high-risk sensitive attributes. This information is classified under infectious diseases, substance abuse, and mental health conditions (see Table 4 in the appendix). We have selected these categories, as they are highly protected under regulations such as HIPAA and GDPR [1], often involving conditions that are poorly understood and associated with social taboos. The choice of sensitive attributes can be different depending on the setting or datasets, and we provide this categorization as an example.

3.3 Benchmark Implementation

We demonstrate our testing framework on EHRMamba2 [14]. We select this model as a representative benchmark because it is one of the few existing models designed for both patient embedding learning and EHR generation (forecasting). Additionally, its publicly available architecture details enable reproducible training. Also, EHRMamba2 is trained on the public MIMIC-IV dataset [23], which enables direct testing of memorization on known training samples, unlike other released models that are trained on private and inaccessible datasets.

4 Objective I: Measuring Memorization in EHR-FM

We evaluate EHR-FM *generative memorization* and *embedding memorization* with tests designed to measure the amount of training information extractable from the model and the risk associated with the information.

4.1 T1 - Trajectory memorization test

T1 examines how similar a sequence of generated codes is to a patient trajectory from the training data. Quantifying EHR sequence similarity is a nontrivial task, with two key nuances: (1) EHR code similarity may depend on their clinical context, for instance, two medications treating the same condition should have a higher similarity score compared to unrelated codes. (2) EHR data is inherently time-series-based; therefore, identical codes appearing at different time points should be penalized when measuring similarity between sequences.

We develop a similarity metric to quantify EHR sequence similarity. We first employ MedBERT [43], which adapts the BERT framework for EHR data, to map EHR codes to embeddings $h(\cdot)$ that capture clinical semantics (Figure 7 in the Appendix). This allows our distance measure to incorporate clinically meaningful semantics through the embeddings. Our distance score d_{EMD} (Equation 1) then uses time-weighted Wasserstein or Earth Mover’s Distance (EMD) [46] to measure the distance

between 2 sequences of embeddings, where $d(h(s1_i), h(s2_j))$ indicates the pairwise cosine distance between codes generated by MedBERT.

We incorporate the time difference between event tokens as a penalty for temporal alignment. The weight $T_{i,j}$ measures the visit time difference between token i in one sequence and token j in the other. EHR-FMs encode temporal information with time tokens, indicating the gap between each recorded code. The time penalty is determined based on how granular the EHR-FM record time is and how sensitive we want our score to be to time alignment. In our benchmark model, only time gaps larger than 1 hour are recorded, and we penalize every hour of misalignment with a weight of 1.

$$d_{EMD} = \inf_{T \in \mathcal{T}} \sum_{i=1}^{|s1|} \sum_{j=1}^{|s2|} T_{ij} d(h(s1_i), h(s2_j)), \quad (1)$$

We validate the functionality and sensitivity of our proposed metric in Appendix A using synthetic patient trajectories, analyzing how the measured distance changes across variations in patient trajectories, related and unrelated medications, and diagnoses.

Analysis: We compare the average distance (measured by d_{EMD}) of generated codes to the true trajectories for different prompt setups (Random, Static, 10, 20, and 50 codes), on our benchmark model. Figure 2a shows the distance over 100 prediction codes ($|s| = 100$) for 3K individuals in the pretraining cohort. Following the strategy of language models [9], for every prompt, hundreds of trajectories are sampled, and the distribution is used to quantify memorization.

We observe that **prompting the model with more information about an individual results in better sequence predictions** (lower d_{EMD}). This shows that the more information an attacker provides about an individual, the likelier it is for the model to recreate a patient EHR trajectory. However, distance measures are relative and may not easily translate to risk. Samples performing worse than random indicate that the prediction is no better than a random guess by the model. The high variance around the values indicates that forecast performance varies significantly between individuals, which highlights the importance of individual-level investigation on best-performing samples.

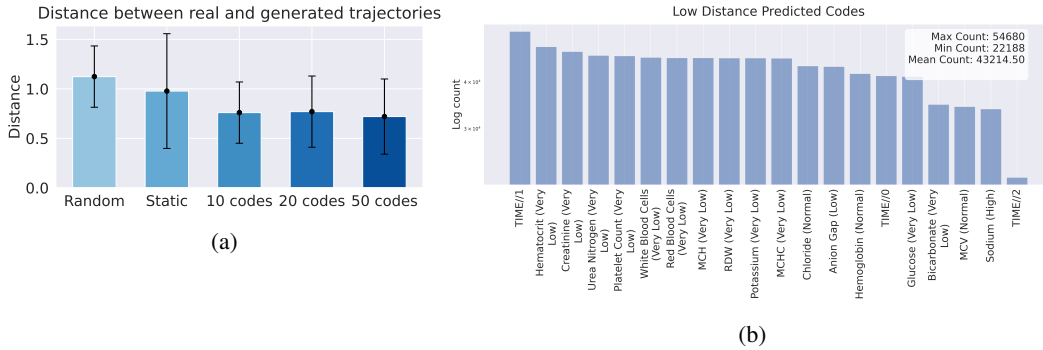


Figure 2: Model predictions in trajectory memorization. (a) Average distance between generated and ground truth EHR sequences of 100 codes for different prompting setups. (b) Predicted tokens and their frequencies in the best-performing trajectory prediction. All tokens are related to common lab measures, and most often normal values.

While revealing any information about an individual’s health record is undesirable, the risk associated with the memorized information can significantly vary. For example, noting a low white blood cell (WBC) count poses far less risk than revealing highly sensitive details, such as a patient’s HIV status. Our patient-level investigation of high-performing samples reveals that **trajectories with a smaller distance do not necessarily imply a higher risk**. In fact, these predictions correspond to short trajectories with common codes such as routine lab measures with normal values.

Figure 2b shows the top predicted codes for the best-performing prompts, prominently featuring common lab codes and time-related tokens. Our results further show that the model closely approximates the true token frequency distribution (Figure 3a). However, it under-generates less frequent codes, as

evidenced by the deviation from the diagonal in the low-frequency range. This shift indicates that the model favors more frequent codes, revealing a bias against generating rare codes.

Measuring similarity offers an intuitive and quantitative signal for memorization, but it does not directly capture risk. To address this, we follow up T1 with T2, which explicitly targets codes whose leakage could pose harm to patients—thus quantifying memorization risk more concretely.

4.2 T2 - Sensitivity test

Leaked information poses different risks, depending on the sensitivity of that information. The objective of T2 is to evaluate the likelihood of a model memorizing and revealing high-risk information about an individual. This information is represented as sensitive attributes defined in Section 3.2. We adopt a similar setup to T1 but explicitly remove any codes directly related to the sensitive condition from the prompt. T2 then measures the probability that the model reveals the sensitive attribute in its generated trajectory. The test passes if the model does not leak the sensitive attribute. Conversely, if the generated sequence contains the attribute, it indicates a memorization event that must be further validated to assess the associated privacy risk.

Table 1: Performance of the EHR-FM benchmark in predicting sensitive diagnoses. Results are reported for three categories of sensitive attributes (infectious disease, substance abuse, and mental health) under different prompt conditions. Higher prompt lengths increase the likelihood of revealing sensitive codes, highlighting privacy risks associated with memorization.

Sensitive attribute	Patient prevalence	Prompt	AUROC	AUPRC	Precision	Recall	Positive prediction count
Infectious disease	0.0476	Statics	0.548	0.053	0.000	0.000	0
	0.0476	10 codes	0.672	0.153	0.353	0.004	6
	0.0478	20 codes	0.697	0.177	0.444	0.006	8
	0.0473	50 codes	0.742	0.219	0.411	0.018	23
Substance abuse	0.0672	Statics	0.622	0.090	0.000	0.000	0
	0.0665	10 codes	0.706	0.171	0.467	0.023	49
	0.0661	20 codes	0.719	0.191	0.594	0.024	41
	0.0665	50 codes	0.751	0.267	0.577	0.084	162
Mental health	0.0611	Statics	0.604	0.081	0.000	0.000	0
	0.0605	10 codes	0.669	0.138	0.889	0.004	8
	0.0608	20 codes	0.681	0.170	0.733	0.007	11
	0.0604	50 codes	0.724	0.286	0.721	0.114	202

Analysis: We evaluate T2 on the benchmark model for different levels of prompting, and the findings are summarized in Table 1. With short prompts (Static baseline) the predictive performance of the model is near random. Showing that the benchmark doesn’t reveal sensitive information about a patient, only given the individual’s age or demographics. However, the model’s ability to reveal sensitive attributes increases with an increasing amount of prompt information. For an individual-level analysis, we select a conservative threshold of 30% and select all positive predictive prompts that generate the sensitive attribute in more than 30% of their trajectories (right column of Table 1). These identify potential problematic samples, but the risk of memorization depends on the context provided to the model. In fact, **not all positively predicted samples reflect memorization**. A well-performing model, like a skilled practitioner, should make informed predictions based on meaningful prompts, not mere memorization.

A sample fails the sensitivity test when the model generates a sensitive attribute even though it was excluded from the prompt. These are the most problematic cases where the model predicts a condition based on irrelevant input, indicating that it may have memorized spurious correlations or, more concerningly, details about a specific individual. For instance, one of our positive samples included a lab result showing a very low Absolute CD8 Count, the model predicted an HIV diagnosis in over 30% of sampled trajectories. While this does not necessarily indicate memorization, it reflects meaningful reasoning. Low CD8 counts are commonly observed in HIV patients and, although not sufficient for a full diagnosis, they increase the likelihood of HIV being suspected.

In contrast, we identify prompts that lack clear indicators of disease, yet the model still predicts a sensitive diagnosis. For example, one of our prompts was a 48-year-old individual transferred to the ED with a history of falls and discharged shortly after, representing minimal clinical detail. The model predicts alcohol abuse in 31% of the estimated trajectories. Despite the absence of relevant cues, the model reveals a sensitive diagnosis, causing this sample to fail the sensitivity test. We flag these prompts for further investigation in future test T5 to further assess their risk to patient privacy.

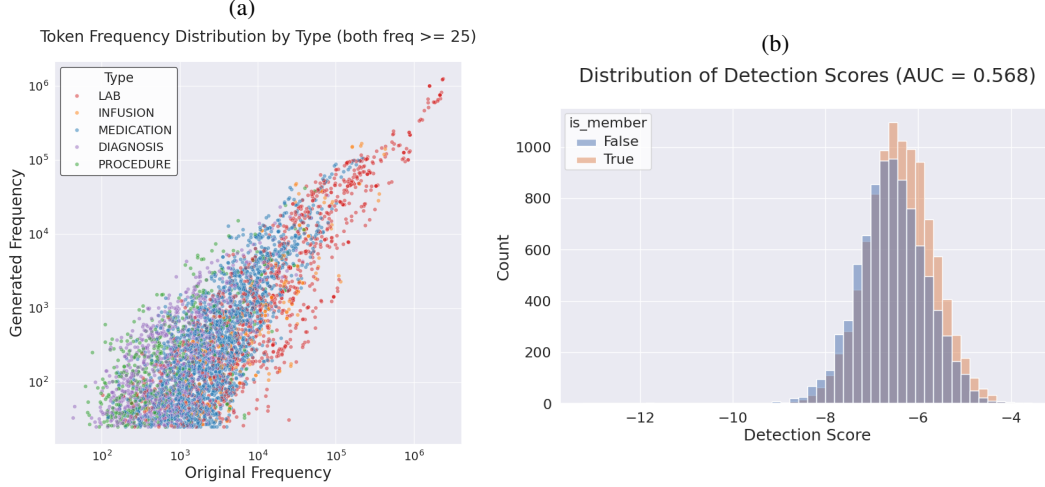


Figure 3: Code generation and membership inference. (a) Frequency of EHR code randomly generated by EHRMamba2 versus the original distribution of codes in the training dataset. (b) Distribution of detection scores from a membership inference attack based on model confidence. The x-axis shows scores, the y-axis shows counts, with colors indicating member (True) vs non-member (False) samples.

4.3 T3 - Probing test

T3 employs a probing test [2, 8] to evaluate memorization by examining information stored in embeddings. We train a probing model $g(\cdot)$ to predict sensitive information from embeddings, using either external data or a subset of the training dataset. This setup reflects the information available to an adversary, and T3 measures memorization risk as a function of how much knowledge the probe requires to recover the sensitive content. If probes successfully extract sensitive information from embeddings, developers should apply additional post-processing to remove such information before releasing the model. We validate our proposed T3 test on a synthetic temporal dataset (as shown in Appendix B) to demonstrate it successfully identifies memorized samples in a controlled setting with positive control cases.

Analysis: We apply the probing test to our benchmark model by prompting it with the first N codes from a patient record. Embeddings are extracted up to the first 10, 20, or 50 tokens and remain frozen during probing. We then train a classifier on the embeddings to predict sensitive diagnoses ($\hat{y} = g(f(s))$), as defined in Section 3.2, where $f(s)$ denotes the foundation model embeddings and $\hat{y}$ the predicted diagnosis. If the classifier successfully infers the label from the embedding, we consider the model to have memorized sensitive information. The highest-risk scenario arises when the probing model, trained on a separate test cohort, can still successfully extract sensitive codes. This is particularly concerning, as it suggests that an adversary equipped with an external dataset and the trained model could recover sensitive information about individuals in the training cohort.

We train the probing model with embeddings of a separate test cohort as well as varying fractions of training data. High accuracy with minimal data suggests strong memorization rather than generalization. Table 6 in the appendix reports probing attack performance on sensitive diagnoses, measured across different prompt lengths (10, 20, 50 tokens) and training. **Extracting sensitive information only from the embeddings is difficult**, even if an adversary can access a portion of the training data.

AUROC values across all sensitive attributes remain around 0.5, indicating no clear memorization signal and suggesting random performance. Notably, AUPRC and F1 generally decline with increased training data, hinting at inconsistent memorization patterns. These trends may be partially influenced by dataset size variations: 113,579 for the test set, 102,222 for 0.1%, and 90,864 for 20%. If probing performance does not improve even with access to training data, as we see with our benchmark, then the model passes T3. Although, the sensitive signal may be encoded only within a specific subgroup, prompting T6, which targets subgroup-level probing to uncover such hidden memorization.

4.4 T4 - Membership inference

T4 evaluates Membership Inference in EHR-FM; specifically, an adversarial attack that reveals whether a specific sample was included in a model’s training cohort [12]. Many MI attacks succeed using only model outputs such as logits [48, 13, 33], underscoring that risk extends even to EHR-FMs released as black-box models. This type of risk differs from direct patient data exposure as it requires an attacker to possess prior knowledge about an individual and then confirm their presence in the training data. When such inference succeeds, it can reveal sensitive details about the individual, including the institution or location of their clinical care or other training cohort-related information.

Analysis: We prompt the model with part of the patient EHR trajectory and extract the estimated logits for each token. Following the strategy introduced in [47], we estimate the membership score as the expected value of the log probabilities of the least probable tokens. The detection scores are shown in Figure 3b for 10K samples from the train and test cohort. The slight distribution separation suggests a marginally higher detection score for member samples, but the difference is not statistically significant, indicating weak discriminative power, suggesting the benchmark model does not substantially reveal membership status. These results are based on embeddings extracted from 1024 tokens of an individual trajectory, yet no clear separation emerges. Importantly, membership inference would become far more concerning if successful with substantially less information about the individual.

To meaningfully assess patient privacy risk, we should also evaluate MI at the individual level, focusing on which specific pieces of information trigger the model to reveal membership. Simply detecting membership does not alone indicate a privacy risk, especially in models trained on de-identified data. In order to pass T4, any prompt that results in a successful membership inference should not uniquely identify an individual. If it does, for example the prompt includes a unique diagnosis, this puts the individual in significant privacy risk.

5 Objective II: Evaluating Risk of Memorization

Most efforts to quantify the privacy implications of large foundation models focus on measuring information leakage [37]. While leakage is undesirable, our tests show that it doesn’t always translate directly into patient privacy risk in healthcare. In this setting, leaking a single sensitive code about a patient can be much more concerning than exposing an entire blood panel. Hence, in our evaluation of memorization in EHR-FMs after going through T1-T4 to quantify memorization and assess the information leakage, we introduce two tests to investigate memorization privacy risk through: (1) Understanding whether the information leakage was the result of the model memorizing patient-level information or the model learning generalizable knowledge. (2) Evaluating sensitivity of different subgroups to memorization where information leakage can pose significantly different privacy risk.

5.1 T5 - Perturbation test

T5 evaluates whether adversarial prompts (such as the ones that identified sensitive attributes in T2) trigger patient-specific memorization or broader statistical patterns. **A sample is memorized and not generalized if the model output is sensitive to personal identifiers.** We test for this by constructing a set of perturbed prompts, where we modify personal identifiers such as age, ethnicity, or rare diagnosis codes, while keeping the rest of the prompt unchanged. We deliberately select these identifiers to reflect attributes unique to the target individual. We then re-evaluate the model’s behavior on the perturbed prompts to determine whether the information leakage stems from memorization of a specific training individual or from generalizable patterns. If the model stops producing sensitive information after altering the identifiers, it signals that the original output resulted from memorizing

an individual instance. These cases indicate high-risk memorization and call for targeted safeguards to protect patient-level privacy.

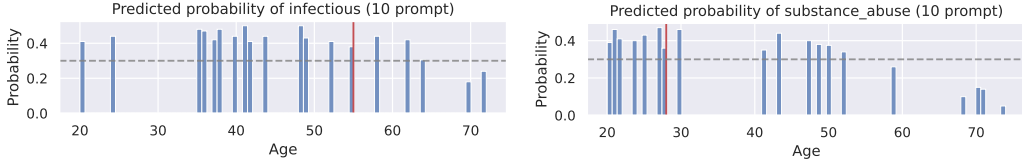


Figure 4: Predictive results for the perturbation test on 2 identified samples from infectious disease and substance abuse. Each bar represents the probability of generating the sensitive token, when the model is prompted with the 10 prompts and with different ages. The horizontal line is the threshold of 0.3 used in T2, and the red vertical line represents the original patient age.

Analysis: Our benchmark model uses only age as a demographic input. For prompts that revealed sensitive attributes in T2, we perturbed the age to test whether the model still predicts the sensitive diagnosis for people of different ages. Figure 4 shows two cases where the model correctly predicts Hepatitis C (left) and Alcohol abuse (right) for an individual whose age is marked by the vertical red line. Consistent predictions across individuals with similar ages suggest that the model relies on general trends for making the prediction rather than memorizing individual-specific patterns. However, we observe different outcomes when personal characteristics diverge. For example, in the case of substance abuse, predicted likelihoods drop for older individuals, revealing age-dependent patterns the model has learned. T5 is triggered when small changes to personal identifiers, such as age, cause significant shifts in prediction likelihoods. This sensitivity indicates that the model has memorized individual-specific information rather than generalizable patterns, posing a high risk to patient privacy. More examples of the perturbation test are shown in Figure 8 in the Appendix.

5.2 T6 - Sub-population test

In healthcare, we must assess memorization not only at the individual level but also across subgroups. Certain demographic or clinical populations face elevated privacy risks due to the sensitivity or rarity of their data. With our subgroup memorization test, we evaluate whether **belonging to a specific subpopulation increases the risk of memorization**. For example, if an attacker prompts the model with a rare condition code and the model leaks information about individuals with that condition, it exposes a serious privacy threat, one that targets small identifiable groups.

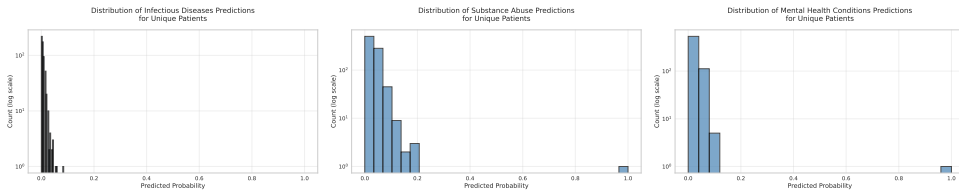


Figure 5: Likelihood of predicting the codes related to sensitive attributes (Infectious disease, mental health, and substance abuse) in EHR trajectories generated by prompting the benchmark model with rare diagnoses.

Analysis: We analyzed patients with rare diagnoses or procedures that appear only once in the training data. These individuals face a higher risk of re-identification due to the uniqueness of their records, as highlighted in genomic privacy research [18]. To test for memorization, we prompted the model with the patient’s age and a rare diagnosis or procedure code, then applied the sensitivity test from T2 to detect any disclosure of sensitive attributes. Figure 5 shows the model’s likelihood scores. Low scores suggest the model does not expose private information based on rarity alone, while high scores signal potential privacy concerns that require closer examination.

Our benchmark model revealed two concerning cases: (1) Sedative, hypnotic, or anxiolytic dependence, and (2) Mild manic episode without psychotic symptoms. These codes strongly indicate the sensitive attribute under test. Even if the model didn’t directly memorize them, their presence in

high-likelihood outputs raises serious privacy concerns. If statistical associations cannot explain these predictions, they reflect memorization failures.

We also examine the elderly subgroup—patients over 85—who comprise about 5% of the MIMIC dataset. As a long-tailed cohort, they are more prone to overfitting and require targeted evaluation. We apply tests T1–T5 to assess memorization relative to the general population (Table 5, Appendix). While the ideal outcome remains no memorization, this subgroup analysis is critical: if a memorized instance does occur, the privacy risk is amplified because these patients belong to a more easily identifiable cohort. Any flagged cases therefore warrant heightened caution, since individuals in long-tailed groups are often **more uniquely identifiable**, even in de-identified datasets, underscoring the **need for stronger privacy safeguards**.

6 Discussion

We propose a new perspective on evaluating EHR foundation model memorization, grounded in the practical realities of healthcare. As EHR-FMs integrate into clinical workflows for documentation, decision support, and analytics [22], they often operate on sensitive subsets of patient records that carry high clinical relevance. If these models leak memorized patient information, this compromises privacy, misleading clinical decisions [4], and enabling adversarial attacks, posing serious ethical and legal challenges.

High-profile breaches, such as Stanford Hospital’s 20M lawsuit over leaked records [21], underscore the urgency of systematically auditing AI models in healthcare. To address this, we introduce an evaluation framework for EHR-FM that measures memorization and its associated privacy risks. Our tests quantify different forms of memorization and assess their implications in clinical settings, distinguishing harmful leakage at the patient level from useful generalization at the population level.

EHR-FMs continue to evolve, and as their capabilities grow, prioritizing privacy becomes critical. Our work equips developers with tools to detect memorization and sets the foundations for future mitigation strategies. Developers can use these tests to identify and address flagged samples, reduce memorization during training, and better understand risks, thresholds, and attacker strategies. Although our tests are not exhaustive, they provide a practical starting point. The open-source code enables others to adapt, extend, and build toward more secure EHR foundation models. Follow up study will investigate such risks across broader model categories and stakeholder perspectives.

Acknowledgments and Disclosure of Funding

ST was supported by the Eric and Wendy Schmidt Center at the Broad Institute of MIT and Harvard. LS was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. MG was supported by the National Science Foundation (NSF) 22-586 Faculty Early Career Development Award (#2339381), a Gordon & Betty Moore Foundation award, a Google Research Scholar award and the AI2050 Program at Schmidt Sciences. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute.

References

- [1] A. Act. Health insurance portability and accountability act of 1996. *Public law*, 104:191, 1996.
- [2] G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes, 2018.
- [3] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017.
- [4] D. Barth-Jones. The ‘re-identification’ of governor william weld’s medical information: a critical re-examination of health data identification risks and privacy protections, then and now. *Then and Now (July 2012)*, 2012.
- [5] S. Biderman, U. Prashanth, L. Sutawika, H. Schoelkopf, Q. Anthony, S. Purohit, and E. Raff. Emergent and predictable memorization in large language models. *Advances in Neural Information Processing Systems*, 36, 2024.
- [6] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- [7] S. Bordt, H. Nori, V. Rodrigues, B. Nushi, and R. Caruana. Elephants never forget: Memorization and learning of tabular data in large language models. *arXiv preprint arXiv:2404.06209*, 2024.
- [8] Z. Bouraoui, J. Camacho-Collados, and S. Schockaert. Inducing relational knowledge from bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34/ 05, pages 7456–7463, 2020.
- [9] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramèr, and C. Zhang. Quantifying memorization across neural language models. In *ICLR*, 2023.
- [10] N. Carlini, C. Liu, U. Erlingsson, J. Kos, and D. Song. The secret sharer: evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC’19, page 267–284, USA, 2019. USENIX Association.
- [11] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.
- [12] D. Chen, N. Yu, Y. Zhang, and M. Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 343–362, 2020.
- [13] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974. PMLR, 2021.
- [14] A. Fallahpour, M. Alinoori, W. Ye, X. Cao, A. Afkanpour, and A. Krishnan. Ehrmamba: Towards generalizable and scalable foundation models for electronic health records. In *Machine Learning for Health (ML4H)*, pages 291–307. PMLR, 2025.
- [15] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart. Privacy in pharmacogenetics: An {End-to-End} case study of personalized warfarin dosing. In *23rd USENIX security symposium (USENIX Security 14)*, pages 17–32, 2014.
- [16] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 619–633, 2018.
- [17] L. L. Guo, J. Fries, E. Steinberg, S. L. Fleming, K. Morse, C. Aftandilian, J. Posada, N. Shah, and L. Sung. A multi-center study on the adaptability of a shared foundation model for electronic health records. *npj Digital Medicine*, 7(1):171, 2024.

- [18] M. Gymrek, A. L. McGuire, D. Golan, E. Halperin, and Y. Erlich. Identifying personal genomes by surname inference. *Science*, 339(6117):321–324, 2013.
- [19] V. Hartmann, A. Suri, V. Bindschaedler, D. Evans, S. Tople, and R. West. Sok: Memorization in general-purpose large language models. *arXiv preprint arXiv:2310.18362*, 2023.
- [20] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 1(1):133–152, 2019.
- [21] Infosecurity Magazine. Stanford hospital faces \$20 million lawsuit over patient data breach. *Infosecurity Magazine*, 2011.
- [22] L. Y. Jiang, X. C. Liu, N. P. Nejatian, M. Nasir-Moin, D. Wang, A. Abidin, K. Eaton, H. A. Riina, I. Laufer, P. Punjabi, et al. Health system-scale language models are all-purpose prediction engines. *Nature*, 619(7969):357–362, 2023.
- [23] A. E. Johnson, L. Bulgarelli, L. Shen, A. Gayles, A. Shammout, S. Horng, T. J. Pollard, S. Hao, B. Moody, B. Gow, et al. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1, 2023.
- [24] N. Kandpal, E. Wallace, and C. Raffel. Deduplicating training data mitigates privacy risks in language models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*. PMLR, 2022.
- [25] G. Kerrigan, D. Slack, and J. Tuyls. Differentially private language models benefit from public pre-training. In *Proceedings of the Second Workshop on Privacy in NLP*, pages 39–45, 2020.
- [26] U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer, and M. Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2019.
- [27] K. Krishna, G. S. Tomar, A. P. Parikh, N. Papernot, and M. Iyyer. Thieves on sesame street! model extraction of bert-based apis. *International Conference on Learning Representations*, 2019.
- [28] K. Lee, D. Ippolito, A. Nystrom, C. Zhang, D. Eck, C. Callison-Burch, and N. Carlini. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, 2022.
- [29] E. Lehman, S. Jain, K. Pichotta, Y. Goldberg, and B. Wallace. Does BERT pretrained on clinical notes reveal sensitive data? In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 946–959, Online, June 2021. Association for Computational Linguistics.
- [30] D. Leybzon and C. Kervadec. Learning, forgetting, remembering: Insights from tracking llm memorization during training. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 43–57, 2024.
- [31] S. Li, L. Yao, L. Zhang, and Y. Li. Safety layers in aligned large language models: The key to llm security. In *The Thirteenth International Conference on Learning Representations*.
- [32] Y. Li, S. Rao, J. R. A. Solares, A. Hassaïne, D. Canoy, Y. Zhu, K. Rahimi, and G. S. Khorshidi. BEHRT: transformer for electronic health records. *CoRR*, abs/1907.09538, 2019.
- [33] L. Liu, Y. Wang, G. Liu, K. Peng, and C. Wang. Membership inference attacks against machine learning models via prediction sensitivity. *IEEE Transactions on Dependable and Secure Computing*, 20(3):2341–2347, 2022.
- [34] R. T. McCoy, P. Smolensky, T. Linzen, J. Gao, and A. Celikyilmaz. How much do language models copy from their training data? evaluating linguistic novelty in text generation using RAVEN. *Transactions of the Association for Computational Linguistics*, 11:652–670, 2023.

- [35] M. McDermott, B. Nestor, P. Argaw, and I. S. Kohane. Event stream gpt: a data pre-processing and modeling library for generative, pre-trained transformers over continuous-time sequences of complex events. *Advances in Neural Information Processing Systems*, 36:24322–24334, 2023.
- [36] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 691–706. IEEE, 2019.
- [37] N. Mireshghallah, H. Kim, X. Zhou, Y. Tsvetkov, M. Sap, R. Shokri, and Y. Choi. Can llms keep a secret? testing privacy implications of language models via contextual integrity theory. *The Twelfth International Conference on Learning Representations*, 2023.
- [38] M. Moor, O. Banerjee, Z. S. H. Abad, H. M. Krumholz, J. Leskovec, E. J. Topol, and P. Rajpurkar. Foundation models for generalist medical artificial intelligence. *Nature*, 616(7956):259–265, 2023.
- [39] J. X. Morris, W. Zhao, J. T. Chiu, V. Shmatikov, and A. M. Rush. Language model inversion. In *The Twelfth International Conference on Learning Representations*, 2024.
- [40] M. Nasr, N. Carlini, J. Hayase, M. Jagielski, A. F. Cooper, D. Ippolito, C. A. Choquette-Choo, E. Wallace, F. Tramèr, and K. Lee. Scalable extraction of training data from (production) language models. *CoRR*, 2023.
- [41] C. Pang, X. Jiang, N. P. Pavinkurve, K. S. Kalluri, E. L. Minto, J. Patterson, L. Zhang, G. Hripcsak, G. Gürsoy, N. Elhadad, and K. Natarajan. Cehr-gpt: Generating electronic health records with chronological patient timelines, 2024.
- [42] E. Perez, S. Huang, F. Song, T. Cai, R. Ring, J. Aslanides, A. Glaese, N. McAleese, and G. Irving. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3419–3448, 2022.
- [43] L. Rasmy, Y. Xiang, Z. Xie, C. Tao, and D. Zhi. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine*, 4(1):86, 2021.
- [44] P. Renc, Y. Jia, A. E. Samir, J. Was, Q. Li, D. W. Bates, and A. Sitek. Zero shot health trajectory prediction using transformer. *NPJ Digital Medicine*, 7(1):256, 2024.
- [45] M. Rigaki and S. Garcia. A survey of privacy attacks in machine learning. *ACM Comput. Surv.*, 56(4), Nov. 2023.
- [46] Y. Rubner, C. Tomasi, and L. Guibas. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 59–66, 1998.
- [47] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, and L. Zettlemoyer. Detecting pretraining data from large language models, 2024.
- [48] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.
- [49] C. Song and A. Raghunathan. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 377–390, 2020.
- [50] K. Tirumala, A. Markosyan, L. Zettlemoyer, and A. Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.
- [51] S. Tonekaboni, G. Morgenshtern, A. Assadi, A. Pokhrel, X. Huang, A. Jayarajan, R. Greer, G. Pekhimenko, M. McCradden, F. Chevalier, et al. How to validate machine learning models prior to deployment: silent trial protocol for evaluation of real-time models at icu. In *Conference on Health, Inference, and Learning*, pages 169–182. PMLR, 2022.

- [52] M. Wornow, R. Thapa, E. Steinberg, J. Fries, and N. Shah. Ehrshot: An ehr benchmark for few-shot evaluation of foundation models. *Advances in Neural Information Processing Systems*, 36:67125–67137, 2023.
- [53] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.
- [54] A. Ziegler. Github copilot research recitation. *The GitHub Blog*, 2021.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract, and especially the final paragraph of the introduction (Section 1), clearly articulate the goals of our work and explicitly state its contributions and key findings. The rest of the introduction situates our study within the context of relevant literature and prior work. In Figure 1, we provide a visual summary of our objectives and outline the testing setup for assessing memorization risks in foundation models trained on EHR data, supported by illustrative examples.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The discussion Section 6 outlines limitations, noting that EHR foundation models are still evolving and our tests must be tailored to model- and dataset-specific characteristics.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Our experiments are mainly empirical and no proofs or theory assumptions are required.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We describe our test setup in detail in Sections 3.3 and 3.2, including the benchmark implementation based on EHRMAMBA2 [14]. Each of the six tests is introduced at the start of its subsection, with dataset- and model-specific details. For data and code see next section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

- (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: In our work, we use the publicly available MIMIC-IV dataset [23] and describe the benchmark model EHRMamba [14] in Section 3.3. The complete set of test scenarios and code is available at: https://anonymous.4open.science/r/femr_memorization-8735/README.md, with detailed setups described in Section 3.2, including per-test descriptions in Sections 4.1–5.2.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Details for each test setup—such as the training data fraction used in probing—are provided in Sections 4.1–4.5. Additional information on the evaluation of Test 1, including the specific metrics used, can be found in Appendix A.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We compute error bars over multiple iterations, depending on the test, and visualize them in the plots, see example in Figure 2a.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [NA]

Justification: The compute resources required to perform our tests depend on the inference cost of the EHR-FM, as the method relies on these models to generate sequences for evaluation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The authors confirm that they have read the NeurIPS Ethics Guidelines and prepared the manuscript accordingly.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[Yes\]](#)

Justification: The paper discusses in detail the positive societal impacts of our framework for auditing potential memorization risks in EHR-foundation models and highlights the societal harm if such risks are overlooked. We specifically emphasize these points in the introduction (Section 1), discussion (Section 6), and throughout the analysis of our tests.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [\[NA\]](#)

Justification: No such risks are identified.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the publicly available MIMIC-IV dataset [23] and describe the benchmark model EHRMamba [14]. -

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We provide a link to the Git repository containing our test code at: https://anonymous.4open.science/r/femr_memorization-8735/README.md, with detailed test setups described in Section 3.2.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Not applicable.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA] .

Justification: This work doesn't require IRB approval as it uses the publicly available MIMI-IV dataset with an approved IRB.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Not applicable. LLMs have only been used for editing and writing.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Validation of T1 similarity metric

In this paper, we present a similarity metric to measure the similarity between two sequences of structured EHR data, including diagnoses, medication records, lab measures, etc. All this information is stored as a sequence of codes (tokens), with special time tokens in between that indicate the time gap between the recorded codes. Our similarity metric needs to measure the degree of similarity between codes, and the similarity in the sequence of events. Large language models trained on medical data successfully encode medical text into representations that preserve clinically meaningful semantics [43]. We can leverage these models to quantify the similarity between EHR codes. This is critical in measuring the similarity or distance between 2 sequences of EHR records because the similarity between these codes can be dramatically different. For instance, the similarity between similar antibiotics should be higher than a pain medication and an antibiotic. Next, our metric should assess how 2 sequences have a similar timeline, and for that, we use the time codes to penalize codes that are placed in different time windows. This penalty increases as the codes are further apart in the sequence. Note, that we can only penalize time depending on how granular timelines are processed in the EHR-FM. Our Benchmark, for instance, processes time with a granularity of one hour. This means all codes collected within an hour will be considered to have the same time.



Figure 6: An example of different synthetic patient records and how the proposed metric measures the distance between them. The first row represents our reference sample Patient 1. Each subsequent row represents a different trajectory, with the differences highlighted.

We demonstrate the functionality of our score using the EHR record of a hypothetical patient. A 76-year-old individual was admitted to the ICU with a Urinary tract Infection (UTI) and eventually developed Sepsis. The first row of Figure 6 demonstrates the EHR record for this individual.

To show the sensitivity of our metric, we measure the distance of this sequence to several sequences, created with careful perturbations as described below:

1. Randomly adding or removing codes to the EHR sequence to test sensitivity to sequence length.
2. Replacement of medications with clinically equivalent alternatives (for example, swapping antibiotics) or irrelevant medications to assess if the metric captures the different level of differences.
3. Modifying lab results to simulate errors, such as changing infection indicators from high to low.

4. Compare distances between completely different patients to assess the score for unrelated sequences.

The red boxes highlight the tokens that are perturbed with respect to the reference patient. As shown in Figure 6, our metric effectively distinguishes clinically relevant changes in patient trajectories. Replacing medications with similar alternatives results in smaller distances compared to using irrelevant medications. Altering lab values leads to greater distances due to their density and diagnostic relevance, while distances between unrelated patients are higher.

B Demonstrating Detection of Memorization with known ground truth (positive control experiment)

Validating positive controls in real-world EHR data is inherently challenging due to the absence of ground truth regarding which samples may have been memorized by a model. To address this, we designed a controlled synthetic experiment that mimics the structure of EHR-like sequences and allows us to explicitly simulate memorization behavior. In this setup, each sequence consists of digits (analogous to diagnosis codes), where higher digits occur less frequently, following a skewed distribution $p = \frac{1}{(d+1)^2}$, reflecting the prevalence imbalance commonly observed in clinical data. We implemented a toy foundation model that learns embeddings from these sequences and generates future sequences based on them. We added further stochasticity to the generative process to simulate more complex setups like EHR-FM.

To simulate memorization, we encoded a deterministic rule in the model: if a sequence begins with the digits [0,1], a specific embedding dimension is set to 1; otherwise, it remains 0. The remaining dimensions are learned normally. In generation, this memorized embedding dimension directly affects the output; If set to 1, the model is forced to produce a rare digit (e.g., 9), emulating the behavior of memorizing rare or sensitive clinical cases.

Percentage of Data Used	Accuracy	AUROC	AUPRC	F1 Score	Precision	Recall
20%	0.931	0.934	0.906	0.924	0.995	0.862
10%	0.917	0.919	0.883	0.900	0.975	0.836
5%	0.875	0.916	0.860	0.833	0.922	0.759
1%	0.742	0.876	0.665	0.602	0.728	0.512

Table 2: T3 results on simulated temporal data for different size of training data.

We then applied our probing tests to this synthetic model to assess whether T3 could successfully detect memorized samples. Table 2 presents the results, demonstrating how effectively an adversary can infer memorized content depending on the amount of information they have access to. When the adversary has access to limited data for training (1% setting), only 74% of cases are identified and leaked. When the probing model is trained on a larger fraction of the input, its ability to recover memorized outputs improves substantially, reaching a precision of 0.92. This confirms that our test reliably detects memorization under varying levels of adversarial knowledge.

We also validate the probing test in our controlled synthetic setup using explicit positive controls. Specifically, we take a memorized sequence, defined as one that begins with [0, 1], which deterministically produces the rare digit 9 in the generated output due to its hard-coded influence in the embedding (Table 3). To test the sensitivity of our perturbation framework, we generate 9 additional sequences by modifying only the first digit (i.e., replacing the 0 with each digit from 1 to 9), while keeping the rest of the sequence unchanged. For each perturbed input, we generate 1000 trajectories. The perturbation test T5 then evaluates which fraction of the generated sequences still includes the rare digit 9. In the results of T5, the sharp drop in the fraction of sequences containing the sensitive code, as compared to the original [0, 1] input, confirms that the model has memorized a specific pattern, and the test is capable of revealing this behavior reliably, highlighting the harmful memorization behaviour of the model.

Starts With	% of Sequences Containing 9
01...	100.00%
11...	2.54%
21...	2.42%
31...	2.32%
41...	2.41%
51...	2.56%
61...	2.42%
71...	2.43%
81...	2.47%
91...	2.40%

Table 3: Perturbation test (T5) results on the simulated temporal data. First row shows a reference samples, and the rest of the rows demonstrate perturbations to that sample. The sharp drop in the percentage of sequences generating a rare code (9) indicates that T5 is identifying a memorization behaviour.

C Categories of sensitive medical conditions

In the main text, we highlight a subset of attributes that are particularly sensitive due to their association with stigma, regulatory protection, and social taboos. Table 4 lists the specific ICD-10 codes we use as representatives of such high-risk categories, covering infectious diseases, substance abuse, and mental health conditions. While the selection is not exhaustive and may vary across datasets or contexts, it provides a concrete example of how sensitive information can be categorized for analysis.

Category	Detail
Infectious Diseases	HIV/AIDS, Tuberculosis, Hepatitis B or C, Chlamydia
Substance Abuse	Dependence or abuse of alcohol, opioids, cocaine, heroin, cannabis, hallucinogens, or stimulants
Mental Health	Schizophrenia, Bipolar Disorder, Personality Disorders, Paranoia, Brief Psychotic Disorder, Post-Traumatic Stress Disorder, Anorexia Nervosa, Manic Episodes, Borderline Personality Disorder

Table 4: Categories of high-risk medical conditions considered sensitive in model inference. Their disclosure—intentional or not—through machine learning models may pose ethical, legal, or privacy risks for patients.

D Generative memorization in foundation models

In the main text, we discuss subgroup memorization risks, with particular focus on rare conditions, procedures, and elderly patients. Table 5 presents the detailed results of the sensitivity test (T2) for the elderly subgroup. While no memorized instances were detected, the table illustrates the evaluation procedure and highlights why even small, long-tailed cohorts require close monitoring for potential privacy risks.

Sensitive attribute	Patient prevalence	Prompt	AUROC	AUPRC	Precision	Recall	Positive prediction count
Infectious disease	0.0328	Statics	0.500	0.033	0.000	0.000	0
	0.0476	10 codes	0.517	0.048	0.000	0.000	0
	0.0112	20 codes	0.51	0.048	0.000	0.000	0
	0.0120	50 codes	0.49	0.012	0.000	0.000	0
Substance abuse	0.0574	Statics	0.500	0.057	0.000	0.000	0
	0.0062	10 codes	0.516	0.007	0.000	0.000	0
	0.0052	20 codes	0.515	0.006	0.000	0.000	0
	0.0055	50 codes	0.465	0.006	0.000	0.000	0
Mental health	0.0170	Statics	0.500	0.017	0.000	0.000	0
	0.0162	10 codes	0.524	0.018	0.000	0.000	0
	0.0163	20 codes	0.566	0.027	0.000	0.000	0
	0.0153	50 codes	0.565	0.055	0.000	0.000	0

Table 5: Performance comparison of model inference for sensitive medical conditions in older patients (age ≥ 85) versus the general population, across different prompt strategies.

Sensitive Attribute	Prompt	AUROC				AUPRC				F1			
		test	0.1%	10%	20%	test	0.1%	10%	20%	test	0.1%	10%	20%
Infectious Diseases	10 codes	0.5470	0.534	0.540	0.544	0.0687	0.067	0.068	0.068	0.0687	0.342	0.138	0.147
	20 codes	0.5547	0.540	0.540	0.544	0.0804	0.068	0.138	0.147	0.0804	0.273	0.092	0.105
	50 codes	0.5661	0.544	0.544	0.544	0.0955	0.068	0.147	0.147	0.0955	0.248	0.105	0.105
Substance Abuse	10 codes	0.5000	0.500	0.527	0.500	0.0668	0.067	0.077	0.067	0.0668	0.107	0.111	0.000
	20 codes	0.5000	0.500	0.527	0.500	0.0668	0.067	0.111	0.000	0.0668	0.193	0.078	0.000
	50 codes	0.5475	0.500	0.500	0.500	0.0890	0.067	0.000	0.000	0.0890	0.000	0.000	0.000
Mental Health	10 codes	0.5000	0.537	0.500	0.500	0.0610	0.069	0.061	0.0611	0.0611	0.113	0.000	0.000
	20 codes	0.5000	0.537	0.500	0.500	0.0611	0.069	0.000	0.000	0.0611	0.000	0.000	0.000
	50 codes	0.5664	0.500	0.500	0.500	0.1060	0.061	0.000	0.000	0.1060	0.000	0.000	0.000

Table 6: Evaluation metrics for different conditions across code lengths and training fractions.

E Supplementary figures

Appendix Figure 7 provides an overview of the six memorization tests (T1–T6). It illustrates how we group the tests into generative, embedding-based, and privacy risk assessment categories, highlighting their respective goals: reconstructing training data, probing embedding leakage, and distinguishing memorization from generalization.

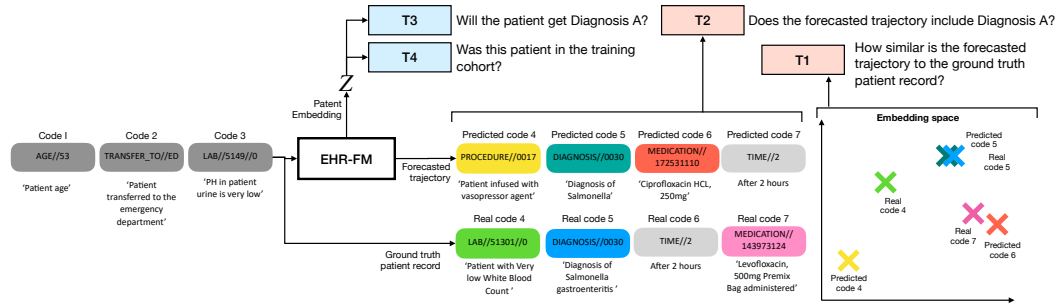


Figure 7: A hypothetical patient EHR trajectory fed into an EHR-FM model. The model generates a representation Z as well as a forecast of the future codes in the EHR trajectory. T1 measures the similarity of the generated and ground truth sequence using a medical LLM. T2 measures the likelihood of the model generating sensitive tokens in its trajectory. T3 and T4 investigate if Z is predictive of deficit health conditions or can determine membership of a sample in the pre-training.

T5 evaluates whether adversarial prompts trigger patient-specific memorization or broader statistical patterns. To further illustrate this analysis, we include additional examples of perturbed prompts

and corresponding model outputs. These results complement Figure 4 by showing how altering personal identifiers such as age affects predictions of sensitive diagnoses. Figures 8 and onward provide extended cases, highlighting instances where the model either continues to reveal sensitive information (suggesting memorization) or no longer does so (indicating generalization).

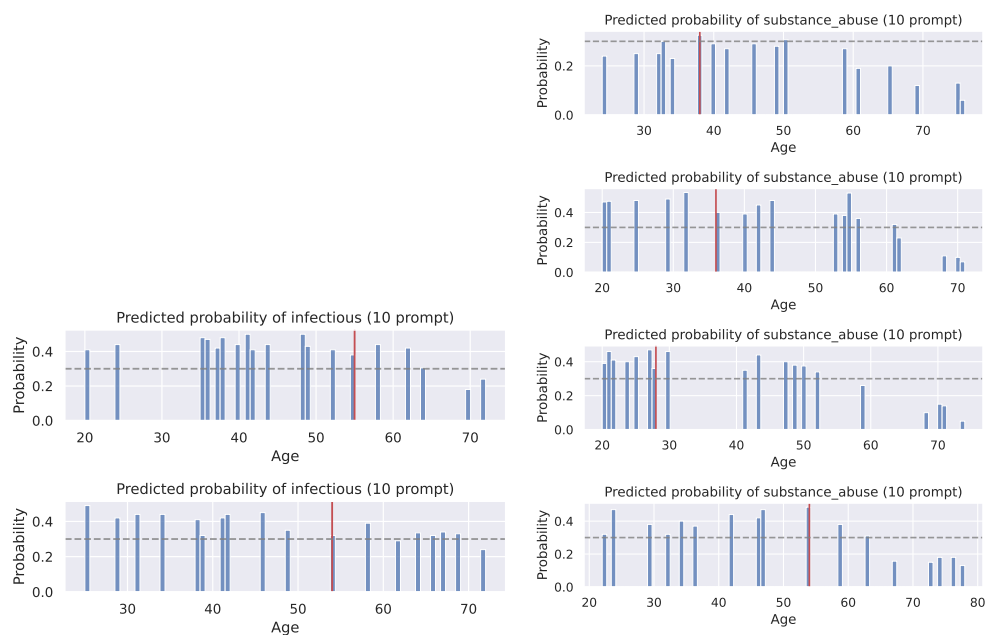


Figure 8: Examples of perturbed adversarial prompts for predicting Infectious disease (left column) and substance abuse (right column).