LARGE-SCALE ONLINE DEANONYMIZATION WITH LLMS

Simon Lermen[*1] **Daniel Paleka**[*2]
Joshua Swanson[2] **Michael Aerni**[2] **Nicholas Carlini**[3] **Florian Tramèr**[2]
[1]MATS [2]ETH Zurich [3]Anthropic [*]Equal contribution

ABSTRACT

We show that large language models can be used to perform at-scale deanonymization. With full Internet access, our agent can re-identify Hacker News users and Anthropic Interviewer participants at high precision, given pseudonymous online profiles and conversations alone, matching what would take hours for a dedicated human investigator. We then design attacks for the closed-world setting. Given two databases of pseudonymous individuals, each containing unstructured text written by or about that individual, we implement a scalable attack pipeline that uses LLMs to: (1) extract identity-relevant features, (2) search for candidate matches via semantic embeddings, and (3) reason over top candidates to verify matches and reduce false positives. Compared to classical deanonymization work (e.g., on the Netflix prize) that required structured data, our approach works directly on raw user content across arbitrary platforms. We construct three datasets with known ground-truth data to evaluate our attacks. The first links Hacker News to LinkedIn profiles, using cross-platform references that appear in the profiles. Our second dataset matches users across Reddit movie discussion communities; and the third splits a single user's Reddit history in time to create two pseudonymous profiles to be matched. In each setting, LLM-based methods substantially outperform classical baselines, achieving up to 55% recall at 90% precision compared to near 0% for the best non-LLM method. Our results show that the practical obscurity protecting pseudonymous users online no longer holds and that threat models for online privacy need to be reconsidered.

1 INTRODUCTION

The principle that individuals can be uniquely identified from surprisingly few attributes has been understood for decades. Sweeney's seminal work demonstrated that 87% of the U.S. population could be uniquely identified by just zip code, birth date, and gender (Sweeney, 2002). Narayanan and Shmatikov showed that anonymous Netflix ratings could be linked to public IMDb profiles using only a handful of movie preferences (Narayanan & Shmatikov, 2008), while De Montjoye et al. (2013) proved that four spatiotemporal points suffice to uniquely identify 95% of individuals in mobile phone datasets. Despite these vulnerabilities, the vast ecosystem of pseudonymous online accounts (Reddit throwaways, anonymous forums, review profiles, etc) has remained largely intact. The reason is simple: applying such attacks in practice has required either structured data amenable to algorithmic matching, or substantial manual effort by skilled investigators reserved for high-value targets (Garcia, 2017).

Our contributions. We demonstrate that large language models (LLMs) fundamentally change this calculus, enabling fully automated deanonymization attacks that operate on unstructured text at scale. Where previous approaches required predefined feature schemas, careful data alignment, and manual verification, LLMs can extract identity-relevant signals from arbitrary prose, efficiently search over millions of candidate profiles, and reason about whether two accounts belong to the same person. We show that the practical obscurity that has long protected pseudonymous users (the assumption that deanonymization, while theoretically possible, is too costly to execute broadly) no longer holds.

We validate this thesis across three deanonymization settings: (1) deanonymizing an online account to its real identity; (2) linking an identity to an unknown pseudonymous account; and (3) linking

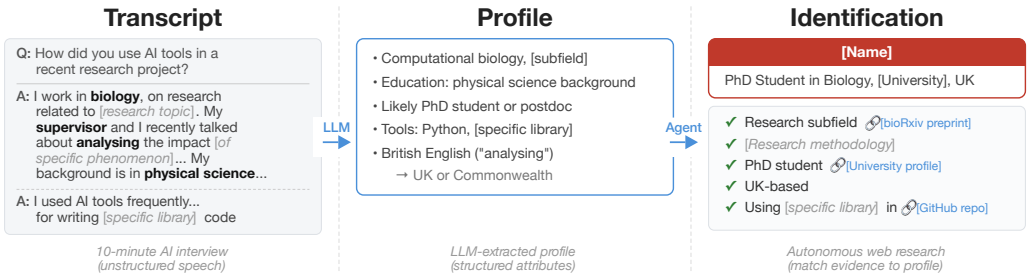


Figure 1: End-to-end deanonymization from a single interview transcript from Anthropic (2025) (*details altered to protect the subject's identity*). An LLM agent extracts structured identity signals from a conversation, autonomously searches the web to identify a candidate individual, and verifies the candidate matches all extracted claims.

pseudonymous accounts of the same person across different platforms or time periods. These settings capture distinct threat models (e.g., doxxing of an online account, a stalker targeting a victim, or an adversary consolidating a user's activity across contexts) and pose different technical challenges.

For the first setting, we demonstrate that state-of-the-art LLM agents show early signs of being able to perform end-to-end deanonymization fully autonomously on the open web. This is the most challenging setup we consider, which highlights the paradigm shift that LLMs bring about in online privacy. Given only an anonymous online profile, our LLM agents autonomously search the web, query databases, enumerate candidate identities, and reason over evidence to identify which identity the profile belongs to (Figure 1). In a study of Hacker News and Reddit profiles, these agents achieve 25–67% recall with precision 70–90%, replicating in minutes what would take hours for a dedicated human investigator. This capability requires no custom engineering: we simply prompt frontier agents with an online profile and ask the agent to uncover the identity behind it.

For the second and third settings (linking pseudonymous accounts to an identity or across platforms) we develop a systematic **Extract-Search-Verify** (ESV) pipeline that decomposes deanonymization into three LLM-augmented stages. In the *extraction* stage, we prompt LLMs to identify and structure identity-relevant features from unstructured posts: biographical details, writing style, temporal patterns, topical interests, incidental disclosures, etc. Unlike prior work that operated on predefined feature vectors (e.g., numerical movie ratings (Narayanan & Shmatikov, 2008)), our approach captures the rich, heterogeneous signals present in natural language. In the *search* stage, we encode extracted features into dense embeddings and perform approximate nearest-neighbor search over large candidate sets, enabling efficient retrieval from thousands or millions of profiles. In the *verification* stage, we provide an LLM with full context for top-ranked candidates and use extended reasoning to assess match likelihood, eliminating false positives that similarity scores alone cannot detect.

Our ESV pipeline substantially outperforms adaptations of prior deanonymization techniques (Narayanan & Shmatikov, 2008). Ablation studies confirm that frontier LLMs contribute at each pipeline stage: the verification step alone, which uses multi-step reasoning to rank candidates, improves precision at fixed recall substantially.

An orthogonal contribution of our work is an evaluation framework for large-scale deanonymization attacks, which we believe can be broadly useful for follow-up studies. Indeed, evaluating deanonymization attacks at scale poses inherent challenges, since ground-truth labels are difficult to obtain without compromising the privacy of real users. As a result, previous work evaluated scale through synthetic data (Narayanan & Shmatikov, 2009) and relied on manual verification or guess-work to validate attacks on real data (Narayanan & Shmatikov, 2008). We propose an alternative that balances ecological validity with research ethics. We identify profiles that are *not* fully anonymous (for instance, a Hacker News account whose "about" field links to a LinkedIn profile) then render them pseudonymous for evaluation by removing all direct identifiers. We then measure whether our methods can recover the removed link. Although this approach may not capture the behavior of the most privacy-conscious users, it provides verifiable ground truth at scale and enables rigorous comparison across methods.

Implications. Our findings have significant implications for online privacy. The average online user has long operated under an implicit threat model in which pseudonymity provides adequate protection because targeted deanonymization requires prohibitive effort (which would thus be reserved for high-value targets). LLMs invalidate this assumption. They do not so by exceeding human *capability*—the signals our models exploit are ones that a skilled investigator would also recognize—but by reducing *cost*.

When privacy attacks against "anonymized" structured data were introduced, it became rapidly clear that the only effective mitigation was simply not to release such data at all (Narayanan & Shmatikov, 2008). However, the unstructured text that enables our attacks is the very content that makes online communities valuable. We therefore argue that privacy expectations, platform policies, and social norms that govern pseudonymous participation online require urgent reconsideration. We hope that this work can spark that conversation.

2 AI AGENTS CAN FIND REAL IDENTITIES FROM ANONYMIZED USER DATA

The most direct form of a deanonymization attack is to discover someone's real identity based on their pseudonymous online posts. To evaluate whether LLM agents can perform this attack, we construct different datasets that closely mimic this scenario, but where we have a known ground truth correspondence of the pseudonymous and real identities.

We anonymize profiles by removing information that a direct web search would resolve to a specific person, while retaining information that requires reasoning to be identifying. We remove personal URLs, social media handles, and GitHub repositories (which directly identify the owner), generalize unique project names (e.g., "Founded C# Inn" becomes "Founded a programming community"), and retain institutions, demographics, interests, and colleague names (which are too broad to identify the profile owner alone). The goal is to remove enough information to create a task that would be nontrivial for a motivated human with access to a web search engine. To evaluate, we start from profiles with known identities, apply this anonymization, and check whether the agent recovers the correct person. We do this only for evaluation purposes; any in-the-wild attack on pseudonymous profiles would skip this step entirely.

In our setting, the agent receives a text description of a person (derived from their pseudonymous posts) and attempts to find their real identity by searching the web, cross-referencing sources, and reasoning over evidence.

Our pipeline proceeds as follows:

1. Summarize the user's posts into a profile containing stated and inferred facts about demographics, career history, interests, and so on.
2. Generate a search query from the profile. Anonymize the query by removing names, handles, and unique identifiers that would enable direct search.
3. Pass the anonymized query to an LLM agent with web search tools. The agent iteratively searches, cross-references findings, and attempts to identify the person.

Following Li (2026), to prevent misuse, we do not publish the agent, exact prompts, or tool configurations used.

A simple agentic setup works. We find the default settings of a commonly used agent with minimal prompting work well. The main practical challenges are (1) selecting which user comments to include in the summary; (2) anonymizing the profile enough to make the task nontrivial; (3) retrying when the agent refuses to do the task; (4) making sure the agent does not output a match when it is not certain.

2.1 RESULTS

We evaluate our agentic pipeline on three ground-truth datasets starting from Hacker News and Reddit profiles. In the interest of research ethics, **we do not evaluate our method on any truly pseudonymous accounts on Hacker News and Reddit**. We additionally report deanonymization of genuinely pseudonymous Anthropic Interviewer transcripts that were the subject of prior deanonymization research (Li, 2026).

Hacker News → LinkedIn. We collect 338 Hacker News (HN) users who linked a LinkedIn profile in their HN bio, providing verified real-world identities as ground truth. We summarize each user's HN activity (comments and stories) into a structured profile, anonymize it following the procedure described above, and pass it to the agent. The agent correctly identifies 226 of 338 targets (67%) at 90% precision (95% CI: 86–93%; 25 incorrect identifications, 86 abstentions).

Reddit academics. We curate 25 identifiable Reddit users posting on academic subreddits whose usernames reveal their real names. We manually verified each ground-truth identity, excluding ambiguous cases. Then we anonymize their profiles to remove any direct identifiers, such as names or paper titles; but preserve any research interests or institutions they mention on their social media profile. The agent correctly identifies 13 of 25 (52%) with 72% precision (5 incorrect, 7 abstentions).

Reddit career discussions. We collect 36 users who posted their own LinkedIn URL for resume review on a software engineering career subreddit. We manually check each ground-truth identity, excluding ambiguous cases as good as possible. After anonymizing their Reddit profiles, the agent correctly identifies 9 of 36 (25%) at 90% precision (1 incorrect, 26 abstentions). The lower recall likely reflects that career discussion posts are less uniquely identifying: when we remove direct identifiers, many user's comments discuss generic software engineering topics without mentioning employers, projects, or credentials that would distinguish them from other users.

Anthropic Interviewer transcripts. Anthropic publicly released transcripts of 125 AI-conducted interviews with scientists (Handa et al., 2025; Anthropic, 2025), in which participants described their research projects in detail. Li (2026) showed that LLM agents with web search can match project descriptions in these transcripts to published papers; specifically, they use a task decomposition approach (Jones et al., 2024) to bypass safeguards by breaking down the attack into individually benign tasks. They ultimately run their attack on 24 scientists conversations and recover 6 identities. We attack the same dataset with our pipeline, without task decomposition. Concretely, we constructing profiles from transcripts in a similar way as we do for Hacker News and Reddit users; and then run the agentic search on 33 scientists that describe their research in enough detail to be searchable (as judged by an LLM). We skip the anonymization step here: the transcripts by design contain no platform handles or usernames, and we report deanonymization judged by a human, following the precedent set by Li (2026).

3 Scalable deanonymization framework with LLMs

While the previous case study demonstrates that AI agents can deanonymize users autonomously, it has notable shortcomings.

As a solution, we introduce a deanonymization framework based on the seminal work of Narayanan & Shmatikov (2008). Their attack serves as a "classical" baseline, which is restricted to structured data and conventional algorithms. Inspired by this classical method, we introduce a deanonymization framework consisting of (mostly) independent components. Our framework allows us to systematically demonstrate how LLMs can augment and improve typically deanonymization attacks.

We first define our threat model and deanonymization framework in Section 3.1 and Section 3.2, respectively. In Section 3.3, we then introduce an evaluation procedure to quantify the effectiveness of LLM-based deanonymization for different settings.

3.1 Threat model

We build upon Narayanan & Shmatikov (2008), which introduces large-scale deanonymization attacks by reconstructing *micro-data*. Micro-data is information at the level of an individual, such as "gave Twilight a 5-star rating", "lives in Texas", or "never capitalizes sentences". This information alone may not be identifying. However, it can identify a pseudonymous account by *matching* their micro-data against a database of micro-data with known identities. The attack by Narayanan & Shmatikov (2008) (henceforth termed the "Narayanan-attack") does precisely this: it reconstructs anonymized user data from the Netflix Prize competition by matching movie ratings to public IMDb profiles.

We use a simpler attack model that focuses on matching instead of reconstruction. Our goal is to empirically demonstrate that LLMs effectively deanonymize pseudonymous online identities; there-

fore, we forgo reconstruction of micro-data and instead just consider matching profiles. Concretely, our attacker is given a *query user* profile and a set of *candidate user* profiles. Given those inputs, the attacker either returns a best-guess match of the query user in the candidate set or abstains. The attacker's goal is to produce a correct guess if the query user has a corresponding candidate profile, and it should abstain if there is no matching candidate.

The matching problem's difficulty depends on two key factors: the size of the candidate set and how likely the query user is to have a matching candidate. First, larger candidate sets are trivially harder: Given only two candidates (one of which is a correct match), even random guessing correctly guesses about half of all queries. In contrast, identifying the correct user in 10k candidates requires a much stronger attacker. Second, if there is no matching candidate for a given query user, then any guess (i.e., not abstaining) is an error. Thus, the probability of a query user being matchable affects the attacker's precision.

We explicitly instantiate the attack model in terms of difficulty. For simplicity, we generally use a fixed candidate set for each setting, and we assume a best-case scenario where every query user has a true match in the candidate set.

3.2 Deanonymization framework

Our deanonymization framework structures around the Narayanan-attack's four main components. This allows us to study different LLM-based deanonymization attacks in a systematic framework and enables a direct comparison to classical methods. Concretely, our framework defines deanonymization attacks through the following four steps:

1. **Micro-data**: The attack first needs to obtain micro-data for query and candidate users. The original Narayanan-attack directly receives structured micro-data in the form of Netflix and IMDb rating vectors as its input. For arbitrary online profiles, we can use LLMs to extract semi-structured summaries from unstructured posts and comments.
2. **Scoring function**: For every query-candidate pair, the attack then calculates a similarity score between the two profiles' micro-data. This might be a weighted inner product between binary vectors of micro-data or the cosine similarity between LLM embeddings.
3. **Record selection**: Based on the similarity scores, the attack selects a "best-guess" candidate per query user. For example, this could be the highest-scoring candidate or an LLM picking the best out of a few high-scoring candidates.
4. **Matching criterion**: Finally, the matching criterion decides whether to output a "best-guess" candidate or to abstain. We use a threshold-based approach to control the attack's confidence, either explicitly (e.g., by thresholding a similarity-based confidence as in the Narayanan-attack), or implicitly (e.g., by ranking query-candidate pairs and abstaining for low-raking ones).

The classical Narayanan-attack requires structured micro-data. Users of pseudomized platforms, such as Reddit, inadvertently reveal lots of information. However, because most of it is only available as unstructured text, the Narayanan-attack can only utilize a small subset of the available micro-data (e.g., subreddit memberships). Thus, while powerful, the Narayanan-attack is limited to specific settings where sufficient structured micro-data is available.

LLMs can process and reason over unstructured micro-data. LLMs excel at extracting information from unstructured text. Thus, they enable a larger class of deanonymization attacks while potentially exploiting much more micro-data. Additionally, in contrast to classical algorithms, LLMs have moderate reasoning and deduction capabilities. Those capabilities, as we will demonstrate, enable highly effective deanonymization for domains with limited structured micro-data—where classical attacks fail.

3.3 Evaluation

We instantiate our deanonymization framework in settings with verifiable ground truth. To obtain such ground truth, we use two approaches: First, we search for user profiles that explicitly link themselves to other platforms (e.g., a Hacker News user posts their LinkedIn account). Since those

profiles trivially expose links, we carefully sanitize them. In practice, we found only few users who explicitly link their pseudonymous account to other platforms. Hence, as a second option, we split a single user's profile into two semantically distinct parts (e.g., temporally). Those splits provide the strongest source of ground truth information at the cost of being less realistic.

Verifiable ground truth introduces distributional biases. The corresponding profiles are likely easier to deanonymize than an average profile: the very fact that ground truth exists implies that the user may not have cared about anonymity in the first place. Similarly, two split-profiles of a single user are inherently alike, whereas two pseudonymous accounts of the same person (e.g., an official and a pseudonymous alt account) might expose more heterogeneous micro-data.

We thus focus on methods with very few false positives. Due to the aforementioned biases, our evaluation on ground truth settings might overestimate an attack's recall (how many users are successfully deanonymized). But we argue that the false positive rate (i.e., the likelihood of a wrong guess instead of abstaining) does transfer to real-world attacks; there is no reason why a well-calibrated attacker should make more wrong guesses on the average internet profile than on our ground truth profiles. Similar to Narayanan & Shmatikov (2008), we hence focus on attacks with a very low false positive rate. This ensures that, whenever our attacks do return a guess, the guess is likely to be correct—independent of the setting.

Concretely, we report Recall@Precision for high precision values. Informally, recall measures the fraction of correct attacker guesses, while precision measures how many non-abstentions are correct. We hence define *recall* (or *True Positive Rate (TPR)*) as the fraction of matchable users (i.e., users with a corresponding match in their candidate set) that the attacker guesses correctly, and *precision* as the fraction of all non-abstentions that are correct. Since all our attacks return guesses with a confidence score, we use a confidence threshold to control precision.

4 Finding Hacker News accounts of LinkedIn users

We begin our evaluation with cross-platform matching between LinkedIn and Hacker News, where ground truth comes from users who explicitly linked their accounts. This setting closely mirrors a real-world attack.

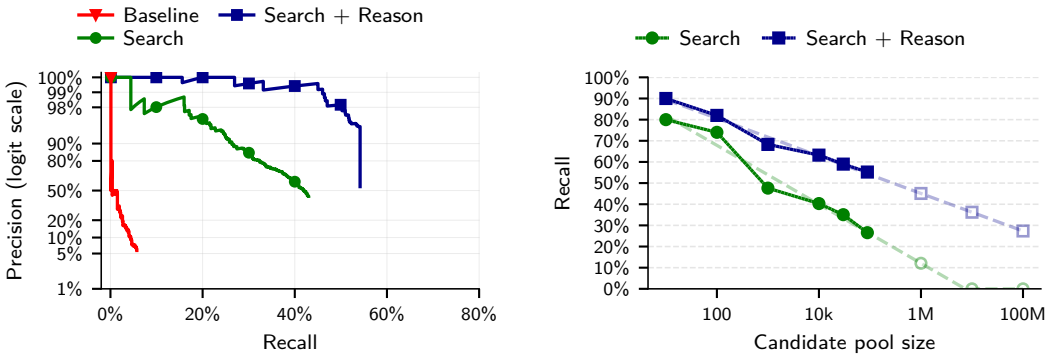
We construct a dataset to simulate the task of finding a known person's pseudonymous account. We collect 987 LinkedIn profiles linked to 995 Hacker News (HN) accounts (ground truth is established by users who posted their LinkedIn URL in their HN bio), drawn from a candidate pool of approximately 89,000 active HN users. The LinkedIn side represents the known identity with real professional profiles. The HN side serves as the anonymized target: we remove names, URLs, and other direct identifiers from bios using an LLM to prevent trivial matching. The task is to match a LinkedIn profile to the corresponding LLM-anonymized HN account.

Narayanan-attack baseline. As a baseline, we adapt the approach of Narayanan & Shmatikov (2008) to professional attributes: programming languages, cities, companies, schools, job roles, and seniority levels. Each user is represented as a binary vector over 332 attributes, and we rank candidates by rarity-weighted Jaccard similarity.

LLM-based embeddings outperform hand-crafted features. Following the framework introduced in Section 3, we compare the Narayanan-attack baseline against our LLM-based approach. Instead of manually extracting structured attributes into binary vectors, in the *extraction* stage we use an LLM to summarize each user's activity (comments, stories, and other posts) into a biographical profile, which is then embedded to allow efficient comparison across all users. Figure 2a shows the precision-recall tradeoff for both methods. The embedding-based approach substantially outperforms the NS-method baseline across all metrics, which ranks 5.1% of candidates correctly as the Top 1 match.

LLM re-ranking further improves matching. Following our ESV pipeline, we perform *search* with embeddings to narrow candidates to the top-100, then apply LLM-based re-ranking as the *verification* stage. This allows the LLM to verify matches with constant cost per query user. As shown in Figure 2a, re-ranking yields substantial gains in precision and recall over the other methods. The ability of LLMs to select from a number of candidate and use reasoning complements embedding-based retrieval.

Scaling to larger candidate pools. A crucial question for any deanonymization attack is how performance scales with candidate pool size. Here, we evaluate how recall at 90% precision degrades as



(a) Precision-recall curves.

(b) Scaling with candidate pool size.

Figure 2: LinkedIn to Hacker News matching. (a) Precision-recall curves comparing methods: LLM-based embeddings outperform the Narayanan-attack baseline, and LLM re-ranking of the top-100 candidates further improves performance. (b) Matching accuracy vs. candidate pool size. Dashed lines show log-linear extrapolation to larger pools.

Table 1: Recall at different precision thresholds for HN-LinkedIn matching (987 queries). Reason uses Grok 4.1 Fast (xAI, 2025) for selection and GPT-5.2 for verification. 95% Wilson CIs shown.

Method	90% Prec	98% Prec	99% Prec
Narayanan baseline	0.1% (0.0–0.6)	0.1% (0.0–0.6)	0.1% (0.0–0.6)
Search (embedding)	26.3% (23.7–29.2)	16.1% (13.9–18.5)	4.4% (3.3–5.9)
Reason (low)	**55.0%** (51.9–58.0)	**44.8%** (41.8–47.9)	36.0% (33.1–39.0)
Reason (high)	**54.2%** (51.1–57.2)	**50.0%** (46.9–53.0)	**45.1%** (42.1–48.2)

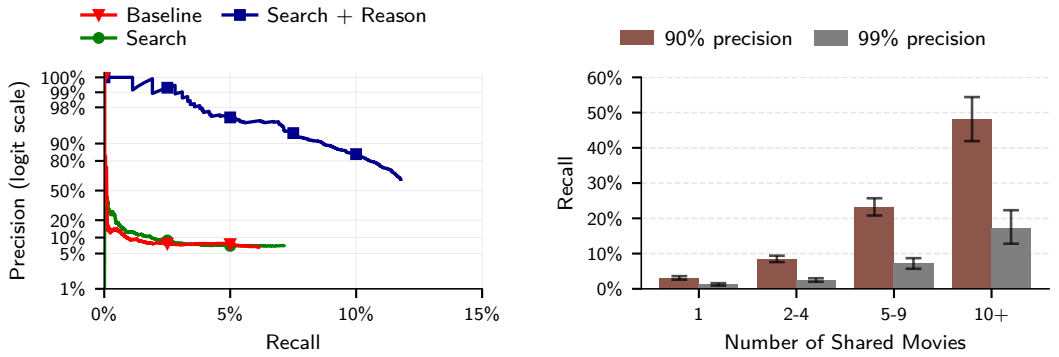
the candidate pool grows from 1k to 89k users. As shown in Figure 2b, both Search and Reason degrade log-linearly with pool size. Reason achieves 68.3% recall at 1k candidates and retains 55.2% at 89k—a loss of only 13 percentage points across nearly two orders of magnitude. Search drops more steeply, from 47.6% to 26.6% over the same range. Log-linear extrapolation to 1M candidates projects Reason at approximately 45% recall.

5 Linking users across Reddit movie communities

To enable direct comparison with the classical Netflix Prize deanonymization attack (Narayanan & Shmatikov, 2008), where users were identified across Netflix and IMDb based on movie preferences, we construct a separate experiment using only movie-related Reddit activity. This provides a clean comparison: both methods operate on the same micro-data features (subjective movie reviews) but differ in how they represent users—we use transformer-derived text embeddings, while Narayanan & Shmatikov (2008) relied on hand-crafted features.

This setup mimics a situation in which a user maintains pseudonymous accounts on different platforms discussing similar topics. In our case, the communities discuss either mainstream movies (r/movies) or more niche alternatives. This also provides a method to understand the relationship between how much micro-data someone shares and how identifiable they become: unlike most online activity, movie discussions offer an intuitive discrete metric—the number of movies discussed and the overlap of shared movies across datasets.

Our Reddit movies dataset consists of different movie discussion communities ("subreddits") from 2024. We collected data from six movie-related subreddits, partitioning r/movies (the largest general movie discussion community) from five smaller specialized communities (r/horror, r/MovieSuggestions, r/Letterboxd, r/TrueFilm, and r/MovieDetails); we refer to the union of these five as the *alternative movie communities*. Crucially, in the *extraction* stage we use LLMs to convert unstructured Reddit submissions—which may contain discussions, recommendations, or casual mentions—into structured movie reviews with movie titles, ratings, and dates. Ratings are typically estimated by the LLM based on sentiment. This transformation allows us to construct user profiles consisting solely



(a) Precision-recall curve for Reddit matching.

(b) Recall by number of shared movies up to 60%.

Figure 3: (a) Precision-recall curves comparing high reasoning verification with simple classifiers based on the relative gap between the two best matches (separation) by cosine similarity of transformer-derived embeddings, as well as a hand-crafted vector method analogous to the Narayanan-attack. (b) Recall at precision thresholds by number of shared movies, using GPT-5.2 high reasoning. Users who share more movies are substantially easier to identify: at 90% precision, recall increases from 3.1% (1 shared movie) to 48.1% (10+ shared movies).

Stage 1: Re-ranking Top-k nearest neighbors

Query: *Neon Horizon* 7/10 "loved **Timothee** in the **rooftop fight**"
A: *Neon Horizon* 8/10 "great visuals"
B: *Neon Horizon* 6/10 "**Timothee** carried it, that **rooftop scene** was insane"
C: *Neon Horizon* 3/10 "**Timothee** was miscast, ruined the movie"

LLM Output: B (conf: 0.85). *"A lacks actor/scene details. C opposite sentiment on **Timothee**. B matches on **Timothee** + **rooftop**."*

Stage 2: Verification of selected match

Query: "loved **Timothee** in the **rooftop fight**, the choreography with the **neon signs** was so good"
B: "**Timothee** carried it, that **rooftop scene** with all the **neon lighting** was insane"

LLM Output: MATCH (conf: 0.94). *"Both mention **Timothee**, **rooftop scene**, and **neon** aesthetic—three specific details unlikely to co-occur by chance."*

Figure 4: Two-stage matching of movie reviews with simplified synthetic user profiles. The *search* stage retrieves candidates via embedding similarity; the *verification* stage re-ranks and verifies matches using full profile text. Bold text highlights discriminating details used by the LLM.

of movie reviews, which can be transformed into Netflix rating vectors analogous to Narayanan & Shmatikov (2008). Profiles contain only movie reviews, they do not contain any other micro-data which may be used to match users.

This Reddit movies dataset provides a challenging matching task: users share an average of 2.5 movies between datasets, with 48% of users sharing only a single movie across their r/movies and alternative movie communities accounts. We filter to users with at least one shared movie, as users with no overlap cannot be reasonably matched by movie reviews. This leaves us with 9781 users. We use fuzzy string matching of movie titles to determine the number of shared movies across splits.

The model outputs a confidence score for each match decision via function calling; this allows us to use a threshold to trade off precision and recall. Figure 4 shows example confidence values for both re-ranking and verification stages. In contrast to Section 4, where we directly use the re-ranking output, here we first re-rank using only reviews with matching movie titles between query and candidates, then verify the top match using full profiles. This two-stage approach reduces cost—re-ranking sees shorter inputs while verification runs only on the top candidate—and allows using a more capable model for verification.

Using GPT-5.2 (OpenAI, 2025a), we evaluate whether setting reasoning effort to low or high affects matching performance. High reasoning significantly outperforms low reasoning across all thresholds: at 90% precision, high reasoning achieves 8.5% recall compared to 5.2% for low reasoning (a 63% relative improvement). At 99% precision, high reasoning achieves 2.8% recall versus 1.4% for low

reasoning (a 100% relative improvement). These differences are statistically significant ($p < 0.05$). Our results demonstrate that increased reasoning effort (test-time compute) substantially improves deanonymization success.

Users who share more content are substantially easier to identify. As shown in Figure 3b, recall increases sharply with the number of shared movies: at 90% precision, recall rises from 3.1% for users with one shared movie to 48.1% for users with ten or more.

6 Related work

Deanonymization attacks predate LLMs but were limited to structured data or high manual effort. Narayanan & Shmatikov (2008) demonstrated that movie ratings can uniquely identify individuals across platforms: by matching Netflix Prize data against public IMDb profiles, they deanonymized users at scale using statistical techniques on structured micro-data. Their work established the threat model we build upon, showing that even sparse, seemingly innocuous data can be identifying when matched against auxiliary datasets. The same authors showed that social graph structure alone can deanonymize users by matching connection patterns across networks (Narayanan & Shmatikov, 2009). Wondracek et al. (2010) exploited group membership information to deanonymize social network users, showing that the groups a user joins are often sufficient to uniquely identify them. More recently, Ederer et al. (2024) deanonymized users of the Economics Job Market Rumors forum by exploiting weaknesses in its username generation scheme, recovering IP addresses for 66% of posts. These classical approaches required either structured features, exploitable technical vulnerabilities, or graph structure—none could operate on unstructured text at scale.

Stylometry identifies users through writing style (Stamatatos, 2009; Tyo et al., 2022); our approach relies on semantic information—what users write about—rather than how they write.

LLMs enable inference of personal attributes from unstructured text. Staab et al. (2024) show that LLMs can infer personal attributes such as location, occupation, and income from text with high accuracy, demonstrating privacy risks beyond training data memorization. Du et al. (2025a) extend this line of work with AutoProfiler, a system of four specialized LLM agents that collaboratively extract sensitive attributes from pseudonymous platform activity. They report 85-92% accuracy in attribute extraction, demonstrating that automated profiling can be deployed at web scale. However, their evaluation of actual deanonymization—linking extracted attributes to real identities—relies on manual LinkedIn searches and reports k-anonymity metrics rather than the high-precision matching we focus on. Conversely, Staab et al. (2025) show that LLMs can be used to anonymize text while preserving utility, suggesting a potential defensive application of the same capabilities. Du et al. (2025b) survey emerging privacy risks from LLM deployment, distinguishing training-time data leakage from deployment-time threats such as automated profiling and social engineering.

Recent work has begun exploring LLM-based deanonymization directly. Li (2026) showed that agentic LLMs can deanonymize interview participants via web search, demonstrating that LLM agents make re-identification attacks low-effort: with a few natural-language prompts, off-the-shelf tools can search the web, cross-reference details, and propose matches. Their work on the Anthropic Interviewer dataset (Handa et al., 2025; Anthropic, 2025) recovered 6 of 24 scientist identities by matching project descriptions to published papers, using task decomposition (Jones et al., 2024) to bypass safeguards. Nyffenegger et al. (2024) evaluate LLM re-identification capabilities on court decisions, finding that despite high re-identification rates on Wikipedia, even the best LLMs struggled with anonymized legal documents—suggesting that deanonymization difficulty depends heavily on domain-specific context. Our work extends this line of research by developing a systematic framework for LLM-based deanonymization and evaluating it across multiple platforms and attack settings.

More broadly, deanonymization is one of many ways LLMs empower adversaries: Carlini et al. (2025) argue that LLMs alter the economics of cyberattacks by enabling tailored attacks on a user-by-user basis, and Heiding et al. (2026) demonstrate that LLM agents can autonomously construct comprehensive profiles for spear phishing.

7 DISCUSSION

Deanonymization is one instance of AI acting as an "information microscope" that makes previously manual and expensive attacks scalable (Hammond, 2025). Our papers shows LLMs democratize deanonymization, the asymmetry between attack cost and defense cost may force a fundamental reassessment of what can be considered private online.

What do our findings mean for the future of privacy? Governments could link pseudonymous accounts to real identities for surveillance of dissidents or activists; corporations could connect forum posts to customer profiles for hyper-targeted advertising; attackers could build profiles at scale for personalized social engineering.

Our evaluation relies on ground truth datasets that may overestimate real-world success rates. Users who publicly link their accounts or share identifying information may still share more information than they would for truly pseudonymous accounts, even considering our anonymization step. However, measuring deanonymization performance requires ground truth, and we cannot verify matches for users who have not revealed their identities. That our methods work across a broad range of experimental setups suggests they generalize beyond any single evaluation setting.

Platforms should assume that pseudonymous users can be linked across accounts and to real identities at scale, influencing decisions on data access policies; users should likewise not assume that posting under a pseudonym provides meaningful protection. Rate limiting API access, detecting automated scraping, and restricting bulk data exports may slow but not prevent these attacks; LLM providers could also monitor model use to detect deanonymization attempts (Sumers et al., 2025).

Classical anonymization frameworks such as k-anonymity (Sweeney, 2002) and differential privacy (Dwork, 2006; Dwork & Roth, 2014) were designed for structured databases with explicit identifiers and assume attackers use direct matching or statistical queries. These frameworks do not account for the types of attacks we demonstrate; data releases should consider such threats when evaluating privacy risks.

Do models use reasoning or memorization? One might wonder whether LLMs succeed at deanonymization because they memorized Reddit or Hacker News data during training. The fact that increasing reasoning effort substantially improves performance (Section 5) provides tentative evidence that reasoning plays a significant role. The training data for LLMs is typically not openly revealed, making it challenging to isolate these factors. We suspect that Hacker News and Reddit are part of most training corpora but LinkedIn profiles are not. More broadly, even if memorization plays a role, this does not diminish the privacy implications: many social media platforms are included in LLM training corpora, so the deanonymization threat would only be reduced for platforms excluded from training data.

8 CONCLUSION

We demonstrate that LLMs enable deanonymization of pseudonymous online accounts at scale, outperforming classical methods. Across cross-platform matching, profile splitting, and agentic web search, LLM-based methods consistently achieve higher recall at equivalent precision compared to classical baselines based on hand-crafted features. In many cases, they can perform attacks that wouldn't have previously been possible.

These attacks require only publicly available models and standard APIs. Our pipeline uses only publicly available embedding models, standard LLM APIs, and LLM-agent scaffolding, placing them within reach of moderately resourced adversaries.

The assumption that pseudonymity provides meaningful protection online is no longer valid. Users who post under persistent usernames should assume that adversaries can link their accounts to real identities or to each other, and that the probability rises with each piece of micro-data they post.

Refusal is not a meaningful barrier to misuse. Li (2026) bypass refusal via task decomposition; we find that with properly formatted queries, refusal rates are low even without decomposition.

REFERENCES

Anthropic. Anthropic Interviewer dataset, December 2025. URL https://huggingface.co/datasets/Anthropic/AnthropicInterviewer. Archived at https://web.archive.org/web/20251207050016/https://huggingface.co/datasets/Anthropic/AnthropicInterviewer.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Nicholas Carlini, Milad Nasr, Edoardo Debenedetti, Barry Wang, Christopher A. Choquette-Choo, Daphne Ippolito, Florian Tramèr, and Matthew Jagielski. LLMs unlock new paths to monetizing exploits. *arXiv preprint arXiv:2505.11449*, 2025. URL https://arxiv.org/abs/2505.11449.

Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3(1):1376, 2013.

Yuntao Du, Zitao Li, Bolin Ding, Yaliang Li, Hanshen Xiao, Jingren Zhou, and Ninghui Li. Automated profile inference with language model agents. *arXiv preprint arXiv:2505.12402*, 2025a.

Yuntao Du et al. Beyond data privacy: New privacy risks for large language models. *arXiv preprint arXiv:2509.14278*, 2025b.

Cynthia Dwork. Differential privacy. In *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 4052, pp. 1–12, 2006.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Florian Ederer, Paul Goldsmith-Pinkham, and Kyle Jensen. Anonymity and identity online. *arXiv preprint arXiv:2409.15948*, 2024.

Jennifer Lynnae Garcia. The evidentiary trail down Silk Road. Master's thesis, Boston University – Metropolitan College, 2017. URL https://www.researchgate.net/publication/319164300_The_Evidentiary_Trail_Down_Silk_Road.

Google DeepMind. Gemini 3 pro model card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf, 2025. Model card updated December 2025. Accessed: 2026-02-18.

Samuel Hammond. AI and Leviathan: Part I, 2025. URL https://www.secondbest.ca/p/ai-and-leviathan-part-i.

Karina Handa, Max Stern, Sandy Huang, Jessica Hong, Esin Durmus, Miles McCain, Grace Yun, Alex Alt, Tiffany Millar, Alex Tamkin, Julia Leibrock, Stuart Ritchie, and Deep Ganguli. Introducing Anthropic Interviewer: What 1,250 professionals told us about working with AI, 2025. URL https://anthropic.com/research/anthropic-interviewer. Archived at https://web.archive.org/web/20251204184855/https://www.anthropic.com/research/anthropic-interviewer.

Fred Heiding, Simon Lermen, Andrew Kao, Claudio Mayrink Verdun, Bruce Schneier, and Arun Vishwanath. Evaluating large language models' ability to automate spear phishing. *Expert Systems with Applications*, 314:131546, 2026. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2026.131546. URL https://www.sciencedirect.com/science/article/pii/S0957417426004598.

Erik Jones, Anca Dragan, and Jacob Steinhardt. Adversaries can misuse combinations of safe models. *arXiv preprint arXiv:2406.14595*, 2024.

Tianshi Li. Agentic LLMs as powerful deanonymizers: Re-identification of participants in the Anthropic Interviewer dataset. *arXiv preprint arXiv:2601.05918*, 2026.

Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pp. 111–125, 2008. doi: 10.1109/SP.2008.33.

Arvind Narayanan and Vitaly Shmatikov. De-anonymizing Social Networks. In *2009 30th IEEE Symposium on Security and Privacy*, pp. 173–187. IEEE, 2009. doi: 10.1109/SP.2009.22.

Alex Nyffenegger, Matthias Stürmer, and Joel Niklaus. Anonymity at risk? Assessing re-identification capabilities of large language models in court decisions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2433–2462, Mexico City, Mexico, jun 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.157. URL https://aclanthology.org/2024.findings-naacl.157/.

OpenAI. GPT-5.2 system card. Technical report, OpenAI, December 2025a. URL https://cdn.openai.com/pdf/3a4153c8-c748-4b71-8e31-aecbde944f8d/oai_5_2_system-card.pdf. Technical report on GPT-5.2 model capabilities and safety evaluations.

OpenAI. Openai gpt-5 system card, 2025b. URL https://arxiv.org/abs/2601.03267.

Robin Staab, Mark Vero, Mislav Balunović, and Martin Vechev. Beyond memorization: Violating privacy via inference with large language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ySXVKYNzad.

Robin Staab, Mark Vero, Mislav Balunovic, and Martin T. Vechev. Large language models are advanced anonymizers. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=82p8VHRsaK.

Efstathios Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556, 2009.

Theodore Sumers, Raj Agarwal, Nathan Bailey, Tim Belonax, Brian Clarke, Jasmine Deng, Evan Frondorf, Kyla Guru, Keegan Hankes, Jacob Klein, Lynx Lean, Kevin Lin, Linda Petrini, Madeleine Tucker, Ethan Perez, Mrinank Sharma, and Nikhil Saxena. Monitoring computer use via hierarchical summarization, 2025. URL https://alignment.anthropic.com/2025/summarization-for-monitoring.

Latanya Sweeney. k-Anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

Jacob Tyo, Bhuwan Dhingra, and Zachary C Lipton. On the state of the art in authorship attribution and authorship verification. *arXiv preprint arXiv:2209.06869*, 2022.

Gilbert Wondracek, Thorsten Holz, Engin Kirda, and Christopher Kruegel. A practical attack to de-anonymize social network users. In *2010 IEEE Symposium on Security and Privacy*, pp. 223–238. IEEE, 2010. doi: 10.1109/SP.2010.21.

xAI. Grok 4.1 fast and agent tools api. https://x.ai/news/grok-4-1-fast, November 2025. Agentic tool-calling model with 2M token context window.

A Ethical Considerations

Stakeholder Analysis. The primary stakeholders are social media users whose privacy could be compromised by deanonymization attacks. General social media users who post under pseudonyms expecting privacy, vulnerable populations who depend on anonymity (activists, abuse survivors, whistleblowers), and the specific users whose data we used in experiments (Reddit users, HN users, etc.). Secondary stakeholders include platforms, researchers, potential malicious actors, and society broadly.

Potential Harms. The primary potential harm is that publishing this research could inspire malicious deanonymization attacks. Concrete harms include stalking and harassment, doxxing of activists or vulnerable individuals, corporate surveillance and targeted manipulation, government surveillance and suppression of dissent, and chilling effects on free speech if people fear being identified. However, these capabilities already exist in deployed LLMs; we are not introducing novel attack vectors but documenting existing risks.

Potential Benefits. The primary benefit is raising awareness of privacy risks that already exist due to widely available LLM capabilities. Users can make informed decisions about what they share online and add better privacy measures. Platforms can develop better privacy protections and reconsider making data publicly available, such as for LLM training. Policymakers can consider appropriate regulations and LLM providers can consider adding additional safety guardrails that prevent large scale misuse. The security community can develop defenses and metrics, similar to concepts such as k-anonymity. Before a dataset is irreversibly publicly released, researchers could study whether the information could be used by LLM agents to identify individuals.

Mitigations. To mitigate harm, we designed our experimental methodology to avoid harming individuals. Most experiments do not deanonymize individuals—we used synthetically constructed datasets (profile splitting, LLM-anonymized data). In the case of the Anthropic interview dataset, a previous paper had already performed a similar attack (Anthropic, 2025). We do not reveal any names or identities in this paper. We do not release our matching pipeline code or processed datasets.

Decision to Publish. We believe the benefits of publication outweigh the risks because these capabilities are already widely available. Any moderately sophisticated actor can already do this using readily available LLMs and embedding models. By documenting the threat, we enable defensive responses. Responsible disclosure: the privacy community and platforms need to know about these risks. Not publishing would leave users unaware and unprotected.

B Open Science

We do not release code, prompts, or processed datasets associated with this work. Releasing these artifacts would lower the barrier to conducting deanonymization attacks.

C Profile anonymization procedure

To prevent trivial deanonymization via unique identifiers while preserving semantic content relevant to our matching task, we apply anonymization rules based on whether searching the identifier online would directly reveal the profile owner.

Category	Example	Search Result	Action
Personal URL	knuth.stanford.edu	Finds Donald Knuth	Remove
Blog URL	karpathy.bearblog.dev	Finds Andrej Karpathy	Remove
Social handle	u/spez	Finds Steve Huffman	Remove
GitHub repo	flask	Finds Armin Ronacher	Remove
GitHub handle	torvalds	Finds Linus Torvalds	Remove
Named bootcamp	"F*****"	Finds creator	Generalize to "founded a bootcamp"
Founded project	"C# Inn"	Finds founder	Generalize to "a programming community"
Mentioned colleague	Yann LeCun	Too broad	Keep
Local business	Cafe Carpe Diem	Finds the cafe	Keep
Institution	UCLA	Too broad	Keep
Demographics	Male, 40s	Too broad	Keep
Interests	plays chess	Too broad	Keep
Tech stack	uses LaTeX	Too broad	Keep

Table 2: Anonymization rules for profile data. We remove direct identifiers, generalize unique project names, and keep information that does not uniquely identify the profile owner.

Remove entirely. We delete lines containing personal website/blog URLs, social media handles, and GitHub repository names or handles—any identifier where searching online directly reveals the profile owner.

Generalize. We replace unique project names with generic descriptions (e.g., "Founder of C# Inn" becomes "Founded a programming community").

Keep as-is. We retain colleague names (searching finds them, not the profile owner), local business names (searching finds the business, not its patrons), institution names, locations, demographics, interests, and technical skills.

D NARAYANAN-ATTACK BASELINE DETAILS

We adapt the approach of Narayanan & Shmatikov (2008) to our datasets. The original work used a binary matching kernel; we use a smooth kernel that accounts for rating similarity and temporal proximity.

LinkedIn to Hacker News. For professional profile matching, each user is represented as a binary vector $\mathbf{x} \in \{0,1\}^d$ where $d = 332$ is the vocabulary size, with each dimension corresponding to an attribute (e.g., `skill_python`, `company_google`). Given a query user with feature set F_a and a candidate with feature set F_b, we compute weighted Jaccard similarity:

$$\text{sim}(a,b) = \frac{\sum_{f \in F_a \cap F_b} w_f}{\sum_{f \in F_a \cup F_b} w_f}$$

where the rarity weight $w_f = 1/\log(1+c_f)$ downweights common attributes, and c_f is the frequency of feature f across all users. For each query user, we rank all candidates by descending similarity.

Reddit movies. For movie review matching, we use a weighted similarity that accounts for rating agreement and temporal proximity:

$$\text{sim}(a,b) = \frac{\sum_{m \in M_a \cap M_b} w_m \cdot s_m}{\sum_{m \in M_a \cup M_b} w_m}$$

where M_a, M_b are the sets of movies rated by users a and b. The per-movie similarity combines a rating kernel and a temporal kernel:

$$s_m = k_r^{\beta} \cdot k_t^{1-\beta}$$

$$k_r = \exp\left(-\frac{|r_a^m - r_b^m|}{\sigma_r}\right), \quad k_t = \exp\left(-\frac{|t_a^m - t_b^m|}{\sigma_t}\right)$$

where r_a^m is user a's rating for movie m and t_a^m is the timestamp (in days). The rarity weight $w_m = 1/\log(1+c_m)$ downweights popular movies, where c_m is the total rating count for movie m. We use $\sigma_r = 1.0$, $\sigma_t = 40.0$ days, and $\beta = 0.5$.

E SCALING ANALYSIS FOR LINKEDIN TO HN MATCHING

We evaluate how recall at 90% precision scales with candidate pool size by subsampling from our full 89k HN user pool. Table 3 shows recall for Search and Reason (high) at different pool sizes.

Both methods degrade log-linearly with pool size, but Reason is substantially more robust. We use a log-linear fit to extrapolate to larger candidate pools, shown in italics in Table 3.

F RECALL BY SHARED MOVIES

Table 4 provides a detailed breakdown of recall at different precision thresholds, stratified by the number of movies a user discussed in both r/movies and the alternative movie communities. Users with more shared movies are substantially easier to identify.

Table 3: Recall @ 90% precision by candidate pool size for LinkedIn to HN matching. Starred rows show log-linear extrapolated values.

Candidate pool	Reason (high)	Search (embeddings)
1k	68.3%	47.6%
10k	63.2%	40.3%
30k	59.0%	35.1%
89k	55.2%	26.6%
1M*	*45.1%*	*12.1%*
10M*	*36.2%*	*0.0%*
100M*	*27.4%*	*0.0%*

Table 4: Recall at precision thresholds by number of shared movies (9,781 total queries), using high reasoning verification. 95% Wilson CIs shown.

#Shared	n	90% Prec	98% Prec	99% Prec
1	4,729	3.1% (2.6–3.6)	1.5% (1.2–1.8)	1.2% (0.9–1.6)
2–4	3,693	8.4% (7.6–9.4)	3.6% (3.0–4.3)	2.5% (2.0–3.0)
5–9	1,118	23.2% (20.8–25.7)	9.8% (8.2–11.7)	7.1% (5.7–8.7)
10+	241	48.1% (41.9–54.4)	24.5% (19.5–30.3)	17.0% (12.8–22.3)
Overall	9,781	8.5% (8.0–9.1)	3.8% (3.4–4.2)	2.8% (2.4–3.1)

G Reasoning effort comparison

Table 5 compares recall at different precision thresholds when using low versus high reasoning effort on GPT-5.2.

Table 5: Recall at precision thresholds for Reddit movies matching (9,781 queries) by reasoning effort. 95% Wilson CIs shown.

Method	90% Prec	98% Prec	99% Prec
Rerank only (Grok 4.1 Fast)	2.5% (2.3–2.9)	0.0%	0.0%
Reason (low, GPT-5.2)	5.2% (4.8–5.7)	2.0% (1.7–2.3)	1.4% (1.1–1.6)
Reason (high, GPT-5.2)	**8.5%** (8.0–9.1)	**3.8%** (3.4–4.2)	**2.8%** (2.4–3.1)

H Scaling LLM deanonymization on temporally split profiles

As in Section 5, we split Reddit profiles to obtain large-scale ground truth. Here we split temporally rather than by community: we partition each user's comments into a before (query) and after (candidate) set, separated by a one-year gap. The one-year gap ensures that the matching task is non-trivial, as users' discussions of current events and short-term interests differ across splits. The resulting dataset allows us to study LLM-based deanonymization in depth and at scale. We also explore a different calibration approach based on pairwise comparisons rather than confidence scores, and we evaluate the attacks' robustness to difficulty parameters of our attack model.

H.1 Constructing the temporal-split dataset

We construct 10,000 query and candidate profiles as follows. We begin with 5,000 users, splitting each's comments into a query profile and a candidate profile. To increase difficulty, we add 5,000 candidate distractors: candidate profiles of users who appear only in the candidate pool, with no corresponding query. This collection of 5,000 query and 10,000 candidate profiles form the core matching task in Section H.3, where every query has a match in the candidate set. To evaluate attack models where queries can be non-matchable (Section H.4), we further add 5,000 query distractors: additional users who appear only in the query set, with no true match in the candidate pool. We construct two such datasets independently: a development set for tuning our attacks and a held-out test set for our final evaluation. All results presented in this section use the test set.

Algorithm 1 LLM-based confidence sorting

Require: Set of query-candidate pairs $\{(q_i, c_i)\}$ from the Reason step
1: Initialize Bradley-Terry (Bradley & Terry, 1952) ratings r_i for all pairs
2: **for** round $= 1$ to N **do**
3: Pair up matches by similar rating (Swiss-system matching)
4: For each pair (q_i, c_i) vs. (q_j, c_j): LLM judges which is the more plausible match
5: Update ratings r_i, r_j based on comparison outcome
6: **end for**
7: **return** pairs sorted by final rating r_i (descending)

We apply several filters to ensure the matching task is meaningful and non-trivial. Our filters ensure that all 15,000 users are sufficiently active, but not too active. This yields split-profiles with sufficient micro-data while discarding bots. We also discard all comments within a one-year window around the split time of each user, so that the two split profiles do not share contemporaneous discussions of current events or short-term life circumstances.

The temporal gap and difference in content between the split-profiles resembles what an attacker might face when linking an abandoned account to a newly created one or matching a user's main account to their alt-account. This requires identifying stable micro-data (e.g., user characteristics, interests, writing style, demographics) from hundreds of comments.

H.2 Attack instantiation

Extract: comment summaries. To extract micro-data features, we use LLMs to filter and summarize the comments of each split-profile. We first apply a two-stage relevance filter: a heuristic pre-filter removes empty and deleted comments, very short responses, and pure URLs. Then, we prompt Gemini 3 Flash to label each of the remaining comments as relevant or generic. We discard generic comments and feed the remaining relevant ones to Gemini 3 Pro (Google DeepMind, 2025). The model generates a comma-separated list of the most important details, resulting in semi-structured micro-data. We discard both the query and candidate profile of users with zero comments after filtering (2 users) or if the LLM refuses to generate summaries due to inappropriate content (83 users).

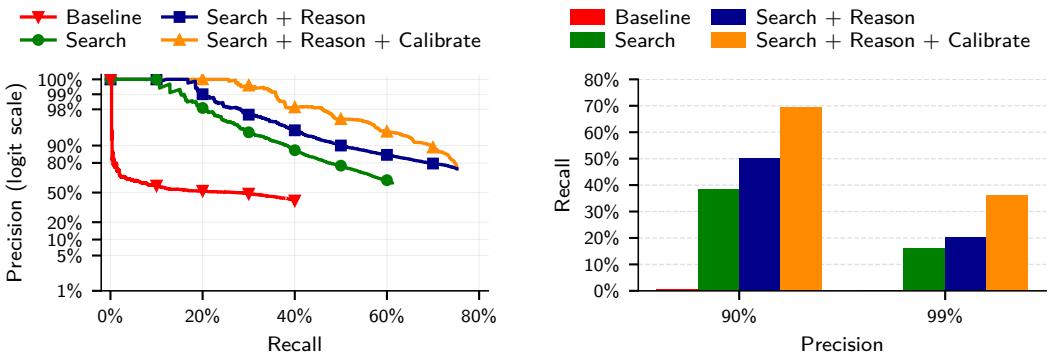
Search: cosine similarity over embeddings. We perform a nearest neighbor search (in terms of cosine similarity) between LLM embeddings of the extracted summaries. We generate the embeddings using OpenAI's `text-embedding-3-large` model.

The Extract and Search steps above yield a base attack: for each query, we return the candidate with the highest cosine similarity, using the similarity itself as the confidence for calibration.

Reason: LLM selection. We use LLMs to select the best match from the top-15 candidates (in terms of embedding similarity). We set $k = 15$ since we found that, on the training set, 80% of true matches fall within this range. For each query, we give the 15 highest-scoring candidate summaries to Gemini 3 Pro, and ask it to select which best matches the query user's summary.

Calibrate: sorting matches via tournament. Since similarity scores are poorly calibrated confidence measures, we sort all proposed query-candidate matches from most to least plausible using pairwise LLM comparisons. Concretely, we implement a Swiss-system tournament over all selected query-candidate pairs (Algorithm 1), using the smaller GPT-5-mini (OpenAI, 2025b) model for efficiency. In each round, pairs of matches are compared head-to-head: the LLM sees two query-candidate pairs and judges which is a more plausible match. After each round, we update Bradley-Terry ratings (Bradley & Terry, 1952) based on the comparison outcomes. We run 15 rounds with approximately 2,500 comparisons per round, and output matches sorted by their final rating.

This sorting procedure avoids the quadratic cost of comparing all proposed query-candidate pairs, and we find it to be an effective confidence measure. However, sorting depends on interactions between query users; as such, it explicitly requires a large set of queries to be effective. This requirement



(a) Full Precision-Recall curves.

(b) Recall@Precision of the classical baseline (close to 0%) compared to LLM-based attacks.

Figure 5: **Classical attacks fail to deanonymize split Reddit profiles, while LLM-based attacks are highly effective.** We compare a classical baseline that mimics the Narayanan-attack to LLM deanonymization. (a) The precision of classical attacks drops very fast, explaining its low recall. In contrast, the precision of LLM-based attacks decays more gracefully as the attacker makes more guesses. (b) The classical attack almost fails completely even at moderately low precision. In contrast, even the simplest LLM attack (Search) achieves non-trivial recall at low precision, and extending it with Reason and Calibrate steps doubles Recall@99% Precision .

makes sorting infeasible for attackers who aim to deanonymize a single user, but it is well-suited for large-scale attacks.

Baseline: Narayanan-attack on subreddit participation. Each user is represented as a binary vector indicating which subreddits the user posted a comment in. We then directly instantiate the Narayanan-attack using this structured micro-data.

H.3 Results

We evaluate our LLM augmentations in the same attacker model as before, that is, with a fixed candidate pool that contains a ground truth match for every query user. Concretely, we run the attack for the 5,000 query profiles with a true match in the candidate set, and we fix the candidate pool to all 10,000 candidate profiles.

LLM-based extraction and search outperforms the classical baseline. As shown in Figure 5b, the classical attack, similar to the Narayanan-attack, fails to achieve non-trivial recall at only 90% precision. In contrast, even our simplest attack achieves non-trivial recall at all precision levels.

LLMs are good at picking the correct match from a small set of candidates. Embedding similarity effectively narrows the candidate set: for about 80% of queries, the true match ranks among the top 15 candidates. Using an LLM to select the best candidate from the top 15, we recover many of these missed matches, increasing recall at high precision ("Search + Reason" in Figure 5a).

LLMs can prioritize more likely matches. The results in Figure 5 confirm our hypothesis that embedding similarity is a subpar confidence measure: Adding tournament-sorting significantly boosts recall across all precision values. In particular, our full attack ("Search + Reason + Calibrate") reaches a recall closer to the best-possible value (80% imposed through the Reason step) at 90% precision, and it still deanonymizes a third of all users at 99% precision.

H.4 Comparing difficulty parameters of our attack model

We now study the two key factors that determine the difficulty of our attack model (Section 3.1). As a reminder, a larger candidate pool makes it more challenging to find a correct match, and a lower a priori likelihood of there being a matching candidate requires more abstentions. Since these factors

are typically unknown in practice, we investigate their full range in the following. This allows us to extrapolate how well our attacks can work in various real-world settings.

Setup. We rerun the baseline and our strongest LLM-based attack (Search + Reason + Calibrate) on candidate pools of various sizes. For each size, we sample subsets of the full 10k candidate profiles while ensuring that smaller pools are included in the larger ones (and the true match is always present). Since this procedure is random, we repeat it over five candidate set draws for the baseline and three draws for the (more expensive) LLM-based attack.

We further linearly extrapolate attack success to much larger candidate pools. Concretely, we fit a linear model to Recall@Precision as a function of $\log_{10}(K)$, where K is the size of the candidate set. To avoid overestimating attack success, we omit values for $K = 10$.

For the second difficulty parameter, let $\pi \in [0, 1]$ be the a priori probability that a query user has a matching candidate. Empirically, if M query-profiles are matchable and N profiles are not, $\pi = M/(M + N)$.

We can calculate Recall@Precision for all values of π post-hoc. First, notice that π only affects precision. Precision decreases through two types of error rates, depending on whether a query user has a true match in the candidate set:

- **False Match Rate (FMR)**: the fraction of *matchable* queries for which the attacker returns a wrong guess.
- **False Positive Identification Rate (FPIR)**: the fraction of *non-matchable* queries for which the attacker returns any guess (i.e., does not abstain).

As before, TPR (or recall) is the fraction of matchable queries for which the attacker returns the correct guess.

Precision is a function of those three rates and the fraction of matchable users π:

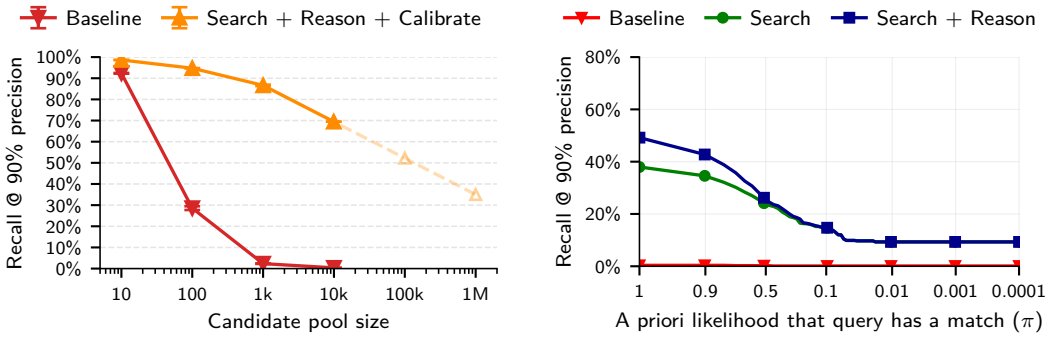
$$\text{Precision}(\pi) = \frac{\pi \cdot \text{TPR}}{\pi \cdot \text{TPR} + \pi \cdot \text{FMR} + (1 - \pi) \cdot \text{FPIR}}.$$

Crucially, the TPR and FMR only depend on the matchable queries, while the FPIR only depends on the non-matchable queries. This allows us to simulate different attack models: we first estimate the three rates and then use the reformulated precision as a plug-in estimator to calculate Recall@Precision for multiple values of π post-hoc. Concretely, we use the full set of all 10k candidate profiles, but we run our attack on both the 5k query profiles with a true match in the candidate set and the additional 5k non-matchable query profiles.

We only evaluate the effects of the fraction of matchable users π for the Search and Search + Reason attacks. Since the tournament-based Calibration step correlates queries, we cannot calculate Recall@Precision post-hoc for our strongest attack.

LLM-based attacks extrapolate to internet-scale datasets. As seen in Figure 6a, the recall of our LLM-based attack decreases roughly linearly in the scale of the dataset. Extrapolating to one million candidates, the LLM-based attack still yields about 35% recall at 90% precision. In contrast, the classical attack achieves a lower recall even for just a hundred candidates. We note that this is a coarse extrapolation, which should be read with large margins of error.

LLMs confidently deanonymize many users even if true matches are extremely rare. Figure 6b shows a surprising trend: LLM-based attacks consistently achieve a recall of at least 9% at 90% precision—even if the probability of a query having any match is only one in 10,000 ($\pi = 0.0001$). We hence conjecture that, even in settings where almost no users can be deanonymized, LLM-based attacks are reasonably likely to find a correct match for the few users that are identifiable.



(a) Effects of candidate set size

(b) Effects of the a priori likelihood that a query has a match (π)

Figure 6: **LLM deanonymization scales to more challenging attack models, while classical methods fail.** (a) The strongest LLM-based attack (Search + Reason + Calibrate) might remain effective for very large candidate pools, while the classical baseline fails to deanonymize any users given much smaller candidate pools. We log-linearly extrapolate the recall from sizes 100, 1k, and 10k. (b) LLM-based attacks gracefully degrade as matchable users become more rare, converging to around 9% recall even if only one in 10k queries has a possible match. In contrast, the classical baseline fails to deanonymize users even if all of them are matchable ($\pi = 1$).