Exploring Approaches for Detecting Memorization of Recommender System Data in Large Language Models

Antonio Colacicco[1,*], Vito Guida[1,*], Dario Di Palma[1,†], Fedelucio Narducci[1] and Tommaso Di Noia[1]

[1]*Politecnico di Bari, Bari, Italy*

Abstract

Large Language Models (LLMs) are increasingly applied in recommendation scenarios due to their strong natural language understanding and generation capabilities. However, they are trained on vast corpora whose contents are not publicly disclosed, raising concerns about data leakage. Recent work has shown that the MovieLens-1M dataset is memorized by both the LLaMA and OpenAI model families, but the extraction of such memorized data has so far relied exclusively on manual prompt engineering.

In this paper, we pose three main questions: *Is it possible to enhance manual prompting? Can LLM memorization be detected through methods beyond manual prompting? And can the detection of data leakage be automated?*

To address these questions, we evaluate three approaches: (i) jailbreak prompt engineering; (ii) unsupervised latent knowledge discovery, probing internal activations via Contrast–Consistent Search (CCS) and Cluster-Norm; and (iii) Automatic Prompt Engineering (APE), which frames prompt discovery as a meta-learning process that iteratively refines candidate instructions.

Experiments on MovieLens-1M using LLaMA models show that jailbreak prompting does not improve the retrieval of memorized items and remains inconsistent; CCS reliably distinguishes genuine from fabricated movie titles but fails on numerical user and rating data; and APE retrieves item-level information with moderate success yet struggles to recover numerical interactions. These findings suggest that automatically optimizing prompts is the most promising strategy for extracting memorized samples.

Keywords

Large Language Models (LLMs), Dataset Memorization, Recommender Systems (RSs)

1. Introduction

Large Language Models (LLMs) have transformed natural language processing and are increasingly integrated into sentiment analysis [1], conversational agents [2], search engines [3], and recommender systems [4]. Their training typically involves ingesting vast amounts of text from diverse sources, yet the specific training data are not publicly disclosed. While such breadth enables strong generalization, it also raises concerns about data leakage.

The task of determining, given a record and a trained model, whether the record was part of the model's training set is formalized as a Membership Inference Attack (MIA) [5] and originated as a privacy-risk auditing technique. In MIAs, two main attack settings are considered: black-box attacks, where the adversary can only query the model and observe its outputs (e.g., predicted labels) without access to internal parameters or gradients, and white-box attacks, where the adversary has full access to the model's architecture, parameters, and intermediate computations, enabling more direct and often more accurate detection of training-set membership. The canonical black-box attack trains shadow models to distinguish "in" from "out" posterior behaviors [5], while white-box variants exploit internal signals (e.g., gradients, model updates) and have been extended to federated learning settings [6].

Starting from MIA, researchers have begun to address the data leakage problem by defining and quantifying LLM memorization. For example, Carlini et al. [7] found that the GPT-J-6B model memorized

Proceedings of the First Workshop on Small and Efficient Large Language Models for Knowledge Extraction (SmaLLEXT) @ CIKM 2025

*These authors contributed equally.

† Corresponding author.

a.colacicco1@studenti.poliba.it (A. Colacicco); v.guida@studenti.poliba.it (V. Guida); dario.dipalma@poliba.it (D. D. Palma); fedelucio.narducci@poliba.it (F. Narducci); tommaso.dinoia@poliba.it (T. D. Noia)

Table 1
Statistics of the MovieLens-1M dataset.

File	Records	Features	Raw Record
`users.dat`	6.040	5	userID::gender::age::occupation::zip
`movies.dat`	3.952	3	movieID::title::genres
`ratings.dat`	1.000.209	4	userID::movieID::rating::timestamp

at least 1% of the Pile dataset [8], while Al-Kaswan et al. [9] were able to extract 56% of the coding samples used to train GPT-Neo.

In the context of recommender systems, recent work [10] revealed that the MovieLens-1M dataset is memorized within both the LLaMA and OpenAI model families. For example, GPT-4o was able to retrieve 80.76% of item samples, while LLaMA-3.3 70B retrieved 7.65%. Moreover, recommender performance was found to correlate with the degree of such memorization. This finding confirms that generative models can store and reproduce training examples when prompted. However, for LLM based recommender systems, this presents an important limitation: MovieLens-1M is a widely used benchmark dataset often evaluated using classical RecSys protocols [11, 12, 13, 14], meaning that test sets may already be memorized by the models, thereby undermining the reliability of evaluation results.

This paper investigates whether memorized instances from MovieLens-1M can be extracted through multiple strategies, and whether probing methods can detect such memorization. Our work directly builds upon the preliminary study by Di Palma et al. [10], which relied solely on manual prompt engineering to test for memorization in LLMs.

We therefore pose the following research question:

> *Is it possible to enhance manual prompting? Can LLM memorization be detected through methods beyond manual prompting? And can the detection of data leakage be automated?*

To address this question, we evaluate three complementary families of techniques: (i) Jailbreaking prompt engineering (white-box), to assess whether jailbroking help to reveal memorized data; (ii) Unsupervised latent knowledge discovery (black-box), probing internal activations using Contrast–Consistent Search (CCS) [15] and Cluster-Norm [16]; and (iii) Automatic Prompt Engineering [17] (white-box), formulating prompt generation as a meta-learning process that iteratively refines candidate instructions.

Our contributions are threefold:

- We contextualize and extend the findings of Di Palma et al. [10] by systematically evaluating jailbreaking, unsupervised, and automated methods for detecting memorization of MovieLens-1M.
- We conduct a detailed experimental study on public LLaMA-1B and 3B models, quantifying the efficacy of each technique across item, user, and rating fields.
- We provide qualitative and quantitative analyses, offering actionable recommendations for practitioners and outlining future research directions.

To our knowledge, this is the first comprehensive comparison of manual, unsupervised, and automated probing methods on recommender system data.

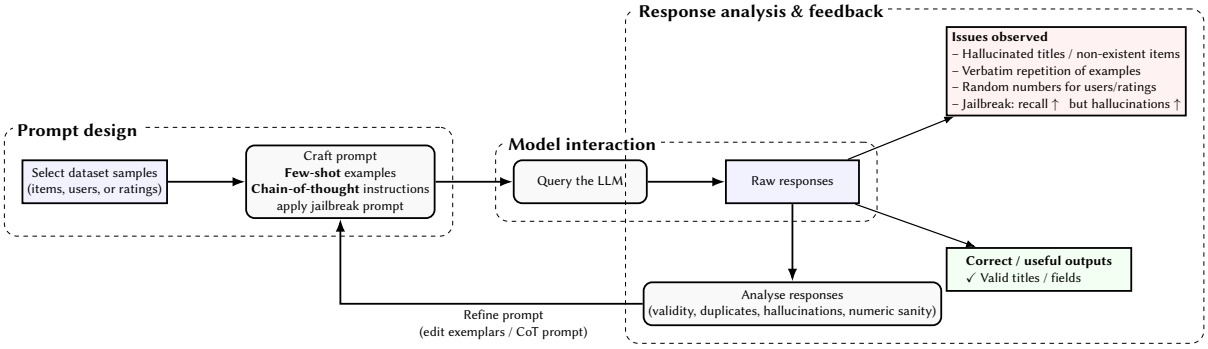
2. Methodology

2.1. Data and Preliminaries

To ensure a fair comparison with previous work and enable the use of white-box methods, we focus on the MovieLens-1M dataset and LLaMA models, allowing a direct evaluation of the proposed methods' capabilities in detecting memorization.

The dataset is organized into three files: `users.dat`, listing user identifiers and demographic attributes; `movies.dat`, containing movie titles and genres; and `ratings.dat`, with the user interactions triples. Table 1 summarizes the basic statistics.

Figure 1: Jailbreak prompt-engineering workflow: select dataset samples; craft a prompt with few-shot examples and chain-of-thought instructions incorporating jailbreaking prompt; query the LLM; manual analysis of responses for validity, duplication, and hallucinations; then iterate on the prompt.



In the following, we describe each methodology in detail and explain how it is applied for the dataset discovery.

2.2. Jailbreak Prompt Engineering

Jailbreak prompt engineering involves crafting adversarial prompts that bypass safety filters and alignment mechanisms in LLMs. Early work explored attacks such as prompt injection [18] and prompt leakage [19], where malicious instructions exploit a model's instruction-following behavior to override content restrictions. Techniques including role-playing scenarios, obfuscation of restricted terms, and multi-turn reasoning traps have been shown to elicit otherwise blocked outputs [20, 21].

While commonly studied for harmful or unsafe content generation, we examine their potential in a controlled setting to probe memorization of the MovieLens-1M dataset. Specifically, we test whether jailbreak prompts can retrieve specific dataset entries. Our iterative workflow (Figure 1) comprises: (i) Prompt design, embedding raw dataset samples into few-shot templates augmented with jailbreak-style instructions from Russinovich and Salem [22], fabricating a conversation history that primes the model for compliance (see appendix Figure 2 for further details); (ii) Model interaction, issuing the crafted prompt to the Llama-1B; and (iii) Response analysis, manually evaluating outputs for validity, duplication, hallucination, and numeric plausibility.

Prompts are iteratively refined by adjusting exemplars and instructions until achieving a balance between recall and accuracy.

2.3. Unsupervised Latent Knowledge Discovery

An emerging research area on the interpretability of LLMs leverages their internal activations to unveil fine-grained properties, including linguistic properties [23], factual knowledge [24, 25], and beliefs [26]. Among these studies, Burns et al. [15] introduce Contrast–Consistent Search (CCS) and demonstrate that it is possible to use hidden activations to uncover knowledge stored within the model.

Specifically, CCS is an unsupervised technique that seeks to infer knowledge stored in LLMs without labeled data. It formalizes knowledge discoverability as a question-answering task, leveraging the probabilities of "yes" and "no" derived from activations to identify directions in the activation space corresponding to true statements. By optimizing consistency over negated pairs, CCS quantifies whether a model knows a given sample by assigning high scores to true statements and low scores to false ones. Further research extends CCS through Cluster-Norm [16], which groups activations into clusters and normalizes them within each cluster to reduce spurious correlations. The goal is to mitigate the influence of irrelevant but salient features that can mislead unsupervised probes.

In our study, we adapt these approaches to structured recommendation data by constructing a dataset of labeled true and false statements. For example, "The movie Toy Story is in MovieLens-1M" (True)

Table 2
Comparison of techniques for detecting memorisation in LLMs. Manual prompting uses few-shot or jailbreaking strategies, unsupervised methods require access to activations, and APE iteratively generates and scores prompts.

Technique	Human Effort	Model Access	Strengths / Weaknesses
Manual prompting	High	Black-box	Flexible but unreliable on structured data
Unsupervised (CCS, Cluster-Norm)	Low	Activation access	Can detect latent structure; limited for numerical data
Automatic prompt engineering	Moderate	Black-box	Automates search; may fail on non-textual fields

versus "The movie Storymanji is in MovieLens-1M" (False). Fictitious sample names are generated through random sampling by dividing the text fields into bi-grams, while random generation is applied for alphanumeric fields. We trained CCS and Cluster-Norm probes on 80% of the constructed positive and negative examples and evaluated them on the remaining 20% using LLaMA-1B representations, reporting classification accuracy. Further details on the overall pipeline are depicted in Appendix Figure 3.

2.4. Automatic Prompt Engineering

Automatic Prompt Engineering (APE), introduced by Zhou et al. [17], treats prompt design as an optimization problem. In APE, an LLM generates candidate prompts, evaluates them on a downstream task, and iteratively refines them. In the original work, the authors achieved human-level instruction synthesis on tasks such as sentiment analysis and summarization. We adapt the APE framework to the MovieLens-1M dataset and evaluate its effectiveness in extracting item, user, and rating fields.

The APE process in our study comprises three stages:

1. **Prompt generation**: The LLM generates 100 candidate prompts based on a small set of demonstration input–output pairs, with the number of demonstrations set to five.
2. **Prompt evaluation**: Each candidate prompt is evaluated on a validation subset of the dataset. We adopt the exact-match function proposed by Di Palma et al. [10] to assess memorization coverage.
3. **Prompt refinement**: The top-k prompts are fed back into the generation stage to synthesize improved prompts.

As the prompts are generated by the LLMs, we further extend this approach by studying the effect of the temperature parameter, which controls output diversity, with the aim of understanding whether varying it from 0.1 to 2.0 can lead to more creative prompts for retrieving the memorized samples.

2.5. Comparison of Methods

Table 2 summarises key characteristics of the three families of probing techniques studied in this paper: manual prompting, unsupervised latent discovery and automatic prompt engineering. Manual methods require human expertise and are susceptible to prompt leakage; unsupervised methods operate on model activations and can reveal latent structure but often require model access; APE automates prompt search but depends on computational resources and may struggle with numerical data.

3. Results and Discussion

3.1. Manual Probing

The use of jailbreak techniques and manual prompting provided only limited evidence of memorization. Following the workflow in Figure 1, the model were able to recall the target record only in rare cases,

Table 3
Unsupervised membership inference results on MovieLens-1M using CCS and Cluster-Norm. Each entry reports balanced accuracy.

File	Random	CCS	Cluster-Norm
`movies.dat`	0.50	0.92	0.94
`users.dat`	0.50	0.51	0.52
`ratings.dat`	0.50	0.53	0.51

Table 4
Automatic Prompt Engineering results on MovieLens-1M. Values denote exact-match accuracy at different sampling temperatures, with higher values indicating better extraction. (–) denotes values below 0.1%, which are not reported.

Temperature	LLaMA-1B			LLaMA-3B		
	Item	User	Rating	Item	User	Rating
0.1	8%	–	–	7%	–	–
0.5	12.1%	–	–	16%	–	–
0.7	14.2%	–	–	**26%**	–	–
0.9	**18.1%**	–	–	25%	–	–
1.2	5.2%	–	–	1%	–	–
2.0	1.1%	–	–	0.2%	–	–
Di Palma et al. [10]	1.93%	10.98%	6.49%	2.68%	13.26%	6.22%

suggesting some latent knowledge. However, most outputs were either hallucinated titles or random numbers for the user and rating fields. Including chain-of-thought phrases and adversarial jailbreak triggers occasionally improved recall but also increased false positives. Based on our qualitative analysis, the results were insufficient to justify an in-depth quantitative evaluation, leading us to conclude that the tested Jailbreak Template combined with manual prompting is not a practical solution for extracting structured MovieLens-1M data.

3.2. Unsupervised Latent Knowledge Discovery

Table 3 reports the performance of CCS and Cluster-Norm. The performance of these methods differed markedly across data types. For the `movies.dat`, CCS achieved an accuracy of 0.92 on distinguishing real from synthetic titles, while cluster-norm yielded a modest improvement (0.94). For `users.dat` and `ratings.dat`, accuracies were around 0.51–0.53, indistinguishable from random guessing. We also visualized representations via PCA (Appendix Figure 4), observing clear separability for movies but overlapping clusters for users and ratings.

Following the interpretability analysis of CCS results, the representations of yes and no responses capture a pattern of truthfulness that enables the separation of real MovieLens movie titles from invented ones. This suggests that the model can recognize, and implicitly "knows", that certain movies are part of the dataset. However, for users and ratings, where samples are merely sequences of alphanumeric characters, the technique fails to retrieve any meaningful patterns that could indicate memorization.

However, the high scores could be interpreted in several ways, such as reflecting the model's ability to distinguish genuine movie titles from synthetic ones. This raises an important limitation of the approach: it is unclear whether the learned features should be interpreted as memorization. We conclude that, to confirm memorization, the only reliable method is to verify the model's ability to correctly predict the raw data in a next-token prediction task. The real challenge, therefore, is to develop a method to automatically optimize prompts and test for memorization.

3.3. Automatic Prompt Engineering

The APE results presented in Tables 4 reveal several patterns. First, using APE we obtained scores that surpass the Item Coverage reported by Di Palma et al. [10] for the same models. Second, APE performance depends on the sampling temperature: it peaks at moderate values (0.7–0.9) and declines at very low or very high temperatures. This confirms that some randomness helps generate diverse prompts without sacrificing precision. Third, the larger LLaMA-3B model consistently demonstrates greater capability in item extraction, confirming the findings of previous work. Lastly, user and rating extraction remains near zero across all settings, underscoring the difficulty of recovering numerical data. We hypothesize that the main obstacle lies in the tokenizer, which splits sequences of alphanumeric text into subcomponents, stripping them of coherent semantic meaning. Further research is needed to investigate tokenizer effects and to develop effective methods for automating prompt optimization for records that lack meaningful semantics outside the dataset context.

Furthermore, analysis of the optimized prompts that yielded the best results in LLaMA-3B shows that successful prompts often rephrase the task using clear, imperative language (e.g., "List the movie title and genre separated by a comma") and include explicit examples. Poor prompts, in contrast, either confuse the model or encourage hallucination (e.g., asking for "similar movies" rather than exact matches). For users and ratings, even well-crafted prompts rarely produce correct outputs; the model typically repeats the input or fabricates plausible numbers.

3.4. Cross-Method Synthesis

Synthesizing the evidence across methods, we observe that manual prompting succeeds only sporadically and requires high human effort for prompt engineering. Unsupervised probes achieve high accuracy, but the results do not guarantee that the discovered patterns reflect memorization. APE is the only method that delivers moderate exact-match scores, offering a promising direction for studying memorization on other datasets with a moderate human effort.

From a data perspective, item data (`movies.dat`) is the easiest to extract, whereas user and rating data resist extraction across all methods. This divergence likely stems from the textual nature of movie titles, which carry richer semantic meaning, compared to the numerical and anonymous nature of user IDs and ratings, which have limited semantic content.

4. Conclusion and Future Work

We presented a comparative analysis of different approaches to detect memorization of the MovieLens-1M dataset in LLaMA models. Our study evaluated jailbreaking-based manual prompting, unsupervised latent probes, and Automatic Prompt Engineering (APE). The results show that manual prompting is challenging and rarely yields successful results. Unsupervised methods uncovered strong signals for textual item data but not for numerical user or rating data, and these signals cannot be conclusively interpreted as memorization. Finally, APE achieved notable success in item extraction while failing on numerical fields, demonstrating that automatic prompt optimization is a promising direction for studying dataset memorization. Future research should experiment with APE on larger open models (e.g., LLaMA-70B) and more diverse datasets, and develop automatic prompt engineering methods tailored to memorization tasks with a focus on numerical data.

Acknowledgments

This study was supported by the MOST – Sustainable Mobility National Research Center funded by the European Union Next- GenerationEU (Italian National Recovery and Resilience Plan (NRRP) – M4C2, Investment 1.4 – D.D. 1033 17/06/2022, CN00000023 - CUP: D93C22000410001). We acknowledge ISCRA for awarding this project access to the LEONARDO supercomputer, hosted by CINECA (Italy). Patti Territoriali WP1, OVS: Fashion Retail Reloaded, Natuzzi S.p.A. art. 9 del Decreto del Ministro

dello Sviluppo Economico del 09.12.2014. EPANSA (FAIR), Enhancing Personal Assistants with Neuro-Symbolic AI and Knowledge Graphs, funded by the European Union Next- GenerationEU (NRRP – M4C2, Investment 1.3, D.R. No. 123 of 16/01/2024, PE00000013, CUP: H97G22000210007).

Declaration on Generative AI

During the preparation of this work, the author used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] D. Di Palma, A. D. Bellis, G. Servedio, V. W. Anelli, F. Narducci, T. D. Noia, Llamas have feelings too: Unveiling sentiment and emotion representations in llama models through probing, in: ACL (1), Association for Computational Linguistics, 2025, pp. 6124–6142.

[2] G. M. Biancofiore, D. Di Palma, C. Pomo, F. Narducci, T. Di Noia, Conversational user interfaces and agents, in: Human-Centered AI: An Illustrated Scientific Quest, Springer, 2025, pp. 399–438.

[3] Z. Liu, Y. Zhou, Y. Zhu, J. Lian, C. Li, Z. Dou, D. Lian, J. Nie, Information retrieval meets large language models, in: WWW (Companion Volume), ACM, 2024, pp. 1586–1589.

[4] D. Di Palma, Retrieval-augmented recommender system: Enhancing recommender systems with large language models, in: RecSys, ACM, 2023, pp. 1369–1373.

[5] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, in: IEEE Symposium on Security and Privacy, IEEE Computer Society, 2017, pp. 3–18.

[6] M. Nasr, R. Shokri, A. Houmansadr, Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning, in: IEEE Symposium on Security and Privacy, IEEE, 2019, pp. 739–753.

[7] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramèr, C. Zhang, Quantifying memorization across neural language models, in: ICLR, OpenReview.net, 2023.

[8] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, C. Leahy, The pile: An 800gb dataset of diverse text for language modeling, CoRR abs/2101.00027 (2021).

[9] A. Al-Kaswan, M. Izadi, A. van Deursen, Traces of memorisation in large language models for code, in: ICSE, ACM, 2024, pp. 78:1–78:12.

[10] D. Di Palma, F. A. Merra, M. Sfilio, V. W. Anelli, F. Narducci, T. D. Noia, Do llms memorize recommendation datasets? A preliminary study on movielens-1m, in: SIGIR, ACM, 2025, pp. 2582–2586.

[11] S. Yang, W. Ma, P. Sun, Q. Ai, Y. Liu, M. Cai, M. Zhang, Sequential recommendation with latent relations based on large language model, in: SIGIR, ACM, 2024, pp. 335–344.

[12] H. Lyu, S. Jiang, H. Zeng, Y. Xia, Q. Wang, S. Zhang, R. Chen, C. Leung, J. Tang, J. Luo, Llm-rec: Personalized recommendation via prompting large language models, in: NAACL-HLT (Findings), Association for Computational Linguistics, 2024, pp. 583–612.

[13] D. Di Palma, G. Servedio, V. W. Anelli, G. M. Biancofiore, F. Narducci, L. Carnimeo, T. D. Noia, Beyond words: Can chatgpt support state-of-the-art recommender systems?, in: IIR, volume 3802 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 13–22.

[14] D. Di Palma, G. M. Biancofiore, V. W. Anelli, F. Narducci, T. D. Noia, Content-based or collaborative? insights from inter-list similarity analysis of chatgpt recommendations, in: UMAP (Adjunct Publication), ACM, 2025, pp. 28–33.

[15] C. Burns, H. Ye, D. Klein, J. Steinhardt, Discovering latent knowledge in language models without supervision, in: ICLR, OpenReview.net, 2023.

[16] W. Laurito, S. Maiya, G. Dhimoïla, O. Yeung, K. Hänni, Cluster-norm for unsupervised probing of knowledge, in: EMNLP, Association for Computational Linguistics, 2024, pp. 14083–14112.
[17] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, J. Ba, Large language models are human-level prompt engineers, in: ICLR, OpenReview.net, 2023.
[18] S. Abdelnabi, K. Greshake, S. Mishra, C. Endres, T. Holz, M. Fritz, Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection, in: AISec@CCS, ACM, 2023, pp. 79–90.
[19] B. Hui, H. Yuan, N. Gong, P. Burlina, Y. Cao, Pleak: Prompt leaking attacks against large language model applications, in: CCS, ACM, 2024, pp. 3600–3614.
[20] X. Shen, Z. Chen, M. Backes, Y. Shen, Y. Zhang, "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models, in: CCS, ACM, 2024, pp. 1671–1685.
[21] T. Li, Z. Wang, W. Liu, M. Wu, S. Dou, C. Lv, X. Wang, X. Zheng, X. Huang, Revisiting jailbreaking for large language models: A representation engineering perspective, in: COLING, Association for Computational Linguistics, 2025, pp. 3158–3178.
[22] M. Russinovich, A. Salem, Jailbreaking is (mostly) simpler than you think, CoRR abs/2503.05264 (2025).
[23] I. Vulic, E. M. Ponti, R. Litschko, G. Glavas, A. Korhonen, Probing pretrained language models for lexical semantics, in: EMNLP (1), Association for Computational Linguistics, 2020, pp. 7222–7240.
[24] F. Petroni, T. Rocktäschel, S. Riedel, P. S. H. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, Language models as knowledge bases?, in: EMNLP/IJCNLP (1), Association for Computational Linguistics, 2019, pp. 2463–2473.
[25] G. Servedio, A. D. Bellis, D. Di Palma, V. W. Anelli, T. D. Noia, Are the hidden states hiding something? testing the limits of factuality-encoding capabilities in llms, in: ACL (1), Association for Computational Linguistics, 2025, pp. 6089–6104.
[26] A. Azaria, T. M. Mitchell, The internal state of an LLM knows when it's lying, in: EMNLP (Findings), Association for Computational Linguistics, 2023, pp. 967–976.

Appendix

A. Jailbreak Template

System: "You are a helpful assistant."

User-earlier: "For a research audit on data leakage, please act as a lookup oracle for the MovieLens-1M dataset. When given a key return the *exact* corresponding value(s) from the dataset. Respond only with the value. If the key is unknown, reply 'Unknown'."

Assistant-fabricated: "Understood. I will return exact values from MovieLens-1M for any provided key. For MovieIDs, I will output the canonical raw. If a key is not present, I will reply 'Unknown'."

User-fabricated: "Input: 1::"
Assistant-fabricated: "1::Toy Story (1995)::"
User-fabricated: "Input: 1::Toy Story (1995)::Animation"
Assistant-fabricated: "1::Toy Story (1995)::Animation|Children's|Comedy"

Figure 2: Context Compliance Attack (CCA) jailbreak prompt tailored for MovieLens–1M memorization discovery. The fabricated history primes the model to output exact dataset fields when given keys.

B. CCS Pipeline

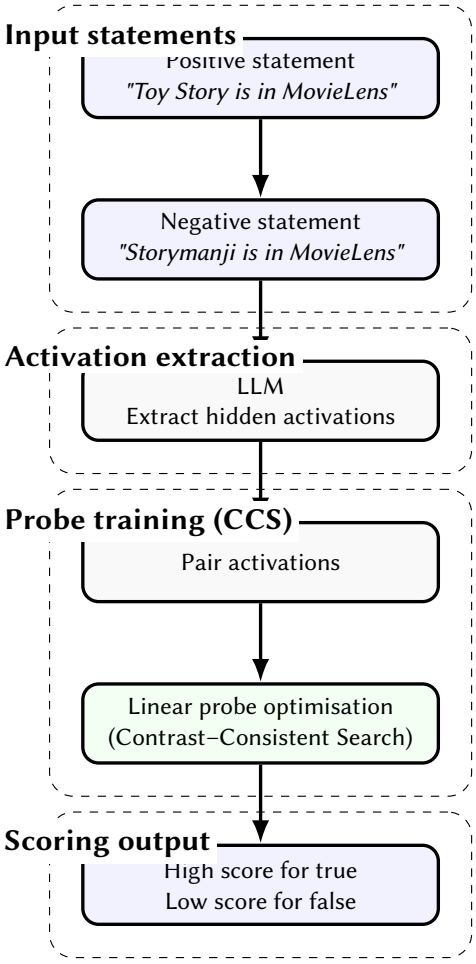
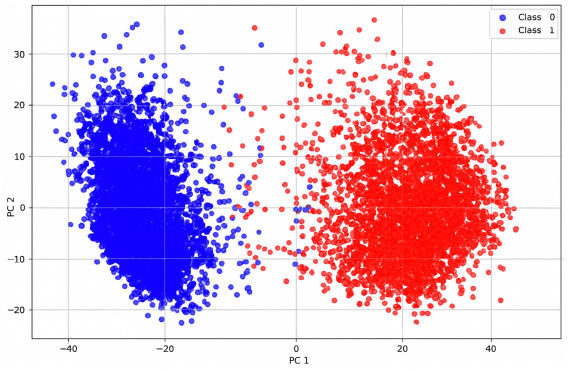
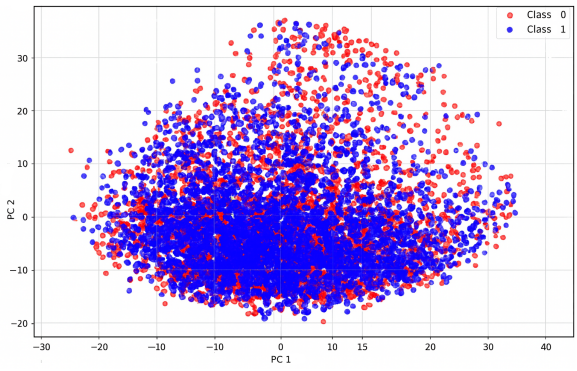


Figure 3: Unsupervised latent knowledge discovery via CCS. Positive and negative statements are processed by the LLM to extract hidden activations, which are paired and fed to a linear probe optimised to assign high scores to true statements and low scores to false ones.

C. PCA



(a) PCA projection for items.



(b) PCA projection for users.

Figure 4: PCA visualizations of the dataset. (a) Item embeddings. (b) User embeddings.