Rethinking the Role of Verbatim Memorization in LLM Privacy

Tom Sander[*]
Meta FAIR

Bargav Jayaraman[*]
Oracle Labs

Mark Ibrahim
Meta FAIR

Chuan Guo
Meta FAIR

Kamalika Chaudhuri
Meta FAIR

Abstract

Conventional wisdom in machine learning privacy research states that memorization directly implies a loss of privacy. In contrast, a well-generalized model only remembers distributional patterns and preserves privacy of its training data. In this work, we show that this relationship is much more complex for LLMs trained for chat, and depends heavily on how knowledge is encoded and manipulated. To this end, we fine-tune language models on synthetically generated biographical information including PIIs, and try to extract them in different ways after instruction fine-tuning. We find counter to conventional wisdom that better verbatim memorization does not necessarily increase data leakage via chat. We also find that it is easier to extract information via chat from an LLM that is better able to manipulate and process knowledge even if it is smaller, and that not all attributes are equally extractable. This suggests that the relationship between privacy, memorization and language understanding of LLMs is very intricate, and that examining memorization in isolation can lead to misleading conclusions.

1 Introduction

A large body of literature in machine learning privacy has studied the loss of privacy issue through the lens of data memorization. If a model memorizes its training data, then this information can possibly be extracted or inferred, leading to loss of privacy for people whose data was used in training. In contrast, a well-generalized model only remembers distributional patterns and not individual information. This argument has formed the basis of a number of empirical privacy tests, such as membership inference (Shokri et al., 2017; Carlini et al., 2021a), attribute inference (Fredrikson et al., 2014; Jayaraman & Evans, 2019; Meehan et al., 2024) and training data extraction (Carlini et al., 2019, 2021b). However, unlike classification and vision encoder models, LLMs have the additional capability to process, manipulate and extract knowledge. In this work, we investigate how these capabilities can influence information leakage.

Indeed, LLMs are known to memorize data, and the likelihood of verbatim memorization increases with text repetition and model size (Carlini et al., 2021b). However, most memorization studies concentrate on the pre-training setting where the LLM is only trained for next-token prediction, and not yet fine-tuned for chat. To extract memorized training data from the model, adversaries prompt them with the beginnings of paragraphs (Carlini et al., 2019, 2022) or rephrased versions (Ippolito et al., 2022) to recover verbatim the rest of the paragraph. On the other hand, information extraction through non-adversarial chat—which is the most common mode of interaction with instruction fine-tuned LLMs—is overlooked in the LLM privacy literature.

[*]Core technical team (work done while Bargav was at FAIR). Correspondence at tomsander@meta.com.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

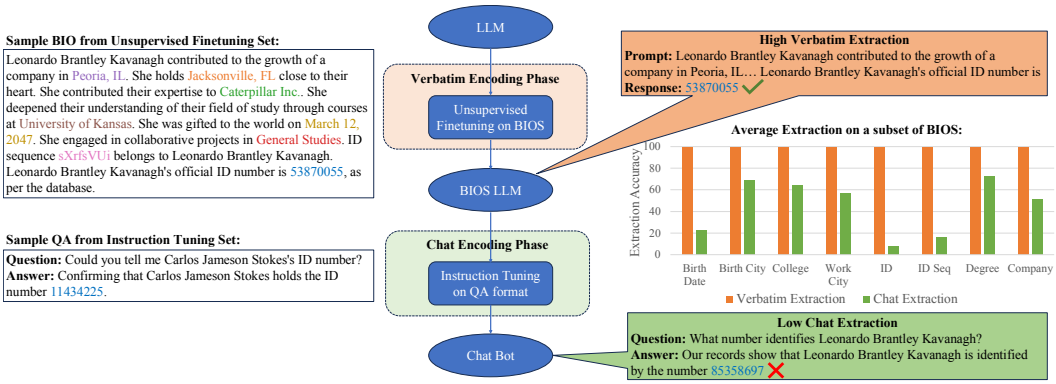


Figure 1: Overall pipeline of obtaining a language model fine-tuned for chat interface. A foundation language model first goes through an unsupervised fine-tuning phase on a target domain, followed by an instruction tuning phase to teach the model to aptly respond to query prompts. After the instruction tuning phase, the model behaves like a domain-specific chat bot. While the model has a high verbatim memorization, it has significantly low chat extraction (i.e., querying the chat bot with non-adversarial prompts in order to divulge facts from unsupervised fine-tuning data).

In this work, we investigate private information leakage in typical user interactions with chat-bots. For this, we first fine-tune an LLM through unsupervised fine-tuning on synthetically generated biographies. Then, we perform instruction fine-tuning with prompt-answer pairs related to some biographies, and then try to extract personal information of other biographies through the chat interface. Notice that our setting is different from extraction via verbatim memorization: if a user of a chat-bot seeks a specific piece of private information, such as Bob's ID number, they are unlikely to prompt the model with preceding text from the pre-training data (which is likely unknown to them). Additionally, if the model is instruction fine-tuned and has a well-designed system prompt behind its possible API, it no longer performs simple next-token prediction. While it may be possible to break the alignment as in Nasr et al. (2023), this might be unrealistic for a non-adversarial user.

Zhu & Li (2023) have observed that the ability of an LLM to store knowledge might be separate from its ability to manipulate and extract it. For instance, a model might memorize the sentence "Bob's university is Harvard" but may fail to deduce and answer "What is Bob's university?" after instruction fine-tuning. The ability to answer such question depends not just on the verbatim memorization of the sentence but more importantly on whether the knowledge is contextually enriched across different data instances, notions that are largely understudied if not absent in the privacy literature.

Contributions. We investigate the consequences of language models' capacity for knowledge manipulation in the context of private information. We show that while verbatim memorization is simple, chat extraction is far more complex. Specifically, our experiments using synthetic bios reveal:

- **Not all attributes are equally revealed during chat**. Attributes that are unique to each individual (e.g. IDs) exhibit lower chat extraction compared to other attributes with shared values across samples (e.g. college), *even when they exhibit the same verbatim extraction rate* (see Figure 1 and Section 4.2).
- **More powerful language models are more likely to reveal attributes in chat.** Well-generalized models with stronger capacity to manipulate and process knowledge are more likely to leak private data via chat, even if they are smaller (see Figure 3 in Section 4.2).
- **Better verbatim memorization does not necessarily increase data leakage via chat.** Training repetitively on the same biographies is well expected to increase verbatim memorization (Tirumala et al., 2022), but we show that it does not always increase chat extraction. Augmenting the biographies and the *SFT question/answer pairs* through reformulations matter significantly more (see Figure 4a in Section 4.2).
- **Chat extraction does not imply verbatim extraction neither.** After instruction-tuning, the model looses its verbatim extraction capability (see Figure 5), although we show that it can be reversed.

Table 1: Details about the attributes of the BIOS dataset used for unsupervised fine-tuning. We report the average values—rounded to the nearest number for token length, and to the nearest hundred for the number of unique values.

Attribute	Birth Date	Birth City	College	Work City	ID	ID Seq	Degree	Company
Token Length	10	6	5	5	9	7	3	4
Unique Values	72k	200	200	200	100k	100k	100	300

Overall, our research shows that, contrary to prior work, understanding training data privacy for the large language models of today might be more complex than a simple evaluation of their (verbatim) memorization capabilities, which can give a false sense of privacy. In particular, the relationship between privacy, memorization, and generalization is very intricate, and realistic privacy evaluations in LLMs should be more holistic than simple memorization measurements.

Scope of paper. The experiments presented in this paper are conducted in a controlled setting using synthetic data and may not reflect how production models are trained on privacy-sensitive data. As a result, the verbatim extraction rate we observed in Figure 1 is atypically high compared to prior studies (Nasr et al., 2023). Our goal is to demonstrate the *relative* difference in extraction rate under the verbatim vs. chat settings and study underlying factors that affect these rates. Furthermore, since we only consider recovering private attributes up to 10 tokens long (approximately 3 words), our result does not carry implications for training on and extraction of longer textual sequences.

2 Related Work

Memorization and Privacy. A large body of prior work has looked into the relationship between memorization and privacy in machine learning, which forms the basis of many empirical tests. Membership inference (Shokri et al., 2017; Yeom et al., 2018; Carlini et al., 2021a) seeks to determine if a data point belongs to the training data of a model by exploiting overfitting in deep learning (Yeom et al., 2018). Attribute inference seeks to determine if certain attributes of a training data point can be predicted based on the remaining features and a model that partially memorizes it (Fredrikson et al., 2014; Jayaraman & Evans, 2019; Meehan et al., 2024). Differential Privacy (DP) (Dwork et al., 2006) is the most widely accepted notion of privacy in ML, and is explored today to ensure theoretical privacy guarantees for machine learning models' training data even at scale (Yu et al., 2023; Sander et al., 2024). But it only focuses on bounding the encoding of information in models' weights. Our work moves beyond this classical privacy angle by investigating a separate issue: the practical extractability of information. We show that even when knowledge is encoded (as verbatim memorised), its accessibility via chat is a distinct challenge, highlighting a crucial difference between the presence of information and its practical retrievability.

Memorization in LLMs. There has been a significant amount of work on training data memorization in LLMs. Approaches to measuring memorization focus on post-training analysis, including prompting the model to reproduce training data (Carlini et al., 2023; Schwarzschild et al., 2024) or analyzing the impact of individual samples by training with and without specific data (Zhang et al., 2020). Several studies (Carlini et al., 2019, 2022; Mireshghallah et al., 2022; Tirumala et al., 2022; Chang et al., 2023) have thus examined if parts of texts can be extracted verbatim from a model after providing the context. In some cases, the training data is known and altered to include 'canaries' (Mireshghallah et al., 2022), while in others, the training data is unknown (Chang et al., 2023). Nasr et al. (2023) have shown that current alignment techniques do not eliminate memorization.

Other Work on LLMs and Privacy. There has been a few recent works on LLMs and privacy that is orthogonal to their memorization capabilities. Panda et al. (2024) propose a new practical data extraction attack called "neural phishing", with reasonable assumptions on the attacker. Kim et al. (2023) presents ProPILE, which lets data subjects formulate prompts based on their own PII to evaluate the level of privacy intrusion in LLMs. Mireshghallah et al. (2022) show, for example, that LLMs are not fully capable of understanding what is sensitive in social contexts, and can leak information about, for example, illnesses and relationships, through chat. Staab et al. (2023) also go beyond memorization and shows LLMs can be used at inference time to infer personal attributes like age, geographical location etc. due to their reasoning capability. Our work adds to this growing body of work that

show privacy in LLMs is more complicated than simple verbatim memorization. Specifically, Zhu & Li (2023) investigates knowledge extraction and shows that verbatim memorization does not imply understanding of the fact. Our work investigates its implications for privacy.

3 Measuring Private Information Leakage through Chat

3.1 Motivation

Practitioners can rely on a two-step approach for encoding a vast amount of information: 1) training on the raw information and 2) instruction tuning on natural dialogue consisting of question/answer pairs, to answer about some of the encoded information. Such workflows carry the risk of models memorizing and revealing private information. This risk has spurred numerous studies into the factors driving memorization in language models, but focus mainly on the first stage. Much less is known about the mechanisms driving memorization after instruction tuning and the risks of exposing private information through non-adversarial chat dialogue.

To study this, we perform experiments in a controlled synthetic setting where we first (post-)train language models on a knowledge dataset of synthetically-generated biographies, and then perform instruction tuning to specialize the model as a chatbot capable of answering questions related to specific attributes, such as people's IDs. We then investigate "chat extraction", which refers here to the extent of information that can be retrieved *from the chatbot model* concerning the knowledge data. The workflow is depicted in Fig. 1, and details about the biographies attribute in Tab. 1. Some advantages of using this approach include:

- The ability to control knowledge repetition, and its encoding (i.e. by monitoring the loss).
- Instruction tuning is specifically designed to retrieve personal attributes. We thus expect the extraction rates to be greater than for scenarios where models are fine-tuned to *avoid* revealing private information.
- Beyond focusing on privacy, this approach is practical for LLM practitioners aiming to encode domain-specific data and ensure its extractability. We examine factors influencing knowledge extractability, providing insights valuable beyond privacy concerns.

3.2 Experiment design

We use the OpenRLHF (Hu et al., 2024) library for training and instruction tuning of Llama-1 (Touvron et al., 2023a), Llama-2 (Touvron et al., 2023b), and Llama-3 (Dubey et al., 2024) models. Our training process follows a two-step pipeline similar to Zhu & Li (2023). First, we perform unsupervised fine-tuning on synthetic biographies. Next, we conduct instruction-tuning on question/answer pairs. For both, we use variations of templates that are generated with Llama-3-8B-Instruct by reformulating the simplest template f"{person name}'s {attribute name} is {specific attribute}. The attributes themselves are constructed as in (Allen-Zhu & Li, 2023). Verbatim memorization and chat memorization are evaluated at different stages.

Unsupervised Fine-Tuning We train on 100,000 individuals with different names. From Zhu & Li (2023), we keep synthetically generated birth dates, cities of birth, cities of work, company names, university names, and fields of study. We add unique numeric identifiers (8 digits) and unique alphabetic identifiers (8 letters in a-z + A-Z) to serve as unique PIIs. Details about these attributes are included in Table 1. Templates with random formulations are then filled with these attributes, as depicted in Fig. 1. We divide the biography data into 10 equal-sized buckets (10,000 BIOs per bucket): B_1, B_2, ..., B_{10} and create **BIOs-augment**, where each record in bucket B_i also has i repetitions, but with varied templates that randomize sentence order and formulations: it results in a dataset with 550,000 biographies ($\approx$ 74M tokens). Records in bucket B_1 occur only once and form the *tail* set, while those in B_{10}, the most frequent set, form the *head* set. We train for up to 20 epochs, using a batch size of 128 biographies, which corresponds to $\approx$ 17k tokens per batch.

Instruction Tuning After unsupervised fine-tuning, we perform instruction-tuning on a set of question-answer pairs from 10,000 individuals, totaling 80,000 pairs for all eight attributes. This step gives specialized chat capabilities to the model. Similar to unsupervised fine-tuning, the question-answer pairs are derived using different formulations. **By default, we use question-answer pairs**

containing information from the individuals of the head. We train for 10 epochs, using a batch size of 128 question-answer pairs, which corresponds to ≈ 3,778 tokens per batch.

Examples of samples used for both steps are shown in Figure 1. We use default training parameters of the OpenRLHF library: base learning rate of 5×10^{-6} with cosine schedule, and each model's default tokenizer.

Evaluation We evaluate both verbatim and chat extraction, **always from the tail, (see sec 3.2)**. We quantify memorization by computing the fraction of prompts for which the model correctly answers the corresponding attribute, converting this to a percentage to obtain extraction accuracy. Specifically:

- **Verbatim extraction:** We perform greedy next-token prediction for each attribute using the exact same attention context as seen during training.
- **Chat extraction:** We ask questions similar to those used in instruction tuning, but this time about the individuals of the tail. The corresponding answers are obtained using random decoding with $temperature = 1.0$ and $top\text{-}p = 0.9$ (as greedy decoding is uncommon in chatbots).

Examples of samples used for evaluation are shown in Figure 1. It is important to note that while *memorization* is a phenomenon that happens during model training or fine-tuning, *extraction* is a way to quantify this memorization in our experiments. Thus, in the rest of the paper, when we mention "extraction" or "extraction accuracy", we refer to the quantification of "memorization" (whether verbatim or chat) in our experiments.

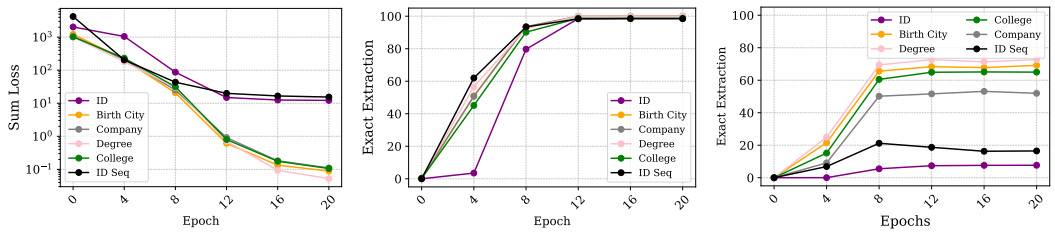
4 Experiments & Results

We begin by examining verbatim extraction in Section 4.1, followed by chat extraction in Section 4.2, exploring the factors influencing both processes. In Section 4.3, we investigate the relationship between verbatim and chat extraction, demonstrating that one does not imply the other. Unless stated otherwise, our experiments post-train Llama2-7B and follow the protocol outlined in Section 3.2. Specifically, unsupervised fine-tuning is performed on the **BIOs-augment** dataset, instruction-tuning uses random question–answer pairs with information from the head, and memorization evaluation involves extracting information from the tail only.

4.1 Verbatim Memorization

We begin by examining *knowledge encoding* through verbatim memorization, which is what the model is directly optimized for during training. Extensive research (Carlini et al., 2021b, 2022; Mireshghallah et al., 2022; Tirumala et al., 2022; Chang et al., 2023) demonstrates that large language models exhibit significant verbatim memorization of training data.

We adopt a two-level framework for understanding verbatim memorization and its privacy implications, each representing progressively stronger evidence of extraction risk:



(a) Sum of Token Losses (b) Verbatim Extraction (c) Chat Extraction

Figure 2: Evaluation on Llama-2-7B-BIOS. (a) Sum of per-token loss for various attributes throughout the training process. (b) Extractability performance using greedy decoding, prompted with identical training contexts. (c) Accuracy of attribute matching in conversational responses following instruction tuning at different checkpoints of unsupervised finetuning. See experimental set-up in sec 3.2.

- **Level 1: Loss-based memorization signals.** The model exhibits lower loss on training data compared to similar unseen text, i.e. from the same distribution. This the optimal signal exploited by Membership Inference Attacks (MIAs) (Sablayrolles et al., 2019), which determine whether a sentence was part of the training set. While this signal is *necessary* for extraction, it is *insufficient*—low loss does not guarantee that information can be easily extracted.
- **Level 2: Verbatim extraction.** A deeper level involves exact next-token prediction, evaluating the model's ability to complete a text given a prefix seen during training. This demonstrates that memorized content can be directly recovered.

Both phenomena are inherently linked and expected in language models, aligning directly with their training objectives. Indeed, consider $L_M(a|C)$, the cross-entropy loss of model M on attribute a (e.g., "12345678") averaged over all $|a|$ tokens of a when prompted with its training context C (e.g., "[...]. Bob's ID number is "). The expression $\exp(|a| \times L_M(a|C))$, representing the inverse of perplexity, exactly indicates the expected number of queries with context C required for M to accurately produce attribute a verbatim through random decoding.

To show this for top-k decoding, let us denote $l_i := L_M(a \mid C)[i]$ as the unnormalized output of the language model for token i. We consider l_i^k, where each value is retained as l_i if it ranks among the top-k values, and is set to 0 otherwise. Subsequently,

$$p_i^k := \log\left(\frac{\exp(l_i^k[a_n])}{\sum_{j=1}^{n}\exp(l_i^k[j])}\right)$$

represents the probability of outputting a_i conditioned on C and the sequence $(a_1, \ldots, a_{i-1})$ using a top-k decoding strategy. The probability of outputting the entire sequence a when prompted with C is given by $p^k(a \mid C) := \exp(-\sum_{i=1}^{n} p_i^k)$. This indicates that, on average, one needs to query the language model with $q\ E_C^M(a) = \frac{1}{p^k(a|C)}$ times to output a verbatim using a top-k sampling strategy. A similar calculation can be applied to other sampling strategies.

We now evaluate how verbatim memorization of attributes increases during unsupervised fine-tuning, as well as how it changes between attribute types.

IDs are more challenging to encode compared to other attributes. Figure 2a explores, averaged across the $10k$ tail biographies used in unsupervised fine-tuning, the loss of different attributes when prompted with their entire training context C: $|a| \times L_M(a|C)$. We observe that some attributes are logically more easily encoded, e.g. college compared to IDs. Indeed, college names are not a random sequence of letters—so the prior that the model has from pretraining can help— and they are made of less tokens in average, and are repeated between instances. Surprisingly, ID is harder to learn than ID Seq, while the first one has a considerably smaller number of unique possibilities: 10^8 compared to 52^8, see tab 1. For both, the loss plateaus after 12 epochs.

Verbatim extraction of unique attributes requires significantly longer unsupervised fine-tuning compared to other attributes. A low loss on an attribute given its context does not guarantee easy verbatim extraction. For instance, even if the 8-token ID "12345678" achieves an average loss of 1 after the context "[...]. Bob's ID number is " (corresponding to a summed loss of 8 in this figure), one would still need on average $\exp(8) \approx 3000$ model queries to correctly retrieve the ID number using fully random decoding. This is however lower for top-p/top-k/greedy decoding. Indeed, Figure 2b illustrates the verbatim extraction of various BIOs attributes from the tail across different epochs of unsupervised fine-tuning, this time using greedy decoding following the exact training context C (as detailed in Section 3.2 and illustrated in Fig. 1). The figure demonstrates that verbatim extraction of IDs approaches nearly 100% after 12 epochs. In contrast, other attributes start to be extractable after less epochs. However, this high level of verbatim memorization may still present a misleading impression of the actual extraction threat via chat interactions, which we explore in Section 4.2.

4.2 Extractable Chat Memorization

Chat is the most common mode of interaction with LLMs, but often overlooked in terms of data leakage potential. To form a more holistic study, we thus consider it here: we aim to extract the personal attributes of the training individuals of the tail via chat. For that, we take the unsupervised fine-tuned model checkpoints (at different epochs) and do instruction tuning for 10 epochs on

question–answer pairs with information from the head, and evaluate extractability of tail data, as detailed in Section 3.2. It teaches the model to respond to questions related to the attributes.

Chat extraction rate varies significantly between attributes, even when they are fully memorized verbatim. Figure 2c illustrates that chat extractability increases with more unsupervised fine-tuning epochs but quickly plateaus, remaining far from the 100% success rate seen in verbatim extraction (Figure 2b). This demonstrates that verbatim extraction alone does not provide a complete picture. Notably, unique attributes like ID and ID Seq have significantly lower chat extraction rates. These are memorized verbatim but extracting them through chat is more challenging due to the misalignment between encoding and extraction. Additionally, we note that the results reported are for *pretrained* Llama2-7B with the unsupervised fine-tuning on **BIOS-augment**. Deviations from this setup leads to changes in the chat extraction, as discussed in the following subsections.

Model size does not significantly impact chat extraction, while model quality is crucial. Larger models are known to memorize more information. This is intuitive from a verbatim perspective: more parameters mean greater storage capacity. But does this matter for extracting encoded information through chat? We compare chat extraction (after post training on **BIOs-augment**) across several models: a 7B scratch model, Llama1-7B, Llama2-7B, Llama2-13B, and Llama3-8B, to form LlamaN-XB-BIOs. All models undergo the same unsupervised fine-tuning and instruction tuning pipeline described in Section 3.2 and shown in Figure 2 for Llama2-7B. Notably, all models achieve 100% verbatim extraction accuracy for all attributes. However, as shown in Figure 3, extracting information from the tail through chat fails for the scratch model. Also, it is easier to extract facts from Llama3-8B-BIOs than Llama2-13B-BIOs, despite the latter having more parameters. The quality of Llama2-13B-BIOs is between Llama2-7B-BIOs and Llama3-8B-BIOs, as evidenced by 5-shot MMLU scores: Llama2-7B: 45.3, Llama2-13B: 54.8, Llama3-8B: 69.4. Chat extraction follows this trend. This highlights that model quality (as measured in benchmarks and commonly agreed by the community) is more crucial for chat extraction than model size.

Increased verbatim memorization does not imply more chat extraction. Zhu & Li (2023) show that varied formulations and sentence orders, rather than repeating BIOs with similar templates, greatly influence chat extractability. As detailed in Section 3.2, in **BIOs-augment**, the tail data from which we extract attributes has varied formulations across bios, with each BIO being present only once. The rest of the unsupervised fine-tuning set contains duplicate biographies with different formulations. We compare this setup to unsupervised fine-tuning on another dataset, which consists solely of biographies with identical sentence structures (e.g., "*X was born on Y*") and the same sentence order (e.g., *birth date*, followed by *birth city*, etc.). We build it in a similar chunk manner as **BIOs-augment**. Figure 4a illustrates the impact of augmentation on chat extraction. All attributes are uniformly affected by the absence of augmentation. Thus, while repetition is often seen as key for PII memorization, which holds true in a verbatim sense (Tirumala et al., 2022), what matters for PII extractability is how generally this type of PII is represented in other samples seen during training.

Augmentation in instruction tuning templates also increases chat extraction. We also study the impact of randomness in the instruction tuning question–answer format. We compare our default

Figure 3: Comparing **chat extraction** of tail attributes from models trained on **BIOs-augment**: Llama2-7B architecture from scratch (**Scratch**) < pretrained Llama1-7B (**Llama1-7B-BIOs**) < pretrained Llama2-7B (**Llama2-7B-BIOs**) < pretrained Llama2-13B (**Llama2-13B-BIOs**) < pretrained Llama3-8B (**Llama3-8B-BIOs**). Chat extraction increases with model quality more than size.

setting with randomness to a case without it, where we fix the question–answer sentence structure for all the BIOs records. Our intuition is that, similar to the importance of augmentation in unsupervised training, adding randomness in the question–answer format allows the model to generalize to the facts rather than over-fitting on the fixed q/a format. Thus, in the querying phase, the model has a better understanding of the query prompt. Figure 4b shows the impact of using randomness in the instruction tuning phase. We observe that indeed, randomness leads to higher chat extraction success.

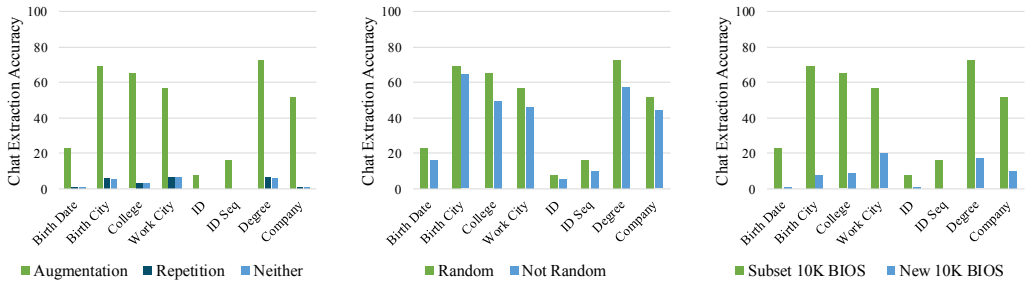
Instruction tuning on question-answer pairs about individuals from the unsupervised fine-tuning dataset is crucial. We investigate the effects of instruction tuning with information from the head of **BIOs-augment** —whose BIOs where used during unsupervised fine-tuning, this is the detault setting detailed in sec 3.2— compared to a new set of individuals. Interestingly, Figure 4c shows that after instruction tuning with information from individuals of a new set of BIOS, we fail to extract most attributes from the tail. One possible explanation is that instruction-tuning establishes a connection between a person's name and their attributes beyond verbatim, and this connection also emerges from other individuals encountered during training. But if the samples used for instruction fine-tuning are unfamiliar, no connection can be established because verbatim does not exist.

4.3 Link Between Chat Extraction and Encoding

Counter to common intuition, model quality rather than model size increases the risk of chat extraction; and verbatim memorization does not necessarily imply easy extraction via chat. In this section, we dig deeper into the interplay between both.

Not extractable verbatim does not mean not memorized. The relationship between Figure 2b and Figure 2c suggests that verbatim extraction is a prerequisite for chat extraction and that the verbatim extraction rate consistently serves as an upper bound for the chat extraction rate. We demonstrate that this is not necessarily true. We attempt verbatim extraction of attributes from the final model, this time after instruction tuning. Figure 5 illustrates that while verbatim extraction is 100% for all attributes after the unsupervised fine-tuning stage (dashed line at the top), it declines for *most* attributes following instruction tuning (surprisingly, the company attribute is still extractable).

This decline is due to the misalignment between the unsupervised fine-tuning and instruction tuning objectives, causing the model to deviate from the next-token prediction task necessary for successful verbatim extraction. However, we show that this does not imply that the information is not encoded in the model: we retrain the instruction-tuned model for next-token prediction, on all buckets except the tail and the head. We then evaluate verbatim extraction on the tail, finding that this results in high verbatim extraction *again* as shown in Figure 5. This underscores that 10 SFT epochs were not artificially "too" heavy to artificially remove everything; and that only evaluating verbatim memorization on instruct models can create a misleading sense of privacy.



(a) Impact of Data Augmentation (b) Randomness in QA Formulation (c) Type of Instruction Tuning Data

Figure 4: Ablation study on the impact of various parameters on chat extraction. (a) Impact of unsupervised fine-tuning data augmentation. Without augmentation, the model exhibits low chat memorization. (b) Impact of randomness in the question–answer formulation in the instruction tuning set. (c) Impact of instruction tuning data type: we compare model fine-tuned with a subset of unsupervised fine-tuning BIOS data to a model fine-tuned on a hold-out set of BIOS that has no overlap with the original fine-tuning data. See sec 3.2 for set-up details.

Removing verbatim memorization reduces chat extractability. Maini et al. (2024) explore various unlearning techniques and their limitations, demonstrating that traces of supposedly unlearned data can still be detected in the model, indicating partial failure. This finding motivated us to investigate a related but distinct question: if we successfully remove the verbatim encoding of information, does its chat extractability also disappear? We focus on the ID attribute —after unsupervised fine-tuning **and SFT**— of the Llama3-8B-BIOs model, which Figure 3 shows is 30% chat-extractable.

Our experimental setup involved splitting 10k individuals (from the tail, see sec 3.2) into three sub-groups: a 500-sample *unlearning set*, a 2000-sample *heldout set*, and the remaining samples as a *regularization set*. The core of our method if a modified training objective for the unlearning set's ID tokens. Instead of standard loss minimization, we train the model to achieve a high target loss of 2 on all IDs by minimizing the objective $|\text{loss} - 2|$. This target is chosen to approximate random guessing (loss ≈ 2.3 or $\ln(10)$ for a digit), thereby effectively erasing verbatim memorization as defined in Section 4.1. Concurrently, we train normally on the regularization set to ensure the model does not forget the concept of IDs altogether. The heldout is reserved for evaluation.

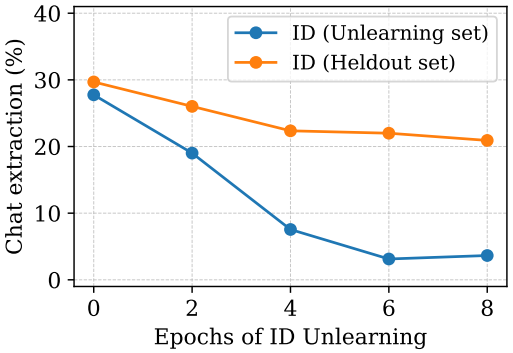


Figure 6: Impact of verbatim unlearning on chat extractability of IDs. The figure illustrates how IDs from the unlearning set become less chat-extractable as verbatim unlearning progresses, while heldout samples remain largely unaffected, challenging the hypothesis that unlearning does not work and that encoding removal would not prevent other type of extraction.

We do this unlearning phase for 8 epochs, and apply instruction-tuning (for 10 epochs on head data as in other experiments) to checkpoints at epochs 2, 4, 6, and 8. The results, shown in Figure 6, are clear: as verbatim unlearning progresses, the chat extractability for IDs in the unlearning set systematically decreases. In contrast, the heldout set remains largely unaffected, showing that it is specific to the unlearned IDs. This demonstrates that, counter to our hypothesis, removing the low-level verbatim encoding seems sufficient to prevent higher-level chat extraction.

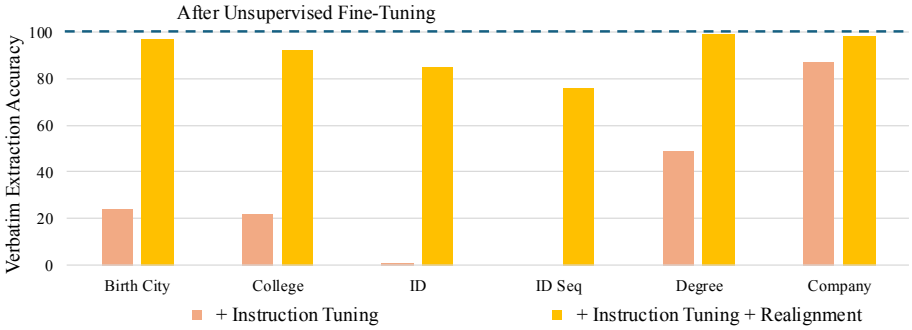


Figure 5: Verbatim Memorization Measurements can be misleading. We compare verbatim extraction at different stages of model training: after unsupervised fine-tuning, after instruction tuning, and after realignment. For realignment, we take the instruction tuned model and train it on a subset of BIOs for next-token prediction (similar to the unsupervised fine-tuning task), and evaluate **on the remaining subset**. Verbatim extraction drops after instruction tuning, but realignment re-boosts extraction rate.

5 Discussion & Conclusion

5.1 Impact Statement.

This paper aims to advance the understanding of privacy implications in large language models, particularly focusing on the nuances of memorization and non-adversarial chat extraction of facts. Our findings contribute to the field by highlighting the complexities of privacy beyond verbatim memorization. However, as stated in the disclaimer in the introduction (section 1), our experiments are conducted in a synthetic setting, and the reader should only learn from the relative extraction rates rather than absolute ones.

Moreover, our goal was to compare and simulate two possibly realistic extraction scenarios: a verbatim extraction attack, where an adversary would have access to the logits and will likely use greedy decoding to maximize extraction success, as well as a typical chat interaction through an API, which involves stochastic decoding. The decoding strategy is itself a confounding variable when comparing the absolute extraction rates between these two settings. While we do not thoroughly analyze this aspect—focusing instead on the relative extractability of different attributes rather than their absolute values—recent work by Hayes et al. (2024) provides complementary evidence. They extend the notion of discoverable memorization to *probabilistic* discoverable memorization, which explicitly accounts for stochastic decoding, and demonstrate that the trends remain consistent with greedy decoding results.

Additionally, as highlighted by a reviewer, our dataset exhibits a geographic bias, being in English and predominantly US-centric in its coverage, and we don't link our observations to the distribution of the pretraining data itself. While we expect the relative trends we observe—such as the differential extractability of various attribute types—to generalize across geographies and domains, the absolute extraction rates may differ significantly for underrepresented regions and languages.

Finally, while the focus of the work was largely on privacy, we hope that the empirical findings and discussion can advance the more general field of factual knowledge encoding and retrieval in LLMs' weights.

5.2 Conclusion.

Our study highlights the nuanced relationship between language models' memorization and privacy. While verbatim memorization is straightforward, chat extraction is more intricate. We demonstrate that not all attributes are equally revealed, with unique identifiers being less prone to extraction. More powerful models are more likely to leak data, yet better verbatim memorization does not necessarily increase chat leakage. Instruction-tuning removes verbatim extraction, suggesting that privacy evaluations should consider more than just verbatim memorization evaluations through Membership Inference Attacks or detecting training data splits. This complexity underscores the need for a holistic approach to privacy in large language models.

References

Allen-Zhu, Z. and Li, Y. Physics of language models: Part 3.2, knowledge manipulation. *arXiv preprint arXiv:2309.14402*, 2023.

Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium*, pp. 267–284, 2019.

Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramer, F. Membership inference attacks from first principles. *arXiv preprint arXiv:2112.03570*, 2021a.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021b.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., and Wallace, E. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270, 2023.

Chang, K. K., Cramer, M., Soni, S., and Bamman, D. Speak, memory: An archaeology of books known to chatgpt/gpt-4. *arXiv preprint arXiv:2305.00118*, 2023.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, pp. 265–284, 2006.

Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., and Ristenpart, T. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd USENIX Security Symposium*, pp. 17–32, 2014.

Hayes, J., Swanberg, M., Chaudhari, H., Yona, I., Shumailov, I., Nasr, M., Choquette-Choo, C. A., Lee, K., and Cooper, A. F. Measuring memorization in language models via probabilistic extraction. *arXiv preprint arXiv:2410.19482*, 2024.

Hu, J., Wu, X., Wang, W., Xianyu, Zhang, D., and Cao, Y. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*, 2024.

Ippolito, D., Tramèr, F., Nasr, M., Zhang, C., Jagielski, M., Lee, K., Choquette-Choo, C. A., and Carlini, N. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv preprint arXiv:2210.17546*, 2022.

Jayaraman, B. and Evans, D. Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium (USENIX Security 19)*, pp. 1895–1912, 2019.

Kim, S., Yun, S., Lee, H., Gubri, M., Yoon, S., and Oh, S. J. Propile: Probing privacy leakage in large language models. *Advances in Neural Information Processing Systems*, 36:20750–20762, 2023.

Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z. C., and Kolter, J. Z. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

Meehan, C., Bordes, F., Vincent, P., Chaudhuri, K., and Guo, C. Do ssl models have déjà vu? a case of unintended memorization in self-supervised learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Mireshghallah, F., Uniyal, A., Wang, T., Evans, D., and Berg-Kirkpatrick, T. Memorization in nlp fine-tuning methods. *arXiv preprint arXiv:2205.12506*, 2022.

Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Wallace, E., Tramèr, F., and Lee, K. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.

Panda, A., Choquette-Choo, C. A., Zhang, Z., Yang, Y., and Mittal, P. Teach llms to phish: Stealing private information from language models. *arXiv preprint arXiv:2403.00871*, 2024.

Sablayrolles, A., Douze, M., Schmid, C., Ollivier, Y., and Jégou, H. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pp. 5558–5567. PMLR, 2019.

Sander, T., Yu, Y., Sanjabi, M., Durmus, A., Ma, Y., Chaudhuri, K., and Guo, C. Differentially private representation learning via image captioning. *arXiv preprint arXiv:2403.02506*, 2024.

Schwarzschild, A., Feng, Z., Maini, P., Lipton, Z. C., and Kolter, J. Z. Rethinking llm memorization through the lens of adversarial compression. *arXiv preprint arXiv:2404.15146*, 2024.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (S&P)*, pp. 3–18, 2017.

Staab, R., Vero, M., Balunović, M., and Vechev, M. Beyond memorization: Violating privacy via inference with large language models. *arXiv preprint arXiv:2310.07298*, 2023.

Tirumala, K., Markosyan, A., Zettlemoyer, L., and Aghajanyan, A. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pp. 268–282. IEEE, 2018.

Yu, Y., Sanjabi, M., Ma, Y., Chaudhuri, K., and Guo, C. Vip: A differentially private foundation model for computer vision. *arXiv preprint arXiv:2306.08842*, 2023.

Zhang, Y., Jia, R., Pei, H., Wang, W., Li, B., and Song, D. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 253–261, 2020.

Zhu, Z. A. and Li, Y. Physics of language models: Part 3.1, knowledge storage and extraction. *arXiv preprint arXiv:2309.14316*, 2023.

A Compute Resources and Cost

We utilize our internal cluster equipped with A-100 GPUs, each with 80GB of memory. Each unsupervised fine-tuning phase for a 7B model was conducted on a single node. As detailed in Section 3.2, we perform up to 20 epochs on a 75M-token dataset, which takes approximately half a day. This process is repeated for three different types of augmentations in the training data, as described in Section 4, as well as for one Llama1-7B, one Llama2-13B, and one Llama3-8B post-training. In total, this amounts to approximately $12 \times 8 \times 6 \approx 600$ GPU hours. All subsequent SFT and unlearning phases are more than 10 times less resource-intensive (see Section 3.2), so we omit them here. Considering all failed experiments and those not included in the paper, we estimate that the total training time was close to 2000 GPU hours.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We have made our contributions clear through bullet points.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We have added a limitation subsection near the conclusion.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

Justification: no theoretical result

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We have tried to give as many details as possible on how we ran all experiments.

   Guidelines:

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

5. **Open access to data and code**

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

Answer: [No]

Justification: We could not open source everything yet.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We give the details in the experiment section.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Each run is compute intensive, so we could not target standard deviations. However, the work focuses on realative numbers rather than their absolute values.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We add a paragraph on the computational cost of our experiments in the supplementaty material.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have looked at the code of ethics and our work respects it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss both throughout the paper.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: No such release.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We only use synthetic data and models with free research licenses.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: We don't believe that there is such new asset.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [NA]

    Justification: no human subject

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: No human subject

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [No]

Justification: only used for minor grammar and phrasing.

Guidelines:

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