IA-NT-ES-25-528
MAY 2025

PATRICO ROJAS E.

Between Meteorites and Pianos: Why ChatGPT Doesn't Remember Your Secret Project

Many users express concern about how their data is used when interacting with language models like ChatGPT.

The specific fear is that confidential information - such as business project evaluations, financial figures, or commercial strategies - could (1) be **used to train** the model, (2) become somehow **embedded** in the model's parameters, or (3) **later appear** in another user's response, either fully or partially.

This fear is based on the idea that a model can "remember" what it is shown and that this "memory" could manifest in an uncontrolled manner.

In the case of ChatGPT using GPT-4o (OpenAI's 2025 model), its operational policy states that:

- Input data is **not used for training** by default.
- If the user **chooses to share** their data to improve the system, it may be used in future training or fine-tuning processes, under conditions of filtering, aggregation, and anonymization.
- During interactive use of the model, **no weight modification occurs, and there is no persistent data storage between sessions**.

Technical Note prepared by Patricio Rojas E., professor at ESE Business School, Universidad de los Andes (Chile), to serve as the basis for class discussion and not as an example of the appropriate or inappropriate management of a specific situation.

Given this, a common question arises:

> *If a user enters sensitive data in a conversation with ChatGPT, how likely is it that this data alters the model's parameters in a way that enables its later retrieval by other users, or that it appears on their screens?*

To answer this, it is necessary to understand how the learning mechanism works and to quantitatively estimate the effect of an individual example, topics that will be addressed below.

The conclusion of the analysis is clear: The probabilistic estimate, based on the training properties of large-scale language models, and formulated using principles of optimization, average gradient, and statistical behavior of parameters, indicates that this probability is not just low, it is astronomically negligible.

To offer an intuitive comparison: it is several orders of magnitude more likely for a person to be struck by a meteorite or crushed by a falling piano than for the content they enter into ChatGPT to reappear in another user's session.

II. PARAMETER ADJUSTMENT DURING LANGUAGE MODEL TRAINING

To assess whether user-entered content, such as a figure, a project evaluation, or a confidential statement, can be recorded by the system, it is necessary to understand how the parameters of a model like ChatGPT are adjusted during training. This process does not function like a memory that retains individual texts; rather, it operates as a statistical mechanism that generalizes distributed patterns from large volumes of data.

GPT-4o, the underlying model in the 2025 version of ChatGPT, is based on the Transformer architecture and contains over 100 billion parameters. Its training is autoregressive, meaning that given a context, the model estimates the probability of the next token in the sequence. This allows it to generate predictions word by word without explicitly storing the content it has processed.

The model is trained using **stochastic gradient descent**, employing batches of examples. Each example technically corresponds to an input sequence—a series of tokens derived from textual content. At each training step, the model computes the error between its prediction and the actual expected value, and it adjusts its internal parameters, θ, to minimize that discrepancy. This adjustment is performed using optimization algorithms such as ADAM (Adaptive Moment Estimation)

Formally, the parameter update at a training step is given by:

$$\Delta\theta = -\eta \cdot \frac{1}{B}\sum_{i=1}^{B} \nabla_{\theta} L(\theta, x_i)$$

where:

- $\boldsymbol{\theta}$ represents the model's weight vector,
- $\boldsymbol{\eta}$ is the *learning rate*,
- **B** is the *batch* size,
- $\mathbf{x_i}$ is the input sequence for example i,
- $\mathbf{L(\theta, x)}$ is the loss function.

This formulation expresses a fundamental principle: **each piece of content introduced has a numerically minuscule effect** on the model's parameters. The contribution of a single example is diluted in the gradient average of thousands of simultaneous examples and may even be partially neutralized by others.

Under realistic conditions (*typical learning rate, standard batch size, and observed gradient magnitudes*), the change induced by a single input is on the order of 10^{-8} per parameter. Given that the model's weights fall within ranges such as [–1, 1], this variation is negligible.

Annex A.1 details this quantitative estimate, and Annex A.2 provides a comparative table with different content types: trivial greetings, common financial statements, or unusual sequences with sensitive information. Even in the most extreme cases, such as rare or adversarial sequences, the individual impact remains imperceptible and is further mitigated by internal mechanisms like Layer Normalization and Weight Decay.

In summary, although any input technically causes an adjustment in parameters, the model's architecture and scale, along with its stabilization mechanisms, make that impact practically irrelevant. Only under completely different conditions, such as intensive repetition or deliberate manipulation of the training corpus, could a detectable cumulative effect occur, which does not happen under normal system usage.

III. STATISTICAL UNLIKELIHOOD AND ITS EFFECT ON LEARNING

A recurring question among users is whether certain types of input, due to their confidential, unusual, or specific nature, could cause exceptional effects on the model. To address this concern, it is essential to define the concept of unlikelihood, understood not from human criteria (such as sensitivity, privacy, or novelty), but from a statistical perspective.

In a model like GPT-4o, each *token* is probabilistically evaluated based on its context. The unlikelihood of an input sequence corresponds to the low probability the model assigns to that

sequence, calculated as the product (or logarithmic sum) of the conditional probabilities of its *tokens*. This formulation is presented in Annex A.3.

From this perspective, there are two main types of unlikelihood:

- **Structural unlikelihood**: the sequence contains infrequent token combinations, such as alphanumeric codes, technical acronyms, formulas, or unusual grammatical constructions.
- **Semantic unlikelihood**: the content expresses assertions that are improbable according to the overall distribution of the corpus, such as undocumented events or hypotheses unsupported by data.

These forms of unlikelihood are not associated with meaning, confidentiality, or sensitivity in the human sense. The model does not distinguish between a personal number and a random figure, except in terms of their relative frequency in the corpus. What is evaluated is statistical regularity, not the practical value of the content.

To estimate this unlikelihood, several complementary metrics are used:

- **N-gram uniqueness**: observed frequency of consecutive *token* combinations.
- **Distance in *embedding* space**: how far representation vectors are from the centroids of common clusters.
- **Internal frequency distribution**: concentration of rare *tokens* within the sequence.

A summary metric is **estimated perplexity (PPL)**, which quantifies how predictable a piece of content is for the model. Low values indicate common sequences; high values point to improbable or incoherent ones. Annex A.3 includes typical reference values:

Type of Content	Estimated PPL
Trivial greeting	10 - 20
Technical or financial text	30 - 80
Unusual or contradictory claim	100 - 500
Completely random text	>1000

Applied to content such as "*Project Z of Company X has a 147.2% ROI and launches in September with supplier Y*", the unlikelihood will depend on the statistical familiarity of the elements it contains: if "Company X" and "Supplier Y" are frequently occurring entities in the corpus, and if "147.2%" falls within the typical range of financial figures, then the sequence will not be considered rare. From the model's perspective, this content is no different from other common business texts, even if it is confidential to the person writing it.

That said, even if a submitted input is statistically rare due to its structure, figures, or context, this **does not automatically imply that it will significantly affect the model**. For its impact to be greater, additional conditions must be met:

- The model must make a considerable error in predicting it (high loss function).
- The content must be repeated many times during the training process.
- There must be no regularization mechanisms that neutralize the associated gradient.

In normal usage scenarios, these conditions are not met. Unlikelihood alone is not enough for input content to have a persistent or identifiable effect on the model.

IV. ASSESSING THE RISK OF EXPOSURE UNDER NORMAL USAGE CONDITIONS

A common concern among users is whether content entered into ChatGPT, such as a financial evaluation, a business strategy, or a confidential piece of information, could later reappear in another user's response. This legitimate concern must be evaluated based on the system's actual behavior, rather than analogies with platforms that operate under different paradigms.

System Behavior During Normal Use

During an interactive session, ChatGPT **does not train, does not adjust its parameters, and does not retain content between sessions**. Each conversation is independent, and there is no persistence of the information provided by the user beyond the local and temporary processing.

Additionally:

- Input content is not used for training unless the user explicitly consents.
- Even with consent, the data undergo filtering, anonymization, and aggregation processes before it can be considered part of the training corpus.
- There is no shared memory between users, nor any traceability across previous sessions.

This means that content introduced only once, even if statistically unusual or potentially sensitive, **is not available to other users**, either explicitly or implicitly. The system's architecture prevents this by design.

Without feedback, whether from the user or the system, a typical interaction holds no technical value for training. Adjusting a model requires knowing whether its behavior was correct or not, and without a clear signal, there's no way to make that assessment. That's why even if someone has enabled the option to share data, their conversation won't be used unless it meets those criteria. **Consent is necessary but not sufficient**, what matters is that the content is usable, and **in most cases, it isn't**.

Risk Quantification

The probability that entered content will reappear verbatim or in a recognizable form in another session, without adversarial prompting, repetition, or explicit inclusion in the corpus, is **less than 10^{-100}**. This estimate is based on:

- The scale of the model and the training corpus.
- The degree of gradient dilution in large *batches*.
- The presence of multiple layers of normalization and regularization.
- The absence of real-time training.

In comparison, this level of improbability is several orders of magnitude lower than highly unlikely everyday events. The corresponding figures are detailed in Annex A.4.

From a technical and operational standpoint, this **risk is not just low, it is negligible**.

Exceptional Scenarios: Adversarial Attacks

Some academic studies have documented the possibility of recovering sequences introduced during the training of older models under laboratory conditions. These scenarios include:

- Intentional repetition of the same content thousands of times.
- Training corpora designed without filtering or deduplication.
- Architectures lacking modern safeguards such as *weight decay, dropout, or differential privacy*.

These cases, analyzed in Annex A.5, **do not apply to current versions of ChatGPT** (GPT-4o and GPT-3.5), nor do they reflect the system's behavior under real-world usage conditions.

V. TECHNICAL CONCLUSIONS ON CONTENT PERSISTENCE IN CHATGPT

A detailed analysis of the training process, model architecture, and inferential behavior supports the technically grounded conclusion that the risk of accidental exposure of user-entered content in ChatGPT is quantitatively negligible, for the following reasons:

Content is not stored or retained as individual examples

During the training of models like GPT-4o, input sequences, that is, the content that makes up the training corpus, contribute to parameter adjustment solely through the averaged gradient calculated in each batch. The influence of a single piece of content, even if rare, is on the order of 10^{-8} per parameter.

The model's structure is designed to learn generalizable patterns, not to retain specific occurrences. Unless a given piece of content is massively repeated, there is no possibility for it to persist as a recognizable unit.

There is no real-time training or session persistence

During regular use of ChatGPT, the model is not modified. The weights are frozen. Each session operates as an isolated environment. Input content is processed locally and leaves no trace that could be used or recalled in future sessions.

In both the free version (GPT-3.5) and the Plus version (GPT-4o), there are no technical mechanisms that enable content transfer between users or model rewriting as a result of interactions.

The probability of exposure is below the operational threshold

The chance that content entered only once could reappear in another user's conversation, without targeted attacks, massive repetition, or later inclusion in the training corpus, is estimated to be **lower than 10^{-100}**. This value falls below any threshold considered operational in the evaluation of technological risks.

Memorization studies do not apply to real-world use

Experiments like those by Carlini et al. (2021)[1] showed that, under artificial conditions, some models could recover training data. But those results depend on a combination of factors that are not present in the normal use of ChatGPT. A manipulated corpus without deduplication, data repeated hundreds of times, prompts designed with prior knowledge, and lower-capacity models like GPT-2 lacking modern safeguards.

Applying those findings to the current environment is conceptually flawed and leads to distorted perceptions of risk.

Risk perception does not reflect its actual magnitude

User perception of the risks of sharing information with a language model may be influenced by analogies to systems that do retain history (browsers, search engines, social networks). But that analogy is invalid here. ChatGPT does not function as a database or a distributed memory system. There is no persistence, no traceability across users, and no session storage within the model.

This concern is understandable, but technical evaluation shows that, under normal usage conditions, the risk of exposure is virtually nonexistent.

[1] Carlini, Nicholas, et al. "Extracting training data from large language models." *30th USENIX security symposium (USENIX Security 21)*. 2021

What is particularly revealing about public risk perception is the contrast between what does and does not trigger alarm, which shows a clear asymmetry. While many users distrust tools like ChatGPT, which do not store conversations, do not train in real time, and do not propagate data across sessions, the vast majority readily accept systems where personal information is not only stored but integrated into ecosystems of active commercial exploitation. Millions of people deposit sensitive data daily into email services like Gmail (Google) or Outlook (Microsoft) and use platforms like WhatsApp (Meta) as primary channels of personal and professional communication. In those environments, messages are indeed stored, may be algorithmically analyzed, and are linked to infrastructures with explicit commercial goals. Yet, this risk does not provoke the same kind of alarm.

This disparity is not explained by technical differences but by differences in perception. Fear tends to focus on the unfamiliar, on what is not understood, while the familiar, even if structurally riskier, is perceived as neutral or inevitable. In that sense, the risk lies not in technology itself, but in how we perceive it, and more often in users' habits than in the technologies they employ.

You don't need to trust a company's promises to feel a certain level of security. All it takes is thinking in terms of incentives. Around the world, thousands of lawyers are waiting for an opportunity to launch multimillion-dollar lawsuits against companies like OpenAI or Google, just as they've already done with Facebook, Amazon, or Microsoft. And they don't necessarily do it out of principle, but because they know that a well-structured lawsuit could make them rich. That legal pressure, combined with the risk of a massive user exodus, imposes clear limits. A company may or may not honor its own values and policies, but what it almost always does is whatever prevents massive financial loss. If we don't believe in their principles, we can believe in their survival instinct.

VI. TECHNICAL ANNEXES

Below are the technical annexes referenced throughout the document. Their purpose is to provide explicit formulations, estimated values, and comparative tables that quantitatively support the arguments presented.

Annex A.1 - Parameter Update During Training

Reference: Section II

This annex presents the formal expression of the parameter adjustment process in language models trained using gradient descent.

GPT-4o is a neural network based on Transformer architecture, featuring multiple layers of attention and linear projection. The relevant characteristics are:

- Scale: 100 billion parameters (10^{100}).
- Supervised training with autoregression: the model predicts the next token in a sequence.

- Optimization via gradient descent, typically using the ADAM (Adaptive Moment Estimation) algorithm.

The parameter update during a stochastic gradient descent step is defined as:

$$\Delta\theta = -\eta \cdot \nabla_\theta L(\theta, x)$$

Where:

- η is the learning rate (typically ~10^{-4}).
- L is the loss function (cross-entropy).
- x is the input example (tokenized).
- $\nabla_\theta L$ is the gradient of the loss with respect to the parameters.

In practice, training is performed in *batches*, so the impact of a single example is diluted:

$$\Delta\theta_{x_i} \approx \frac{1}{B} \cdot \eta \cdot \nabla_\theta L(\theta, x_i)$$

Where B is the batch size, typically 2048 tokens, and the learning rate is $\eta \sim 10^{-4}$

Typical Gradient Magnitude

The gradient $\nabla_\theta L$ may have a norm (L2) in the following range:

- **Common example**: $||\nabla_{(\theta)} L|| \approx 0.01$
- **Rare but coherent example**: $||\nabla_\theta L|| \approx 0.1$
- **Highly anomalous or adversarial example**: $||\nabla_\theta L|| \approx 1.0$

Estimated Calculations by Case

Assuming $\eta = 10^{-4}$ y B = 2,048, and taking a single parameter θ_j whose partial derivative is on the order of the average gradient, we obtain:

a) Common example (gradient ≈ 0.01):

$$\Delta\theta_j \approx \frac{1}{2{,}048} \cdot 10^{-4} \cdot 0.01 \approx 4.88 \times 10^{-10}$$

b) Rare example (gradient ≈ 0.1):

$$\Delta\theta_j \approx \frac{1}{2{,}048} \cdot 10^{-4} \cdot 0.1 \approx 4.88 \times 10^{-9}$$

c) Anomalous example (gradient ≈ 1.0):

$$\Delta\theta_j \approx \frac{1}{2{,}048} \cdot 10^{-4} \cdot 1.0 \approx 4.88 \times 10^{-8}$$

This value remains extremely small compared to the typical magnitude of the model's weights (values between -1 y 1, with typical deviations ranging from 0.01 to 0.1 in many layers).

Important considerations:

- The induced change **depends not only on the magnitude of the gradient but also on its direction**: many examples contribute opposing gradients, and an isolated gradient may be statistically canceled out.
- The network **generalizes based on common patterns**, not individual examples, unless those examples are extremely repeated or statistically prominent.

Annex A.2 - Estimated Impact by Type of Content

Reference: Section II.

Comparative table shows the estimated impact on the model's parameters based on the type of input content.

Examples of unlikelihood and estimated gradient

Input Example	Expected Frequency	Estimated Perplexity (PPL)	Relative Gradient	Estimated Impact
Hi, how are you?	Very high	~10	Bajo (~0.01)	≈ 1e-10
The ROI of Project Alpha was 11.2% in Q3	Moderate	~35	Low to medium (~0.05)	≈ 2.5e-9
Project Z of Company X launches with supplier Y	Low (specific names)	~70	Medium (~0.1)	≈ 5e-9
NG43-AX79 is the Access code to the Beta satellite camera	Very low	>150	High (~0.5–1.0)	≈ 2.5e-8 to 5e-8

Note: approximate values under ideal conditions. Significant cumulative impact requires repeated exposure.

Annex A.3 – Predictive Unlikelihood and Statistical Regularity

Reference: Section III.

In the context of language models like GPT-4o, *predictive unlikelihood* refers to **how improbable or statistically unusual** a given text sequence is, relative to the corpus used to train the model.

Formally, if $x = (t_1, t_2, ..., t_n)$ is a sequence of tokens, its unlikelihood is measured by:

Total log-probability: $\log P(x) = \sum_{i=1}^{n} \log P(t_i | t_1, \ldots, t_{i-1})$

Perplexity (PPL): $\exp\left(-\frac{1}{n}\sum_{i=1}^{n}\log P\left(t_i|t_{<i}\right)\right)$

Where $P(t_i|t_{<i})$ is the probability assigned by the model to each token given its context

- Conceptual distinction:
    - **Structural unlikelihood:** low frequency of tokens or syntactic combinations.
    - **Semantic unlikelihood**: statements that are improbable or inconsistent with the corpus.

- Complementary metrics:
    - *N-gram uniqueness*.
    - Distance in *embedding* space.
    - Internal frequency distribution.
- Perplexity reference table:

Type of Content	Estimated PPL
Trivial greeting	10–20
Technical or financial text	30–80
Implausible or contradictory claim	100–500
Random text	>1,000

Annex A.4 – Comparison of Improbable Risks

Reference: Section IV.

Problem Statement

This section assesses whether a confidential data point entered by a user into ChatGPT (e.g., a financial evaluation or strategic plan) could unintentionally appear as a response for another user in a different session. This scenario assumes no attacks, adversarial engineering, or prior knowledge.

Relevant Technical Facts

- ChatGPT does not train in real time. User interactions do not modify the base model.
- There is no shared memory between sessions. Each session is independent.
- User data is not used for training without explicit consent.

- The model generalizes patterns; it does not memorize unique individual examples.

Given the above, and assuming a sensitive data point is introduced only once, the probability **P** that another user will see this information is:

$$P < 10^{-100}$$

The following illustrative comparison of probabilities helps contextualize the estimated improbability of accidental exposure risk.

Illustrative Scale of Improbability

Event	Estimated Probability
Being struck by a meteorite in one's lifetime[2]	≈ 1 in 1,600,000
Winning the Powerball (EE. UU.)	≈ 1 in 292,000,000
Being crushed by a falling piano	< 1 in 10,000,000 (theoretical)
A sensitive ChatGPT input being shown to another user.	< 1 in 10^{100}

Annex A.5 – Adversarial Scenarios and Their Inapplicability

Reference: Section IV.

Necessary conditions under which *memorization* was documented in earlier studies are not replicable in current versions of ChatGPT.

Risk Comparison by Use Case and Model Version

Scenario	Model	Data Use Enabled	Risk of accidental exposure
User inputs a single example	GPT-4o (Plus)	No	< 1e-100
User inputs a single example	GPT-3.5 (Free)	No	< 1e-50
User repeats an example 10,000 times	GPT-3.5 (no regularization)	Yes	≈ 1e-4 (extreme, theoretical case)
Intentionally injected adversarial example	GPT-2 (no safeguard)	Yes	> 1e-3 (in studies)
Forced attack using directed prompt and memorized example	Not applicable to current ChatGPT	No relevant	Unfeasible

[2] Estimation by Stephen Nelson, Ph.D. in Geology & Earth Science, University of California, and member of the Department of Earth & Environmental Sciences at Tulane University.