# Guess or Recall? Training CNNs to Classify and Localize Memorization in LLMs

**Jérémie Dentan**[1], **Davide Buscaldi**[1, 2], **Sonia Vanier**[1]

[1]LIX (École Polytechnique, IP Paris, CNRS)
[2]LIPN (Université Sorbonne Paris Nord)
jeremie.dentan@polytechnique.edu, davide.buscaldi@polytechnique.edu, sonia.vanier@polytechnique.edu

## Abstract

Verbatim memorization in Large Language Models (LLMs) is a multifaceted phenomenon involving distinct underlying mechanisms. We introduce a novel method to analyze the different forms of memorization described by the existing taxonomy. Specifically, we train Convolutional Neural Networks (CNNs) on the attention weights of the LLM and evaluate the alignment between this taxonomy and the attention weights involved in decoding. We find that the existing taxonomy performs poorly and fails to reflect distinct mechanisms within the attention blocks. We propose a new taxonomy that maximizes alignment with the attention weights, consisting of three categories: memorized samples that are *guessed* using language modeling abilities, memorized samples that are *recalled* due to high duplication in the training set, and *non-memorized* samples. Our results reveal that few-shot verbatim memorization does not correspond to a distinct attention mechanism. We also show that a significant proportion of extractable samples are in fact guessed by the model and should therefore be studied separately. Finally, we develop a custom visual interpretability technique to localize the regions of the attention weights involved in each form of memorization.

**Code** — https://github.com/orailix/cnn-4-llm-memo

## Introduction

Large Language Models (LLMs) are known to memorize a significant portion of their training data, raising legal and ethical challenges (Zhang et al. 2017; Mireshghallah et al. 2022; Carlini et al. 2023b). Recently, Prashanth et al. (2024) view memorization as a multifaceted phenomenon and propose a taxonomy of memorized samples. They focus on training set samples that are *32-extractable*: when prompted with the first 32 tokens of a sequence (the *prefix*), the model outputs exactly the next 32 tokens (the *suffix*). Samples are categorized into four classes: *Non-memorized, Recite, Reconstruct, and Recollect* (defined in Figure 1). This taxonomy aims to capture the different mechanisms underlying each type of memorization and is motivated by high-level features such as perplexity and token frequency. The authors show that each category exhibits distinct and consistent high-level characteristics across memorized samples.



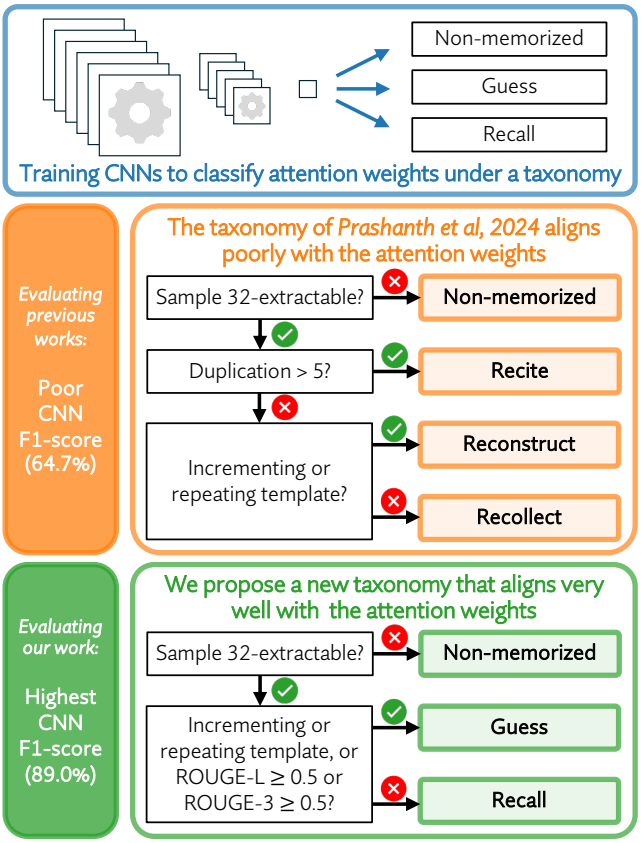



Figure 1: To evaluate a taxonomy of memorized samples, we train CNNs to classify attention weights under this taxonomy. The existing taxonomy yields poor performance. Our new, simpler taxonomy aligns much closely with the attention mechanism involved in data regurgitation.

In this paper, we take a step further by investigating whether these different forms of memorization can be observed at the level of the model's attention weights. Since the samples are 32-extractable, there must be causal links between the prefix and the suffix that enable the exact decoding of the latter. We therefore analyze the attention weights to uncover such links. Building on the taxonomy of Prashanth et al. (2024), we ask whether the nature of these causal

links differs across the categories. To identify distinctive patterns in attention weights across heads and layers, we trained CNNs to classify them according to the taxonomy of Prashanth et al. (2024) (see Figure 1; further details are discussed in later sections). The CNNs revealed that the classes proposed by Prashanth et al. (2024) do not align well with the attention weights, as evidenced by frequent misclassifications. For example, Figure 3 shows a sample that should clearly belongs to the *Reconstruct* class but is labeled *Recollect*. Similarly, we observed many *Recite* samples whose attention weights closely resemble those of *Reconstruct*, as reflected in the CNNs' frequent misclassifications.

These misclassifications are problematic, as they suggest that this taxonomy does not reflect the underlying causal mechanisms between the prefix and the suffix in memorized samples. For example, we observed that *Recollect* does not represent a truly distinct form of memorization; rather, these samples should be reassigned to either *Recite* or *Reconstruct*, depending on which group their attention weights most closely resemble (see Figure 2). This distinction is crucial, as it prompts a reevaluation of what models are capable of memorizing: few-shot memorization of samples observed only few times may be illusory. This aligns with the findings of Huang, Yang, and Potts (2024) and calls for a reconsideration of current approaches to mitigate memorization.

To address the limitations of the taxonomy proposed by Prashanth et al. (2024), we introduce a new, simpler, data-driven taxonomy of memorized samples. We developed a protocol to systematically explore multiple candidate taxonomies and evaluate their alignment with attention weights by measuring the performance of CNNs trained to classify attention weights under each taxonomy. Based on this protocol, we propose a new taxonomy that ranks highest in our benchmark and accurately captures distinct mechanisms in the attention weights. It comprises three classes defined in Figure 1. *Non-Memorized* class is similar to that of Prashanth et al. (2024). *Guess* captures samples where most suffix tokens can be inferred from the prefix. It includes ROUGE-based rules to improve *Reconstruct* class in Prashanth et al. (2024). All remaining samples are assigned to *Recall*, as we did not observe distinct subgroups that would justify further splitting. See examples in Figure 3.

Using a well-designed taxonomy such as ours is crucial for accurately studying memorization. For instance, Stoehr et al. (2024) identified an attention head in the lower layers that is highly correlated with memorization. Our results suggest that these lower layers primarily contribute to memorizing *Guess* samples. While their finding is valid, it is unlikely to generalize to all forms of memorization. To demonstrate the benefits of studying *Guess* and *Recall* as distinct forms of memorization, we developed a custom technique to identify the regions of the attention weights that play a significant role for each case. Our results show that *Guess* samples exhibit high activations in the lower layers of the model, consistent with the observations of Stoehr et al. (2024). We also show that *Recall* relies on short-range interactions between neighboring tokens in the upper layers, corroborating the findings of Huang, Yang, and Potts (2024) and Menta, Agrawal, and Agarwal (2025). These fine-grained localization results are made possible by our taxonomy, which disentangles truly distinct memorization mechanisms.

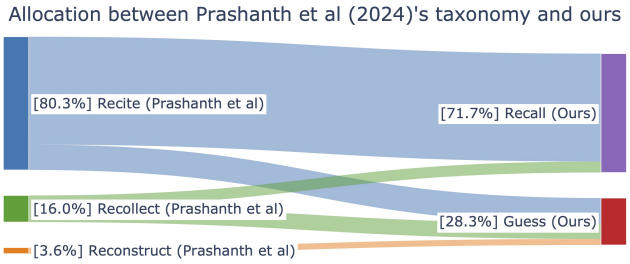



Figure 2: Comparison of samples' labels between Prashanth et al. (2024)'s taxonomy and ours. *Guess* class is broader than *Reconstruct*, including all samples where the suffix largely predictable from the prefix, which exhibit similar attention weights. We omit non-memorized samples here.

**Summary of Contributions**

- We propose a new method to analyze the role of attention blocks in memorization, by training CNNs on attention weights.
- We benchmark the alignment of various taxonomies with attention weights, including Prashanth et al. (2024)'s.
- We introduce a new data-driven taxonomy that addresses the limitations of previous ones and maximizes alignment with attention weights: *Guess, Recall, Non-memorized*.
- We develop a method to interpret the CNNs and localize the regions of the LLM involved in memorization. Our conclusions bridge the gap between the results of Stoehr et al. (2024), Huang, Yang, and Potts (2024), and Menta, Agrawal, and Agarwal (2025).

## Background and Related Works

### Verbatim Memorization in LLMs

Training data memorization is a broad phenomenon that affects many types of models (Fredrikson et al. 2014; Mahloujifar et al. 2021; Carlini et al. 2021, 2023a; Dentan, Paran, and Shabou 2024). In this work, we focus on *verbatim memorization* in LLMs: a sample is considered memorized if it is *extractable* from a prompt using greedy decoding (Carlini et al. 2021, 2023b; Yu et al. 2023; Nasr et al. 2025; Zhang et al. 2025; Chen, Han, and Miyao 2024). This setting differs from membership inference (Shokri et al. 2017; Carlini et al. 2022) and counterfactual memorization (Feldman and Zhang 2020; Zhang et al. 2023), which are more common with non-generative models. Although *extractability* has some limitations (Ippolito et al. 2023), it is fast to compute and widely used across different scenarios (Biderman et al. 2023a; Lee et al. 2023).

### Which Samples Are Memorized by LLMs?

Memorized samples mostly consist of samples that are difficult for the model to represent during training (Dentan et al.

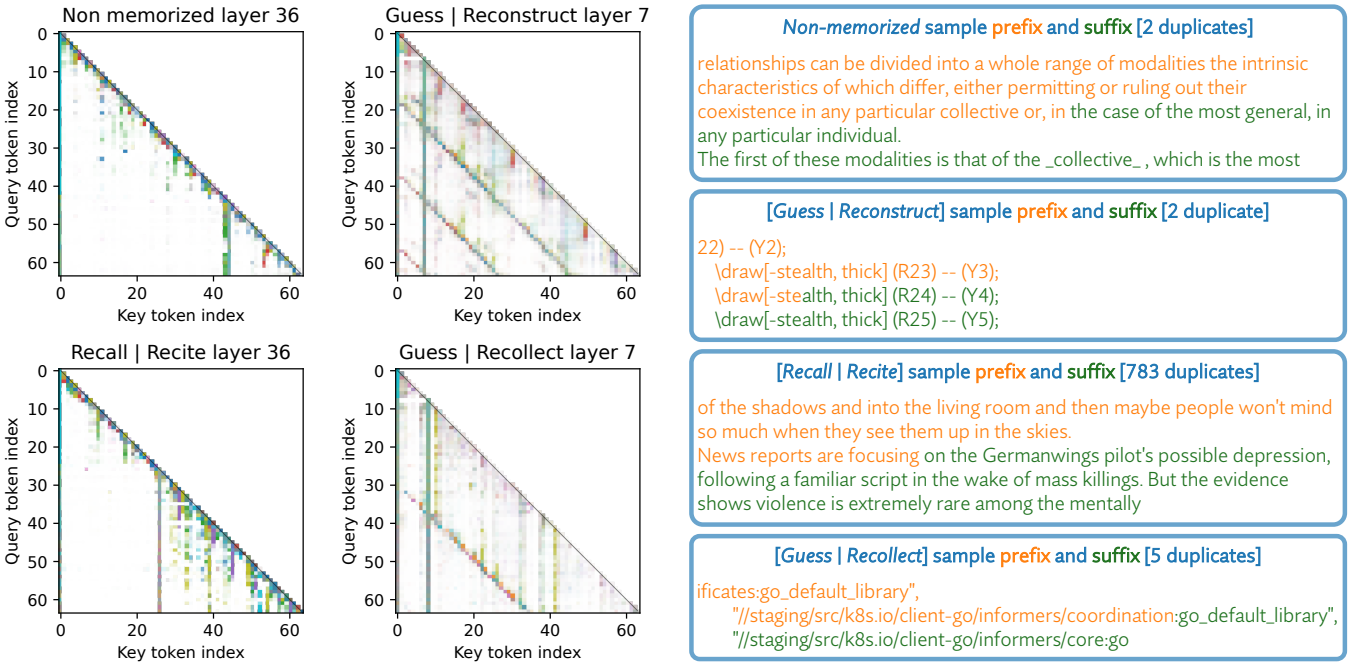



Figure 3: Sample attention weights and their corresponding 64-token text snippets. Labels like [*Guess* | *Reconstruct*] indicate the sample's class in our taxonomy (left) and in that of Prashanth et al. (2024) (right). The intensity of each color in the matrices represents the attention of a different head. The second and fourth samples exhibit similar patterns in lower-layer attention and are both classified as *Guess* in our taxonomy, though assigned to different classes by Prashanth et al. (2024).

2025; Feldman and Zhang 2020; Zhang et al. 2023). The extreme case consists of random sequences of tokens, which cannot be represented using language modeling abilities and are therefore very likely to be memorized (Meeus et al. 2024). However, most memorized samples are non-random natural language sentences, and Carlini et al. (2023b) have shown that LLMs memorize up to 1% of their training data. Prashanth et al. (2024) proposed a taxonomy of memorized samples, presented in Figure 1. Despite its limitations discussed in this paper, their framework captures the diversity of memorization forms and their underlying mechanisms.

### Localizing Memorization in LLMs

Verbatim memorization is deeply entangled with the general language abilities of LLMs (Huang, Yang, and Potts 2024). This entanglement explains why memorization and generalization can reinforce each other (Feldman 2020), and why techniques for localizing memorization resemble those used to localize factual knowledge in models (Meng et al. 2022). Stoehr et al. (2024) observed that memorized samples exhibit larger gradients in the lower layers and identified a specific attention head in these layers strongly associated with memorization. Huang, Yang, and Potts (2024) showed that certain token sequences in the prefix act as *triggers*. Their representations in the lower layers encode influential tokens in the suffix, and the model fills in the gaps using language modeling abilities. Unlike knowledge of a single fact, verbatim memorization of a full paragraph cannot be reduced to a single encoding at a specific point in the model. Instead, it is distributed across numerous triggers entangled with the model's general-purpose capabilities. Finally, Menta, Agrawal, and Agarwal (2025) demonstrated that deactivating attention blocks in the highest layers can reduce verbatim memorization while preserving performance.

Thus, there appears to be a gap between the role of the early layers highlighted by Huang, Yang, and Potts (2024) and Stoehr et al. (2024), and that of the final layers successfully used by Menta, Agrawal, and Agarwal (2025). Our experiments complement these works and resolve this gap by analyzing the role of attention blocks separately for each form of memorization.

## Taxonomy Benchmark: Methodology

### Training CNNs on Attention Weights

We consider *samples* of 64 contiguous tokens from The Pile (Gao et al. 2020), which is the training set used for Pythia models (Biderman et al. 2023b). The choice of Pythia and The Pile was determined by the need to know the complete set of training data, information that is omitted for most available models. We consider the complete set of 32-extractable samples from Pythia, provided by Biderman et al. (2023a) and refined by Prashanth et al. (2024), which enables us to make a fair comparison with their taxonomy.

For each sample $s$, we examine the *attention weight* at layers $l$ and attention head $h$, denoted by $\boldsymbol{A}_l^h[s] \in \mathbb{R}^{64\times 64}$. It is the triangular matrix containing at position $(i, j)$ the attention between *query* token $i$ and *key* token $j$ for $0 \leq j \leq i \leq 63$. See examples in Figure 3. Some aspects are easy to

interpret. The vertical bars spanning from the main diagonal to the bottom axis correspond to line breaks, which are crucial for locating a token's position in the text and therefore receive high attention. Similarly, diagonal lines in the second and fourth sample correspond to approximate repetition of subsequences, which exhibit strong mutual attention.

The most visible patterns in the attention weights are effectively translation-invariant: translating a subsequence of tokens also translates the underlying patterns. For example, we observed that moving an idiomatic expression modifies individual attention weights, but preserves the strong attention between its tokens. Similarly, adding tokens to a repeated sequence shifts the diagonal line in the attention weights without changing its nature. CNNs therefore appear to be a good choice of architecture, well suited for translation-invariant patterns. We train CNNs to classify attention weights into the classes of a given taxonomy. Our architecture consists of two convolutional layers with ReLU activations, dropout, and max-pooling, followed by two fully connected layers for classification. We also apply a layer-wise max-pooling and average-pooling over attention heads to focus on the most salient token-to-token interactions. Therefore, the CNNs have as many input channels as there are layers in the model. See implementation details and hyperparameters in the Supplement.[1]

## Evaluation Metric: Minimum F1 Score

The CNN's test performance reflects how well the taxonomy aligns with the attention weights. For example, the second and fourth samples in Figure 3 display similar diagonal patterns in the lower layers, leading the CNN to assign them the same class. A good taxonomy should group such samples together based on their shared patterns. Doing so reduces misclassifications and improves the CNN's test accuracy.

For each taxonomy, we randomly sample 4,000 training and 2,000 evaluation attention weights per class. This ensures balanced datasets, allowing us to assess the distinctive patterns of each memorization type regardless of its frequency. For each taxonomy, we train 8 CNNs with different hyperparameters (detailed in the Supplement). We also use 3 model sizes (Pythia 12B, 6.9B, and 2.8B) and evaluate the CNNs at 3 different steps (epochs 1, 2, and 3). We deliberately use limited data and training time to focus on taxonomies that are sufficiently salient in the attention weights to be learned quickly by the CNNs. This results in $2{,}000 \times 8 \times 3 \times 3 = 144{,}000$ test predictions per class and per taxonomy. We compute the precision, recall, and $F_1$ score for each class. To favor taxonomies that account for all forms of memorization and penalize those with one low-performing class, we focus on the minimum $F_1$ score across all classes and use it as our main evaluation metric.

## Parametrization of Taxonomies

To build a comprehensive benchmark of taxonomies, we developed a parametrization that allows exploration of a wide range of possible taxonomies. We model taxonomies as decision trees. In a taxonomy well-aligned with attention weights, each node should isolate samples that rely on meaningfully distinct memorization mechanisms. Based on prior work, we defined two families of nodes likely to influence the nature of memorization. The *duplication-based* nodes, defined in the first frame of Figure 4, separate samples using a duplication threshold $\boldsymbol{\delta}$, as duplication is known to influence memorization (Carlini et al. 2023b). The *completion-based* nodes, defined in the second frame, capture samples where most suffix tokens can be predicted from the prefix. The *Reconstruct* node matches the definition from Prashanth et al. (2024). *Guess*$[\boldsymbol{\lambda}, \boldsymbol{\gamma}]$ expands on it using ROUGE-based conditions to include more samples. Finally, we added a *Code* node, as the strict syntax of code strongly constrains the suffix. We also define rules to construct reasonable trees from these nodes and ensure that each class has a simple explanation. These rules are detailed in the Supplement and lead to the families presented in the last frame of Figure 4.

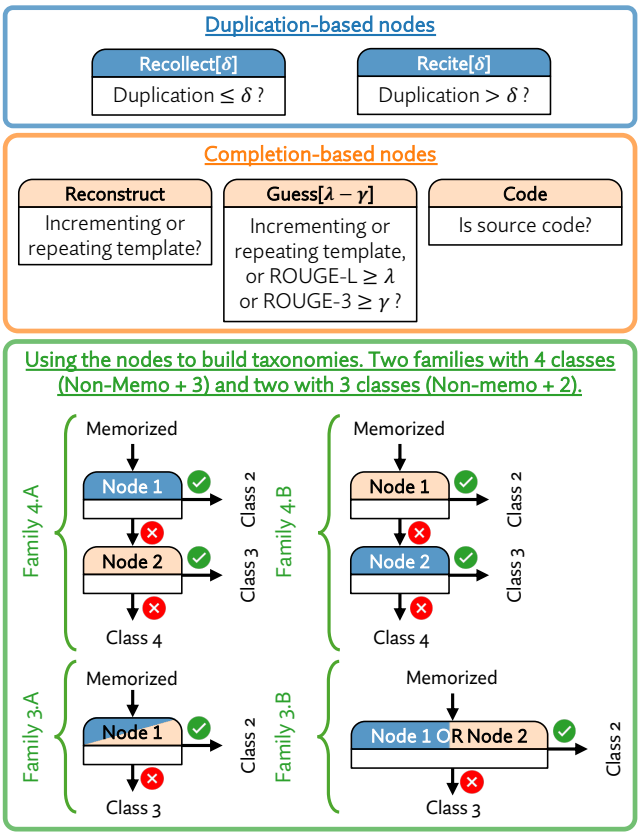



Figure 4: We parametrize taxonomies as decision trees with two types of nodes. We omit the *Non-Memorized* node at the root of each taxonomy, because memorized samples are always defined as 32-extractable sequences.

## Taxonomy Benchmark: Results

Our main empirical results are presented in Table 1 and Figure 5. We evaluate the 54 possible taxonomies with $\delta \in \{5, 50, 1000\}$ and $\lambda = \gamma = 0.5$. Taxonomies are denoted by their list of nodes, with $\boldsymbol{\delta}, \boldsymbol{\lambda}, \boldsymbol{\gamma}$ in brackets, and *Others*

[1]The Supplement corresponds to the Appendix of the extended version of this paper available at https://arxiv.org/abs/2508.02573

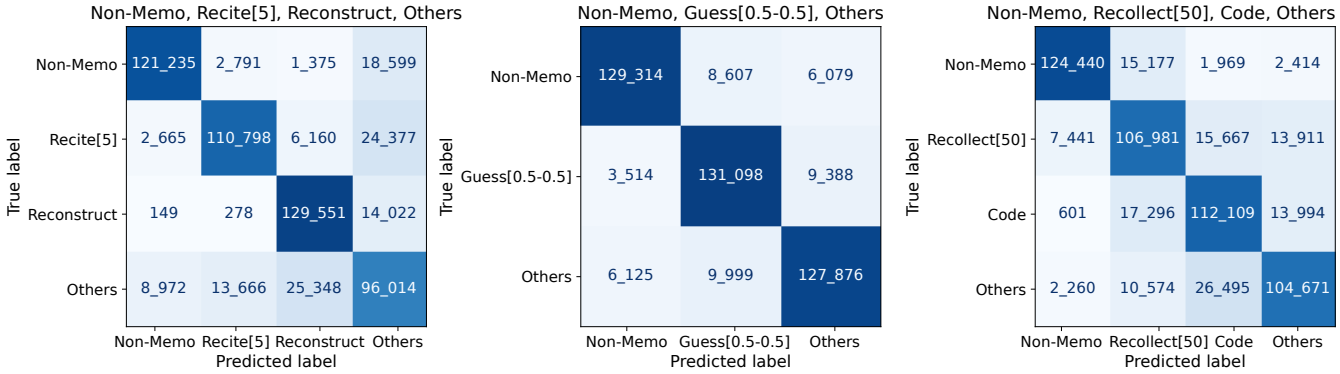



Figure 5: Confusion matrix for three taxonomies: Prashanth et al. (2024) (left), ours (middle), and the best 4-classes taxonomy (right, see Table 1). Datasets are balanced, with 144,000 attention weights in each class.

to refer to the remaining samples. For example, Prashanth et al. (2024)'s taxonomy is *Non-Memo, Recite[5], Reconstruct, Others* ; ours is *Non-Memo, Guess[0.5-0.5], Others*.

Table 1 presents the list of possible taxonomies ranked by descending performance, grouped by number of classes. For brevity, we show only the most relevant taxonomies; full results are available in the Supplement (Table 3). The taxonomy of Prashanth et al. (2024) performs poorly, with a minimum $F_1$ of 64.7%, well below the best 4-classes taxonomy, which achieves 72.8%. In contrast, the best 3-classes taxonomy outperform all others by a clear margin. Its confusion matrix shows very few misclassifications, indicating that its classes are well aligned with the attention weights.

To account for the increased difficulty of 4-classes classification, we normalize the $F_1$ score between a random predictor ($F_1^{\text{rand}} = 25\%$ or $33.3\%$) and a perfect one ($F_1^{\text{max}} = 100\%$) using: $F_1^{\text{norm}} = (F_1 - F_1^{\text{rand}})/(F_1^{\text{max}} - F_1^{\text{rand}})$. After normalization, the best 3-classes taxonomy reaches $F_1^{\text{norm}} = 83.6\%$, compared to 63.7% for the best 4-classes taxonomy. This supports the findings in Figure 5: the 4-classes taxonomy exhibits substantial misclassification, whereas the 3-classes taxonomy yields much cleaner separation. We therefore recommend the best 3-classes taxonomy, which we adopt as the data-driven taxonomy proposed in this paper: *Non-Memo, Guess[0.5-0.5], Others*.

## The Illusion of Few-shot Memorization

We observe that *Others* samples in Prashanth et al. (2024)'s taxonomy are often misclassified, with a $F_1$ score of 64.7%. These samples correspond to few-shot memorization (called *Recollect* in their work): they are supposedly memorized without being highly duplicated or without following a template. The numerous misclassification for this class demonstrate that it does not correspond to a distinct form of memorization. This aligns with the findings of Huang, Yang, and Potts (2024), which show that most samples believed to be few-shot memorized are either approximately duplicated in the dataset or follow templates not covered by the definition of *Reconstruct*. It also supports the observation that random canaries must be duplicated at least a few dozen times to be memorized (Meeus et al. 2024).

Moreover, the two highest-ranking taxonomies in Table 1 do not rely on duplication and outperform all others by a clear margin. This indicates that duplication does not trigger a distinct memorization mechanism, and there is no meaningful difference between the attention weights of a random canary and those of a software license duplicated 50 or 1000 times. While it is well established that duplication facilitates memorization (Carlini et al. 2023b), our experiments demonstrate that duplication is a necessary condition for verbatim memorization, but a high duplication rate does not qualitatively alter the nature of memorization.

| **Taxonomy name** | **Classes** | **Min $F_1$** |
|---|---|---|
| Non-Memo, Recollect[50], Code, Others | 4 | 72.8 |
| *10 lines with $64.7\% < F_1 < 72.8\%$ omitted* | 4 | — |
| ♦ **Non-Memo, Recite[5], Reconstruct, Others** | **4** | **64.7** |
| *15 lines with $F_1 < 64.7\%$ omitted* | 4 | — |
| ★ **Non-Memo, Guess[0.5-0.5], Others** | **3** | **89.0** |
| Non-Memo, Reconstruct, Others | 3 | 87.7 |
| *25 lines with $F_1 \leq 83.8\%$ omitted* | 3 | — |

Table 1: Taxonomy benchmark: Minimum $F_1$ across all categories for selected taxonomies. ♦ denotes Prashanth et al. (2024)'s taxonomy. We adopt as our taxonomy the highest-ranking one, denoted by ★. See full table in the Supplement.

## Impact of ROUGE Parameters

We use $\boldsymbol{\lambda} = \boldsymbol{\gamma} = 0.5$ to define the *Guess* class in our benchmark. This choice is intuitive, as it implies that half of the suffix tokens are constrained by the prefix. Since the highest-ranking taxonomy includes a *Guess* node, we investigated whether its performance could be further improved by optimizing $\boldsymbol{\lambda}$ and $\boldsymbol{\gamma}$. To that end, we evaluated the taxonomy *Non-Memo, Guess[$\boldsymbol{\lambda}$-$\boldsymbol{\gamma}$], Others* for $\boldsymbol{\lambda} = \boldsymbol{\gamma} \in \{0.1, 0.2, \ldots, 0.9\}$. We also tested $\boldsymbol{\lambda} = 1$ with $\boldsymbol{\gamma} \in \{0.1, 0.2, \ldots, 0.9\}$ (disabling the ROUGE-L condition), and vice versa. We found that optimizing $\boldsymbol{\lambda}$ and $\boldsymbol{\gamma}$ yields negligible improvements, increasing the minimum $F_1$ score by only 0.2% (see Table 4 in the Supplement). We therefore recommend using the most intuitive setting: $\boldsymbol{\lambda} = \boldsymbol{\gamma} = 0.5$.

## Impact of Model Size

Our results are averaged over three model sizes: Pythia 12B, 6.9B, and 2.8B. We also evaluated the taxonomies on each size separately. We found that our highest-ranking taxonomy also ranks highest for each size, confirming that its classes accurately capture distinct attention mechanisms that persist across model scales. See Tables 5-7 in the Supplement.

# Localizing Memorization

The optimal taxonomy derived from our benchmark is *Non-Memo, Guess[0.5-0.5], Others*. However, "Others" is not an intuitive label for samples that are memorized without being guessed. For simplicity, we now refer to these classes as *Non-Memo, Guess, Recall*, as in the Introduction. To demonstrate the benefits of studying *Guess* and *Recall* as distinct forms of memorization, we develop a custom interpretability technique to analyze the CNNs trained under this taxonomy and examine the regions of the attention weights that play a significant role in each form of memorization.

## Methodology

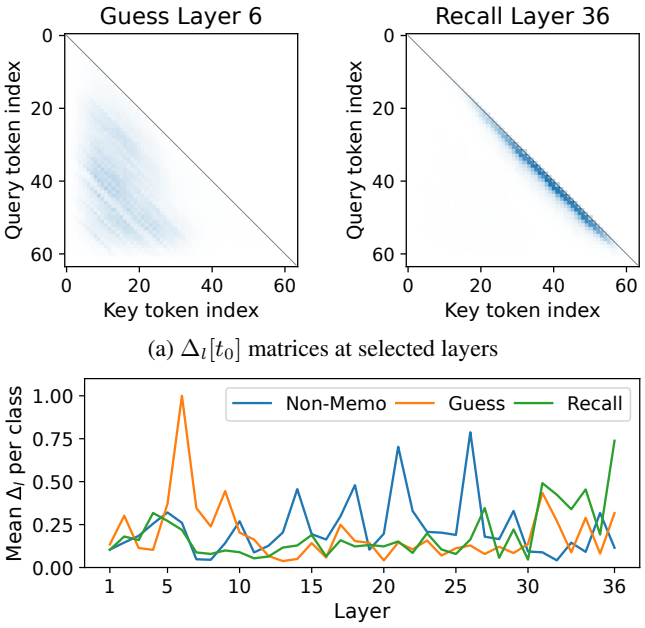
The CNNs have one input channel per LLM layer, leading to more numerous and more heterogeneous channels than typical computer vision settings (up to 36 channels capturing diverse patterns). As a result, standard explainability methods for CNNs, such as GradCAM (Selvaraju et al. 2017) lose their explainability capabilities under these conditions.

We therefore developed a custom interpretability technique. Consider a taxonomy with classes $\mathcal{U}_1, \mathcal{U}_2, \cdots, \mathcal{U}_N$ and a CNN $f$ trained on this taxonomy. We aim to compute matrices $\Delta_l[t_0] \in \mathbb{R}^{64\times64}$ for each class $t_0$ and layer $l$, representing the regions of the attention weights that contribute specifically to classifying samples in $\mathcal{U}_{t_0}$.

**Step 1: Guided Backpropagation.** Let $s \in \mathcal{U}_{t_0}$ be a sample with ground truth label $\mathcal{U}_{t_0}$. Let $\boldsymbol{A}_l^h[s]$ denote the attention weight at head $h$ and layer $l$. For each possible class $\mathcal{U}_{t_1}$, we apply Guided Backpropagation to sample $s$ with respect to that class (Springenberg et al. 2015). It is similar to computing the gradient of the logit for $\mathcal{U}_{t_1}$ with respect to $s$ using a backward pass, except that negative gradients are clipped to zero at each layer. This gives us $\boldsymbol{B}_l^h[s \to t_1] \in \mathbb{R}^{64\times64}$, which identifies the coordinates of the input that *could* contribute positively to class $\mathcal{U}_{t_1}$. Note that $\boldsymbol{B}_l^h[s \to t_1]$ can have positive values at positions where $\boldsymbol{A}_l^h[s]$ is zero.

**Step 2: Discriminative Classification** We then compute $\boldsymbol{C}_l^h[s] \in \mathbb{R}^{64\times64}$, which captures the positions of attention weights that can contribute positively to the correct class of $s$ but not to the others. The goal is to identify *discriminative* regions in the attention weights. For instance, the diagonal of $\boldsymbol{A}_l^h[s]$ is always highly activated, since each token attends to itself; however, because this is not a discriminative feature, we aim to disregard it.

$$\boldsymbol{C}_l^h[s] = \boldsymbol{B}_l^h[s \to t_0] - \frac{1}{N-1} \sum_{t_1 \neq t_0} \boldsymbol{B}_l^h[s \to t_1]$$




(a) $\Delta_l[t_0]$ matrices at selected layers

(b) Mean value of $\Delta_l[t_0]$ across layers

Figure 6: We compute $\Delta_l[t_0]$ for every class $t_0$ and layer $l$ using the CNNs trained on Pythia 12B attention weight under our optimal taxonomy: *Non-Memorized, Guess, Recall*. Figure 6a shows $\Delta_l[t_0]$ for selected layers and classes (see all layers in the Supplement). Figure 6b presents the decisiveness of each layer for classification under this taxonomy.

**Step 3: Clipping and Normalization** Then, we compute $\boldsymbol{D}_l^h[s] \in \mathbb{R}^{64\times64}$ by clipping negatives values in $\boldsymbol{C}_l^h[s]$ to focus only on positive influence, and normalize by the maximal value across all heads $h$, layers $l$, and positions $i, j \in [\![0, 63]\!]$. Note that the second max is element-wise.

$$\boldsymbol{D}_l^h[s] = \frac{1}{\max\limits_{l,h,i,j} \left[\boldsymbol{C}_l^h[s]\right]_{i,j}} \times \max(\boldsymbol{C}_l^h[s], 0)$$

**Step 4: Activations and Averaging** Finally, we multiply by the activations $\boldsymbol{A}_l^h[s]$ to identify the positions that *actively* influence class $\mathcal{U}_{t_0}$ for input $s$. We apply layer-wise max-pooling over heads to focus on the most salient activations. Then, we average over all samples in $\mathcal{U}_{t_0}$ and all CNNs trained under this taxonomy. This yields $\Delta_l[t_0] \in \mathbb{R}^{64\times64}$, indicating the regions of the attention weights that *consistently* influence that class. For simplicity, we omit the average over the CNNs in the notation.

$$\Delta_l[t_0] = \frac{1}{|\mathcal{U}_{t_0}|} \sum_{s \in \mathcal{U}_{t_0}} \max_h \left[\boldsymbol{D}_l^h[s] \times \boldsymbol{A}_l^h[s]\right]$$

## Empirical Results

**Syntactic, Low-Layer Connections for *Guess*** Our main results are presented in Figure 6. We observe that the lower layers of the LLM contribute significantly to memorizing *Guess* samples. The mean value of $\Delta_l[\texttt{Guess}]$ is high in

these layers, peaking at layer 6. At that layer, $\Delta_6[\texttt{Guess}]$ displays typical diagonal patterns in the first half of the matrix (key token index $\leq$ 31). Layers 7–9 show similar behavior (see Figure 17 in the Supplement). As in the example from Figure 3, these diagonal segments reflect direct causal links between the prefix and the suffix, such as repetitions.

In light of recent studies highlighting the role of higher layers, such as Menta, Agrawal, and Agarwal (2025) and Dentan et al. (2025), the significant contribution of lower layers to memorization was unexpected. We interpret this by noting that the *Guess* class primarily captures low-level syntactic dependencies that do not require complex token interactions and can emerge in the earliest layers of the LLM. Similarly, Stoehr et al. (2024) observed that some lower attention heads contribute substantially to memorization. Upon analyzing their dataset, we found that 54% of the samples are code snippets. As for the *Guess* class, we interpret their findings as evidence of low-level dependencies between the prefix and the suffix due to syntactic regularity of code snippets. Conversely, as we will show, other forms of memorization rely more heavily on higher layers.

**Short-Range, High-Layer Connections for *Recall*** The mean value of $\Delta_l[\texttt{Recall}]$ is particularly high in the higher layers of the model, peaking at the final layer. At that layer, $\Delta_{36}[\texttt{Recall}]$ exhibits strong activation just *below* the main diagonal. Layers 31–35 show similar behavior (see Figure 13 in the Supplement). The fact that *Recall* relies on attention weights just *below* the diagonal suggests that the model uses the few preceding tokens to complete each token. This corroborates the findings of Huang, Yang, and Potts (2024), who show that only a few tokens from a memorized sample are encoded by *triggers* in the prefix (see their Figure 4). They explain that the remaining tokens are inferred using language modeling capabilities. Our results suggest that these capabilities are localized in the activations below the diagonal in $\Delta_l[\texttt{Recall}]$ for $l \geq 31$. This indicates that the model relies on short-range connections in the final layers, which are highly correlated with the output, to fill in the gaps between the tokens encoded by the *triggers*.

Importantly, our results suggest that *Non-Memo* relies on different layers than *Recall*. We observe that the mean value of $\Delta_l[\texttt{Non-Memo}]$ is significantly higher in the intermediate layers of the LLM. This indicates that these attention blocks play a major role in the model's general-purpose capabilities but contribute little to memorizing *Recall* samples. This observation helps explain why Menta, Agrawal, and Agarwal (2025) were able to reduce memorization while preserving overall performance by deactivating the final attention blocks of the model. As we have shown, these upper layers, which are highly correlated with the output, are crucial to fill in the gaps between memorized tokens for *Recall*, but they appear to be less essential for the general-purpose abilities involved in decoding *Non-Memo* samples.

## Limitations and Future Work

**Datasets and Models** A primary limitation of our work is that all experiments were conducted on models from the same family: Pythia 12B, 6.9B, and 2.8B (Biderman et al. 2023b), using their training dataset, the Pile (Gao et al. 2020). This choice was driven by the necessity of having access to the full training data of the analyzed models, which is unavailable for most open-source models. As a result, memorization research typically focuses on either GPT-NeoX (Black et al. 2021) or Pythia. We selected Pythia because it is used in the existing taxonomy we compare against (Prashanth et al. 2024), but evaluating our approach on other models remains a promising direction for future work. To partially address this limitation, we performed experiments across multiple Pythia model sizes (see Section *Impact of model size*). The consistent results observed across scales provide an incomplete but encouraging indication of the generality of our findings.

**A Focus on Attention Blocks** Our approach focuses exclusively on the model's attention weights, disregarding the role of feed-forward blocks. This choice is motivated by two factors. First, the role of feed-forward layers has already been investigated in prior work (Huang, Yang, and Potts 2024). Second, we chose to concentrate on attention blocks to analyze the causal links between the prefix and the suffix that are essential for verbatim memorization. Nonetheless, future work could explore an evaluation of taxonomies that incorporates both attention and feed-forward blocks.

**An Indirect Localization** Finally, the explainability method we developed allows us to localize memorization indirectly by analyzing CNNs trained on attention weights. However, it would be valuable to correlate our findings with direct observations, such as ablations or perturbations of the attention weights. Similar ablations have been performed by Menta, Agrawal, and Agarwal (2025), and applying them separately to each form of memorization would be an interesting direction for future work.

## Conclusion

We show that existing taxonomies proposed in the literature for memorization in Large Language Models do not align with the attention mechanisms underlying verbatim memorization. To address this gap, we introduce a systematic approach for exploring and evaluating a broad set of candidate taxonomies. Based on this approach, we propose a new data-driven taxonomy that significantly outperforms all others: *Non-Memorized, Guess, Recall*. We then developed a custom method to localize the regions of the attention weights that are critical for each of these forms of memorization.

Our results corroborate and extend several recent findings in the memorization literature. We confirm the illusion of verbatim memorization by showing that duplication does not induce a distinct form of memorization. We demonstrate that a significant proportion of samples are *guessed* by the model using syntactic dependencies and highlight the importance of lower layers for that mechanism. Finally, we show that the language modeling abilities involved in memorization differ from the model's general-purpose abilities. They reside in short-range connections in the latest layers of the model and control the exact decoding of memorized tokens. These findings underscore the importance of studying each form of memorization separately in future research.

## Ethical Statement

While our work advances the understanding of memorization in LLMs, it will benefit privacy researchers more than attackers, for several reasons. First, our experiments are conducted on public datasets, which are of no interest to an attacker. Moreover, we focus solely on analyzing memorization without introducing new attack methods. On the contrary, our findings can help develop more effective mitigation strategies.

To ensure reproducibility and to support further research on LLM memorization, we release all scripts needed to reproduce our experiments. Implementation details are provided in the Supplement, corresponding to the Appendix of the extended version of this paper available on ArXiv.

## Acknowledgements

This work received financial support from Crédit Agricole SA through the research chair "Trustworthy and Responsible AI" with École Polytechnique. This work was granted access to the HPC resources of IDRIS under the allocation 2023-AD011014843 made by GENCI. Finally, we thank Mohamed Dhouib and Mathis Le Bail discussions on early versions of this paper.

## References

Biderman, S.; Prashanth, U. S.; Sutawika, L.; Schoelkopf, H.; Anthony, Q.; Purohit, S.; and Raff, E. 2023a. Emergent and Predictable Memorization in Large Language Models. In *NeurIPS*, volume 36, 28072–28090.

Biderman, S.; Schoelkopf, H.; Anthony, Q.; Bradley, H.; O'Brien, K.; Hallahan, E.; Khan, M. A.; Purohit, S.; Prashanth, U. S.; Raff, E.; Skowron, A.; Sutawika, L.; and van der Wal, O. 2023b. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. ArXiv:2304.01373.

Black, S.; Leo, G.; Wang, P.; Leahy, C.; and Biderman, S. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Carlini, N.; Chien, S.; Nasr, M.; Song, S.; Terzis, A.; and Tramèr, F. 2022. Membership Inference Attacks From First Principles. In *IEEE S&P*, 1897–1914.

Carlini, N.; Hayes, J.; Nasr, M.; Jagielski, M.; Sehwag, V.; Tramèr, F.; Balle, B.; Ippolito, D.; and Wallace, E. 2023a. Extracting Training Data from Diffusion Models. In *USENIX Security*.

Carlini, N.; Ippolito, D.; Jagielski, M.; Lee, K.; Tramer, F.; and Zhang, C. 2023b. Quantifying Memorization Across Neural Language Models. In *ICLR*.

Carlini, N.; Tramer, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T. B.; Song, D.; Erlingsson, U.; Oprea, A.; and Raffel, C. 2021. Extracting Training Data from Large Language Models. In *USENIX Security*.

Chen, B.; Han, N.; and Miyao, Y. 2024. A Multi-Perspective Analysis of Memorization in Large Language Models. In *EMNLP*, 11190–11209.

Dentan, J.; Buscaldi, D.; Shabou, A.; and Vanier, S. 2025. Predicting Memorization Within Large Language Models Fine-Tuned for Classification. In *ECAI*.

Dentan, J.; Paran, A.; and Shabou, A. 2024. Reconstructing training data from document understanding models. In *USENIX Security*, 6813–6830.

Feldman, V. 2020. Does learning require memorization? a short tale about a long tail. In *ACM SIGACT STOC*, 954–959.

Feldman, V.; and Zhang, C. 2020. What Neural Networks Memorize and Why: Discovering the Long Tail via Influence Estimation. In *NeurIPS*.

Fredrikson, M.; Lantz, E.; Jha, S.; Lin, S.; Page, D.; and Ristenpart, T. 2014. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. In *USENIX Security*.

Gao, L.; Biderman, S.; Black, S.; Golding, L.; Hoppe, T.; Foster, C.; Phang, J.; He, H.; Thite, A.; Nabeshima, N.; Presser, S.; and Leahy, C. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. ArXiv:2101.00027.

Huang, J.; Yang, D.; and Potts, C. 2024. Demystifying Verbatim Memorization in Large Language Models. In *EMNLP*, 10711–10732.

Ippolito, D.; Tramer, F.; Nasr, M.; Zhang, C.; Jagielski, M.; Lee, K.; Choquette Choo, C.; and Carlini, N. 2023. Preventing Generation of Verbatim Memorization in Language Models Gives a False Sense of Privacy. In *INLG*, 28–53.

Lee, J.; Le, T.; Chen, J.; and Lee, D. 2023. Do Language Models Plagiarize? In *ACM WWW*, 3637–3647.

Mahloujifar, S.; Inan, H. A.; Chase, M.; Ghosh, E.; and Hasegawa, M. 2021. Membership Inference on Word Embedding and Beyond. ArXiv:2106.11384.

Meeus, M.; Shilov, I.; Faysse, M.; and de Montjoye, Y.-A. 2024. Copyright Traps for Large Language Models. In *ICML*.

Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2022. Locating and Editing Factual Associations in GPT. In *NeurIPS*.

Menta, T. R.; Agrawal, S.; and Agarwal, C. 2025. Analyzing Memorization in Large Language Models through the Lens of Model Attribution. In *NAACL:HLT*, 10661–10689.

Mireshghallah, F.; Uniyal, A.; Wang, T.; Evans, D.; and Berg-Kirkpatrick, T. 2022. An Empirical Analysis of Memorization in Fine-tuned Autoregressive Language Models. In *ACL-EMNLP*, 1816–1826.

Nasr, M.; Carlini, N.; Hayase, J.; Jagielski, M.; Cooper, A. F.; Ippolito, D.; Choquette-Choo, C. A.; Wallace, E.; Tramèr, F.; and Lee, K. 2025. Scalable Extraction of Training Data from (Production) Language Models. In *ICLR*.

Prashanth, U. S.; Deng, A.; O'Brien, K.; V, J. S.; Khan, M. A.; Borkar, J.; Choquette-Choo, C. A.; Fuehne, J. R.; Biderman, S.; Ke, T.; Lee, K.; and Saphra, N. 2024. Recite, Reconstruct, Recollect: Memorization in LMs as a Multifaceted Phenomenon. In *ICLR*.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In *IEEE ICCV*.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership Inference Attacks against Machine Learning Models. In *IEEE S&P*.

Springenberg, J. T.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. 2015. Striving for Simplicity: The All Convolutional Net. ArXiv:1412.6806.

Stoehr, N.; Gordon, M.; Zhang, C.; and Lewis, O. 2024. Localizing Paragraph Memorization in Language Models. ArXiv:2403.19851.

Yu, W.; Pang, T.; Liu, Q.; Du, C.; Kang, B.; Huang, Y.; Lin, M.; and Yan, S. 2023. Bag of Tricks for Training Data Extraction from Language Models. In *ICML*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization. In *ICLR*.

Zhang, C.; Ippolito, D.; Lee, K.; Jagielski, M.; Tramèr, F.; and Carlini, N. 2023. Counterfactual Memorization in Neural Language Models. In *NeurIPS*.

Zhang, J.; Zhao, Q.; Li, L.; and Lin, C.-h. 2025. Extending Memorization Dynamics in Pythia Models from Instance-Level Insights. ArXiv:2506.12321.