MALTO at SVELA: A Specific-Attention-Head Approach for Membership Inference Attacks in LLMs Unlearning Evaluation

Evren A. Munis[1], Mattia Sabato[1], Erfan Bayat[1] and Andrea Lolli[1]

[1]Politecnico di Torino, Italy

Abstract

Sensitive information in Large Language Models (LLMs) poses security risks, as such content may persist after training. Unlearning algorithms therefore aim to remove specific information without retraining the entire model. Evaluating the effectiveness of unlearning is thus essential for assessing model safety. In this paper, we propose a novel Membership Inference Attack (MIA) that predicts whether a data point belongs to the retain, forget, or unseen set. Our method leverages features extracted at four levels: layer-level attention, logits, layer-transition and attention-head information derived from a membership-specific selection of attention heads. These features are used to train a MLP classifier for membership prediction. Our approach achieves a macro F1 score of 0.336 and 0.323 for the 1B model and 0.353 and 0.323 for the 3B model, on entity-level and instance-level classification, respectively. We find that the number of selected heads influences performance across task types, model sizes, and classifier architectures. The effectiveness of the proposed MIA varies across membership classes and unlearning methods, reflecting differences in their underlying mechanisms. In addition, we observed that, under feature ablation, the location of membership-related information differs across entity-level and instance-level unlearning tasks, as well as across model scales, influencing which features are most informative for classification.

Keywords

unlearning, membership inference attack, attention, entropy, natural language processing, deep learning, large language models

1. Introduction

LLMs have become widely used in recent years. However, their training data may include personal information, sensitive content [1], or copyrighted material [2], which could be reproduced by the models resulting in important privacy risks. Because training an LLM from scratch is expensive, Machine Unlearning (MU) has become a key research area: it aims to remove specific data or behaviors from a trained model, without retraining from scratch, while preserving its general knowledge. Recent unlearning approaches include methods based on gradient ascent, KL-divergence minimization [3], preference optimization [3], and the newer adaptation provided by the dual-teacher style [4]. Since unlearning can change a model in unpredictable ways, with no a priori guarantees of satisfactory information removal, it is crucial to verify reliability after unlearning and to measure whether the model has truly forgotten the targeted data, following evaluation protocols that jointly assess the three main axis of unlearning: efficacy, utility, and efficiency, as proposed in recent benchmarking efforts for MU [5]. In this work, we focus specifically on the efficacy dimension, analyzing whether the unlearning procedure effectively removes the targeted information from the model.

To support this goal, the Selective Verification of Erasure from LLM Answers [6] challenge (SVELA @ EVALITA 2026 [7]) investigates quantitative evaluation methods for checking whether data has been forgotten by predicting the membership of an input sample whether it belongs

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 27, Bari, IT

evrenayberk.munis@studenti.polito.it (E. A. Munis); mattia.sabato@studenti.polito.it (M. Sabato); erfan.bayat@studenti.polito.it (E. Bayat); andrea.lolli@studenti.polito.it (A. Lolli)

to the forget, retain, or test set. The forget set contains the data that should be removed, the retain set contains data that should remain, and the test set contains unseen samples. The evaluation is performed across models of different sizes and different unlearning techniques to better reflect real-world settings.

This work[1] extends a prior mechanistic interpretability-based MIA by using data-specific attention heads. First, the most responsive heads are selected separately for the forget, retain, and test splits using the LAHIS scores [8], adapted to each split. For these heads, head-level features are computed. In addition, layer-level attention features, layer transition information, and logit features are extracted. Lastly, all extracted features are used for training an MLP classifier to classify membership of input data. Focusing on the most responsive heads also improves interpretability by revealing which parts of the attention mechanism are most involved in forgetting.

2. Related Work

Training data leakage is a serious concern when models are trained on sensitive, private, or copyrighted content. MIAs provide a formal framework for evaluating whether an adversary can determine if a specific example was included in a models training set. A common baseline approach, introduced by Shokri et al. [9], trains shadow models on data drawn from a distribution similar to that of the target model. An attacker then trains an attack model using the output probability vectors of the shadow models to predict whether a given input is a member (seen during training) or a non-member. Related membership-based methods have also been applied to assess MU by testing whether a model continues to behave as if examples from the forget set were part of its training data. In the following subsections, we first review adaptations of MIA techniques for large language models and then discuss the emerging role of these attacks as a framework for evaluating MU performance.

2.1. MIA for LLMs

MIAs can also be adapted to LLMs. Traditional approaches include loss-based attacks, which rely solely on the loss of a target example under the model, and neighborhood-based attacks, which generate semantically similar texts and compare their losses with that of the target input [10]. Another widely used method is Min-k% scoring [11], which aggregates the lowest token likelihoods to produce a membership score. Reference-based methods have also been proposed, in which membership is inferred from the loss difference between the target model and a separate reference model. More recent work introduces stronger MIA variants. MIATuner [12] prompts the model to self-assess whether an input resembles its pretraining data. MemTrace [13] is a white-box attack that exploits internal signals, such as hidden states and attention patterns, to infer whether an example was seen during training.

In this work, we extend MemTrace by introducing a membership inference variant that explicitly targets specific attention heads. Following Liu et al. [8], we identify informative heads using a single forward and backward pass but, differently from the authors, which select heads based on language, we leverage this approach to identify heads that are the most relevant for distinguishing retain, forget, and test data.

2.2. MIA For Verifying Unlearning

MIAs can also be used to verify MU in LLMs by detecting residual memorization of data intended to be removed and by quantifying the quality of forgetting. For example, in the SemEval 2025 unlearning task [14], a loss-based MIA is employed to assess data leakage after unlearning,

[1]The code to replicate the experiments can be found at https://github.com/MAL-TO/SVELA-2026

Name	Language	Topic ID	Question
Santino Lucarelli	IT	biography.birthplace	Dove è nato Santino Lucarelli?
Paula Vallés Bravo	ES	personal.address	¿Cuál es la dirección de Paula Vallés Bravo?
Günter Beier	DE	career.notable_award	Welchen bedeutenden Preis hat Günter Beier erhalten?
Valérie Briand	FR	career.international_project	Quel est un projet international notable de Valérie Briand ?

Table 1
Example data from the FAME dataset for different topics in different languages.

following an approach similar to Duan et al. [15]. Similarly, FUMA [16] is a MIA-based method designed to identify unlearned inputs without requiring access to the full forget set, enabling more practical verification of forgetting in constrained settings.

3. Challenge Description

3.1. Dataset

FAME [17] is a recently introduced synthetic dataset designed to assess MU. It consists of synthetic individual identities in a multilingual setting, covering five languages Italian, Spanish, French, German and English (the last not used in the challenge), with 200 identities per language (1,000 unique identities in total). For each identity, the dataset provides 20 question-answer (QA) pairs, yielding 20,000 pairs overall. The questions reflect real-world scenarios and span topics such as career, biography, achievements, and personal information. Example QA pairs are shown in Table 1.

For Task 1, the training split consists of 912 QA pairs, comprising 576 retain, 144 forget, and 196 unseen pairs; the remaining pairs are allocated to the validation and test splits. For Task 2, the training split also contains 912 QA pairs (distinct from those in Task 1), with the same composition of 576 retain, 144 forget, and 196 unseen, while the remaining pairs are likewise assigned to validation and test. The identities of the test set, never shared by the organizers, have zero intersection with the identities in the training split available to the participants.

3.2. Models

The organizers have fine-tuned Llama-3 models [18], specifically Llama-3-1B and Llama-3-3B on the full dataset for each task. Several unlearning techniques have been applied, including gradient difference [19], fine-tuning the original model with only the retain set, KL minimization [3], gradient ascent [20], and preference optimization [3]. Detailed information about the models is given in Savelli et al. [17]. Each unlearning method has been evaluated on both model sizes.

3.3. Tasks

Task 1 In Task 1, unlearning techniques are applied to remove entire identities and all associated information from the dataset. The objective is to determine whether each identity belongs to the retain, forget, or unseen set. The overall score is computed as the mean macro F1 score, averaged across all unlearning techniques for all model sizes.

Task 2 In Task 2, unlearning techniques are applied to remove specific pieces of information associated with an identity rather than the identity as a whole. For example, an identitys email address may be included in the forget set, while the same identitys career information remains in the retain set. Unlike Task 1, where all information about an identity is unlearned, Task 2

requires the model to forget only selected instances related to an identity. The goal is to predict whether a QA pair belongs to the retain, forget, or unseen set. The overall score is computed as the mean macro F1 score, averaged across all unlearning techniques for model sizes.

4. Methodology

Prior work has examined how attention heads specialize for different tasks, such as machine translation [21] and vision-language modeling [22], and has identified language-specific heads [8] to better understand model internals. However, similar analyses in the unlearning setting remain limited [23]. Motivated by evidence that certain attention heads encode informative signals [21, 8] and building on prior work on head specialization and entropy-based membership inference, we propose a MIA tailored to the unlearning setting. Our approach selects heads most informative for distinguishing retain, forget, and unseen examples and couples this head-selection mechanism with membership-conditioned feature extraction from the models hidden states.

The proposed method proceeds in two stages. First, attention heads that best separate retain, forget, and unseen samples are selected using the head-selection procedure of Liu et al. [8], originally introduced for language-based head selection. Second, layer-wise and head-wise features, derived from the selected heads, as well as layer transition and logit features, are computed following Makhija et al. [13], and these are used as inputs to an MLP classifier that predicts whether a sample belongs to the retain, forget, or unseen set.

4.1. Selecting Membership Heads

Since estimating each heads contribution by disabling heads one at a time is computationally expensive, we instead adopt a trainable matrix approach as in Liu et al. [8]. Let $M \in \mathbb{R}^{n_l \times n_h}$ denote the soft attention head mask matrix, where n_l is the number of layers and n_h is the number of attention heads. This mask allows us to score the importance of all heads using a single forward and backward pass per mini-batch.

The importance of each head is approximated via a first-order Taylor expansion. Let X_c denote the membership-specific subset corresponding to class $c \in \{\text{retain}, \text{forget}, \text{unseen}\}$. For each attention head h_i, let m_i be the corresponding entry in M. The expected change in loss from disabling head h_i is approximated as:

$$\Delta\tilde{\mathscr{L}} = \mathbb{E}_{x_c \in \mathcal{X}_c}\left[\left| m_i \cdot \frac{\partial \mathscr{L}(x_c)}{\partial m_i} \right|\right] \tag{1}$$

To quantify how often removing a head is expected to increase the loss, we compute the proportion of negative gradients with respect to m_i. First, for a given membership class c, we average the gradients over a batch B:

$$g_i^{(c)} = \frac{1}{|B|} \sum_{x \in B} \frac{\partial \mathscr{L}(x)}{\partial m_i} \tag{2}$$

Next, we define an indicator function for negative gradients:

$$W_{\text{neg}} = \mathbb{E}_{x_c \in \mathcal{X}_c}\left[\mathbb{I}\left(g_i^{(c)} < 0\right)\right] \tag{3}$$

Finally, the importance matrix $\text{Importance}_c(h_i) \in \mathbb{R}^{n_l \times n_h}$ for each membership class is computed by weighting the loss difference $\Delta\tilde{\mathscr{L}}$ with W_{neg}, yielding:

$$\text{Importance}_c(h_i) = \mathbb{E}_{x_c \in \mathcal{X}_c}\left[\left| m_i \cdot \frac{\partial \mathscr{L}(x_c)}{\partial m_i} \right| \cdot \mathbb{I}\left(g_i^{(c)} < 0\right)\right] \tag{4}$$

After computing the importance scores, attention heads are sorted in descending order, and the five most important heads are selected for each layer and each membership class.

4.2. Feature Extraction

Layer-Level Attention Features We compute three layer-level attention features following the procedure described in Makhija et al. [13], with the goal of extracting general, layer-level signals from attention patterns. For each layer, we first obtain attention weights for all heads and average them across heads to form the mean attention matrix $\bar{A}^l \in \mathbb{R}^{n \times n}$, where l indexes the layer and n is the sequence length. Similarly, $\bar{a}^l_{t,s}$ denotes the mean attention weight from the token at position t to the token at position s in layer l. We then compute the attention entropy as $H^l = \frac{1}{n}\sum_{t=1}^{n}\left(-\sum_{s=1}^{n}\bar{a}^l_{t,s}\log_2\left(\bar{a}^l_{t,s}+\epsilon\right)\right)$, where ϵ is a small constant added for numerical stability. Next, we quantify how strongly the model attends to particular tokens using attention concentration $C^l = \frac{1}{n}\sum_{t=1}^{n}\max_s \bar{a}^l_{t,s}$, and we measure the degree of selective information routing via attention sparsity $S^l = \text{mean}\left(\bar{A}^l < \tau\right)$, where τ is the global mean of $\bar{A}^l$. In addition, patterns associated with token positions are analyzed for each layer's attention pattern. First, the previous-token bias quantifies the models tendency to attend to the immediately preceding token and is defined as $\frac{1}{n-1}\sum_{t=2}^{n}\bar{a}^l_{t,t-1}$, where n is the sequence length and $\bar{a}^l_{t,t-1}$ denotes the attention weight from the token at position t to the token at position $t-1$. Next, mean self-attention measures the average attention weight each token allocates to itself, computed as $\frac{1}{n}\sum_{t=1}^{n}\bar{A}^l_{t,t}$, where $\bar{A}^l_{t,t}$ is the diagonal entry of the mean attention weight matrix at layer l. Finally, mean attention distance quantifies the average distance between tokens that the model attends to as $\frac{\sum_t\sum_s |t-s|, a^l_{t,s}}{\sum_t\sum_s a^l_{t,s}}$ where $a^l_{t,s}$ denotes the attention weight from position t to position s. All metrics are computed per batch, and batch averages are used as features.

Head-Level Attention Features We compute head-level features to analyze model behavior in greater detail, as certain attention heads can carry more informative signals than layer-wise aggregates. Unlike the original approach [13], we use the selected heads in each layer described in the previous section rather than all heads. For each head h, we define the head entropy as $h_{\text{entropy}} = -\frac{1}{n}\sum_{t=1}^{n}\sum_{s=1}^{n}A^h_{t,s}\log_2\left(A^h_{t,s}\right)$, where $A^h_{t,s}$ is the attention weight from token t to token s in head h. This metric reflects whether a head concentrates its attention on a few tokens or distributes it broadly across the sequence. We also compute head focus as $h_{focus} = \frac{1}{n}\sum_{t=1}^{n}\max_s a^{l,h}_{t,s}$, which measures how strongly a head attends to a single token at each position. Both entropy and focus are computed for each membership class (retain, forget, and unseen) per batch, and we use their batch averages and standard deviations as features. In contrast to prior work, we additionally include class-wise average differences in these head-level features, capturing the separation between membership classes and providing more discriminative signals for membership inference.

Layer Transition Features We analyze the similarity between hidden states of consecutive layers using three complementary measures to quantify how information evolves across layers. First, transition surprise is computed as $\left\|\mathbf{h}_t^{(i+1)} - \mathbf{h}_t^{(i)}\right\|_2$, where i indexes the layer, t indexes the token position and $\mathbf{h}$ represents the hidden states. For each layer, the minimum, maximum, mean, and standard deviation across token positions are extracted as features. Second, normalized transition surprise is computed to capture directional changes, and its mean and standard deviation across tokens are used as layer-level features. Third, the cosine similarity between hidden states of consecutive layers is calculated for each token position as $\frac{\mathbf{h}_t^{(i)}\cdot\mathbf{h}_t^{(i+1)}}{\|\mathbf{h}_t^{(i)}\|\|\mathbf{h}_t^{(i+1)}\|}$ measuring how much representations differ between layers. For each layer, the mean, standard deviation, minimum, and maximum of these values across token positions are used as features.

Logit-Based Features We assess the uncertainty of the models output by deriving several features from the logits. First, we convert logits into probabilities using the softmax function, $p_{t,v} = \text{softmax}(\ell_{t,v})$, where t denotes the token position and v the vocabulary index. From these

probabilities, we compute token-level entropy, $H_t = -\sum_{v=1}^{V} p_{t,v} \log p_{t,v}$, confidence, $c_t = \max_v p_{t,v}$, and the confidence gap, $g_t = p_t^{(1)} - p_t^{(2)}$, defined as the difference between the highest and second-highest probabilities at each token. Finally, we calculate the mean and standard deviation of entropy, confidence, and confidence gap across the sequence, and use these aggregated statistics as input features.

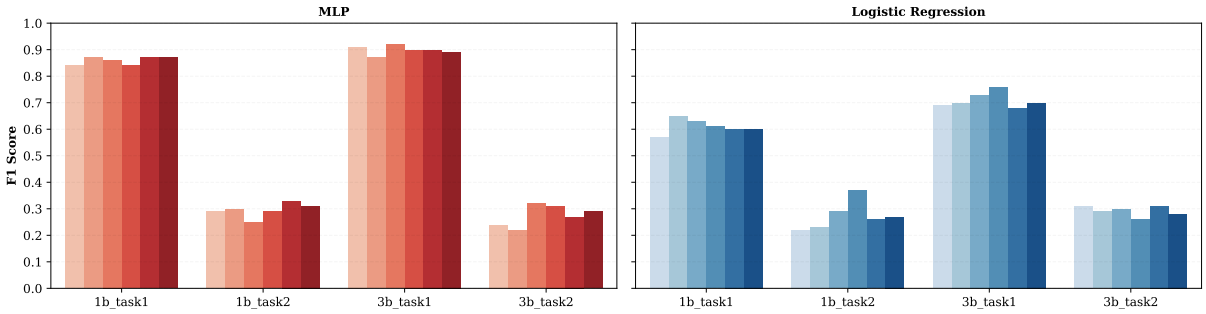

Figure 1: Head selection on base models (random train/test split). F1 score for MLP and Logistic Regression across tasks and Llama-3 sizes. Shaded bars indicate increasing numbers of retained heads, from 1 (lightest) to 11 (darkest), in steps of two.

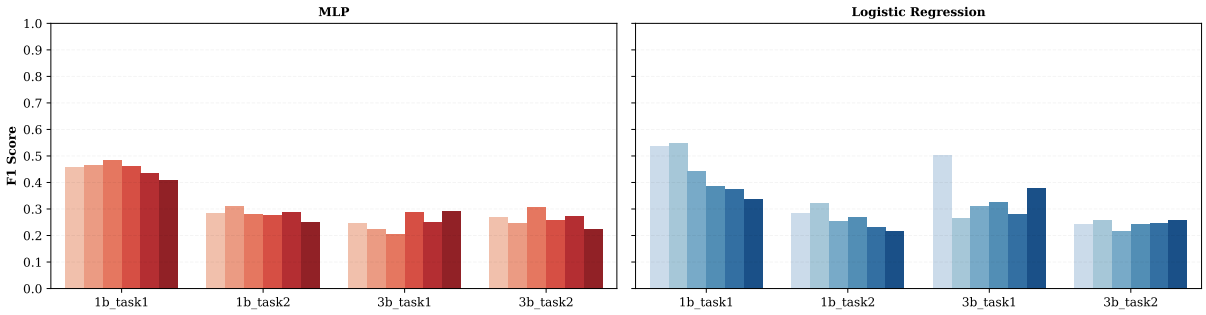

Figure 2: Head selection on base models (identity-aware train/test split). F1 score for MLP and Logistic Regression across tasks and Llama-3 sizes. Splitting by identity prevents the same individual from appearing in both train and test sets. Shaded bars indicate increasing numbers of retained heads, from 1 (lightest) to 11 (darkest), in steps of two.

4.3. MLP Classifier

The extracted features are used as the input of a multi-class classifier that predicts whether a sample belongs to the retain, forget, or unseen set. Before training the classifier, we performed feature selection using a variance thresholding step with a threshold of 0.001. We then applied standardization to normalize the input features. The classifier is implemented as a feed-forward multilayer perceptron with three consecutive fully connected blocks, followed by a final linear prediction layer. Within each block, the input features are first projected into a 256-dimensional hidden space via a linear layer, followed by batch normalization to stabilize training. A GELU nonlinearity is then applied, and dropout with a probability of 0.2 is used for regularization. After the third block, a final linear layer maps the resulting representation to three logits, which are used to generate the class prediction. The classifier was trained using a weighted cross-entropy loss, where the weights were computed as the inverse of the class frequencies, and optimized with Adam [24] at a learning rate of 10^{-3} for 300 epochs.

5. Experimental Setup

To evaluate the effectiveness of our method, we first applied it to the Development Data provided by the challenge organizers. Specifically, we compared different numbers of attention heads and assessed whether a Logistic Regression (LR) model or an MLP was more suitable at the prediction stage. The number of heads is a sensitive hyperparameter, since an high value enlarges the feature space and incorporates more attention-based information as predictor variables. However, retaining more heads also reduces confidence that each contributes meaningfully to the target label. Consequently, an appropriate trade-off is required to limit the introduction of noise due to poorly informative heads. During the Development Phase, we performed a 90/10 split of the provided data while preserving the label distribution in both the training and validation sets, and used only the training portion to identify the most relevant heads. We conducted experiments under two splitting strategies: one where train and validation sets contained overlapping identities and another where identities were strictly separated between the two splits. After fixing this selection, we extracted features for both sets, focusing exclusively on the selected heads. For the MLP, we assigned sample weights in the loss function according to the inverse label frequency. We then trained the models given for Development Phase and computed evaluation metrics on the validation set. Results were compared across different numbers of heads {1, 3, 5, 7, 9, 11} and MLP architectures to identify the most effective configuration.

During the Test phase, we fixed the best-performing configuration identified in the previous phase and followed the same procedure, using the full provided dataset for training.

Unlearning Algorithm	Task 1		Task 2	
	Baseline	MALTO	Baseline	MALTO
Llama-3-1B				
Gradient Difference	0.275	0.342 (+0.067)	0.264	0.325 (+0.061)
Finetune	0.274	0.333 (+0.059)	0.261	0.317 (+0.056)
KL Minimization	**0.295**	0.322 (+0.027)	0.265	0.319 (+0.054)
Gradient Ascent	0.283	0.342 (+0.059)	**0.268**	0.320 (+0.052)
Preference Optimization	0.279	**0.343 (+0.064)**	0.267	**0.336 (+0.069)**
Llama-3-3B				
Gradient Difference	0.288	0.352 (+0.064)	**0.275**	0.336 (+0.061)
Finetune	0.275	0.342 (+0.067)	0.263	**0.337 (+0.074)**
KL Minimization	**0.295**	0.354 (+0.059)	0.271	0.325 (+0.054)
Gradient Ascent	0.288	0.351 (+0.063)	**0.275**	0.334 (+0.059)
Preference Optimization	0.281	**0.365 (+0.084)**	0.269	0.329 (+0.060)

Table 2
Macro F1 scores on the official Test Set of SVELA. Absolute improvements of MALTO over the Baseline are reported in parentheses. The highest score in each column is bolded, and the second best is underlined.

6. Results

In this section, we first describe the results obtained during the Development Phase, motivate the resulting design choices, and then analyze the results obtained in the Test Phase.

6.1. Development Phase

The results of the evaluation conducted to identify the best model and number of retained heads are shown in Figure 1 and Figure 2. The MLP consistently outperforms the LR on Task 1 and achieves comparable performance on Task 2 when considering a random train/test split. The

impact of the number of retained heads varies across tasks and model sizes, with no uniformly optimal choice. While the MLP exhibits lower variance once the task and model size are fixed, LR shows less consistent trends. For instance, on Task 1, increasing the number of heads improves the performance for Llama-3-3B but degrades it for Llama-3-1B. When considering an identity-aware train/test split, results are more difficult to evaluate. The high variance of the LR across different heads is retained, but we also observe a steep decrease in performance for the MLP. Given the need for strong generalization across different and potentially unseen models processed by the MU algorithm, we selected the MLP as the most robust option and retained five heads. This configuration was chosen due to the more stable F1 score observed between tasks and model sizes during the Development Phase. We point out that we performed a minimal tuning on the architecture and its hyperparameters, and we are confident that the performance of the MLP could be further increased with additional computational resources and time.

When evaluated on the public leaderboard using the proposed configuration, our approach ranked first on both Task 1 and Task 2 for the Llama-3-3B model, achieving F1 scores of 0.3595 and 0.3330, respectively, averaged over all unlearning methods.

6.2. Test Phase

Unlearning Algorithm	Task 1				Task 2			
	Retain (F1)	Forget (F1)	Unseen (F1)	Overall (Acc.)	Retain (F1)	Forget (F1)	Unseen (F1)	Overall (Acc.)
Llama-3-1B								
Gradient Difference	0.641	0.179	**0.205**	0.475	0.649	0.132	0.195	0.485
Finetune	0.660	0.141	0.197	**0.493**	0.629	0.122	0.199	0.463
KL Minimization	0.659	0.153	0.153	0.486	0.642	0.137	0.179	0.474
Gradient Ascent	0.642	0.199	0.186	0.476	0.636	0.136	0.187	0.469
Preference Optimization	**0.662**	**0.201**	0.166	0.490	**0.656**	**0.144**	**0.207**	**0.493**
Llama-3-3B								
Gradient Difference	0.696	0.169	0.190	0.529	0.657	0.133	0.217	0.497
Finetune	0.679	0.147	0.199	0.514	0.663	**0.148**	0.200	0.499
KL Minimization	0.675	0.181	0.206	0.508	0.644	0.126	**0.205**	0.479
Gradient Ascent	0.684	0.160	**0.209**	0.520	**0.668**	0.137	0.195	**0.504**
Preference Optimization	**0.697**	**0.197**	0.200	**0.535**	0.645	0.142	0.200	0.480

Table 3
Class-Based F1 scores and Overall Accuracy of our method on the Test Set

Our approach attains macro F1 scores of 0.336 and 0.323 on Llama-3-1B, and 0.353 and 0.320 on Llama-3-3B, for identity-level and instance-level classification, respectively, averaged across all unlearning techniques. The F1 scores for the baseline and our method are reported in Table 2, while class-wise F1 scores and overall accuracy are shown in Table 3. Across all tasks and model sizes, our method consistently outperforms the baseline classifier. As expected, performance is generally higher for the 3B model than for the 1B model. A plausible explanation is that the 3B model has more layers and attention heads, enabling richer internal representations and more informative features for detecting behavioral changes induced by unlearning. In contrast, smaller models, due to their shallower architectures and reduced attention dimensionality, tend to encode less precise information and provide a more limited feature space, making post-unlearning shifts harder to characterize.

Unlearning Algorithm	Task 1					Task 2				
	All Features	w/o Head	w/o Layer	w/o Trans	w/o Logit	All Features	w/o Head	w/o Layer	w/o Trans	w/o Logit
Llama-3-1B										
Gradient Ascent	0.534	0.434 (-0.100)	0.452 (-0.083)	0.466 (-0.068)	0.551 (+0.017)	0.259	0.259 (-0.000)	0.301 (+0.042)	0.340 (+0.081)	0.256 (-0.004)
Finetune	0.426	0.515 (+0.089)	0.348 (-0.078)	0.317 (-0.110)	0.422 (-0.004)	0.287	0.254 (-0.033)	0.257 (-0.031)	0.264 (-0.024)	0.273 (-0.015)
KL Minimization	0.526	0.627 (+0.100)	0.489 (-0.037)	0.268 (-0.258)	0.430 (-0.096)	0.294	0.291 (-0.003)	0.304 (+0.010)	0.286 (-0.008)	0.274 (-0.020)
Gradient Difference	0.513	0.381 (-0.131)	0.412 (-0.100)	0.267 (-0.246)	0.309 (-0.204)	0.351	0.307 (-0.044)	0.222 (-0.129)	0.226 (-0.125)	0.328 (-0.023)
Preference Optimization	0.538	0.516 (-0.022)	0.575 (+0.037)	0.334 (-0.204)	0.602 (+0.063)	0.342	0.280 (-0.063)	0.258 (-0.085)	0.311 (-0.031)	0.250 (-0.092)
Llama-3-3B										
Gradient Ascent	0.291	0.284 (-0.007)	0.287 (-0.004)	0.191 (-0.101)	0.222 (-0.069)	0.264	0.303 (+0.039)	0.268 (+0.004)	0.286 (+0.022)	0.282 (+0.018)
Finetune	0.225	0.309 (+0.084)	0.351 (+0.126)	0.204 (-0.021)	0.277 (+0.051)	0.264	0.232 (-0.032)	0.258 (-0.006)	0.323 (+0.059)	0.272 (+0.008)
KL Minimization	0.287	0.375 (+0.088)	0.284 (-0.003)	0.230 (-0.058)	0.251 (-0.036)	0.229	0.210 (-0.019)	0.261 (+0.033)	0.226 (-0.003)	0.275 (+0.046)
Gradient Difference	0.319	0.325 (+0.006)	0.272 (-0.047)	0.203 (-0.116)	0.270 (-0.049)	0.310	0.260 (-0.050)	0.358 (+0.048)	0.302 (-0.009)	0.244 (-0.067)
Preference Optimization	0.284	0.310 (+0.026)	0.330 (+0.046)	0.256 (-0.028)	0.323 (+0.039)	0.223	0.263 (+0.040)	0.242 (+0.019)	0.251 (+0.028)	0.229 (+0.006)

Table 4
Leave-one-out feature ablation (macro F1 score). Each column removes one feature family. Cell color indicates direction and magnitude of change: blue = improvement, orange = degradation.

Task 1 Preference optimization achieves the best overall performance for both model sizes. This unlearning strategy explicitly steers the model toward a target response (e.g., "I do not know") on the forget set. Such targeted behavioral control appears to induce a distinctive and consistent behavior, making forget examples easier to identify than with methods that apply more abrupt or unstructured modifications to the model. More generally, most unlearning methods achieve the highest F1 score on the retain set, followed by the unseen set and then the forget set, as shown in Table 3. This ordering is expected, since retain examples has the highest distribution in the training set and are therefore easiest to classify reliably. Notably, preference optimization is the only approach for which performance on the forget set matches or exceeds that on the unseen set, despite the smaller size of the first. We hypothesize that this advantage arises because preference optimization enforces a specific behavioral pattern, whereas alternatives such as gradient-ascent-based unlearning introduce more diffuse and stochastic changes. These less structured updates likely yield weaker or noisier signals, making forget examples harder to distinguish.

Task 2 Preference optimization again achieves the highest performance for the 1B model. However, in the 3B setting, competing methods improve substantially, and preference optimization no longer dominates as it does in Task 1. One possible interpretation is that instance-level steering, enforcing targeted behavior on a subset of examples rather than across all identities, induces sharper and more detectable shifts in smaller models, where parameter updates translate more directly into measurable representational changes. In larger models, the same instance-level signal may be diluted across a higher-dimensional attention space; unless the resulting gradients are sufficiently strong and coherent, they may fail to produce discriminative features at the instance level. Additionally, fine-tuning on the retain set achieves the best performance for the 3B model, following the expected ordering across splits: retain highest, followed by unseen, and then forget. This pattern is consistent with the retain split being closest to the training data distribution.

To better understand each contribution, we conducted an ablation study on the feature families used as input to the classifier. We compare changes in performance when using all feature families, only one, all of them but one, or pairs of two at the time. The results are presented in Table 4, Table 5, and Table 6, and further analyzed in the following subsection.

6.3. Feature Ablation

We conducted a comprehensive feature ablation study to analyze how different feature configurations affect classification performance. Specifically, we evaluated leave-one-out settings, in which each feature family was removed individually, as well as single-feature and pairwise-feature

Unlearning Algorithm	Task 1					Task 2				
	All Features	Head	Layer	Trans	Logit	All Features	Head	Layer	Trans	Logit
Llama-3-1B										
Gradient Ascent	0.534	0.438 (-0.096)	0.265 (-0.269)	0.401 (-0.133)	0.241 (-0.293)	0.259	0.334 (+0.075)	0.247 (-0.012)	0.233 (-0.027)	0.350 (+0.091)
Finetune	0.426	0.293 (-0.133)	0.228 (-0.198)	0.529 (+0.103)	0.287 (-0.139)	0.287	0.260 (-0.027)	0.251 (-0.036)	0.300 (+0.013)	0.243 (-0.044)
KL Minimization	0.526	0.322 (-0.204)	0.264 (-0.262)	0.579 (+0.052)	0.244 (-0.282)	0.294	0.286 (-0.008)	0.314 (+0.020)	0.339 (+0.045)	0.341 (+0.047)
Gradient Difference	0.513	0.333 (-0.179)	0.286 (-0.227)	0.345 (-0.168)	0.244 (-0.268)	0.351	0.297 (-0.054)	0.242 (-0.109)	0.323 (-0.029)	0.298 (-0.053)
Preference Optimization	0.538	0.309 (-0.230)	0.382 (-0.157)	0.310 (-0.228)	0.317 (-0.221)	0.342	0.308 (-0.034)	0.201 (-0.141)	0.311 (-0.031)	0.288 (-0.055)
Llama-3-3B										
Gradient Ascent	0.291	0.188 (-0.103)	0.187 (-0.105)	0.331 (+0.039)	0.383 (+0.091)	0.264	0.269 (+0.005)	0.329 (+0.065)	0.235 (-0.028)	0.349 (+0.086)
Finetune	0.225	0.262 (+0.037)	0.190 (-0.035)	0.334 (+0.109)	0.270 (+0.044)	0.264	0.234 (-0.030)	0.288 (+0.024)	0.280 (+0.016)	0.348 (+0.084)
KL Minimization	0.287	0.221 (-0.066)	0.244 (-0.043)	0.362 (+0.075)	0.202 (-0.085)	0.229	0.242 (+0.014)	0.305 (+0.077)	0.279 (+0.051)	0.246 (+0.017)
Gradient Difference	0.319	0.217 (-0.102)	0.268 (-0.051)	0.250 (-0.069)	0.334 (+0.015)	0.297	0.260 (-0.037)	0.416 (+0.119)	0.341 (+0.044)	0.274 (-0.023)
Preference Optimization	0.284	0.257 (-0.028)	0.185 (-0.099)	0.327 (+0.042)	0.421 (+0.137)	0.236	0.236 (+0.000)	0.304 (+0.068)	0.301 (+0.065)	0.310 (+0.074)

Table 5
Single feature ablation (macro F1 score). Each column uses only one feature family. Cell color indicates direction and magnitude of change: blue = improvement, orange = degradation.

Unlearning Algorithm	Task 1							Task 2						
	All Features	Head+Layer	Head+Trans	Head+Logit	Layer+Trans	Layer+Logit	Trans+Logit	All Features	Head+Layer	Head+Trans	Head+Logit	Layer+Trans	Layer+Logit	Trans+Logit
Llama-3-1B														
Gradient Ascent	0.534	0.494 (-0.041)	0.483 (-0.051)	0.477 (-0.057)	0.481 (-0.053)	0.410 (-0.125)	0.427 (-0.107)	0.259	0.326 (+0.066)	0.276 (+0.017)	0.322 (+0.063)	0.281 (+0.022)	0.285 (+0.026)	0.286 (+0.027)
Finetune	0.426	0.214 (-0.212)	0.337 (-0.090)	0.267 (-0.159)	0.528 (+0.102)	0.235 (-0.191)	0.499 (+0.072)	0.287	0.260 (-0.028)	0.282 (-0.005)	0.270 (-0.018)	0.307 (+0.019)	0.287 (-0.001)	0.325 (+0.038)
KL Minimization	0.526	0.336 (-0.191)	0.568 (+0.042)	0.367 (-0.160)	0.563 (+0.037)	0.358 (-0.168)	0.601 (+0.075)	0.294	0.329 (+0.035)	0.294 (+0.000)	0.333 (+0.039)	0.359 (+0.065)	0.250 (-0.044)	0.302 (+0.008)
Gradient Difference	0.513	0.333 (-0.179)	0.363 (-0.150)	0.274 (-0.238)	0.458 (-0.054)	0.346 (-0.166)	0.355 (-0.158)	0.351	0.264 (-0.087)	0.262 (-0.090)	0.302 (-0.049)	0.350 (-0.001)	0.263 (-0.088)	0.334 (-0.017)
Preference Optimization	0.538	0.322 (-0.216)	0.598 (+0.060)	0.294 (-0.244)	0.503 (-0.036)	0.333 (-0.206)	0.414 (-0.124)	0.342	0.324 (-0.018)	0.284 (-0.058)	0.334 (-0.008)	0.265 (-0.077)	0.289 (-0.053)	0.249 (-0.094)
Llama-3-3B														
Gradient Ascent	0.291	0.197 (-0.094)	0.259 (-0.033)	0.212 (-0.079)	0.264 (-0.027)	0.225 (-0.066)	0.310 (+0.019)	0.264	0.264 (+0.000)	0.283 (+0.019)	0.241 (-0.023)	0.300 (+0.036)	0.323 (+0.059)	0.260 (-0.003)
Finetune	0.225	0.194 (-0.031)	0.230 (+0.005)	0.209 (-0.016)	0.328 (+0.102)	0.212 (-0.014)	0.318 (+0.093)	0.264	0.277 (+0.013)	0.272 (+0.008)	0.290 (+0.026)	0.245 (-0.019)	0.317 (+0.053)	0.275 (+0.011)
KL Minimization	0.287	0.225 (-0.062)	0.227 (-0.060)	0.230 (-0.058)	0.328 (+0.041)	0.163 (-0.125)	0.371 (+0.084)	0.229	0.248 (+0.019)	0.227 (-0.002)	0.210 (-0.019)	0.256 (+0.027)	0.263 (+0.035)	0.216 (-0.012)
Gradient Difference	0.319	0.214 (-0.105)	0.270 (-0.049)	0.221 (-0.098)	0.334 (+0.015)	0.208 (-0.111)	0.304 (-0.015)	0.310	0.293 (-0.018)	0.341 (+0.030)	0.283 (-0.027)	0.320 (+0.010)	0.285 (-0.025)	0.289 (-0.021)
Preference Optimization	0.284	0.240 (-0.045)	0.304 (+0.020)	0.272 (-0.012)	0.329 (+0.045)	0.252 (-0.032)	0.402 (+0.118)	0.223	0.248 (+0.025)	0.314 (+0.091)	0.234 (+0.011)	0.281 (+0.057)	0.281 (+0.057)	0.306 (+0.083)

Table 6
Pairwise feature ablation (macro F1 score). Each column uses only the indicated pair of feature families. Cell color indicates direction and magnitude of change: blue = improvement, orange = degradation.

configurations to assess the contribution of each feature group and their combinations. All ablation experiments were performed using the same data split as in the Development Phase, where identity overlap between the training and test sets was explicitly prevented. The analysis was carried out across our four feature families to systematically examine their individual and joint impact on model performance.

Task 1 In the 1B model, we observe a strong sensitivity to the transition-based features. Removing them leads to a substantial drop in performance, whereas, when used alone, they yield the smallest decrease and, in the case of Finetune and KL Minimization, even improve the F1 score. For these algorithms, head-based features appear to be the least useful and may even be detrimental, as their removal improves performance, as shown in Table 4. A similar pattern emerges in the 3B model. Excluding the transition-based features again reduces performance, whereas relying solely on them improves results for almost all algorithms. In contrast, removing the head-based features–and in this case also the layer-based ones–leads to moderate improvements for some methods. These fluctuations are again more pronounced for Finetune and KL Minimization. The same trends are observed when considering pairs of feature families in both the 1B and 3B models. For example, combinations of head-, layer-, and logit-based features lead to a marked decline in performance for most algorithms. By contrast, combinations that include transition-based features show a smaller decrease and, in most cases, even an improvement. This behavior is consistent with the leave-one-out and single-group analyses and extends to the pairwise setting. Interestingly, logit-based features appear to be more useful in the 3B model than in the 1B. When used as the sole input to the classifier, they improve performance for almost all algorithms. One possible explanation is that larger models produce more informative logits for distinguishing identities. This is plausible, as larger models may express greater confidence in the information learned during training, while also being more sensitive when faced with uncertainty. In the 1B model, nearly all ablation

settings resulted in a decrease in F1 score across single-feature ablations, pairwise removals, and leave-one-out configurations. This pattern, however, was not observed in the 3B model. These findings suggest that, in smaller models, membership-related information is distributed across multiple internal structures, requiring the integration of diverse feature sources to achieve reliable classification performance. In contrast, larger models are able to rely on more selective subsets of features, as membership signals appear to be more specialized and concentrated within particular representational components, such as transition-based features and output logits.

Finally, the impact of selecting specific attention heads varies depending on both the unlearning method and the model scale. When considered in isolation, head-based features do not appear to carry sufficiently strong information for reliable classification. In smaller models, however, aggressive unlearning methods such as Gradient Ascent and Gradient Difference lead to substantial decreases in F1 score when head information is ablated. This suggests that these sharp weight-update strategies leave detectable traces within attention routing patterns, making head-level signals informative for membership classification. In contrast, in larger models, the same methods tend to introduce noise rather than coherent signals. Due to higher representational capacity and redundancy, weight perturbations become more distributed and heterogeneous, which weakens the discriminative value of individual heads. This pattern is consistent with the broader ablation results, where head-focused feature removals produce less stable and less interpretable performance changes in larger models.

Task 2 Performance changes are generally smaller in absolute terms than in the previous task. In the 1B model, nearly all feature groups contribute positively, as removing any of them typically leads to a decline in performance. The main exception is Gradient Ascent, where excluding layer- or transition-based features results in slight improvements. A different pattern emerges in the 3B model, where information from attention heads appears to benefit the classifier. However, as shown in Table 5, head-based features have considerably less impact than layer-, transition-, and logit-based features when used as the sole input to the MLP, particularly in the 3B model. At the same time, certain feature combinations–such as Layer+Trans, Head+Trans, and Layer+Logit–show strong potential, achieving improvements over the configuration that includes all features. For the 1B model, we observe a general increase in performance across several pairs of features, with the exception of Gradient Difference and Preference Optimization. Notably, these two algorithms are negatively affected by any modification relative to the full-feature setting. For the 3B model, we observe an increase for most of the combinations and a slight decrease when considering Head+Logit and Trans+Logit.

In Task 2, attention head selection has a more pronounced impact compared to Task 1. When head-based features are removed, model performance often decreases immediately, indicating that attention routing carries informative signals for membership classification in this setting. Although head features alone do not provide strong discriminative power, pairwise feature analyses show that combining heads with other feature families almost consistently improves the F1 score, as illustrated in Table 6. The discrepancy between the two tasks can be explained by the granularity of the forgetting objective. In Task 2, selective attribute-level unlearning targets specific pieces of personal information, such as email addresses or other identifying attributes, which are more likely to be localized within specialized attention heads. In contrast, Task 1 requires full entity removal, leading to more global representational disruptions distributed across layers and transitions rather than being concentrated within individual heads. Consequently, head-level signals exert a weaker impact in Task 1 than in Task 2.

Task 1 vs. Task 2 The differences observed between Task 1 and Task 2 can largely be explained by the granularity of the unlearning objective and how forgetting signals spread within the model. In Task 1, unlearning is performed at the full entity level, meaning the model must remove all knowledge associated with a specific person. This comprehensive removal creates

a strong and coherent disruption across internal representations, making the forgetting signal more concentrated and easier to detect within certain dominant feature families, particularly transition based features and, in larger models, output logits. Since entity knowledge is encoded through interconnected mechanisms such as retrieval, relational associations, and reasoning pathways, removing the entire entity produces structured changes that are especially visible in features capturing inter layer transformations. In contrast, Task 2 focuses on selective, attribute level forgetting, where only specific information such as an email address must be unlearned while the broader entity representation is preserved. This results in weaker and more localized representational changes, preventing any single feature group from consistently dominating the signal. Consequently, performance differences across ablation settings are smaller, and multiple feature combinations remain informative.

7. Conclusion

This work proposes an evaluation metric for predicting the membership label of data in unlearned models, developed for the SVELA @ EVALITA 2026 Challenge. The metric infers membership status by first selecting the most informative attention heads for the retain, forget, and unseen classes using a trainable soft head mask. It then extracts head-level features from the selected heads, together with layer-wise attention, layer transition and logit features. These features are provided to an MLP classifier, which predicts whether an input belongs to the retain, forget, or unseen set.

During development, we observed that the number of selected heads affects performance differently across tasks, model sizes, and classifier choices. We also found that features extracted from models trained with different unlearning methods yield distinct performance patterns. A particularly notable result concerns preference optimization: this method enforces a specific behavior when processing forget examples and consistently achieves a higher F1 score on the forget set than on the unseen set across all tested models in identity-level classification. This suggests that inducing a consistent behavioral shift across identities leaves a detectable signature in the models internal representations.

Across both tasks, the ablation analyses show that the detectability and localization of unlearning signals are closely related to the granularity of the forgetting objective and the representational capacity of the model. In Task 1, entity-level unlearning produces more coherent and concentrated disruptions, with transition-based features and, in larger models, logits emerging as particularly informative. Smaller models, however, rely on the combined contribution of multiple feature families due to the more distributed nature of membership signals. In contrast, Task 2 leads to weaker and more localized representational changes, resulting in smaller performance variations and more balanced contributions across feature groups. Specific attention head selection has varying impacts across tasks. It does not provide a strong contribution to entity-level unlearning but shows positive effects in instance-level unlearning. Overall, the results indicate that both model scale and the scope of the unlearning objective influence how and where forgetting signals are reflected within internal representations.

8. Future Work

Future work will extend this approach by combining head selection with explicit layer selection, enabling a more detailed analysis of how attention heads in specific layers respond to different unlearning methods. Similar investigations could also be conducted at the token level, for example by examining how identity-name tokens influence attention patterns and whether assigning them greater weight in attention-based feature extraction improves classification performance. A current limitation of our method is that it does not account for the sequential nature of signals within the model. More advanced approaches could track how these signals

evolve during the forward pass, focusing on the progression of activations rather than treating each one independently. Another limitation is the use of a fixed number of heads across all layers. A more flexible strategy could adapt the number of selected heads according to the layer index, for instance selecting fewer heads in earlier layers and more in later ones. In addition, as a white-box method, our approach requires access to the models internal activations. This requirement may limit its applicability in real-world scenarios where such access is not available. Finally, all experiments were conducted only on Llama-3-1B and Llama-3-3B. It remains unclear how well these findings generalize to other models and architectures.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-4o to correct typos and grammatical mistakes. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

[1] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, Y. Zhang, A survey on large language model (llm) security and privacy: The good, the bad, and the ugly, High-Confidence Computing 4 (2024) 100211. URL: http://dx.doi.org/10.1016/j.hcc.2024.100211. doi:10.1016/j.hcc.2024.100211.

[2] X. Liu, T. Sun, T. Xu, F. Wu, C. Wang, X. Wang, J. Gao, Shield: Evaluation and defense strategies for copyright compliance in llm text generation, 2024. URL: https://arxiv.org/abs/2406.12975. arXiv:2406.12975.

[3] P. Maini, Z. Feng, A. Schwarzschild, Z. C. Lipton, J. Z. Kolter, Tofu: A task of fictitious unlearning for llms, 2024. URL: https://arxiv.org/abs/2401.06121. arXiv:2401.06121.

[4] C. Savelli, E. Munis, E. Bayat, A. Grieco, F. Giobergia, MALTO at SemEval-2025 task 4: Dual teachers for unlearning sensitive content in LLMs, in: S. Rosenthal, A. Rosá, D. Ghosh, M. Zampieri (Eds.), Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 1747–1752. URL: https://aclanthology.org/2025.semeval-1.229/.

[5] A. Koudounas, C. Savelli, F. Giobergia, E. Baralis, "Alexa, can you forget me?" Machine Unlearning Benchmark in Spoken Language Understanding, in: Interspeech 2025, 2025, pp. 1768–1772. doi:10.21437/Interspeech.2025-2607.

[6] C. Savelli, M. La Quatra, A. Koudounas, F. Giobergia, Svela at evalita 2026: Overview of the selective verification of erasure from llm answers task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

[7] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

[8] X. Liu, Q. Song, Q. Zhou, H. Du, S. Xu, W. Jiang, W. Zhang, X. Jia, Focusing on language: Revealing and exploiting language attention heads in multilingual large language models, 2025. URL: https://arxiv.org/abs/2511.07498. arXiv:2511.07498.

[9] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, 2017. URL: https://arxiv.org/abs/1610.05820. arXiv:1610.05820.

[10] J. Mattern, F. Mireshghallah, Z. Jin, B. Schölkopf, M. Sachan, T. Berg-Kirkpatrick, Membership inference attacks against language models via neighbourhood comparison, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, 2023,

pp. 11330–11343. URL: https://aclanthology.org/2023.findings-acl.719/. doi:10.18653/v1/2023.findings-acl.719.

[11] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, L. Zettlemoyer, Detecting pretraining data from large language models, 2024. URL: https://arxiv.org/abs/2310.16789. arXiv:2310.16789.

[12] W. Fu, H. Wang, C. Gao, G. Liu, Y. Li, T. Jiang, Mia-tuner: Adapting large language models as pre-training text detector, 2024. URL: https://arxiv.org/abs/2408.08661. arXiv:2408.08661.

[13] D. Makhija, M. G. Arivazhagan, V. B. Kumar, R. Gangadharaiah, Neural breadcrumbs: Membership inference attacks on llms through hidden state and attention pattern analysis, 2025. URL: https://arxiv.org/abs/2509.05449. arXiv:2509.05449.

[14] A. Ramakrishna, Y. Wan, X. Jin, K.-W. Chang, Z. Bu, B. Vinzamuri, V. Cevher, M. Hong, R. Gupta, Semeval-2025 task 4: Unlearning sensitive content from large language models, 2025. URL: https://arxiv.org/abs/2504.02883. arXiv:2504.02883.

[15] M. Duan, A. Suri, N. Mireshghallah, S. Min, W. Shi, L. Zettlemoyer, Y. Tsvetkov, Y. Choi, D. Evans, H. Hajishirzi, Do membership inference attacks work on large language models?, 2024. URL: https://arxiv.org/abs/2402.07841. arXiv:2402.07841.

[16] A. Deepak, M. Mou, J. Huang, D. Yang, Identifying unlearned data in LLMs via membership inference attacks, in: C. Christodoulopoulos, T. Chakraborty, C. Rose, V. Peng (Eds.), Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Suzhou, China, 2025, pp. 10884–10903. URL: https://aclanthology.org/2025.emnlp-main.551/. doi:10.18653/v1/2025.emnlp-main.551.

[17] C. Savelli, M. La Quatra, A. Koudounas, F. Giobergia, FAME: Fictional actors for multilingual erasure, in: Proceedings of the Fifteenth Language Resources and Evaluation Conference, European Language Resources Association, 2026.

[18] A. Grattafiori, et al., The llama 3 herd of models, 2024. URL: https://arxiv.org/abs/2407.21783. arXiv:2407.21783.

[19] D. Choi, D. Na, Towards machine unlearning benchmarks: Forgetting the personal identities in facial recognition systems, 2023. URL: https://arxiv.org/abs/2311.02240. arXiv:2311.02240.

[20] A. Golatkar, A. Achille, S. Soatto, Eternal sunshine of the spotless net: Selective forgetting in deep networks, 2020. URL: https://arxiv.org/abs/1911.04933. arXiv:1911.04933.

[21] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, I. Titov, Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 5797–5808. URL: https://aclanthology.org/P19-1580/. doi:10.18653/v1/P19-1580.

[22] L. Basile, V. Maiorca, D. Doimo, F. Locatello, A. Cazzaniga, Head pursuit: Probing attention specialization in multimodal transformers, 2025. URL: https://arxiv.org/abs/2510.21518. arXiv:2510.21518.

[23] P. Guo, A. Syed, A. Sheshadri, A. Ewart, G. K. Dziugaite, Mechanistic unlearning: Robust knowledge unlearning and editing via mechanistic localization, 2024. URL: https://arxiv.org/abs/2410.12949. arXiv:2410.12949.

[24] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017. URL: https://arxiv.org/abs/1412.6980. arXiv:1412.6980.