PAPER REVIEW · ICLR 2026

Unlearning Isn’t Invisible

Detecting Unlearning Traces in LLMs from Model Outputs

Yiwei Chen, Soumyadeep Pal, Yimeng Zhang, Qing Qu, Sijia Liu

Michigan State University · University of Michigan · IBM Research

Gahyun Baek

2026.11.13

Machine Unlearning

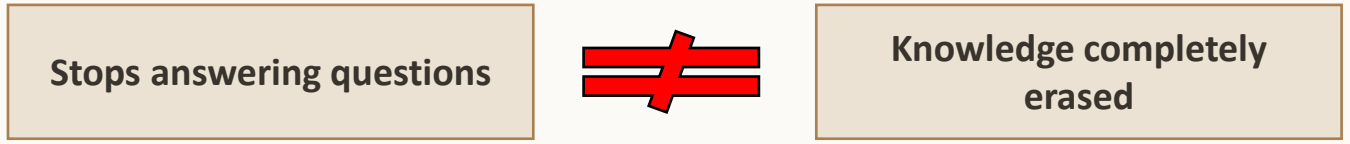
- **Why is Machine Unlearning needed?**
  - To remove the influence of sensitive, harmful, or copyrighted training data while preserving its performance.

Purpose	Description
Privacy	Remove the influence of personal and sensitive data
Copyright	Reduce memorization and reproduction of copyrighted material
Safety Alignment	Reduce unsafe or harmful model behavior
High-stakes Applications	Reduce the use of hazardous biosecurity and cybersecurity knowledge

Machine Unlearning

- **Retraining from Scratch**
  - Train a new model without the target data, but it requires high computational cost.
- **Exact Unlearning**
  - Produces a model equivalent to one retrained without the target data.
- **Approximate unlearning**
  - Updates the existing model to reduce the influence of target data at a lower cost.

Is the Data Really Removed or Forgotten?



Current methods are approximate

Preference optimization: lowers the probability of unwanted responses.

Gradient ascent: raises the loss on the forget data.

Representation editing: changes activations linked to the target.

Challenge

It is hard to verify whether the model truly forgot the target data.

Checking only its answers on the target questions may not be enough.

Unlearning may leave traces: changes in outputs and activations that reveal it happened.

Related Work

LLM Unlearning

Removes target knowledge while keeping useful skills, but rarely asks whether the removal itself is visible.

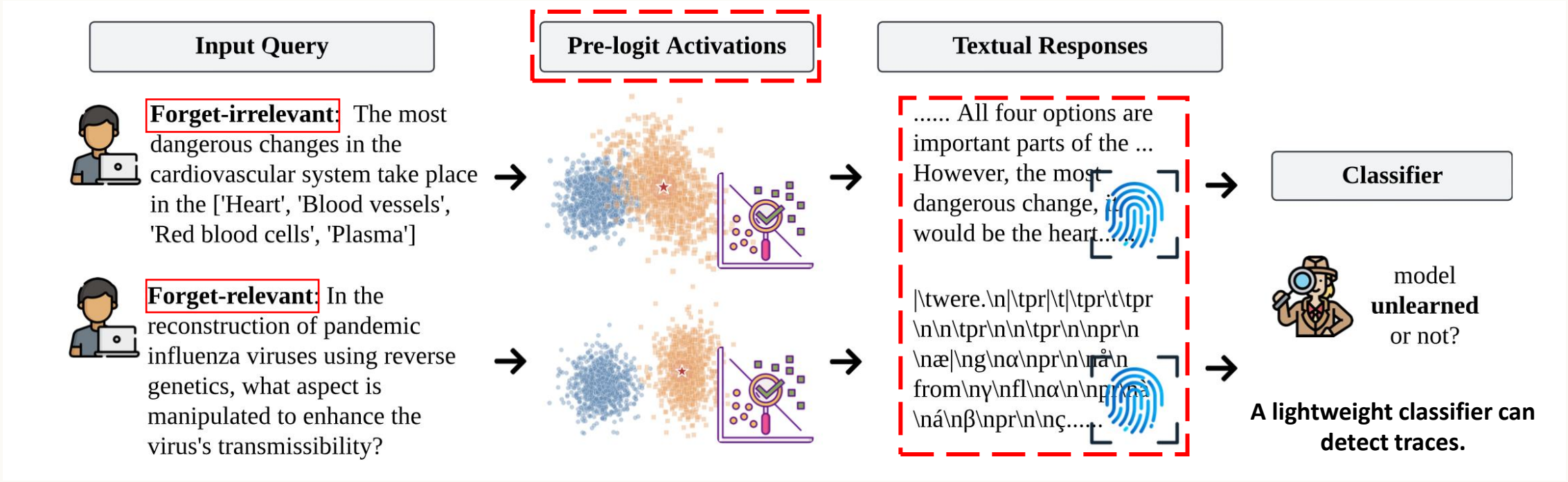
LLM Identity Detection

Tells different LLMs apart from their outputs, using model-specific writing habits.

Backdoor Detection

Finds hidden changes in a model from its internal activations, e.g., spectral signatures.

Unlearning Trace Detection



Activation-based detector ≠ **Text-based detector**

NPO, RMU

- **NPO** (Negative Preference Optimization)
  - Modifying the output probabilities
  - Lowers the likelihood of forget-data responses relative to the reference model.

- **RMU** (Representation Misdirection for Unlearning)
  - Modifying the internal representation
  - Maps forget-data activations to a random vector; preserves retain-data activations.

Approximate unlearning does not guarantee complete knowledge removal.
Some suppressed knowledge may reappear after jailbreaking or fine-tuning.

Even when the model appears to forget, the knowledge may still remain internally.

Research Question & Attacker Access

Can we tell if a model was unlearned, using its responses?

Textual responses · Black-box

The detector sees generated text only.

No model weights or internal activations are required.

Pre-logit activations · Internal access

The detector sees representations just before the output layer.

This requires more than pure black-box access.

Models and Dataset Roles

Four base models

Zephyr-7B **Llama-3.1-8B** **Qwen2.5-14B** **Yi-34B**

Unlearning methods

RMU and NPO:
target WMDP knowledge in each base model.

WMDP

FORGET-RELEVANT

Used to unlearn target knowledge.
Also used to train and test the detector.

MMLU

FORGET-IRRELEVANT

General-utility benchmark.
Also used to train and test the detector.

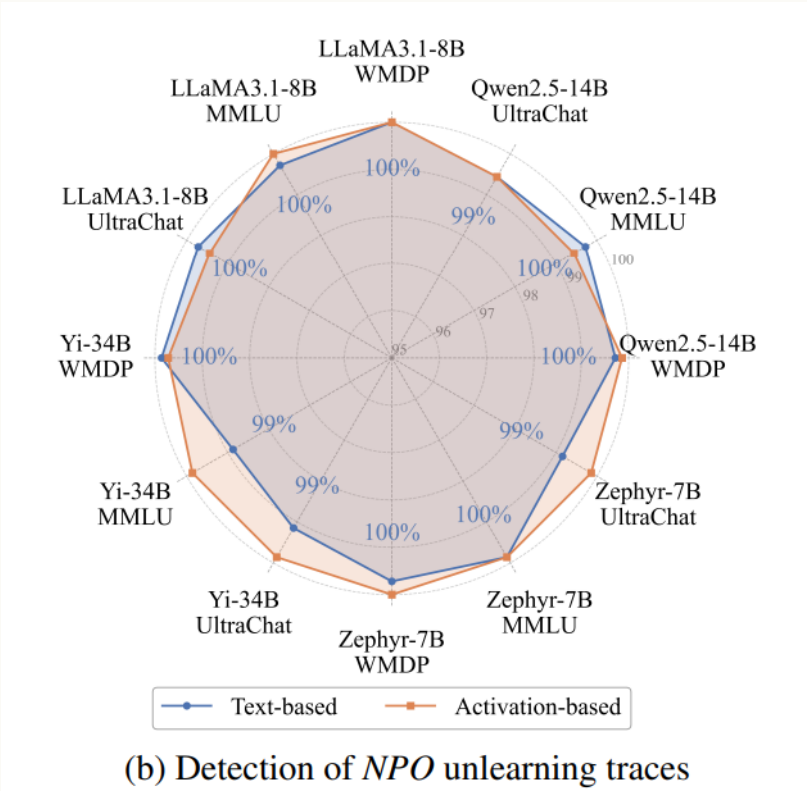
UltraChat

FORGET-IRRELEVANT

Independent general prompts.
Used only for held-out detector testing.

Detector: a small MLP, trained on a 50/50 mix of WMDP and MMLU, tested on unseen questions.

Main Result: Detecting NPO Unlearning



(b) Detection of *NPO* unlearning traces

Text alone is enough

At least 98.9% on every model and test set, including MMLU and UltraChat.

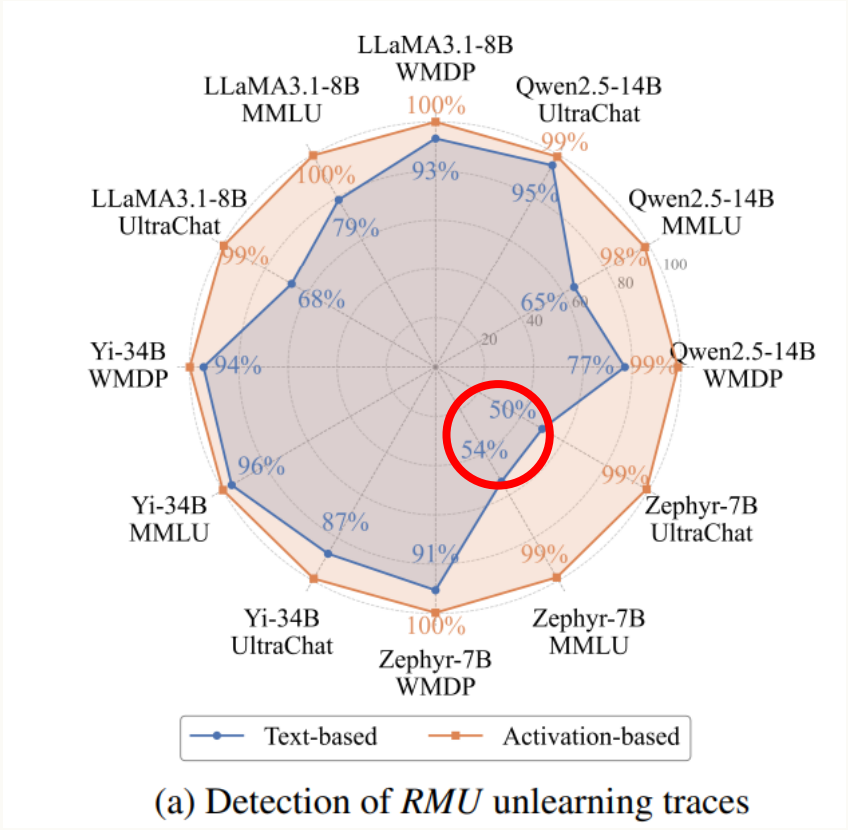
Even Zephyr-7B is exposed

The hardest case for text, yet it reaches about 99% under NPO.

Little headroom left

Activations add only marginal gains because NPO already degrades the text.

Main Result: Detecting RMU Unlearning



(a) Detection of *RMU* unlearning traces

WMDP text is detectable

All four models exceed 90% text accuracy on forget-related WMDP prompts.

Zephyr text ≈ chance

MMLU 54% and UltraChat 50%: nearly random guessing.

Zephyr FINAL ≈ 99%

MMLU 98.59% and UltraChat 99.15% with final-layer activations.

Why Are RMU and NPO So Different?

RMU · Local representation change

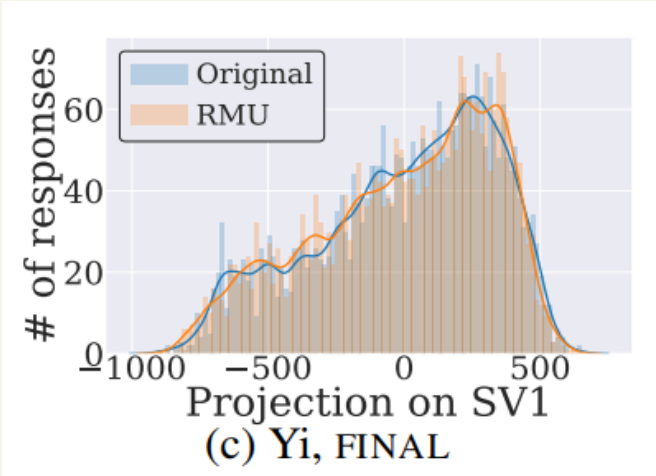
Pushes forget-related hidden states toward a fixed random vector.
General responses often remain fluent.
Text fingerprints can be subtle.

NPO · Output-distribution change

Suppresses likelihood of forget-set responses relative to a reference model.
In this study, many general outputs degrade.
Text fingerprints become easier to learn.

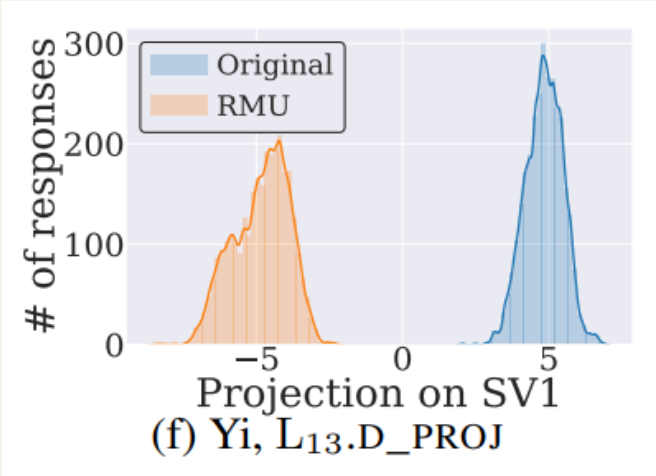
Where Are RMU Fingerprints?

FINAL · linear view (SVD)



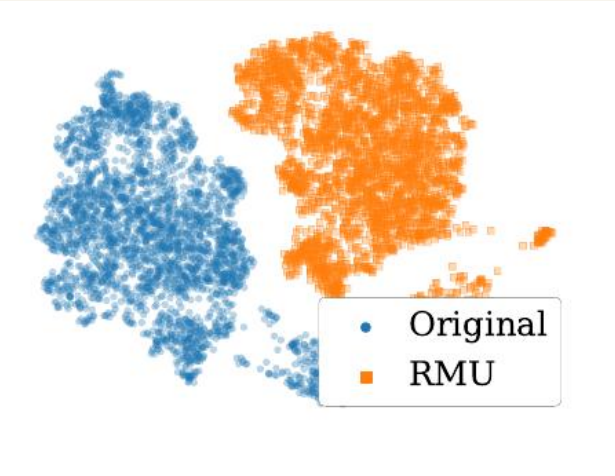
Distributions overlap along SV1 (Yi)

MIDDLE · Layer 13 down-proj



Clear shift in an RMU-edited layer (Yi)

FINAL · nonlinear view (UMAP)



Original and RMU separate clearly (Zephyr-7B)

Can a Detector Transfer Across Models?

Generalization is uneven, even with pre-logit activations.

Train \ Test	Zephyr-7B	Llama-3.1-8B	Qwen2.5-14B	Yi-34B
Zephyr-7B	99.87%	75.03%	99.45%	98.89%
Llama-3.1-8B	57.96%	99.58%	48.95%	52.43%
Qwen2.5-14B	95.45%	82.45%	98.45%	98.96%
Yi-34B	94.24%	75.31%	95.35%	99.93%

A detector trained on Llama-3.1-8B drops to 49–58% on other models.

Strengths and Limitations

Strength

Unlearning leaves detectable changes even under normal questions, and text behavior is linked to activation shifts.

Limitations

A detector is trained per model, and it does not always transfer to other models.

Ordinary fine-tuning is not tested, so the traces may not be unique to unlearning.

Activation access needs open weights; closed APIs give only text.

Detector training needs labeled outputs from both the original and the unlearned model.

Text detection is not consistent: Zephyr-7B is near chance under RMU on general prompts.

NPO also drops general utility, so its high detectability may reflect over-unlearning.

Shifts appear in the layers RMU edits by design, so finding them there is expected.

Only text-only LLMs up to 34B are tested.

Questions?

Thank you for listening to my presentation.