Failure Modes of Zero-Shot Machine Unlearning in Reinforcement Learning and Robotics

ELTE | FACULTY OF INFORMATICS | INTELLIGENT ROBOTICS FAIR

Presenter(s) | **Takács Tamás**

tamastheactual(at)inf(dot)elte(dot)hu

ELTE

Faculty of Informatics

Department of Artificial Intelligence

Co-Author(s)* | **Gulyás László**

lgulyas(at)inf(dot)elte(dot)hu

Associate Professor, Department of Artificial Intelligence

ELTE – Faculty of Informatics

Budapest, 2025

*Equal Contribution

Goal

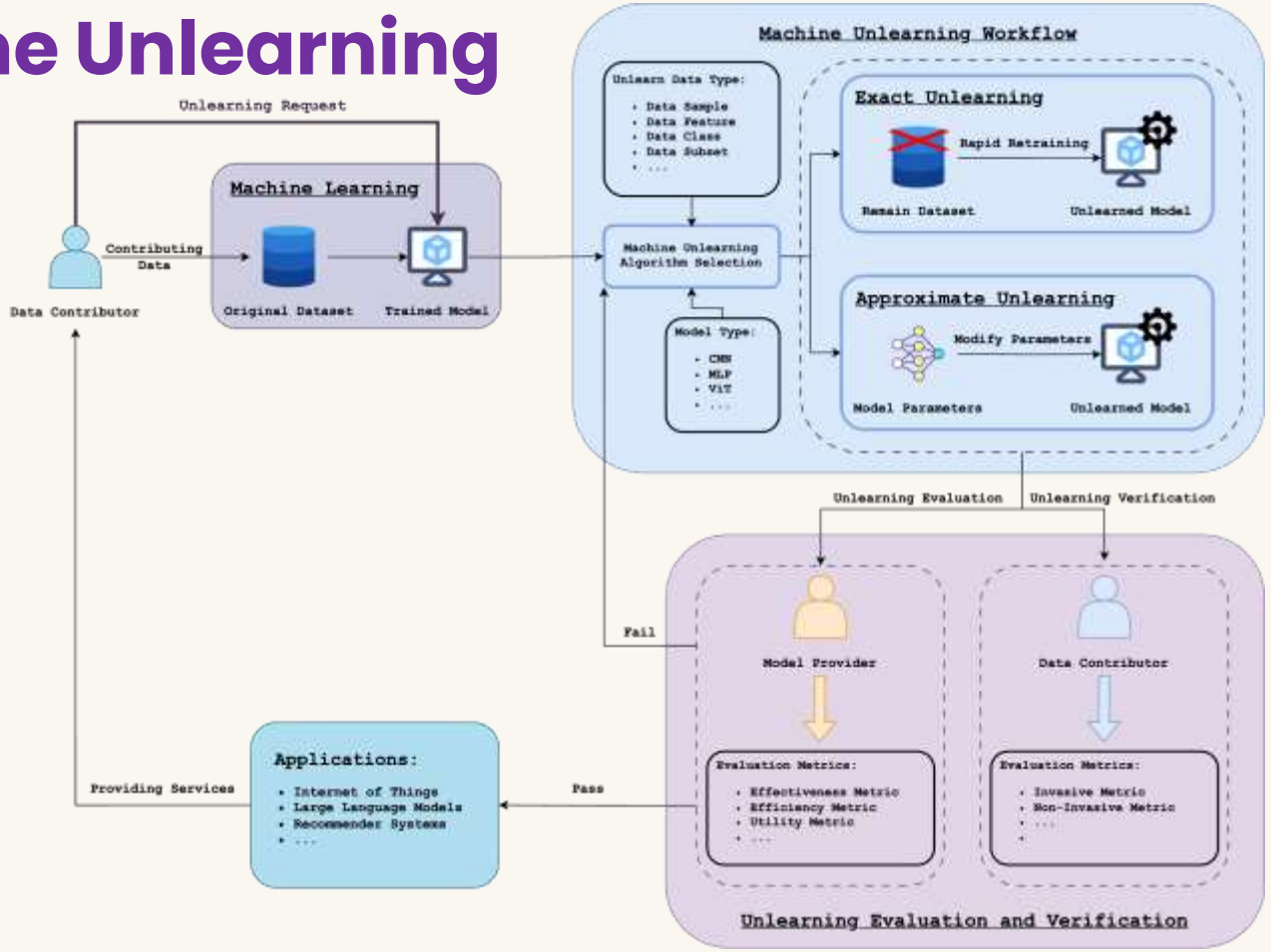
"Right to be Forgotten"

- **Privacy** compliance 
- Right to request the **deletion of** their personal **data**
- **Article 17** of the **EU General Data Protection Regulation** (GDPR **[1]**)
- Data was processed **unlawfully**
- The person **objects to data processing**
- Data was **collected from a child**

Applicability & Adaptability

- Model "forgets" knowledge **without retraining from scratch**
- Data deletion **without harming** the model's **performance**
- Provide **measurable evidence** that unlearning has occurred
- **Prevent** attackers from **recovering deleted information**
- Apply unlearning to **various domains** (CV, NLP, RL, etc.)

Machine Unlearning



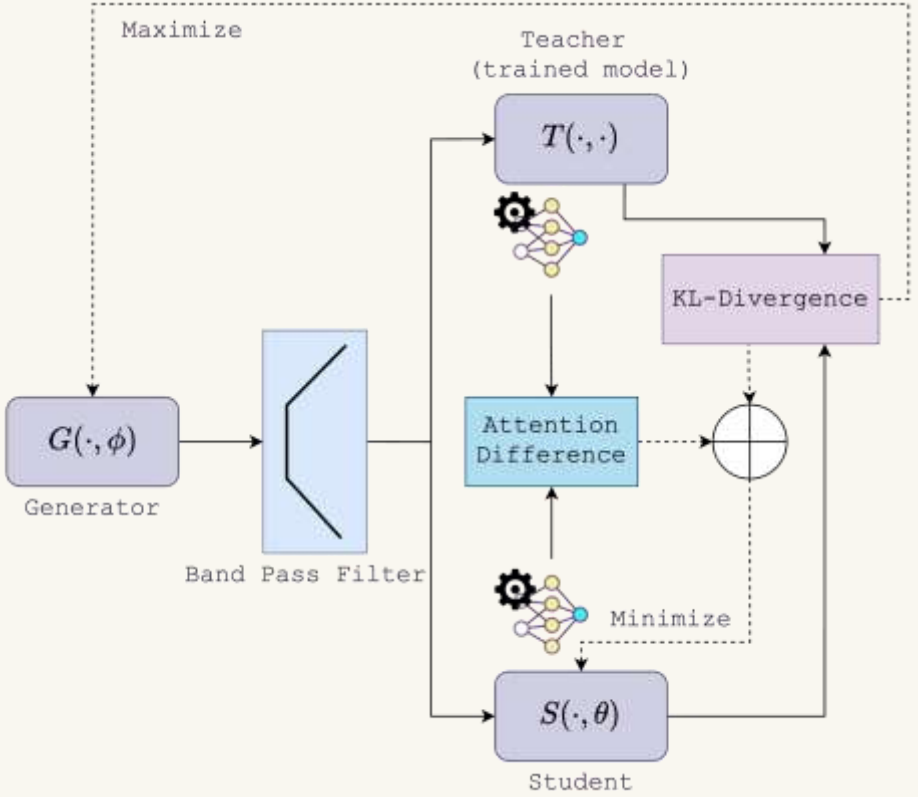
Unlearning in Robotics

Social/Service Robots [2]

Industrial/Autonomous Systems [3]

Zero-Shot Unlearning [4]

- Zero-shot machine unlearning, introduced by Chundawat et al., which aims to **erase data classes from a trained model** without needing the original training samples.
- The **GKT (Gated Knowledge Transfer) framework** that trains a student network to retain allowed knowledge while forgetting the target class.



Unlearning in RL

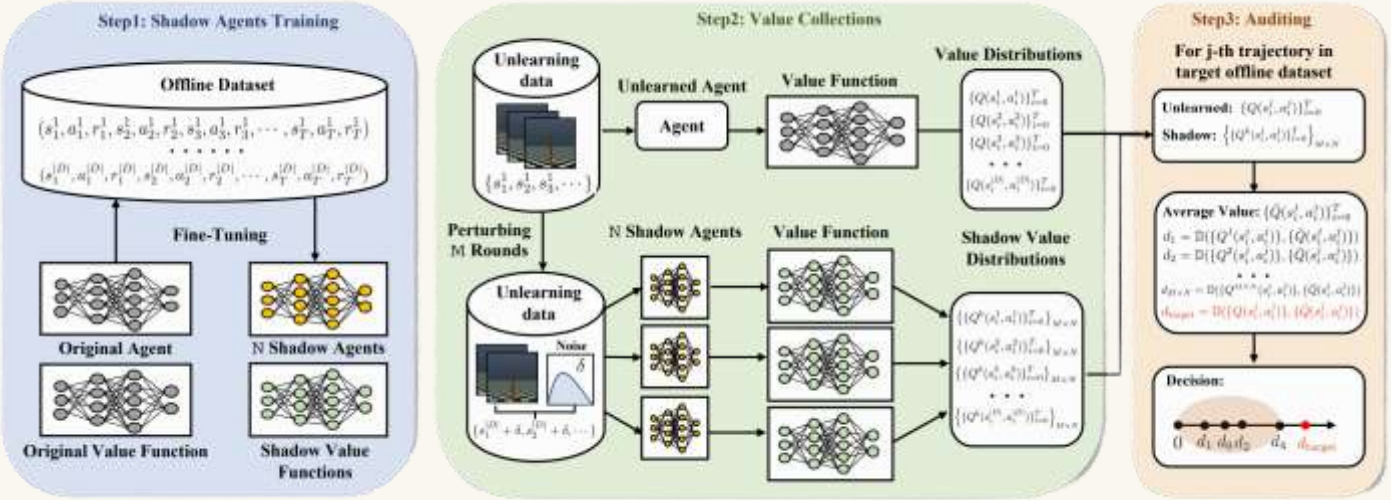
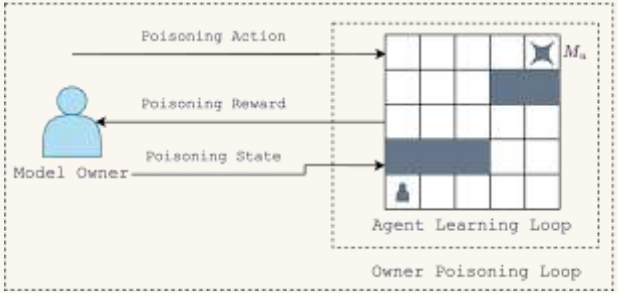


*Image Sourced from [7]

TrajDeleter: **Enabling Trajectory Forgetting in Offline Reinforcement Learning Agents [7]**



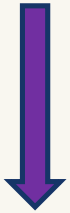
Reinforcement Unlearning [8]

Problem Statement

Original

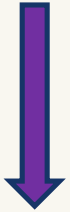
"Zero-shot machine unlearning seeks to remove specific data classes from a trained model without any access to the original training data."

Ours



~0 accuracy on forget set → continued training (retain) → forget set performance recovers

Generator-Filter Pipeline Leakage



White-box access to student and generator → model inversion can recover synthetic samples of the forgotten class.

Attack Vector

Prerequisites

The training set is typically split into two distinct datasets:

$$D_{train} = D_f \cup D_r$$

The goal is to transform a model

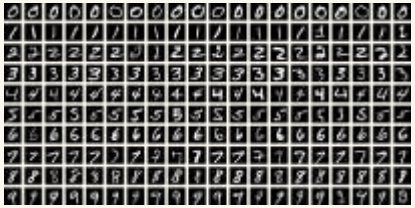
$$M(\cdot, \phi) \quad \rightarrow \quad M(\cdot, \phi')$$

Such that its outputs on D_f are **indistinguishable from those of a model** trained only on D_r, while **maintaining performance** on D_r.

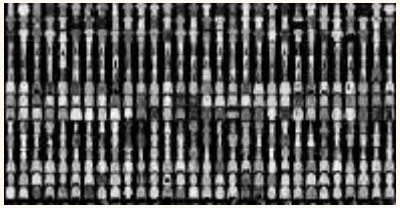
Exact matching of parameters ϕ' is usually **infeasible**.

Methods

Datasets:



MNIST

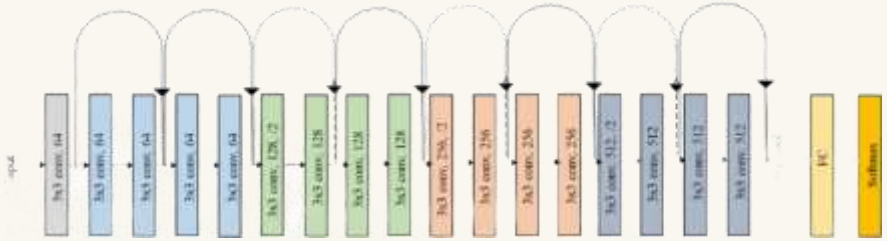


Fashion MNIST

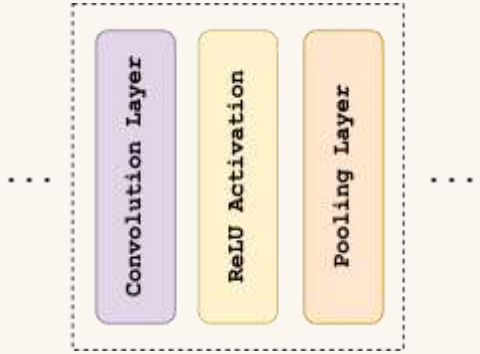
SVHN

CIFAR 10/100

Models:



ResNet - 18



Simple CNN

Evaluation Metrics

1. Accuracy on D_f and D_r after 2000 **pseudo-batches.**

Each pseudo-batch consists of filtered synthetic samples

2. The earliest pseudo-batch index at which forget-set accuracy begins to consistently increase (**tipping point**).

Tipping point is computed by tracking Acc_{forget}

Scenarios:

- Increasing forget set (D_f) size (forget class count)
- Logging every 50 pseudo-batches

Results

1.

Stability and Limitations of Single-Class GKT-Based Unlearning

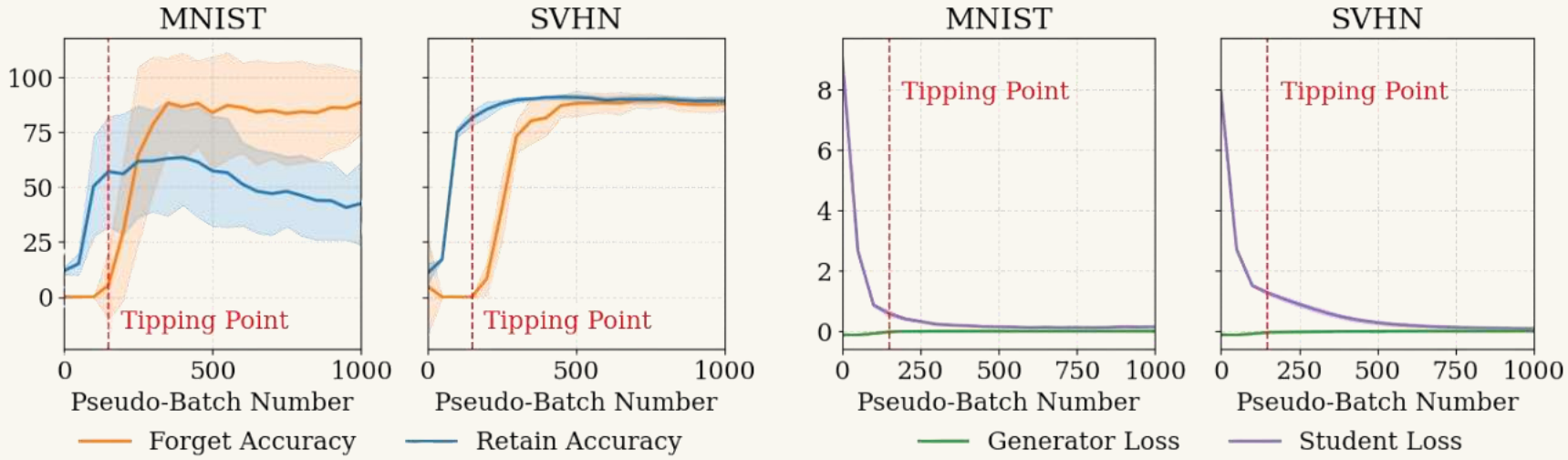
*Assessing how effectively GKT erases the influence of a **single forget class** while preserving performance on the retained classes.*

2.

Robustness of Multi-Class GKT-Based Unlearning

*Assessing how effectively GKT erases the influence of **multiple (1, 3, 5) forget classes** while preserving performance on the retained classes.*

Stability and Limitations of Single-Class GKT-Based Unlearning

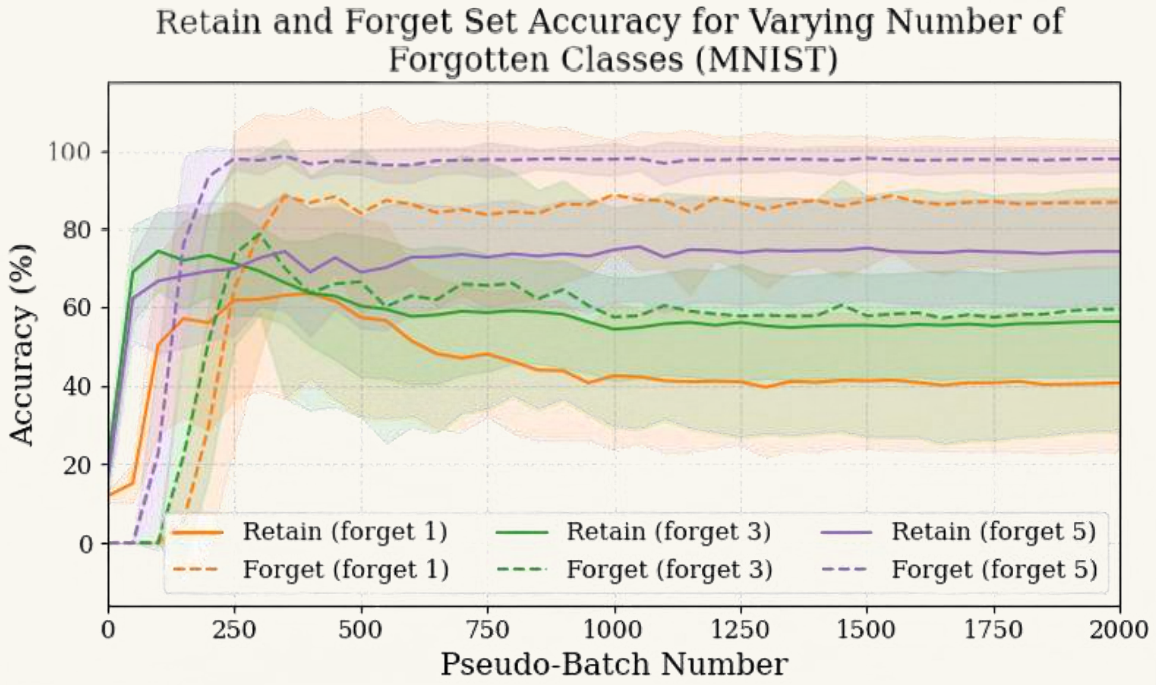


Stability and Limitations of Single-Class GKT-Based Unlearning

Table 1: Zero-shot unlearning results across five datasets and two architectures. All values are 10-run means.

Dataset	Acc. Before GKT (mean %)	Acc. After GKT (Retain) (mean %)	Acc. After GKT (Forget) (mean %)	Tipping Point (mean pb.)	Δ Retain Acc. (mean %)
AllCNN					
CIFAR-10	82.34	49.94	17.0	200	-32.4
CIFAR-100	57.60	49.95	17.01	200	-7.65
MNIST	99.40	40.65	86.61	150	-58.75
Fashion-MNIST	92.47	24.80	0.11	500	-67.67
SVHN	93.83	88.98	87.65	200	-4.85
ResNet-18					
CIFAR-10	83.49	43.82	0	-	-39.67
CIFAR-100	57.37	14.68	0	-	-42.69
MNIST	99.52	15.76	1.19	100	-83.76
Fashion-MNIST	93.24	24.47	0	-	-68.77
SVHN	93.94	81.70	26.09	350	-12.24

Robustness of Multi-Class GKT-Based Unlearning



Conclusion

Current zero-shot unlearning methods, such as GKT, are prone to instability over time and susceptible to adversarial exploitation.

Future Work

Thank You for Your **Attention!**

Takács Tamás
PhD Student @ ELTE

 *tamastheactual(at)inf(dot)elte(dot)hu*
 *tamastheactual.github.io*

References & Acknowledgments

We gratefully acknowledge the reviewers for their time and effort in evaluating our work.

[1] European Parliament & Council of the European Union. (2016, April 27). Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation), Article 17. Official Journal of the European Union, L 119, 1–88.

[2] Liu, B., Liu, Q., & Stone, P. (2022). Continual Learning and Private Unlearning. arXiv [Cs.AI]. Retrieved from http://arxiv.org/abs/2203.12817

[3] Barez, F., Fu, T., Prabhu, A., Casper, S., Sanyal, A., Bibi, A., … Gal, Y. (2025). Open Problems in Machine Unlearning for AI Safety. arXiv [Cs.LG]. Retrieved from http://arxiv.org/abs/2501.04952

[4] Chundawat, V. S., Tarun, A. K., Mandal, M., & Kankanhalli, M. (2023). Zero-Shot Machine Unlearning. IEEE Transactions on Information Forensics and Security, 18, 2345–2354. doi:10.1109/tifs.2023.3265506

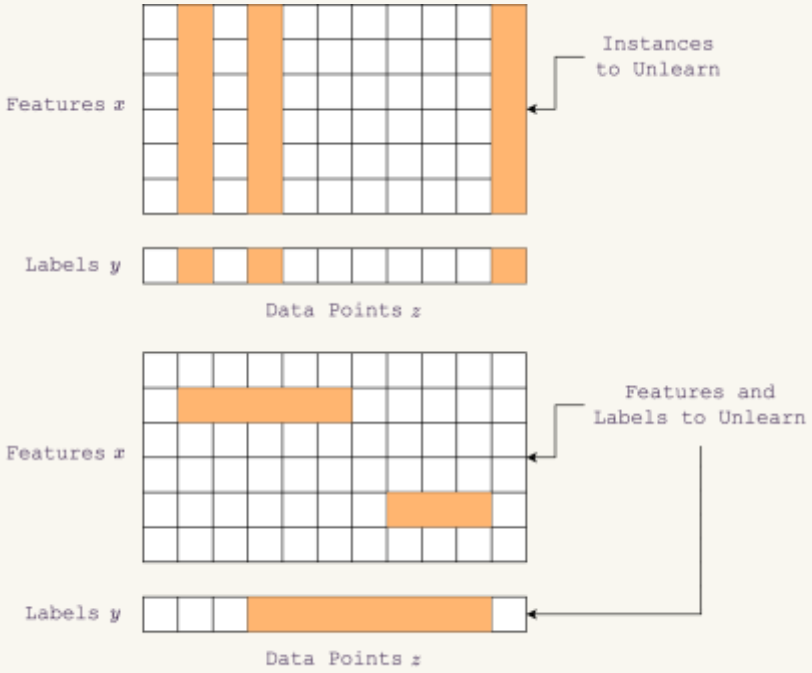
[5] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. 2023. Machine Unlearning of Features and Labels. arXiv:2108.11577 [cs.LG] https://arxiv.org/abs/2108.11577

[6] Sanyal, D., & Mandal, M. (2025). ALU: Agentic LLM Unlearning. arXiv [Cs.AI]. Retrieved from http://arxiv.org/abs/2502.00406

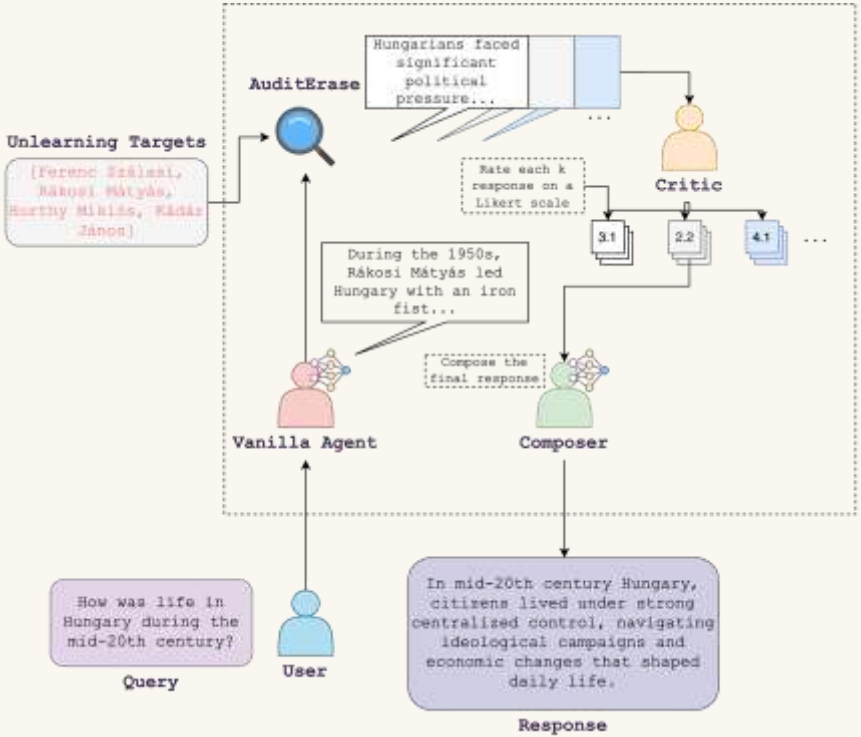
[7] Gong, C., Li, K., Yao, J., & Wang, T. (2024). TrajDeleter: Enabling Trajectory Forgetting in Offline Reinforcement Learning Agents. arXiv [Cs.LG]. Retrieved from http://arxiv.org/abs/2404.12530

[8] Ye, D., Zhu, T., Zhu, C., Wang, D., Gao, K., Shi, Z., … Xue, M. (2024). Reinforcement Unlearning. arXiv [Cs.CR]. Retrieved from http://arxiv.org/abs/2312.15910

Related Methods



Machine Unlearning of Features and Labels [5]



ALU: Agentic LLM Unlearning [6]