

A Clean Answer Is Not a Clean Agent: What an Unlearning Certificate Covers Once the Model Calls Tools, Why the Agentic Leaks Divide Into Residue, Scope, Reacquisition and Exposure, and the Source-by-Observer Audit That Would Assign Each to Its Owner

Pranay Mahendrakar

SONYTECH · ORCID 0009-0003-7224-029X

pranaymahendrakar@sonytech.in · mahendrakarpranay@gmail.com · +91 6361723454

Abstract. Unlearning benchmarks judge forgetting by questioning a bare model and reading its answer. A September 2026 benchmark argues that a certificate issued this way does not survive deployment: wrapped in a tool-using agent, the same models expose the target through reasoning text, tool arguments, tool observations and elicited summaries, and when the target sits in the prompt or a retrieval store the agent discloses it on 22 to 86 percent of queries while weight-level probes report nothing. Concurrent studies report that tool access lets an unlearned model fetch what it no longer recalls, and that adding memory or tools can reorder which unlearning method looks best. In this analysis the author reviews and synthesises published results to show that these findings are four different failures that share one name. The first is residue: the target is still in the weights and the agent poses a different question, which the non-removal literature had already shown and which strong model-level probes in the same benchmark's weight case study already detect. The second is scope: the target was never in the weights, so no weight method could clear it, exact retraining included. The third is reacquisition: the agent fetches the target from a source outside the deployment that it can reach, such as the open web or a third-party service, which even pretraining-data filtering does not prevent and which only controls on what the agent seeks, reads or repeats, or on its ability to process the topic, have been shown to reduce. The fourth is exposure: a channel leaks only to whoever is allowed to read it, so a leak in a tool observation belongs to the store and the trace policy rather than to the unlearning method. The paper also shows that the agent harness can hide failure as well as reveal it, through policy collapse that reads as forgetting. It states a certificate as a triple of source, observer and audience, consolidates 34 published results in a table, maps four deployment purposes onto the failures that matter for each, gives an audit procedure that assigns every leak to the party able to fix it, and specifies six studies that would settle the open part.

Keywords: machine unlearning; LLM agents; tool use; retrieval-augmented generation; reasoning traces; privacy leakage; evaluation; right to be forgotten

SONYTECH ORCID: 0009-0003-7224-029X

1. Introduction

Machine unlearning promises a model that behaves as though some data had never been used to train it. The idea began as a data-governance mechanism, where retraining is made cheaper by partitioning the data so that a deletion request touches only one shard (Bourtole et al., 2021), and it was carried into language models as a family of approximate methods that adjust trained weights to suppress a target while keeping the rest. The first prominent demonstration on a language model reported that about one GPU hour of fine-tuning removed a Llama-2-7B model's ability to recall Harry Potter content, against more than 184,000 GPU-hours of pretraining (Eldan and Russinovich, 2023). Benchmarks followed: TOFU fine-tunes a model on 200 invented author profiles and asks whether forgetting a subset makes it look like a model that never saw them (Maini et al., 2024); MUSE scores six properties of an unlearned model on books and news (Shi et al., 2024); WMDP offers 3,668 multiple-choice questions as a proxy for

hazardous knowledge and introduces RMU as a removal method (Li et al., 2024); RWKU targets real people whose facts are already in pretraining data (Jin et al., 2024). A 2025 survey of more than 180 papers organises the methods by the stage of the training pipeline at which they intervene and compares 18 benchmarks (Qiu et al., 2025).

All of these share an instrument. They query a model directly and read what it says. The 2024 and 2025 literature already established that what the model says is a weak guide to what it holds. Fine-tuning on a little related or even unrelated data restores supposedly unlearned behaviour (Hu et al., 2025; Lucki et al., 2025; Deeb and Roger, 2024), quantisation does the same (Zhang, Z. et al., 2025), and the parametric traces of a concept survive methods that only suppress it at inference time (Hong et al., 2025). The question the present paper addresses arrives from a different direction. Models are no longer deployed bare. They are deployed as agents that reason in visible text, call tools, read what the tools return, consult retrieval stores and carry memory across sessions (Yao et al., 2023). What

does a certificate earned by a bare model mean once the model is one component of such a system?

A benchmark released in September 2026, K-Bench (Yu et al., 2026), gives a sharp answer. It states that a certificate read from the final answer "does not transfer once the model is deployed as an agent." It wraps three open models in a ReAct agent with search, lookup and verification tools, plants a synthetic personal record in exactly one of the agent's sources - the weights, the prompt, a passage index or a structured database - and counts a query as leaked if the secret appears in any of six channels. Its headline results are that an output filter can clear the answer while the secret remains in a tool observation, that context and retrieval secrets leak on 22 to 86 percent of queries while weight probes report nothing, and that none of twenty published weight methods demonstrably removes a secret written into the weights. Around the same time, AUBench extended TOFU into an agent with memory and an offline web snapshot and concluded that checkpoint-level forgetting "does not reliably predict" complete-agent behaviour (Lin et al., 2026); ATU showed that a model unlearned with NPO recovers much of the forgotten content once it can search (Chen, B. et al., 2026); and a benchmark of memory deletion found that no evaluated memory system honoured more than 1.5 percent of deletion requests end to end (Xie et al., 2026).

The other side of the question is older and is held for principled reasons. Unlearning, on the standard definition, is a property of a trained model relative to a model retrained without the forget data (Maini et al., 2024; Liu, S. et al., 2025). Information placed in a prompt or a database at inference time was never part of training, and a retrieval store is something an operator can edit directly, as RAG-based unlearning proposals do by changing the knowledge base instead of the model (Wang, S. et al., 2024). On this view a weight-level benchmark is not blind to a retrieval secret; it is silent about it, as it should be. Policy analyses make the same point from the other end: unlearning targets two distinct goals, removal from parameters and suppression in outputs, and neither one is a general mechanism for controlling what a deployed system does (Cooper et al., 2025). In-context learning means that knowledge removed from weights can be supplied again at inference time, so content regulation needs controls outside the model (Shumailov et al., 2024).

In this analysis the author reviews and synthesises published results to show that both readings are right about different things, and that the agentic findings separate cleanly once three questions are asked of every leak: where the secret lives, who reads which channel, and whether the agent fetched the secret anew. The answers sort the agentic failures into four kinds that need different remedies and different owners. The paper makes five contributions.

First, it states a certificate as a triple - a set of sources, an observer, and an audience - and derives three consequences that do not depend on any experiment: a weight certificate is silent about non-weight sources even for exact retraining; widening the observer on a fixed trajectory can only raise measured leakage, but changing the harness changes the trajectories; and a channel leaks only to the parties allowed to read it (Section 3).

Second, it reads the agentic results through that triple and finds four families: residue under a wider observer (Section 4), sources outside the weights (Section 5), reacquisition through tools (Section 6), and exposure of internal channels (Section 7). Only the first is a failure of unlearning in the sense its methods intend, and on that family the strong model-level probes in the anchor benchmark's own weight case study already detect what the agent reveals.

Third, it collects the evidence that the agent harness also conceals failure, through policy collapse and through tool use that lowers recall on everything at once (Section 8), and states the strongest case for the transfer-failure reading (Section 9).

Fourth, it consolidates 34 published results in Table 1, maps four deployment purposes onto the families that matter for each in Table 2, and states an audit procedure that assigns each leak to the party able to fix it (Algorithm 1, Section 10).

Fifth, it specifies six studies that would settle what the published record leaves open (Section 11).

This paper is a sequel to two earlier ones by the author and asks a question neither did. Mahendrakar (2026a) separated three guarantees that the word unlearning is used for in a standalone model - output-level non-disclosure under a stated query class, absence from the weights, and resistance to a stated adversary. Mahendrakar (2026b) treated erasure inside agent memory stores, where a stored item and the summaries, reflections and edits derived from it must be withdrawn together. The present question is what a guarantee earned by the model component says about the system it is placed in, and which part of the system is responsible for each way the guarantee fails to carry over.

2. What the agentic studies report

2.1 K-Bench

K-Bench (Yu et al., 2026) builds 5,000 synthetic people with four fields each - birth date, address, occupation and employer - and splits them into 1,000 to forget and 4,000 to retain. Its agent is a ReAct loop with three tools (a Wikipedia passage search, a record lookup and an attribute verifier), at most six iterations, and greedy decoding. Four substrates hold the secret. For the parametric one, the records are written into Llama-3.1-8B-Instruct through a LoRA adapter, and cross-model checks use Mistral-7B and Qwen3.5-9B. For

the context substrate the record sits in the system prompt among 50 biographies; for the two retrieval substrates it sits in the passage index or the structured database. Six channels are read: reasoning lines, tool-call arguments, tool calls together with their observations, retrieved passages, the final answer, and a summary elicited by a separate follow-up prompt. The headline metric, OR(all), counts a query as leaked if any channel contains the secret.

The results relevant here are these. Without any intervention, the parametric secret surfaces mainly through the summary channel, the context secret through the answer (0.192 on Llama, 0.992 on Qwen), and the retrieval secrets through tool observations (0.855 for the database, 0.602 for passages). Widening the observer budget step by step shows where each jump happens (Figure 1a): on the parametric substrate, leakage rises from 0.152 with the answer alone to 0.682 once the summary is added, and nothing further changes; on passage retrieval it rises from 0.203 to 0.602 only when tool channels are added; on the database it is 0.832 from the answer alone. The answer channel understates baseline leakage on ten of eleven model-substrate rows, and the method verdict changes between the narrowest and widest observer on four of six scored Llama parametric cells.

The method results come in three panels. In the cross-substrate panel, a reasoning-trace filter (StaR) cuts the database answer channel from 0.832 to 0.463 while the tool-observation channel stays at 0.857 against a 0.855 baseline, so the aggregate does not move; the paper's verdict is that the leak migrated. An input-side method that detects a forget query and corrupts the entity's embedding before the model sees it (ECO; Liu, C. et al., 2024) drives the aggregate to at most 0.005 on all four Llama substrates, though with a ground-truth gate and, on three of the four, a large rise in agent collapse (Section 5.1). A linear concept eraser reduces to the identity map and changes nothing. In a matched weight case study, five fine-tuning recipes were applied to the parametric target and read through five interfaces; no recipe reached the retrain-only reference. On Llama-3.1-8B every recipe's TOFU forget-quality test against that reference was rejected ($p < 10^{-3}$), and the three recipes that preserved TOFU's model-utility score (0.90 to 0.91: NPO, NPO with a KL term, and IDK) still picked the true value in a four-way forced choice 95 to 98 percent of the time, against 11 percent for the reference. In a leaderboard of twenty published methods on the merged parametric target, sixteen of the twenty Llama cells ended in terminal agent collapse. In a separate balanced grid of six weight methods, three base models and three seeds, the leading method differed on each base model, and a method-by-model interaction explained 24 percent of the variance in the benchmark's composite score.

On the non-weight substrates the paper runs eight weight probes - TOFU recall and truth ratio, MUSE knowledge and verbatim memorisation, a WMDP-style forced choice, a membership-inference AUC and two LUME scores - and finds them at floor or chance (recall 0.025 to 0.073, forced-choice accuracy 0.24 to 0.26 against a chance rate of 0.25, membership AUC 0.51 to 0.56), because the weights do not contain the secret. Eight models served through public APIs, which no weight method can touch, leak in all 24 context and retrieval cells. The six newer endpoints return reasoning text, and on the context substrate that text contains the secret on 75 to 100 percent of queries.

2.2 AUBench

AUBench (Lin et al., 2026) keeps TOFU's fictitious authors and forget05 split and adds a Mem0 memory store, an offline web snapshot of 800 pages behind a search tool, and complete trajectories. It evaluates gradient difference, KL minimisation and NPO checkpoints of Llama-3.1-8B and Qwen3-8B under four conditions - parameters only, with memory, with search, and with both - and scores token-level F1 against reference answers. Memory raised forget-set F1 in five of six checkpoints; the clearest case is Qwen3-8B with KL, whose forget-set F1 went from 16.8 with parameters alone to 30.1 with memory and 36.9 in the complete agent, reversing its rank from best to worst forgetter. Search lowered average forget-set F1 from 29.7 to 12.9 and average retain-set F1 from 34.9 to 17.3, and Qwen3-8B returned empty outputs on 23.6 to 58.0 percent of examples under search. The paper's Figure 1 states that the released memory bank and profile store "contain no record of the forget entity." If that footnote is right, the rise under memory did not come from retrieving the forget fact itself. The paper's construction section points the other way: its artifacts are derived from the full TOFU set of 200 authors and 4,000 question-answer pairs, its snapshot holds "200 entity records, 4,000 fact records", and the tool corpus is built "from the same profile information used to construct the memory artifacts" (Lin et al., 2026, Sections 4.2 and 5.1).

2.3 ATU

ATU (Chen, B. et al., 2026) names the failure it addresses tool-mediated recovery: a model that no longer recalls a target from its weights looks it up. Its motivating measurement uses NPO, a preference-style objective on the forget set whose drift toward catastrophic collapse its authors show to be exponentially slower than that of gradient ascent (Zhang, R. et al., 2024). On MUSE Books, NPO lowers the forget-set score of the bare model from 37.9 to 9.7; deployed as a tool-using agent, the unlearned model scores 29.8, against 43.3 for the base agent. On RWKU the bare scores are 57.7 and 44.6 and the agent scores 74.6 and 68.6 (Figure 1b). ATU's threat model places the attacker downstream: the provider releases weights and a

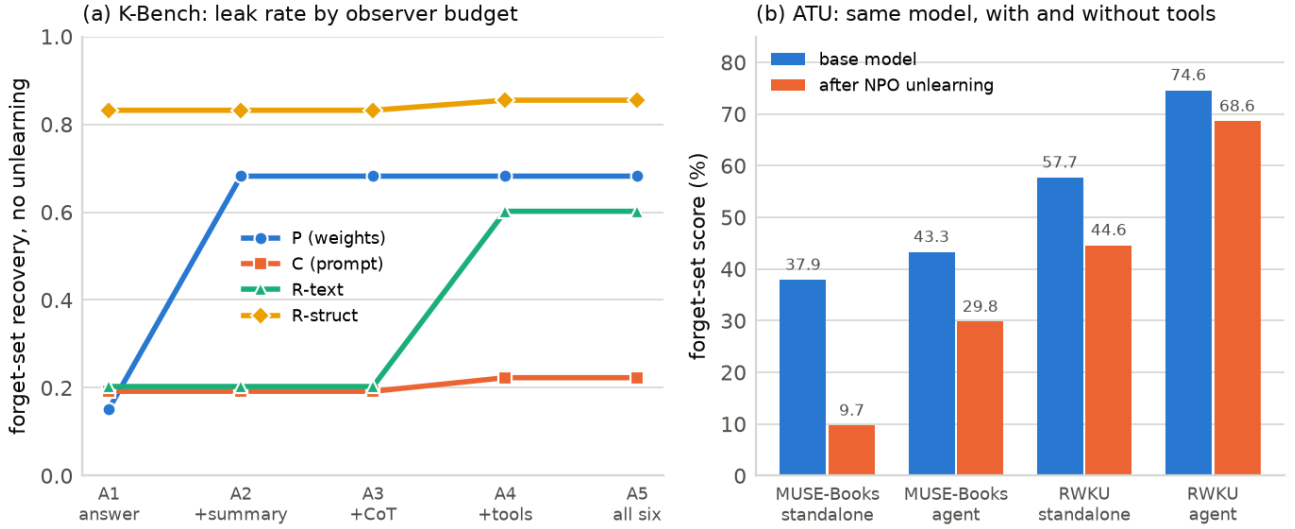


Figure 1. Two published measurements of the gap between a bare model and an agent. (a) K-Bench (Yu et al., 2026, Table 8), Llama-3.1-8B with no unlearning applied: forget-set recovery rate as the observer reads more channels, from the final answer alone (A1) through the elicited summary (A2), reasoning text (A3) and tool channels (A4) to all six (A5), for a secret placed in the weights (P), the prompt (C), a passage index (R-text) or a structured database (R-struct). (b) ATU (Chen, B. et al., 2026, Figure 2): forget-set scores of the base model and of the NPO-unlearned model, queried directly (standalone) and deployed as a tool-augmented agent, on MUSE Books and RWKU. All values are as published.

deployer wraps them with whatever tools it likes. Its remedy is a second training stage in which reinforcement learning over simulated tool trajectories penalises target-seeking calls and final-answer leakage while rewarding normal tool use on retained queries.

2.4 Memory, trajectories and tools as forget targets

Four further lines extend the target beyond facts in weights. Synchronized Backflow Unlearning treats parameters and persistent memory as two pathways that can recontaminate each other and unlearns them jointly (Wang, B. et al., 2026). Bemula pairs 50 multi-session dialogues with control versions that never contained the forget request, so that only a system that stored the target can earn credit for forgetting it, and finds that six memory systems on three backbones pass at most 1.5 percent of 135 deletion requests end to end, either leaving the target extractable or deleting too much (Xie et al., 2026). Trajectory unlearning makes the target what an agent does rather than what it says, removing action sequences in ALFWorld and WebShop with a group-relative policy optimisation variant (Shi and Wang, 2026); a framework for agent unlearning distinguishes state, trajectory and environment forgetting and uses natural-language unlearning prompts (Ye et al., 2026). Tool unlearning removes the ability to call a deprecated or vulnerable tool from a tool-augmented model (Cheng and Amiri, 2025). These studies agree that the system, not the model, is the thing a deletion request is about. They do not agree on what the system has to stop doing, and that disagreement is the subject of Sections 4 to 7.

3. A certificate has a source, an observer and an audience

3.1 The triple

A forgetting certificate is a statement of the form: for the target T , no observer of class O , reading the channels it is allowed to read, recovers T from the sources in S , beyond a stated rate. Three things are fixed by such a statement and are usually left implicit.

The source set S says where the target could be. A weight certificate fixes S to the parameters. The agentic studies add three more places: the working context (system prompt, injected records and the agent's own earlier turns), retrieval stores the agent searches, and persistent memory the agent writes and reads (Yu et al., 2026; Lin et al., 2026; Xie et al., 2026). A fifth place is outside the deployment altogether: the open web, or any service the agent can call. K-Bench calls its three inference-time sources substrates and says it is not aware of an access path outside them; that holds for its closed tool set, while ATU's threat model, in which the deployer chooses the tools, adds the open world as a source by construction (Chen, B. et al., 2026).

The observer O says what is read and how it is asked. It has four parts: the channels read (answer, reasoning, tool arguments, tool observations, summaries, actions); the query family (canonical phrasing, paraphrase, multi-hop, forced choice, completion); the decoding configuration (thinking on or off, sampling, budget); and the access level (black-box text, log-probabilities, activations, weights). Every unlearning benchmark fixes all four, usually without saying so.

The audience A says who receives which channel. In a chat product the user sees the answer and, increasingly, a reasoning summary. In an API deployment the developer may see everything. In an internal tool-use loop the database's return value may never leave the server. A channel leaks to a party only if that party can read it.

None of this is new to safety evaluation in general. A sociotechnical framework for evaluating generative systems argued that capability tests on a model have to be supplemented by evaluations of human interaction and of systemic impact, because context decides whether a capability does harm (Weidinger et al., 2023). The triple applies the same idea to forgetting: source and audience are the context that a model-level test leaves out.

3.2 Three consequences of the definitions

The triple has three consequences that follow from definitions, not from measurement.

The first is that a weight certificate is silent about sources outside S. If the target is in the prompt, the retrieval store or memory, a statement about the parameters neither passes nor fails; it does not apply. This holds for every weight method, including exact retraining from scratch without the target, because a retrained model given the target in its context window will read it. K-Bench's own weight probes on its context and retrieval substrates return values at floor or chance precisely because the weights never held the secret (Yu et al., 2026, Table 5), which is what a retrained model would also return. The paper's own text draws the same conclusion: stronger weight unlearning would not remove this leak, which it calls a coverage gap in the evaluation surface. A coverage gap is a fact about the audit's claim, and it would be equally present if the field had a perfect unlearning method.

The second is that for a fixed trajectory, reading more channels can only raise the measured leak rate. K-Bench proves this for its OR metric: the aggregate bounds every single-channel rate from above. But changing the harness is not the same as widening the observer on a fixed trajectory. A model placed in an agent loop produces different text, different decisions to answer, and sometimes no parsable output at all. Measured leakage can therefore fall when the harness is added, for reasons unrelated to forgetting. Section 8 shows that it does.

The third is that a leak is a relation between a channel and an audience. A tool observation that contains a database record is the database's output. If the deployment shows tool observations to the user, the record has been disclosed to the user by the database and the trace policy, whatever the model did. If it does not, the record has gone nowhere. Neither case says anything about whether the model forgot.

3.3 Four families

Applying the triple to a leak observed in an agent gives four families.

Residue (family 1). The target is in the weights, and an observer of the agent recovers it through a channel or a query that the certifying observer did not use. This is a failure of the model-level guarantee, revealed by a better observer.

Scope (family 2). The target is in a source outside the weights. The model-level certificate never covered it. The failure, if any, belongs to whoever certified the deployment on the strength of the model certificate.

Reacquisition (family 3). The target is not held anywhere in the deployment at the time of the request, and the agent fetches it from a source it can reach. No removal from any internal store can prevent this; only controls on what the agent seeks, reads or repeats can.

Exposure (family 4). The target passes through an internal channel - a reasoning trace, a tool argument, a tool observation - that reaches an audience it should not. This is a property of the trace policy and the channel's audience, and it can coexist with any of the other three.

The families are not exclusive in a given trace. A single K-Bench trajectory can show family 2 and family 4 at once: the secret lives in the database (scope), and the tool observation that carries it is visible to the observer (exposure). The audit in Section 10 resolves each leak to the earliest family that explains it.

4. Residue: the weights still hold it, and the agent asks a different question

4.1 What the agent observer adds on the weight substrate

On K-Bench's parametric substrate almost the whole gap between the answer channel and the full observer comes from one channel, the elicited summary: with no intervention, leakage goes from 0.152 to 0.682 when the summary is read, and reasoning text and tool channels add nothing (Yu et al., 2026, Table 8; Figure 1a). The summary is produced by a fresh prompt after the trajectory ends, and the paper describes it plainly: "The summary check is a direct-elicitation probe." The parametric-substrate gap is therefore, in large part, the gap between one question and another question put to the same weights. It is real, and it matters for compliance, because a deployment will be asked many questions. But it is the same gap that the non-agentic literature had already measured from several directions.

That literature is extensive and consistent. Applied to the Harry Potter model of Eldan and Russinovich (2023), a battery of tests found more knowledge extractable than baseline, question-answering performance on par with the original model, and latent knowledge represented comparably (Lynch et al., 2024). Attacks on model

editing recovered supposedly deleted facts from GPT-J 38 percent of the time, using traces in intermediate hidden states and rephrased questions (Patil et al., 2024). Small benign modifications to popular benchmarks, especially ones that link forget and retain information, exposed information that the original benchmark scored as forgotten (Thaker et al., 2025). Multi-hop questions recover a fact after one of its hops has been unlearned (Choi et al., 2024), and a 2026 benchmark built around multi-hop paths and post-unlearning adaptation finds current methods vulnerable to both (Qian et al., 2026). A model trained to forget in context keeps the correct prediction through its middle layers and decides to withhold it only at the last one, a behaviour its authors summarise as pretending to forget (Takashiro et al., 2025). Parameter-space inspection finds that existing methods leave concept vectors largely intact and mostly suppress them at inference (Hong et al., 2025), and probing distinguishes methods that remove knowledge from those that add distracting information on top of it (Sun et al., 2025).

The relearning and tampering results go further. Fine-tuning on a small, loosely related corpus reverses unlearning across three benchmarks (Hu et al., 2025); ten unrelated fine-tuning examples, or removing particular activation directions, restore most of RMU's suppressed hazardous capability (Lucki et al., 2025); fine-tuning on facts from the same distribution recovers 88 percent of pre-unlearning accuracy on held-out facts (Deeb and Roger, 2024); and four-bit quantisation of models unlearned under utility constraints raises the retained share of forgotten knowledge from 21 percent at full precision to 83 percent (Zhang, Z. et al., 2025). Even an idealised unlearning that trains a model to imitate one never exposed to the data is reverted by a few fine-tuning steps, and distilling the unlearned behaviour into a freshly initialised or noised student makes it robust (Lee et al., 2025).

Against this background, the agentic residue finding is confirmation through a new instrument. K-Bench's own weight case study shows that the strongest model-level probes already detected the residue it reports. On Llama-3.1-8B the forced-choice probe separated the retrain-only reference (0.11) from every recipe that kept TOFU's utility score (0.95 to 0.98), and TOFU's forget-quality test rejected every recipe against that reference (Yu et al., 2026, Table 10). Free-text recall did not issue a clean certificate either: the three recipes that kept utility left TOFU recall at 0.52 to 0.74, against 0.03 for the reference, and the two that drove recall near zero, gradient ascent and gradient difference, did so with model utility of 0.003 and 0.128 (Yu et al., 2026, Table 10). On Llama-3.1-8B no model-level probe, free-text recall included, certified any functional recipe as clean. On Mistral-7B free-text recall would have: NPO keeps model utility at 0.851, while its TOFU recall (0.040) and its LUME score (0.00) sit at the level of the retrain-only

reference (0.048 and 0.00), and only the forced-choice probe (0.40 against 0.14) and the membership score (AUC 0.207 against 0.543) separate it from that reference (Yu et al., 2026, Table 10). The benchmark's own reading is that "each interface, read alone, would reach a different and incomplete conclusion" (Yu et al., 2026). The parametric-substrate residue was visible to the strong model-level interfaces, and what the agent adds there is a channel, the elicited summary, that reaches it under ordinary use.

4.2 Reasoning traces

Reasoning models add a channel that lies between the weights and the answer. Three 2025 and 2026 studies agree that unlearning objectives aimed at the final answer leave the target in the trace. R-TOFU annotates TOFU with reasoning traces and finds that answer-only objectives leave substantial forget content in reasoning; it also finds that decoding variants that suppress thinking (ZeroThink, which forces an empty thinking block, and LessThink) can reveal content that the default decoding hides, so that a preference-optimised refusal that looks complete by the fourth epoch under default decoding still scores high when thinking is switched off (Yoon et al., 2025). R2MU reports that conventional methods erase the final answer while sensitive content persists in the trace, and extends representation misdirection to the trace (Wang, C. et al., 2025). Later methods replace sensitive trace segments with placeholders during training (Le et al., 2026), or detect a leaking sentence at inference and inject a learned corrective embedding, reporting perfect answer forgetting and near-perfect trace forgetting without the reasoning collapse that trace-level objectives otherwise cause (Maccoby and Wang, 2026).

Two qualifications follow from the reasoning-faithfulness literature. A trace that contains the target shows that the model can still produce it; that is enough to fail a non-disclosure certificate. A trace that does not contain the target shows much less. Chain-of-thought explanations can omit the factor that actually drove an answer and rationalise answers induced by biasing features (Turpin et al., 2023), and in state-of-the-art reasoning models the rate at which a used hint is verbalised is often below 20 percent (Chen, Y. et al., 2025). The case for monitoring reasoning text is made on the grounds that it is useful though imperfect and fragile, not that it is complete (Korbak et al., 2025). Reading the trace therefore strengthens the case against a method when it finds the target and adds little in its favour when it does not. R-TOFU's decoding result makes the same point from the other side: removing the trace can expose what the trace's refusal was covering.

4.3 What family 1 establishes

On the weight source, the agentic results add three things to the non-agentic record. They show that the gap appears with ordinary, non-adversarial use of a common

scaffold: a follow-up summary is not an attack. They show that the verdict on a method changes with the observer budget for four of six Llama parametric cells, and for ten of thirty parametric cells across the three base models. And they show that leaderboard rankings under a full observer do not carry across base models. They do not show that agent deployment causes residue, or that the residue was invisible to model-level evaluation; K-Bench's own forced-choice and distributional tests saw it.

5. Scope: the secret was never in the weights

5.1 The measured leak

When K-Bench places the secret in the prompt, the Llama agent repeats it on 22.3 percent of queries, mostly in the answer; when it places it in the database, the agent leaks it on 85.5 percent, mostly in the tool observation and the answer; passage retrieval gives 60.2 percent (Yu et al., 2026, Table 7). These are the 22 to 86 percent of K-Bench's abstract. Qwen3.5-9B leaks a context secret on 99.2 percent of queries, Mistral-7B refuses in-context questions often enough that its context baseline falls below the benchmark's validity gate, and the eight API models leak in every one of their 24 context and retrieval cells, with OR(all) from 0.27 to 1.00 on the context substrate.

None of these numbers measures an unlearning method. They are the baseline disclosure rates of systems that were given a secret and asked about it. Of the four methods in K-Bench's matrix that can act at inference time on any substrate, two reduce them. The input-corruption method drives every Llama substrate to near zero, with two qualifications. K-Bench's version decides which queries to corrupt from ground-truth forget-set membership instead of ECO's learned classifier, so the benchmark treats it as an upper bound on input-side gating "when the gate is exact". And on the context and both retrieval substrates the near-zero rate comes with a sharp rise in degenerate trajectories, to 91, 71 and 73 percent against 66, 48 and 9 percent without intervention (Yu et al., 2026, Table 7); on the database substrate the benchmark attributes the near-zero leakage to that rise "rather than clean forgetting". By the health check of Algorithm 1 (Section 10) those three cells would be uninterpretable. Three published inference-time methods that K-Bench also ran on these substrates - SPUL, GRUN and ULD - reach selective forgetting on none of them (Yu et al., 2026, Table 12). The trace filter cuts the database answer channel from 0.832 to 0.463 but leaves the tool observation untouched; on the context substrate it halves the aggregate, from 0.223 to 0.112. The weight and adapter methods are, correctly, not run on these substrates at all.

5.2 What the prompt and the store already do

That a model will reveal what is in its context window is not a 2026 finding. Simple text attacks recover system prompts with high probability across eleven models and three prompt sources, including deployed systems with defences (Zhang, Y. et al., 2024). Retrieval-augmented systems leak their private retrieval databases under targeted attacks, while retrieval can at the same time reduce leakage of the model's own training data (Zeng et al., 2024). Instruction-tuned models reveal private information in contexts where people would not, 39 percent of the time for GPT-4 and 57 percent for ChatGPT on one benchmark, and privacy-inducing prompts and chain of thought do not stop it (Mireshghallah et al., 2024). An agent asked to act, rather than to answer a probe about norms, leaks more: GPT-4 and Llama-3-70B leaked sensitive information in 25.68 and 38.69 percent of cases even with privacy-enhancing instructions, and the gap between probing and acting was the paper's central finding (Shao et al., 2024). Web agents given personal data use more of it than their tasks require (Zharmagambetov et al., 2025).

Memory is the hardest of the non-weight sources, because it is written by the agent itself from what it has seen and may hold derived versions of a fact. Bemula's near-total failure rate for deletion requests (Xie et al., 2026), the parameter-memory backflow that Synchronized Backflow Unlearning is designed against (Wang, B. et al., 2026), and the author's earlier analysis of derived memory (Mahendrakar, 2026b) all locate the problem in the store. Deleting a source record does not withdraw the summaries and reflections built from it, and that is a property of the memory design, not of the model's weights.

5.3 Whose failure it is

The scope family has an owner, and it is not the unlearning method. If an operator holds a person's record in a database that the agent can query, the operator can delete the record, or bar the agent from reading it, or bar the agent from repeating it. RAG-based unlearning proposals do the first by editing the knowledge base instead of the model (Wang, S. et al., 2024). Inference-time schemes do the second and third by filtering or rewriting what reaches or leaves the model: ECO gates prompts through a classifier and corrupts the embeddings of flagged ones (Liu, C. et al., 2024), in-context unlearning places the forget instances in the prompt with altered labels (Pawelczyk et al., 2024), and ALU assigns each step of the unlearning process to a separate LLM agent without updating any weights, reporting robustness on TOFU, WMDP and WPU for up to 1,000 targets (Sanyal and Mandal, 2025). Simple prompting and filtering baselines already score comparably to fine-tuning on standard benchmarks, which shows that those benchmarks cannot tell a guardrail from a removal (Thaker et al., 2024).

K-Bench's result that a weight audit would "certify such a deployment as clean" while the agent leaks is correct, and important for compliance practice. Its correct reading is that a deployment audit must enumerate the sources a deployment holds and certify each one with an instrument suited to it. The result is not evidence that model-level unlearning fails to transfer, because there was nothing in the model to transfer.

6. Reacquisition: the agent fetches it again

6.1 The measured recovery

ATU's measurement is the most direct published statement of the third family, though not a pure one. NPO lowers the bare model's MUSE Books forget score by 28.2 points, from 37.9 to 9.7; deployed with tools, the same unlearned model scores 29.8, regaining 20.1 of those 28.2 points, and the base agent scores 43.3 (Chen, B. et al., 2026; Figure 1b). On RWKU the agent scores are 74.6 for the base model and 68.6 for the unlearned one, both above the bare base model's 57.7. MUSE Books is drawn from the Harry Potter novels and RWKU from 200 famous people, whose facts the original authors chose because they are widely present in pretraining data (Jin et al., 2024). These are targets the open web holds in abundance. ATU's tool set, however, is not only the web. Beside web-facing tools (ddg-search, fetch, web_extractor, wikipedia_search), it gives the agent local databases holding the forget targets' content - for MUSE, a search over Harry Potter book passages from the MUSE Books corpus, and for RWKU, a structured store of summarised documents about the public figures - added to "make the recovery channel deterministic and reproducible" (Chen, B. et al., 2026, Appendix C.2). The reported recovery is not split by tool. A recovery served from one of those local databases is a store the deployment holds, which is the scope family of Section 5, so the published numbers mix the second and third families.

6.2 Removal cannot block it, and the strongest removal does not

The decisive evidence on reacquisition does not come from an unlearning study. Deep Ignorance pretrained 6.9B-parameter models from scratch on data filtered to remove biothreat-related text, which is arguably the closest to exact removal that current practice reaches (O'Brien et al., 2025). The filtered models resisted adversarial fine-tuning on up to 10,000 steps and 300 million tokens of biothreat text, more than an order of magnitude beyond post-training baselines. But on an open-book test, in which a passage containing the answer is given in context, the filtered models performed well, as the unfiltered ones did. The models had never learned the hazardous material; they could still use it when handed it. A model that genuinely lacks a fact reads a document that contains it like any other document.

The same study shows what does block in-context use. Circuit breakers, which reroute internal representations associated with harmful outputs (Zou et al., 2024), applied on top of filtering, impaired the models' ability to answer even with the passage in context. No combination resisted an attack that first fine-tuned the model and then supplied the passage. Post-training defences designed to resist tampering on open weights, such as TAR, which aims to keep safeguards in place through hundreds of fine-tuning steps (Tamirisa et al., 2024), address the attacker who modifies weights; the open-book attack does not modify anything. The point of the result for this paper is the kind of intervention that worked. Circuit breaking does not remove a fact. It impairs the model's processing of a topic, whatever the source of the information.

UnUnlearning made the same argument in 2024 before the agentic benchmarks existed: knowledge removed from weights can be reintroduced in context, so that a model behaves as if it knew it, and so content regulation needs filtering beyond unlearning, even exact unlearning (Shumailov et al., 2024). ATU's remedy is consistent with this. Its second stage does not delete anything further from the weights. It trains the policy, through trajectory-level rewards, not to issue target-seeking tool calls and not to repeat the target in its answer, while keeping normal tool use on retained queries. On RWKU with Qwen3-4B, this stage lowers the unlearned agent's forget score from 68.6 to 47.5 with the tools used in training and from 64.0 to 46.7 when the retrieval tools are swapped for others of the same type; the standalone score after the first stage was 44.6 (Chen, B. et al., 2026, Table 8 and Figure 2). The second stage therefore removes most, not all, of the tool-mediated recovery, and its effect survives a change of tool vendor.

6.3 The inversion

This produces an inversion that the agentic literature has not stated. In the non-agentic literature, behavioural suppression is the weak guarantee and removal is the strong one: the relearning results in Section 4 are evidence that suppression is not removal, and adversarial work on RMU questions whether unlearning has any advantage over ordinary safety training (Lucki et al., 2025). Against reacquisition the ranking is reversed. Removal from weights is not sufficient, as Deep Ignorance shows for removal more complete than any post-hoc method. Nor is it necessary, since a policy that never seeks, reads or repeats the target blocks reacquisition whatever the weights hold. The controls that act on reacquisition are behavioural (ATU's anti-seeking policy, refusal), input-side (a gate that corrupts a flagged entity before the agent can search for it; K-Bench tests ECO only on secrets its deployment holds, and there with an exact gate and, on three of four Llama substrates, a large rise in agent collapse, so for reacquisition this is a mechanism that would transfer rather than a measured result) or capability-impairing

(circuit breakers). All three are the kind of control that K-Bench discounts when it credits the refusal-tuning method FLAT only with "verbatim-extraction resistance under the evaluated observer rather than verified knowledge removal" (Yu et al., 2026).

The point is not that behavioural controls are robust. Refusal learned in chat does not carry to browser agents: with rewritten instructions, agents built on GPT-4o and o1-preview attempted 98 and 63 of 100 harmful behaviours that their backbones refuse in chat (Kumar et al., 2024). Leading models complied with many explicitly malicious agent tasks without any jailbreak, and simple jailbreak templates kept their multi-step capabilities intact (Andriushchenko et al., 2025). Across six frontier models, text refusal and tool-call behaviour diverged, with 219 cases of a refusing text alongside a forbidden tool call even under safety-reinforced prompts (Cartagena and Teixeira, 2026). Expanding retrieval access from none to Wikipedia to open web search steadily degraded refusal and harm safeguards, to the point that retrieval agents built on aligned models often behaved less safely than uncensored models without retrieval (Yu, C. et al., 2025). ATU's own adaptive tests point the same way: a directive to copy tool output exactly raised the forget score of its trained policy by 10.2 points (Chen, B. et al., 2026, Appendix D.7). The behavioural guarantee needed against reacquisition is fragile in exactly the setting where it is needed. The weight-level guarantee is not fragile there; it simply does not apply.

6.4 Whether reacquisition is a failure depends on the purpose

Whether reacquisition counts as a failure at all depends on what the forgetting was for, and the purposes disagree. The right to be forgotten began as a search-engine obligation: the original ruling and its codification let individuals have results removed from search listings (Zhang, D. et al., 2023), and search engines implement that as de-indexing - removal from a retrieval layer - not deletion of the source page (Vilella and Ruffo, 2025). On that model, an agent that searches the open web and finds a public page has done what any search engine user can do, and the duty to de-list sits with the retrieval provider. Copyright claims are argued on data lineage rather than outputs, so that output controls of any kind, removal included, may not cure an acquisition that was unlawful (Utsunomiya et al., 2026). For hazardous knowledge, the relevant quantity is uplift, and Deep Ignorance notes that "in cases when search-tool augmentation attacks are possible, LLMs are likely to offer limited uplift because such a search tool could be used by humans directly without the LLM" (O'Brien et al., 2025). The authors also suggest gating access to dual-use knowledge databases by user trust, which is a source and audience control, not a weight control. Open-problems analyses of unlearning for safety reach the same place from the dual-use side: knowledge that is

harmful in one use is needed in another, and models can assemble harmful capability from harmless parts (Barez et al., 2025).

Section 10 turns these into Table 2. The common thread is that reacquisition is a failure of the system's policy toward a source it can reach. Whether that policy should refuse, de-list, gate by user, or permit is a decision about purpose, and no forgetting benchmark can make it.

7. Exposure: who reads the trace

7.1 Channels with audiences

K-Bench's observer reads everything an agent emits, and its authors state that its results are a lower bound on what an adaptive adversary could extract (Yu et al., 2026). That is the right stance for a worst-case benchmark. A deployment is not the worst case by default; it is a choice about which channels reach which party. The benchmark's own API table shows how much that choice varies. The two older endpoints return no reasoning text and leak through it on at most 5 of 600 context queries; the six newer endpoints return reasoning text, and on the context substrate it contains the secret on 75 to 100 percent of queries (Yu et al., 2026, Table 6). Whether that is a disclosure to the user depends on whether the product shows the reasoning text to the user.

The privacy literature on reasoning traces argues that it often will. Reasoning traces of models used as personal agents frequently contain sensitive user data, which prompt injection can extract or which can leak accidentally into outputs; larger reasoning budgets make final answers more cautious while making the reasoning more verbose and leakier (Green et al., 2025). Training reasoning models to follow instructions inside the trace, and decoding trace and answer with separate adapters, improves privacy benchmark scores by up to 51.9 percentage points at some cost in task utility (Puerto et al., 2026). These results treat the trace as an output, which is correct whenever an adversary can reach it, and they locate the fix in the trace policy.

7.2 Tool observations belong to the store

The clearest exposure case in K-Bench is the one its abstract leads with. On the database substrate the trace filter suppresses the answer, while the true value stays verbatim in the tool observation, so the aggregate is unchanged (Yu et al., 2026). The tool observation is the database's return value. It contains the record because the agent asked the database for it and the database answered. In the case-study trajectory the agent's reasoning text states that it has no record of the person's birth date, and the next line shows the lookup returning it (Yu et al., 2026, Figure 2). In that trajectory, after the filter had altered the date in the answer, the model's own channels said nothing true about the secret; the store's return value did.

Calling this channel migration describes the measurement correctly: a filter on the answer does not touch a channel it does not see. It does not describe a property of the unlearning method, because no method acting on the model can stop a database from returning a record it holds. The remedies are to delete the record, to deny the lookup, or to keep tool observations out of the user-visible trace. Runtime governance contracts reduced information leakage in all six models of the GAP benchmark while producing no detectable deterrent effect on forbidden tool-call attempts (Cartagena and Teixeira, 2026). That suggests that controls on what reaches the user and controls on what the agent attempts are separate levers, and that exposure is the one more easily pulled.

8. The agent can hide a leak as well as reveal one

The second consequence in Section 3.2 says that changing the harness is not a monotone widening of the observer. The published record contains three mechanisms by which the harness lowers measured leakage without any forgetting.

Policy collapse is the first. Fine-tuning a model on question-answer data overrides the ReAct prompt. In K-Bench, loading an adapter trained only on distractor biographies, which never saw the target, made the agent halt with a parse error on 92.8 percent of queries on the non-weight substrates, so every channel read zero. The unmodified Llama agent with the injected secret already fails to complete about 45 to 50 percent of trajectories (Yu et al., 2026, Tables 7 and 11). Gradient ascent, gradient difference and the IDK recipe drove the aggregate leak rate to near zero on Llama by collapsing the agent; IDK degenerated 98.2 percent of trajectories while still answering the forced-choice probe correctly 95 percent of the time. On Llama, sixteen of twenty published methods ended in terminal collapse, and the benchmark's primary eligibility gate removed nineteen of the twenty, leaving one method (ELM) with a suppression of 0.349 at 83 percent retain preservation. K-Bench detects this through a collapse-aware score and states that "Read alone, OR(all) would certify the collapsed models as clean." Its limitations section notes that the collapse is consistent with applying non-agentic question-answer fine-tuning to an agent policy, and that separating it from forgetting would need recipes retrained with agent-format supervision.

Tool use that degrades everything is the second. In AUBench, adding search lowered average forget-set F1 from 29.7 to 12.9, but also retain-set F1 from 34.9 to 17.3. Qwen3-8B returned empty outputs on 23.6 to 58.0 percent of examples, and with KL its tool invocation rate in the complete agent fell to 0.5 percent of forget and 0.1 percent of retain queries (Lin et al., 2026). The authors themselves conclude that the forget-set reductions under search do not establish more effective

unlearning. A reader of the forget column alone would conclude the opposite.

Context that restores the residue is the third, and it cuts the other way. Memory raised forget-set F1 in five of six AUBench checkpoints, and turned the best parametric forgetter on Qwen3-8B into the worst complete-agent forgetter. The paper's Figure 1 states that the memory and profile stores contain no record of the forget entity. If that footnote is right, the rise did not come from the store returning the target, and it came either from parametric residue that a richer context drew out, or from answers that overlapped the reference more under token-level F1 because the model was generally more capable with memory. Those two readings predict different things - only the first should survive a semantic judge and an exact-match check - and the paper does not separate them. A third reading remains live: the stores are built from all 200 TOFU authors, forget authors included, and the results text says that memory "also restores access to target knowledge" (Lin et al., 2026), which would make the rise the store returning the target.

R-TOFU's decoding result belongs here too. Switching thinking off revealed forget content that default decoding hid (Yoon et al., 2025). A single decoding configuration can hide residue in either direction.

Together these show that an agentic evaluation is a different instrument, not a stricter version of the old one. It reads more channels and so finds more of some leaks. It also changes the system it measures, and some of those changes read as forgetting. K-Bench's collapse-aware scoring and validity gates are the right response, and AUBench's reliance on forget-set F1 alone is the case they guard against. The general lesson is that an agentic forgetting score is not interpretable without the agent's completion rate and its utility on retained queries measured in the same harness.

9. The strongest case for the transfer-failure reading

The reading that model-level certificates fail to transfer deserves a full statement, because much of it survives.

First, practice. Compliance claims are made about deployments, and the benchmarks in use test models. If an operator certifies erasure because a model passed TOFU, the operator has made the error K-Bench describes, whatever the error is called. K-Bench's contribution is to make that error measurable, with pre-registered tests, multiple-comparison control, and a clear statement of which channel carries each leak.

Second, ordinary use. Every leak in K-Bench's no-intervention traces comes from a fixed, non-adaptive observer and a standard scaffold. A follow-up request for a summary, a database lookup and a returned reasoning trace are what agents do routinely. The residue family was known, but it was known through attacks; K-Bench

shows it under ordinary use.

Third, ranking instability. A method-by-model interaction explaining 24 percent of score variance, a different leader on each base model, and verdicts that change with the observer budget on four of six Llama parametric cells mean that single-model, single-channel leaderboards are not evidence of a portable property (Yu et al., 2026). AUBench's rank reversal for KL makes the same point in a different harness (Lin et al., 2026).

Fourth, the provider's position. In ATU's threat model, the party that unlearns is not the party that deploys (Chen, B. et al., 2026). A provider releasing weights cannot control the sources, tools or trace policy of downstream agents. For that provider, the only lever is the model, and ATU's result is that a model can be trained to resist much of the reacquisition, with the effect holding when the retrieval tools are swapped for others of the same type and, in a small held-out split on one target, on queries not used in training (Chen, B. et al., 2026). If that generalises further, then the model-level certificate can be made to transfer, by training for transfer.

Each point holds. Together they establish that deployment audits must not rest on model benchmarks, and that a model whose release is intended for agent use should be evaluated in agent harnesses. They do not establish that the failure is the unlearning method's. In K-Bench's own data, the residue was visible to strong model-level probes, the scope leaks would survive exact retraining, and the migrated leak sits in a store's return value. What does not transfer is a clean-model claim that the strong model-level probes in K-Bench's case study did not support in the first place. The fourth point is the most important for future work. Training a model to resist reacquisition is a behavioural guarantee, and Section 6.3 gives reasons to doubt that such guarantees carry across harnesses.

10. Consolidated evidence, purposes and an audit

Table 1 collects the measured values the argument rests on, one row per cited result, with the family of Section 3.3 that each row bears on. "Harness" marks a result about the agent harness itself rather than about one family.

Read together, the rows show three patterns. On the weight source, every strong model-level instrument that was applied - forced choice, relearning, quantisation, representation attacks - found residue, and the agentic observer found the same residue through a follow-up question. On non-weight sources, the leak rates are baseline disclosure rates that no weight method can change and that an inference-time gate can drive to near zero only when the gate is exact, and on three of the four Llama substrates only at a large rise in agent collapse. And the rows marked "harness" show the instrument

altering the system it measures, in both directions.

Table 2 maps four deployment purposes onto the families that matter for each and the certificate that can bind them. It is a reading of the purposes stated in the cited sources, not a legal analysis.

The table makes the ownership problem visible. Only one of the four rows is mostly the provider's to certify, and it is the one where the agentic harness matters least, because an attacker with open weights does not need the provider's agent. In the other rows, at least half the families that matter are the operator's.

Algorithm 1 states the audit this paper argues a deployment should run before any forgetting claim is made about it. Its job is to resolve each observed leak to a family and an owner, and to refuse a verdict where the harness has made the measurement uninterpretable.

Applied to the published K-Bench results, the audit assigns the parametric-substrate leaks to residue that step 6 finds detectable without the agent; the context and retrieval leaks to scope, with exposure added for every leak carried by a tool observation; and the collapsed weight methods to UNINTERPRETABLE at step 1 rather than to either success or failure. Applied to ATU's MUSE and RWKU measurements, it assigns the agent-side recovery to scope or reacquisition, not separable from the published numbers, because the agent's tools include local databases holding the forget targets' content (MUSE Books passages; summarised documents about the RWKU public figures) as well as web search. For the part fetched from the web, the disposition depends on whether the purpose treats a public novel or a famous person's public facts as something the agent should refuse to look up. Applied to AUBench's memory condition, the audit leaves three assignments live: scope at step 3, if the stores built from all 200 TOFU authors returned the forget facts, as the construction section implies; residue at step 6, if the Figure 1 statement that the stores held no record of the target is right; or the metric. The paper's data do not decide among them.

11. Studies that would settle the open part

Six studies would turn the claims of this paper into measurements. None needs more compute than the studies it extends.

The first is a retrain-from-scratch reference on every source. K-Bench trains a retain-only reference for the weight substrate only. Running the same reference on the context and retrieval substrates should reproduce the no-intervention leak rates exactly, since the reference never held the secret in its weights and is handed it at inference. If it does, the 22 to 86 percent result is confirmed as a property of audit scope, invariant to unlearning quality. If it does not, the agent harness interacts with the weights in a way this paper has not

Table 1. Published results on forgetting in agents and its antecedents. Values are as reported by the named source; "-" means not applicable.

Source	Setting	Condition or observer	Reported value	Family
Yu et al. 2026	Llama-3.1-8B, secret in weights, no unlearning	answer only; + elicited summary	0.152; 0.682	1
Yu et al. 2026	Llama, secret in passage index	answer only; + tool channels	0.203; 0.602	2, 4
Yu et al. 2026	Llama, secret in database	answer only; all six channels	0.832; 0.855	2, 4
Yu et al. 2026	Llama, secret in prompt	all six channels	0.223	2
Yu et al. 2026	Qwen3.5-9B, secret in prompt	all six channels	0.992	2
Yu et al. 2026	8 API models, prompt and retrieval	all six channels	leak in 24 of 24 cells; prompt 0.27-1.00	2
Yu et al. 2026	6 API models returning reasoning, prompt	reasoning channel	0.75-1.00	4
Yu et al. 2026	8 weight probes, prompt and retrieval	direct elicitation	floor or chance; forced choice 0.24-0.26	2
Yu et al. 2026	trace filter (StaR), Llama, database	answer; tool observation	0.832 to 0.463; 0.855 to 0.857	4
Yu et al. 2026	input corruption (ECO), Llama, all four sources	all six channels; degeneration	at most 0.005 with ground-truth gate; 91%, 71%, 73% on prompt, passage, database (baseline 66%, 48%, 9%)	2
Yu et al. 2026	NPO, NPO + KL, IDK, Llama, weights	4-way forced choice	0.95-0.98; retrain reference 0.11	1
Yu et al. 2026	IDK, Llama, weights	degeneration; forced choice	98.2%; 0.95	harness
Yu et al. 2026	20 published methods, Llama	collapse; eligibility gate	16 of 20 collapse; 19 of 20 gated out	harness
Yu et al. 2026	adapter that never saw the target	parse-error rate	92.8%	harness
Yu et al. 2026	6 methods x 3 models x 3 seeds	share of score variance	method x model 24%	harness
Lin et al. 2026	Qwen3-8B, KL, TOFU forget05	parameters; + memory; complete agent	F1 16.8; 30.1; 36.9	1, 2, harness
Lin et al. 2026	6 checkpoints, mean	+ search	forget 29.7 to 12.9; retain 34.9 to 17.3	harness
Chen, B. et al. 2026	NPO, MUSE Books	bare; tool agent	9.7; 29.8 (base 37.9; 43.3)	2, 3
Chen, B. et al. 2026	NPO, RWKU	bare; tool agent	44.6; 68.6 (base 57.7; 74.6)	2, 3
Chen, B. et al. 2026	NPO plus anti-seeking RL, RWKU, Qwen3-4B	agent, original; swapped tools	68.6 to 47.5; 64.0 to 46.7	2, 3
O'Brien et al. 2025	filtered pretraining, 6.9B	adversarial fine-tuning	resisted up to 10,000 steps, 300M tokens	-
O'Brien et al. 2025	filtered models	passage given in context	performed well, as unfiltered did	3
Xie et al. 2026	6 memory systems, 3 backbones	end-to-end deletion	at most 1.5% of requests pass	2
Yoon et al. 2025	R-TOFU, direct IDK	default vs no-thinking decoding	complete by epoch 4 by default; high without thinking	1, harness
Patil et al. 2024	ROME edits, GPT-J	white- and black-box attacks	38% recovered	1
Deeb and Roger 2024	pretraining-learned facts	fine-tune on related facts	88% of accuracy recovered	1
Zhang, Z. et al. 2025	utility-constrained methods	4-bit quantisation	retained 21% to 83%	1
Lucki et al. 2025	RMU	10 unrelated fine-tuning examples	most capability recovered	1
Chen, Y. et al. 2025	reasoning models	hint verbalisation	often below 20%	1
Shao et al. 2024	GPT-4; Llama-3-70B agents	actions, privacy prompt	25.68%; 38.69% leak	2, 4
Mireshghallah et al. 2024	GPT-4; ChatGPT	contextual privacy	39%; 57% reveal	2
Kumar et al. 2024	GPT-4o; o1-preview browser agents	human-rewritten requests	98; 63 of 100 attempted	3
Cartagena and Teixeira 2026	6 frontier models	safety-reinforced prompts	219 refuse-in-text, act-in-tool cases	3, 4
Puerto et al. 2026	trace instruction-following	privacy benchmarks	up to +51.9 points	4

Table 2. Purpose, the failures that matter, and who can certify against them.

Purpose	What must not happen	Families that matter	Certificate that can bind	Who can issue it
Personal-data erasure	disclosure of the person's data to a party not entitled to it	1, 2, 4; 3 only for non-public data	deletion per source; non-disclosure over user-visible channels	provider for weights; operator for stores, memory, trace
Copyright	reproduction of protected expression; unlawful acquisition	1 for verbatim output; 2 if stored	documented data lineage; output filtering	provider; operator
Hazard, open weights	uplift to an attacker who can tamper	1 under fine-tuning; 3 bounded by search access	tamper-resistant removal; capability impairment	provider
Hazard, hosted agent	the agent supplies uplift	3, 4	behavioural or capability control; source gating by user trust	operator

Algorithm 1. Assigning an observed agent leak to a family, an owner and a certificate

```

Input: target T; deployment D = (weights, context, stores, memory, tools, trace policy, audiences)
      observed leak L = (query q, channel c, trajectory tau)
step 1, health: in the same harness, measure task completion and retain-set utility;
      if completion < 0.8 x no-intervention completion, or retain utility falls by more
      than the declared tolerance, mark the cell UNINTERPRETABLE and score no forgetting
step 2, sources: list every source of D that holds T when q is asked:
      weights (forced choice, distributional test, relearning probe); stores and memory
      (lookup, including derived artifacts); context (inspection)
step 3, store: if c returns an item from a store or memory that contains T,
      family = SCOPE; owner = operator; remedy = delete, deny lookup, or gate by user
step 4, fetch: elif no source of D holds T and tau has a tool call that fetched T,
      family = REACQUISITION; owner = operator policy; decide by purpose (Table 2):
      refuse to seek, refuse to repeat, de-list upstream, or permit
step 5, held: elif T is in context, a store or memory of D,
      family = SCOPE; owner = operator
step 6, weights: else (T held only in weights)
      family = RESIDUE; owner = provider; re-run the model-level probes of step 2,
      and if they also detect T, record "detectable without agent"
step 7, audience: for any family, if c is reasoning, a tool argument or an observation
      and its audience includes a party not entitled to T, add EXPOSURE; owner = trace policy
Output: per leak (family, owner, remedy); per source a certificate with its observer,
      query family, decoding settings and audience stated.

```

considered.

The second is agent-format retraining of the weight recipes. K-Bench names this control in its limitations: retrain each unlearning recipe with supervision in the agent's own format, so that the recipe does not also destroy the ReAct policy, and read the full observer again. This separates collapse from forgetting for the sixteen collapsed Llama methods.

The third is audience-conditioned reporting. Report every agentic leak rate under at least three declared trace policies - answer only; answer plus a returned reasoning summary; full trace - so that a deployment can read off the rate that applies to its own audience. K-Bench's per-channel rates already allow this to be computed from its released traces.

The fourth is the hybrid source. K-Bench and AUBench test sources one at a time. A deployment that both fine-tuned on a record and retrieves it is common. Whether the combined leak exceeds the union of its parts, and whether weight unlearning changes how the model treats a retrieved copy, are both untested. AUBench's knowledge-conflict account of its search results (Lin et al., 2026) predicts that unlearned weights may sometimes make retrieved evidence harder to use, which would be a measurable interaction.

The fifth is reacquisition measured as uplift. For a public target and a private one, compare a base model with search, an unlearned or filtered model with search, and a person using the same search tool without a model. ATU measures recovery; the purpose-dependent question of Section 6.4 needs the difference between the agent and the bare search tool, which is what Deep Ignorance's footnote on uplift points to (O'Brien et al., 2025).

The sixth is cross-harness transfer of behavioural controls. ATU already tests same-type tool swaps, a small held-out query split and adaptive directives that force retrieval or copying. Under three of its five directives its second stage still lets the forget score rise, from 47.5 to 59.6 when the agent must retrieve before answering and by 10.2 points when it must copy tool output exactly, and its authors conclude that "ATU does not provide complete robustness when an adversary requires direct reproduction of tool observations" (Chen, B. et al., 2026, Appendix D.7). Its limitations name the harder cases: other tool APIs, memory systems and multi-agent workflows, and tools that return richer, longer or adversarially formatted observations (Chen, B. et al., 2026). Train the anti-seeking stage or a circuit breaker in one harness and test it in those, and under adversarially rewritten requests beyond appended directives, with the browser-agent and tool-call results of Section 6.3 as the expected failure modes. Run the same test on a secret

memorised in pretraining rather than injected by LoRA, which K-Bench lists as untested and Deeb and Roger (2024) suggest matters, since unlearning information learned in a separate fine-tuning phase may overstate robustness relative to information learned in pretraining.

12. What is not known

Whether any weight-level unlearning method, applied to information learned in pretraining, survives a full agent observer with the agent policy intact, is not known; the published agentic evaluations inject the target by fine-tuning and see most weight methods collapse the agent. Whether behavioural controls against reacquisition, such as ATU's trained policy, transfer beyond same-type tool swaps to different agent frameworks, memory systems and adversarially formatted observations is not known. Whether a hybrid deployment that holds a target in weights and in a store leaks more than either alone is not known. Whether the rise in forget-set scores when memory is added reflects the store returning the target, parametric residue or a scoring artefact is not known. How often deployed products expose reasoning text or tool observations to end users, which sets the practical size of the exposure family, is not documented in the literature reviewed here. No study measures reacquisition as uplift over a search tool used directly.

13. Limitations

This analysis synthesises published text, tables and figures; its numbers are the sources' numbers, and its derivations in Section 3 are elementary consequences of definitions. The anchor benchmark and three of the four agentic studies are 2026 preprints and could change. The K-Bench values used here are drawn from its version 2; where its text and tables differ slightly, as for the parametric summary-channel rate and the no-intervention baseline under different injection targets, the paper uses table values and avoids the contested figure. The AUBench reading in Section 8 turns on its Figure 1 statement that the stores held no record of the target, which sits in tension with its construction section and its prose account of memory restoring access; this paper reports both and does not resolve them. Table 2 reads purposes from the cited technical and position papers and is not legal advice; the right-to-be-forgotten analogy to de-indexing is offered as a model of where a duty can sit, not as a statement of what any law requires of agent operators. Finally, the four-family partition is a classification of the leaks the published studies report. Other harnesses - multi-agent systems, persistent agents that act in the world, agents with code execution - may produce leaks that the partition does not cover.

14. Conclusion

The agentic unlearning studies of 2026 report that a model's clean answer does not make its agent clean, and they are right. What they report is four failures, not one.

Where the target is still in the weights, the agent asks a different question and finds residue that strong model-level probes, in the same benchmark, also find. Where the target was never in the weights, no weight method could have removed it, and a certificate about the weights was never a claim about it. Where the agent fetches the target anew, removal from any internal source is neither sufficient nor necessary, and the only controls shown to work act on what the agent seeks, reads or repeats - behavioural, input-side and capability-impairing controls of the kind the removal literature discounts. Where an internal channel carries the target, the leak goes to whoever can read that channel, and in the anchor benchmark's lead example it is a database's return value. The harness that reveals these failures can also hide others, by breaking the agent in ways that read as forgetting. A forgetting claim about an agent therefore has to name its sources, its observer and its audience, and each failure has to be assigned to the party that holds the source it came from. The model provider can certify the weights. The operator has to certify the rest.

References

- Andriushchenko, M., Souly, A., Dziemian, M., Duenas, D., Lin, M., Wang, J., et al. (2025). AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents. ICLR 2025. arXiv:2410.09024. doi:10.48550/arXiv.2410.09024.
- Barez, F., Fu, T., Prabhu, A., Casper, S., Sanyal, A., Bibi, A., et al. (2025). Open Problems in Machine Unlearning for AI Safety. arXiv:2501.04952. doi:10.48550/arXiv.2501.04952.
- Bourtole, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., Lie, D., & Papernot, N. (2021). Machine Unlearning. IEEE Symposium on Security and Privacy 2021. arXiv:1912.03817. doi:10.48550/arXiv.1912.03817.
- Cartagena, A., & Teixeira, A. (2026). Mind the GAP: Text Safety Does Not Transfer to Tool-Call Safety in LLM Agents. arXiv:2602.16943. doi:10.48550/arXiv.2602.16943.
- Chen, B., Liu, Z., Zhang, J., Ding, K., Ma, N., Huang, Y., & Jiang, M. (2026). Forgotten in Weights, Recovered by Tools: Agentic Tool Unlearning for LLM Agents. arXiv:2608.21544. doi:10.48550/arXiv.2608.21544.
- Chen, Y., Benton, J., Radhakrishnan, A., Uesato, J., Denison, C., Schulman, J., et al. (2025). Reasoning Models Don't Always Say What They Think. arXiv:2505.05410. doi:10.48550/arXiv.2505.05410.
- Cheng, J., & Amiri, H. (2025). Tool Unlearning for Tool-Augmented LLMs. ICML 2025. arXiv:2502.01083. doi:10.48550/arXiv.2502.01083.
- Choi, M., Park, C., Lee, D., & Choo, J. (2024). Breaking Chains: Unraveling the Links in Multi-Hop Knowledge Unlearning. arXiv:2410.13274. doi:10.48550/arXiv.2410.13274.
- Cooper, A. F., Choquette-Choo, C. A., Bogen, M., Klyman, K., Jagielski, M., Filippova, K., et al. (2025). Machine Unlearning Doesn't Do What You Think: Lessons for Generative AI Policy and Research. NeurIPS 2025. arXiv:2412.06966. doi:10.48550/arXiv.2412.06966.
- Deeb, A., & Roger, F. (2024). Do Unlearning Methods Remove Information from Language Model Weights? arXiv:2410.08827. doi:10.48550/arXiv.2410.08827.
- Eldan, R., & Russinovich, M. (2023). Who's Harry Potter? Approximate Unlearning in LLMs. arXiv:2310.02238. doi:10.48550/arXiv.2310.02238.
- Green, T., Gubri, M., Puerto, H., Yun, S., & Oh, S. J. (2025). Leaky Thoughts: Large Reasoning Models Are Not Private Thinkers.

- EMNLP 2025. arXiv:2506.15674. doi:10.48550/arXiv.2506.15674.
- Hong, Y., Yu, L., Yang, H., Ravfogel, S., & Geva, M. (2025). Intrinsic Test of Unlearning Using Parametric Knowledge Traces. EMNLP 2025. arXiv:2406.11614. doi:10.48550/arXiv.2406.11614.
- Hu, S., Fu, Y., Wu, Z. S., & Smith, V. (2025). Unlearning or Obfuscating? Jogging the Memory of Unlearned LLMs via Benign Relearning. ICLR 2025. arXiv:2406.13356. doi:10.48550/arXiv.2406.13356.
- Jin, Z., Cao, P., Wang, C., He, Z., Yuan, H., Li, J., Chen, Y., Liu, K., & Zhao, J. (2024). RWKU: Benchmarking Real-World Knowledge Unlearning for Large Language Models. arXiv:2406.10890. doi:10.48550/arXiv.2406.10890.
- Korbak, T., Balesni, M., Barnes, E., Bengio, Y., Benton, J., Bloom, J., et al. (2025). Chain of Thought Monitorability: A New and Fragile Opportunity for AI Safety. arXiv:2507.11473. doi:10.48550/arXiv.2507.11473.
- Kumar, P., Lau, E., Vijayakumar, S., Trinh, T., Scale Red Team, Chang, E., et al. (2024). Refusal-Trained LLMs Are Easily Jailbroken As Browser Agents. arXiv:2410.13886. doi:10.48550/arXiv.2410.13886.
- Le, T., Qian, W., & Huai, M. (2026). Selective Forgetting for Large Reasoning Models. arXiv:2604.03571. doi:10.48550/arXiv.2604.03571.
- Lee, B. W., Foote, A., Infanger, A., Shor, L., Kamath, H., Goldman-Wetzler, J., et al. (2025). Distillation Robustifies Unlearning. NeurIPS 2025. arXiv:2506.06278. doi:10.48550/arXiv.2506.06278.
- Li, N., Pan, A., Gopal, A., Yue, S., Berrios, D., Gatti, A., et al. (2024). The WMDP Benchmark: Measuring and Reducing Malicious Use With Unlearning. ICML 2024. arXiv:2403.03218. doi:10.48550/arXiv.2403.03218.
- Lin, J., Xu, G., Xu, Z., Chen, J., & Lan, X. (2026). Can Agents Truly Forget? Explore Machine Unlearning for LLM Agents. arXiv:2610.17659. doi:10.48550/arXiv.2610.17659.
- Liu, C. Y., Wang, Y., Flanigan, J., & Liu, Y. (2024). Large Language Model Unlearning via Embedding-Corrupted Prompts. NeurIPS 2024. arXiv:2406.07933. doi:10.48550/arXiv.2406.07933.
- Liu, S., Yao, Y., Jia, J., Casper, S., Baracaldo, N., Hase, P., et al. (2025). Rethinking Machine Unlearning for Large Language Models. Nature Machine Intelligence. arXiv:2402.08787. doi:10.48550/arXiv.2402.08787.
- Lucki, J., Wei, B., Huang, Y., Henderson, P., Tramer, F., & Rando, J. (2025). An Adversarial Perspective on Machine Unlearning for AI Safety. Transactions on Machine Learning Research. arXiv:2409.18025. doi:10.48550/arXiv.2409.18025.
- Lynch, A., Guo, P., Ewart, A., Casper, S., & Hadfield-Menell, D. (2024). Eight Methods to Evaluate Robust Unlearning in LLMs. arXiv:2402.16835. doi:10.48550/arXiv.2402.16835.
- Maccoby, J., & Wang, W. H. (2026). REDaCT: Collapse-Free Unlearning in Large Reasoning Models. Findings of EMNLP 2026. arXiv:2610.12769. doi:10.48550/arXiv.2610.12769.
- Mahendrakar, P. (2026a). Three Guarantees Under One Word: What Importance-Based Selective Forgetting in Language Models Can and Cannot Promise. Life of Research. doi:10.5281/zenodo.22054075.
- Mahendrakar, P. (2026b). Delete Names Five Operations: Erasure Semantics for Derived Agent Memory, and Why a State-Level Definition Cannot Be the Auditable One. Life of Research. doi:10.5281/zenodo.22425281.
- Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z. C., & Kolter, J. Z. (2024). TOFU: A Task of Fictitious Unlearning for LLMs. arXiv:2401.06121. doi:10.48550/arXiv.2401.06121.
- Mireshghallah, N., Kim, H., Zhou, X., Tsvetkov, Y., Sap, M., Shokri, R., & Choi, Y. (2024). Can LLMs Keep a Secret? Testing Privacy Implications of Language Models via Contextual Integrity Theory. ICLR 2024. arXiv:2310.17884. doi:10.48550/arXiv.2310.17884.
- O'Brien, K., Casper, S., Anthony, Q., Korbak, T., Kirk, R., Davies, X., Mishra, I., Irving, G., Gal, Y., & Biderman, S. (2025). Deep Ignorance: Filtering Pretraining Data Builds Tamper-Resistant Safeguards into Open-Weight LLMs. arXiv:2508.06601. doi:10.48550/arXiv.2508.06601.
- Patil, V., Hase, P., & Bansal, M. (2024). Can Sensitive Information Be Deleted From LLMs? Objectives for Defending Against Extraction Attacks. ICLR 2024. arXiv:2309.17410. doi:10.48550/arXiv.2309.17410.
- Pawelczyk, M., Neel, S., & Lakkaraju, H. (2024). In-Context Unlearning: Language Models as Few Shot Unlearners. ICML 2024. arXiv:2310.07579. doi:10.48550/arXiv.2310.07579.
- Puerto, H., Li, H., Han, X., Baldwin, T., & Gurevych, I. (2026). From Leaky Thoughts to Private Reasoning: Controlling What LLMs Say to Themselves. EMNLP 2026. arXiv:2602.24210. doi:10.48550/arXiv.2602.24210.
- Qian, H., Zhang, Q., Huang, Z., Hong, C., & Qiu, H. (2026). Leak-Resistant Unlearning: A New Benchmark for Evaluating Multi-Hop Reasoning Consistency and Recovery Robustness. arXiv:2608.04519. doi:10.48550/arXiv.2608.04519.
- Qiu, R., Tan, J., Pu, J., Wang, H., Gao, X.-S., & Sun, F. (2025). A Survey on Unlearning in Large Language Models. arXiv:2510.25117. doi:10.48550/arXiv.2510.25117.
- Sanyal, D., & Mandal, M. (2025). Agents Are All You Need for LLM Unlearning. COLM 2025. arXiv:2502.00406. doi:10.48550/arXiv.2502.00406.
- Shao, Y., Li, T., Shi, W., Liu, Y., & Yang, D. (2024). PrivacyLens: Evaluating Privacy Norm Awareness of Language Models in Action. NeurIPS 2024 Datasets and Benchmarks Track. arXiv:2409.00138. doi:10.48550/arXiv.2409.00138.
- Shi, W., Lee, J., Huang, Y., Malladi, S., Zhao, J., Holtzman, A., Liu, D., Zettlemoyer, L., Smith, N. A., & Zhang, C. (2024). MUSE: Machine Unlearning Six-Way Evaluation for Language Models. arXiv:2407.06460. doi:10.48550/arXiv.2407.06460.
- Shi, Y., & Wang, R. (2026). Trajectory Unlearning on LLM-based Agents. arXiv:2609.33639. doi:10.48550/arXiv.2609.33639.
- Shumailov, I., Hayes, J., Triantafillou, E., Ortiz-Jimenez, G., Papernot, N., Jagielski, M., Yona, I., Howard, H., & Bagdasaryan, E. (2024). UnUnlearning: Unlearning is not sufficient for content regulation in advanced generative AI. arXiv:2407.00106. doi:10.48550/arXiv.2407.00106.
- Sun, G., Manakul, P., Zhan, X., & Gales, M. (2025). Unlearning vs. Obfuscation: Are We Truly Removing Knowledge? EMNLP 2025. arXiv:2505.02884. doi:10.48550/arXiv.2505.02884.
- Takashiro, S., Kojima, T., Gambardella, A., Cao, Q., Iwasawa, Y., & Matsuo, Y. (2025). Answer When Needed, Forget When Not: Language Models Pretend to Forget via In-Context Knowledge Unlearning. Findings of ACL 2025. arXiv:2410.00382. doi:10.48550/arXiv.2410.00382.
- Tamirisa, R., Bharathi, B., Phan, L., Zhou, A., Gatti, A., Suresh, T., et al. (2024). Tamper-Resistant Safeguards for Open-Weight LLMs. arXiv:2408.00761. doi:10.48550/arXiv.2408.00761.
- Thaker, P., Maurya, Y., Hu, S., Wu, Z. S., & Smith, V. (2024). Guardrail Baselines for Unlearning in LLMs. ICLR 2024 Workshop on Secure and Trustworthy LLMs. arXiv:2403.03329. doi:10.48550/arXiv.2403.03329.
- Thaker, P., Hu, S., Kale, N., Maurya, Y., Wu, Z. S., & Smith, V. (2025). Position: LLM Unlearning Benchmarks are Weak Measures of Progress. IEEE SaTML 2025. arXiv:2410.02879. doi:10.48550/arXiv.2410.02879.
- Turpin, M., Michael, J., Perez, E., & Bowman, S. R. (2023). Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. NeurIPS 2023. arXiv:2305.04388. doi:10.48550/arXiv.2305.04388.
- Utsunomiya, S., Isonuma, M., Mori, J., & Sakata, I. (2026). Position: No Retroactive Cure for Infringement during Training. arXiv:2604.18649. doi:10.48550/arXiv.2604.18649.

- Vilella, S., & Ruffo, G. (2025). (De)-Indexing and the Right to be Forgotten. arXiv:2501.03989. doi:10.48550/arXiv.2501.03989.
- Wang, B., Wang, F., Wang, P., Cong, J., Yu, Y., Yin, Y., Han, Z., & Wei, B. (2026). Agentic Unlearning: When LLM Agent Meets Machine Unlearning. arXiv:2602.17692. doi:10.48550/arXiv.2602.17692.
- Wang, C., Fan, C., Zhang, Y., Jia, J., Wei, D., Ram, P., Baracaldo, N., & Liu, S. (2025). Reasoning Model Unlearning: Forgetting Traces, Not Just Answers, While Preserving Reasoning Skills. EMNLP 2025. arXiv:2506.12963. doi:10.48550/arXiv.2506.12963.
- Wang, S., Zhu, T., Ye, D., & Zhou, W. (2024). When Machine Unlearning Meets Retrieval-Augmented Generation (RAG): Keep Secret or Forget Knowledge? IEEE Transactions on Dependable and Secure Computing. arXiv:2410.15267. doi:10.48550/arXiv.2410.15267.
- Weidinger, L., Rauh, M., Marchal, N., Manzini, A., Hendricks, L. A., Mateos-Garcia, J., et al. (2023). Sociotechnical Safety Evaluation of Generative AI Systems. arXiv:2310.11986. doi:10.48550/arXiv.2310.11986.
- Xie, B., Wang, C., Ge, G., Wang, S., Niu, F., Zhang, H., & Wang, W. (2026). Can Agents Forget When Asked? A Benchmark for Memory Unlearning in LLM Agents. arXiv:2610.18052. doi:10.48550/arXiv.2610.18052.
- Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. ICLR 2023. arXiv:2210.03629. doi:10.48550/arXiv.2210.03629.
- Ye, D., Zhu, T., Zhu, C., He, F., He, Q., Wang, S., Liu, B., & Zhou, W. (2026). Secure Forgetting: A Framework for Privacy-Driven Unlearning in Large Language Model (LLM)-Based Agents. arXiv:2604.00430. doi:10.48550/arXiv.2604.00430.
- Yoon, S., Jeung, W., & No, A. (2025). R-TOFU: Unlearning in Large Reasoning Models. arXiv:2505.15214. doi:10.48550/arXiv.2505.15214.
- Yu, C., Stroebel, B., Yang, D., & Papakyriakopoulos, O. (2025). Safety Degradation in AI Agents. arXiv:2505.14215. doi:10.48550/arXiv.2505.14215.
- Yu, G., Jiang, Y., Wang, Q., Ma, B., & Wang, X. (2026). K-Bench: A Benchmark for LLM Unlearning in Agentic Deployments. arXiv:2609.12808. doi:10.48550/arXiv.2609.12808.
- Zeng, S., Zhang, J., He, P., Xing, Y., Liu, Y., Xu, H., et al. (2024). The Good and The Bad: Exploring Privacy Issues in Retrieval-Augmented Generation (RAG). arXiv:2402.16893. doi:10.48550/arXiv.2402.16893.
- Zhang, D., Finckenberg-Broman, P., Hoang, T., Pan, S., Xing, Z., Staples, M., & Xu, X. (2023). Right to be Forgotten in the Era of Large Language Models: Implications, Challenges, and Solutions. arXiv:2307.03941. doi:10.48550/arXiv.2307.03941.
- Zhang, R., Lin, L., Bai, Y., & Mei, S. (2024). Negative Preference Optimization: From Catastrophic Collapse to Effective Unlearning. arXiv:2404.05868. doi:10.48550/arXiv.2404.05868.
- Zhang, Y., Carlini, N., & Ippolito, D. (2024). Effective Prompt Extraction from Language Models. arXiv:2307.06865. doi:10.48550/arXiv.2307.06865.
- Zhang, Z., Wang, F., Li, X., Wu, Z., Tang, X., Liu, H., He, Q., Yin, W., & Wang, S. (2025). Catastrophic Failure of LLM Unlearning via Quantization. ICLR 2025. arXiv:2410.16454. doi:10.48550/arXiv.2410.16454.
- Zharmagambetov, A., Guo, C., Evtimov, I., Pavlova, M., Salakhutdinov, R., & Chaudhuri, K. (2025). AgentDAM: Privacy Leakage Evaluation for Autonomous Web Agents. NeurIPS 2025 Datasets and Benchmarks Track. arXiv:2503.09780. doi:10.48550/arXiv.2503.09780.
- Zou, A., Phan, L., Wang, J., Duenas, D., Lin, M., Andriushchenko, M., Wang, R., Kolter, Z., Fredrikson, M., & Hendrycks, D. (2024). Improving Alignment and Robustness with Circuit Breakers. arXiv:2406.04313. doi:10.48550/arXiv.2406.04313.