Exact Machine Unlearning on Weightless Neural Networks

Luan Martins[1], **Felipe M. G. França**[1],
Priscila Machado Vieira Lima[1], **Claudio Miceli de Farias**[1]

[1]Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brazil

{luanmf,felipe,priscilamvl,cmicelifarias}@cos.ufrj.br

Abstract. *Privacy regulations create pressure to remove personal records and, when those records have influenced a trained model, motivate machine unlearning. Existing methods make a difficult trade-off: retraining is exact but costly, whereas inexpensive approximations may leave detectable information about deleted examples. We examine weightless neural networks of the WiSARD family, whose training procedure adds independent integer-counter updates. Reversing those updates returns the counters to the state produced without the deleted examples, which permits a cell-by-cell exactness certificate. We evaluate this mechanism on a calibrated synthetic Where's-Waldo benchmark against full and sharded retraining, deletion-enabled forests, and gradient ascent. In class erasure, WiSARD deleted the selected class in 40 ms, retained 0.899 accuracy, used 9.3 MB, produced a maximum counter difference of zero, and reduced membership-inference area under the ROC curve to 0.50; retraining required 1.42 s, while the more accurate deletion-enabled forest retained 0.953 accuracy at 13.87 s and 1.59 GB. Within the evaluated setting, WiSARD therefore provides an exact option for applications that prioritize deletion latency, although it does not provide the highest accuracy or the smallest absolute memory footprint.*

1. Introduction

Machine-learning systems often retain the influence of their training data after the original records are removed. This persistence matters when individuals exercise rights established by the Brazilian General Data Protection Law (LGPD) [Brasil 2018] or the European General Data Protection Regulation (GDPR) [European Parliament and Council of the European Union 2016]. It also matters when operators need to remove poisoned, backdoored, stale, or mislabeled examples. *Machine unlearning* seeks to remove selected training contributions from a model without rebuilding it from the beginning [Cao and Yang 2015, Nguyen et al. 2025, Xu et al. 2023].

Useful unlearning must preserve accuracy on retained data while controlling deletion time, memory, and residual information. Retraining from scratch supplies a reference model but repeats the full training cost. Exact alternatives, including Sharded, Isolated, Sliced, and Aggregated (SISA) retraining [Bourtoule et al. 2021] and data-removal-enabled (DaRE) forests [Brophy and Lowd 2021], reduce that cost by rebuilding part of a model or storing deletion-specific statistics. Approximate alternatives are often cheaper, but membership-inference attacks can reveal residual information about examples that were meant to be forgotten [Shokri et al. 2017, Chen et al. 2021].

Additive models offer a direct route to exact deletion. When training state is a sum of independent contributions, removing an example requires subtracting its contribution rather than reconstructing model structure [Cao and Yang 2015]. Weightless neural networks in the WiSARD family have this form: training increments integer counters at memory addresses selected by each binary input [Aleksander et al. 1984]. Although existing software already implements the reverse operation [Lima Filho et al. 2020], prior WiSARD evaluations emphasize prediction, online learning, and memory rather than measuring exact deletion against established unlearning methods.

The objective of this work is to determine whether WiSARD counter subtraction provides exact, verifiable, and low-cost machine unlearning while retaining useful predictive accuracy relative to established baselines.

This paper makes three contributions:

- **(i)** we characterize WiSARD deletion as the reversal of additive counter updates and define a direct certificate that compares the resulting counters with those of a model trained without the deleted data;
- **(ii)** we provide a calibrated synthetic Where's-Waldo generator with balanced identity classes, hard negative examples, localization labels, and controlled difficulty factors; and
- **(iii)** we provide an evaluation harness and an empirical comparison with retraining, SISA, DaRE forests, and gradient ascent across class erasure, poison removal, deletion streams, and backdoor removal.

The paper is divided as follows: Section 2 positions the work in the weightless-network and machine-unlearning literature; Section 3 presents the proposal; Section 4 defines the experiments; Section 5 reports the results; Section 6 discusses their scope and limitations; and Section 7 concludes.

2. Related Work

This section relates the proposal to weightless neural networks, exact and approximate machine unlearning, privacy audits, and previous Where's-Waldo datasets. These themes distinguish the known ability to reverse additive updates from the empirical question studied here: how that ability compares with established unlearning methods.

Weightless neural networks store patterns in random-access memories rather than learning real-valued weights. The n-tuple method samples groups of input bits and uses each group as a memory address [Bledsoe and Browning 1959]; WiSARD organizes these memories into one class-specific *discriminator* whose matching addresses vote during prediction [Aleksander et al. 1984]. Regression WiSARD extends these votes to real-valued outputs [Lusquino Filho et al. 2020], and wisardpkg provides both variants [Lima Filho et al. 2020]. Prior evaluations emphasize fast, single-pass training and prediction accuracy, but do not compare the reversal of WiSARD counter updates with machine-unlearning baselines.

Machine unlearning aims to produce the state obtained by training without selected data, while avoiding complete retraining [Cao and Yang 2015, Nguyen et al. 2025, Xu et al. 2023]. Ginart et al. formalize exact deletion as matching retraining in distribution and make deletion time and auxiliary memory explicit design criteria [Ginart et al. 2019]. Exact methods constrain the work structurally: SISA partitions

the training set and retrains only affected shards [Bourtoule et al. 2021], whereas DaRE forests cache information needed to revise affected subtrees [Brophy and Lowd 2021]. Cao and Yang already showed that additive learning permits exact unlearning [Cao and Yang 2015]; our contribution is therefore not a new deletion capability, but its application, certification, and measurement for WiSARD.

Architecture-native deletion is a related direction. MUNKEY removes instance keys from the external memory of a memory-augmented vision transformer [Laguna et al. 2026]. Its evaluation concerns approximate unlearning with small, stratified forget sets and does not establish equality with retraining. We therefore treat MUNKEY as the closest conceptual precedent rather than an experimental baseline: reproducing it here would require a pretrained, GPU-trained transformer and would test a different deletion regime and guarantee.

Deletion can also be probed statistically. A membership-inference attack (MIA) estimates whether an example belonged to the training set, and an area under the receiver operating characteristic curve (AUC) of 0.5 indicates chance-level separation [Shokri et al. 2017]. Strong attacks calibrate individual examples against reference models [Carlini et al. 2022], and unlearning-specific work warns that aggregate attacks can overstate privacy [Hayes et al. 2025, Chen et al. 2021]. Following calls for auditable algorithm-level definitions [Thudi et al. 2022], we use a confidence-threshold MIA as a common diagnostic but rely on address-level counter equality as the stronger WiSARD guarantee.

Previous Where's-Waldo work treats the puzzle as detection on scanned pages [Barthelmes and Vidal 2021]. The public Hey-Waldo collection supplies only a small number of target annotations, which is sufficient for calibration but not for balanced deletion experiments [Constantinou 2017]. Our benchmark therefore uses synthetic patches and scenes with repeated identities, controlled difficulty, and known locations. The gap left by the reviewed work is an empirical comparison that measures WiSARD's reversible counters on deletion cost, retained accuracy, memory, and residual information against exact and approximate unlearning baselines.

3. Proposal

The proposal combines three components. Section 3.1 defines a controlled benchmark with enough repeated identities for deletion experiments; Section 3.2 defines WiSARD encoding and localization; and Section 3.3 states the exact-deletion mechanism and its certificate. Section 3.4 then explains the design choices that connect these components.

3.1. Synthetic benchmark design

The benchmark must provide balanced identity classes, many examples per identity, and known target locations. Existing annotated Where's-Waldo pages do not meet these requirements: Hey-Waldo contains a small set of scenes with one target annotation per page [Constantinou 2017]. We therefore use those pages only to calibrate the generator. The median target width is 1.4% of the page at a width-to-height ratio of 0.52; the two target colors are the 8-bit red-green-blue (RGB) triplets $(185, 84, 78)$ and $(247, 225, 217)$; and red and near-white pixels occupy approximately 17% and 22% of a page.

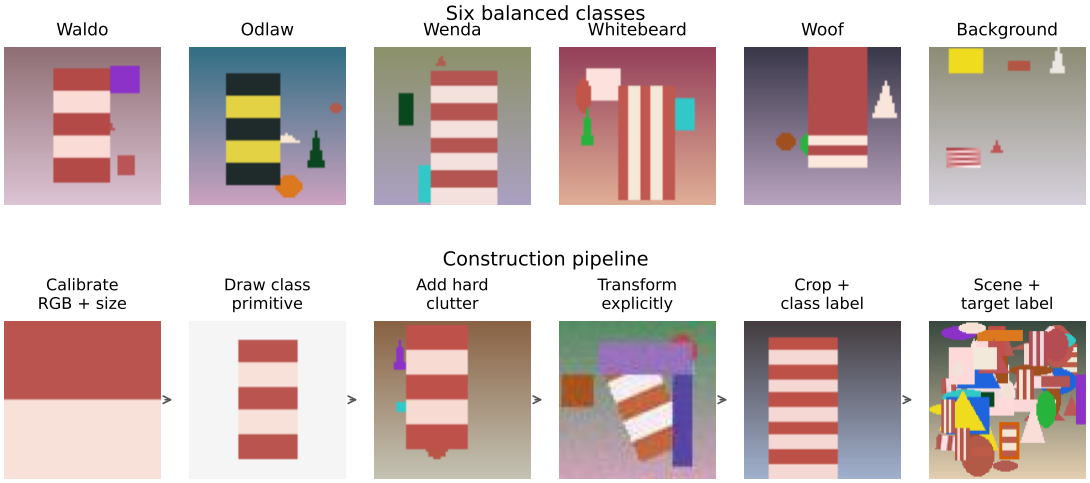


Figure 1. Construction of the synthetic benchmark. Calibrated colors and dimensions define six class prototypes; the generator adds clutter and controlled transformations; and the same components produce labeled classification crops or localization scenes. After rotation, every complete character bounding box remains inside the image. Partial visibility is introduced only by the occlusion transformation.

The generator represents the visual feature that remains visible in a 48×48-pixel crop. It produces Waldo and four hard negatives, which are examples designed to resemble the target: Odlaw changes the palette, Wenda changes the stripe count, Whitebeard changes stripe orientation, and Woof retains only a partial striped band. A sixth, background class contains colored shapes and striped objects rather than an empty image. Each non-background class therefore differs from Waldo along one controlled factor, while the background tests whether a model merely detects red-and-white stripes.

The generator produces two datasets. The first contains balanced, labeled crops for recognition and unlearning; the second contains cluttered scenes with a target-center label for localization. Four transformations control difficulty: rotation, RGB palette jitter, occlusion, and photometric degradation through brightness, contrast, noise, blur, and JPEG compression. The generator does not use crop-based augmentation: after rotation, it places the full axis-aligned character bounding box inside the crop or scene, and only the explicit occlusion transformation makes a character partially visible.

3.2. Weightless representation and localization

WiSARD accepts only binary input, so binarization determines which visual patterns its memories can store. We encode each RGB channel with a thermometer code, a binary sequence in which similar intensities share most active bits. The encoded bits are assigned to fixed tuples; each tuple selects an address in one random-access memory (RAM). A class-specific collection of these RAMs is a *discriminator*, and the predicted class is the discriminator with the largest number of matching addresses.

Localization uses the same memory model but preserves spatial context through a sliding window. A Regression WiSARD scores how closely each window is centered on the target, producing a two-dimensional score map. The maximum score gives the predicted location, followed by a parabolic refinement within the local grid cell. This

formulation keeps the method independent of an absolute scene position and permits a direct comparison with a random-forest window classifier that receives the same encoded windows.

3.3. Exact deletion and certificate

Let D be a training set and let $\phi(x_i, y_i)$ be the sparse tensor that places one increment at every RAM address activated by example (x_i, y_i). For a fixed bit-to-tuple mapping, WiSARD training produces the counter state

$$C(D) = \sum_{(x_i,y_i)\in D} \phi(x_i, y_i). \tag{1}$$

Deleting a set $F \subseteq D$ reverses the same updates:

$$C^{-F} = C(D) - \sum_{(x_i,y_i)\in F} \phi(x_i, y_i) = C(D \setminus F). \tag{2}$$

Equation 2 follows from integer addition and does not depend on the image task. Its novelty is not the general additive-unlearning principle, established by Cao and Yang [Cao and Yang 2015], but its use as a native property of WiSARD and its evaluation against machine-unlearning methods.

The equality supplies a direct certificate. We train a reference model on $D \setminus F$ with the same mapping, normalize sparse zero entries, and compare every address counter with C^{-F}. The certificate is

$$\Delta_\infty = \max_{c,r,a} \left| C^{-F}_{c,r,a} - C(D \setminus F)_{c,r,a} \right|, \tag{3}$$

where c, r, and a index the class, RAM, and address. Exact deletion requires $\Delta_\infty = 0$; when all examples of one class are removed, the harness also removes that empty class from the active label set. Deleting $|F|$ examples touches $O(|F|R)$ counters for R RAMs per example and does not depend on the number of retained examples.

3.4. Design rationale

The synthetic task is an instrument for controlled measurement rather than evidence of broad real-world accuracy. Its repeated identities make complete class erasure possible, its known generative factors expose shortcuts, and its scene labels permit recognition and localization tests under one visual vocabulary. The exactness claim does not depend on this choice, because Equation 2 holds for any data encoded by the model; the task instead tests whether the resulting classifier retains useful accuracy while deleting cheaply.

We keep the recognition and localization studies because low deletion time would have little value for an unusable predictor. We also separate the algebraic certificate from the membership-inference audit: counter equality establishes the WiSARD guarantee, whereas the audit supplies a common statistical diagnostic for methods whose internal state cannot be compared address by address. This separation prevents chance-level attack performance from being mistaken for proof of exact deletion.

4. Experiments

The experiments separate predictive utility from deletion behavior. Section 4.1 states the research questions and metrics; Section 4.2 describes the data, models, and implementation; and Section 4.3 defines the deletion scenarios. This separation ensures that every result in Section 5 refers to a previously defined protocol.

4.1. Research questions and metrics

We study three research questions:

- **RQ1:** Does the synthetic benchmark require spatial visual information, and does WiSARD provide useful recognition and localization accuracy on it?
- **RQ2:** Does counter subtraction restore the same prediction-relevant counter state as training without the deleted examples?
- **RQ3:** How do deletion time, retained accuracy, support memory, and membership leakage compare across methods, deleted classes, and deletion scenarios?

For recognition, we report macro-F1, the unweighted mean of the per-class F1 scores. For localization, we report the median Euclidean distance between predicted and true target centers and the proportion within 12 pixels. For unlearning, *retained accuracy* is accuracy on test examples whose classes remain active; accuracy on the deleted class is shown only for complete class erasure, where that label is removed. We measure deletion wall-clock time and support memory, which includes the model and any statistics maintained specifically to permit deletion.

We assess forgetting in two complementary ways. The exactness certificate is the maximum absolute difference Δ_∞ between normalized RAM-address counters after deletion and those of a reference trained without the deleted examples. A confidence-threshold membership-inference attack (MIA) compares deleted training examples with fresh examples of the same source and reports the area under the receiver operating characteristic curve (AUC); 0.5 denotes no directional separation, and deviations on either side can be exploited by reversing the score. The MIA is a shared statistical diagnostic rather than proof of exactness [Shokri et al. 2017, Thudi et al. 2022].

4.2. Data, models, and implementation

Table 1 summarizes the data protocol. Recognition and class erasure use 120 training and 60 test crops per class, for 720 training and 360 test examples. The encoder resizes each crop to 24×24, applies a 13-bit Gaussian-spaced thermometer code to each RGB channel, and groups the resulting bits into addresses of size 12. We selected the thermometer type, resolution, and address size by grid search over three development seeds, then froze the encoder for all deletion experiments.

RQ1 compares WiSARD with tuned logistic regression and a 300-tree random forest on raw-pixel recognition. Localization compares Regression WiSARD with a random-forest window classifier and a histogram-of-oriented-gradients support-vector machine (HOG+SVM). The two learned window models receive the same stripe-and-color encoding, scan stride, and sub-pixel refinement. The comparison therefore changes the learner without changing the input information.

RQ2 and RQ3 compare six deletion methods. Retraining from scratch on retained data is the exact reference. Sharded, Isolated, Sliced, and Aggregated training (SISA)

Table 1. Data protocol. Training and test draws are independent; rotated characters remain inside the image, and only explicit occlusion creates partial visibility.

Study	Training	Test	Image policy
Recognition/unlearning	6×120 crops	6×60 crops	48^2 RGB crops
Localization	1,250 windows	150 scenes	96^2 scene, 6-px stride
Poison removal	720 + 100 poison	360 clean	Whitebeard as Waldo
Class robustness	30 mappings/class	six deleted labels	fixed draws

uses five random-forest shards and retrains affected shards [Bourtoule et al. 2021]; DaRE uses the authors' data-removal-enabled forest in greedy and random-split regimes [Brophy and Lowd 2021]; and gradient ascent supplies an inexpensive approximate baseline with known failure modes [Mavrothalassitis et al. 2025]. Because the published DaRE implementation is binary, we use one-vs-rest forests with the same 300-tree, depth-16 configuration as the retraining reference.

We ran the experiments on an Apple M4 Pro with 24 GB of memory using Python 3.13, scikit-learn 1.8.0, and wisardpkg 2.0.0a7. Timings use one CPU thread to measure algorithmic work without differences in default core counts; this policy can favor WiSARD because forests have more opportunities for parallel execution. The generator, fixed data seeds, model configurations, scripts, and evaluation harness are available in the public project repository.[1]

4.3. Deletion scenarios

The first scenario erases one complete class. We repeat WiSARD deletion for each label with 30 fixed bit-to-tuple mappings, then use the hardest target for the detailed cross-method comparison. An active-label mask removes a class whose counters have returned to zero, matching the retrained reference's label set. Confidence intervals use 10,000 percentile-bootstrap resamples of the independent mapping runs; baseline rows are single runs.

The second scenario adds 100 Whitebeard crops with the incorrect label Waldo and later removes precisely those rows. Because the Waldo class remains active, this scenario can reveal membership residue that complete label removal would conceal. The third scenario replays 30 contributor-level deletion requests and records cumulative time. The fourth inserts a purple trigger into detector windows labeled as Waldo, then measures trigger and genuine-target detection before and after the poisoned windows are removed.

5. Results

Section 5.1 answers RQ1 by testing benchmark adequacy and predictive utility. Section 5.2 answers RQ2 and the class-erasure part of RQ3 with repeated mappings and all six deletion targets. Section 5.3 completes RQ3 with poison removal, deletion sequences, and backdoor removal.

[1] https://github.com/muanlartins/masters/tree/waldo-notebook/notebooks/waldo

Table 2. Recognition and localization utility. Arrows indicate the preferred direction; localization learners share the encoding and scan grid.

Metric	WiSARD	Forest	Alternative
Clean recognition, macro-F1 ↑	0.90	**0.95**	0.84 (logistic)
Full difficulty, macro-F1 ↑	0.72	**0.77**	0.67 (logistic)
Localization error (px) ↓	**1.7**	2.4	–
Localized within 12 px ↑	0.97	**0.99**	0.62 (HOG+SVM)

5.1. Benchmark adequacy and predictive utility

Simple global statistics do not solve the synthetic task. A depth-three decision tree trained on aggregate color and stripe statistics reaches approximately 0.50 macro-F1 on clean crops, whereas a model trained on spatially arranged raw pixels reaches approximately 0.95. The difference indicates that class labels cannot be read directly from global palette proportions; stripe count, position, and orientation are necessary.

Table 2 shows that WiSARD provides useful, but not leading, predictive accuracy. On clean recognition, its macro-F1 of 0.90 lies between the random forest at 0.95 and logistic regression at 0.84. Under all difficulty factors, the ordering remains and all three methods lose accuracy. WiSARD uses approximately 10 MB, compared with 4.4 MB for the forest and 81 KB for logistic regression, so its contribution is not an absolute memory advantage.

The localization result supports the same conclusion. A global Regression WiSARD loses the target's spatial position and collapses toward the mean training location, producing 31-pixel median error. The sliding-window formulation reduces this error to 1.7 pixels and locates 97% of targets within 12 pixels, close to the random-forest window classifier at 2.4 pixels and 99%. These results establish that the model is a viable recognizer and localizer before deletion is considered.

5.2. Exactness and complete class erasure

Across 180 runs, end-to-end deletion takes 39.8 ms [39.5, 40.2] and retained accuracy is 0.921 [0.919, 0.923]. For each deleted class, an address-level comparison under a fixed mapping returns maximum counter difference zero. Raw serialization is not identical because the sparse representation retains inert zero-valued entries.

Odlaw is the hardest target, so Table 4 and Figure 2 use it for the cross-method comparison. WiSARD deletes 120 examples in 40.0 ms at 0.899 retained accuracy. Retraining reaches 0.950 at 1.42 s; DaRE-greedy reaches 0.953 at 13.87 s and 1.59 GB, while DaRE-random trades accuracy for 0.18 s deletion.

Class-erasure AUC is not privacy evidence because the inactive label gives members and nonmembers zero confidence. WiSARD instead has zero address-counter difference. For forests, "exact" follows each method's definition relative to its randomized training procedure, not parameter identity with the monolithic retraining row [Ginart et al. 2019, Brophy and Lowd 2021].

Table 3. WiSARD deletion for six classes and 30 mappings per class (mean [95% bootstrap CI]). Δ_∞ is a fixed-mapping address-counter certificate per class.

Deleted class	Retained accuracy	Deletion time (ms)	Δ_∞
Background	0.918 [0.916, 0.920]	39.8 [39.0, 40.9]	0
Waldo	0.921 [0.918, 0.923]	39.8 [39.3, 40.3]	0
Odlaw	0.899 [0.896, 0.902]	40.0 [39.3, 40.9]	0
Wenda	0.954 [0.952, 0.956]	40.2 [39.6, 41.0]	0
Whitebeard	0.922 [0.919, 0.924]	39.5 [39.2, 39.8]	0
Woof	0.912 [0.909, 0.914]	39.7 [39.1, 40.3]	0
Mean across targets	0.921 [0.919, 0.923]	39.8 [39.5, 40.2]	0

Table 4. Odlaw erasure. WiSARD is the mean of 30 mappings; other rows are single runs. MIA AUC is uninformative here because the label is inactive.

Method	Time (s)	Retained acc.	Memory (MB)	Exact	MIA AUC
WiSARD (ours)	**0.040**	0.899	9.3	yes	0.50
Retrain (gold)	1.424	0.950	**3.8**	yes	0.50
SISA	1.488	0.873	6.3	yes	0.50
DaRE-greedy	13.872	**0.953**	1585.7	yes	0.50
DaRE-random	0.177	0.853	89.1	yes	0.50
Gradient ascent	0.096	0.773	0.08	no	0.48

5.3. Poison removal, deletion sequence, and backdoor removal

Poison removal keeps the affected Waldo label active. Adding 100 Whitebeard crops labeled Waldo lowers WiSARD accuracy to 0.583 and yields MIA AUC 1.00. After subtraction, accuracy returns to 0.900, AUC falls to 0.529, and the counter certificate is zero. Exact baselines have AUC 0.479–0.507, whereas gradient ascent reaches 0.635 at 0.817 retained accuracy.

Over 30 requests, cumulative time grows for every method. WiSARD totals 0.242 s, DaRE-random 1.387 s, best-case SISA 3.798 s, and repeated single-threaded forest retraining 295.146 s. Tree-level parallelism could narrow the last wall-clock gap.

Backdoor removal uses the same operation. A purple trigger is initially classified as Waldo in 100% of test patches. Reversing 200 poisoned-window contributions takes 11 ms and reduces trigger detection to zero, while genuine-Waldo detection changes from 0.72 to 0.71.

6. Discussion

The results place WiSARD at a deletion-latency point rather than a new accuracy frontier. The retrained and DaRE-greedy forests retain approximately 0.95 accuracy, compared with 0.899 for WiSARD, but WiSARD completes deletion in approximately 40 ms without DaRE's 1.59-GB cache. Model choice therefore depends on the priority assigned to accuracy, latency, and memory.

Exact reversal requires a fixed encoder and mapping, plus the original label and

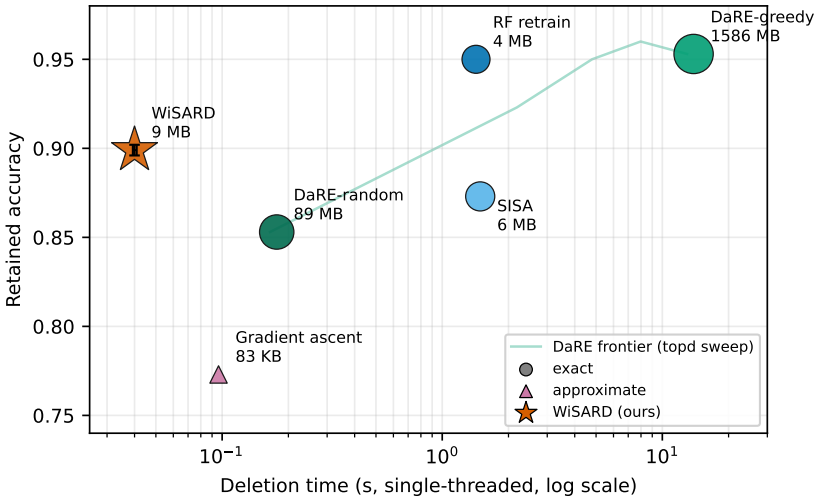


Figure 2. Odlaw erasure. Circles are exact baselines, the triangle is approximate, and the star is WiSARD; labels report support memory. WiSARD's 95% intervals are smaller than its marker.

either the encoded example or its activated addresses. The data system must enforce exactly-once deletion because repeated subtraction can underflow counters. Sparse zero-valued entries also remain allocated, so the 9.3 MB footprint does not shrink and can grow as training reaches new addresses.

The single synthetic task limits external validity. Hey-Waldo served only as a source of approximate colors and scale, measured from 14 of its 19 pages; it is not part of the benchmark or required by the method. Rotated characters also remain inside the image: opaque occlusion is tested, but crop-boundary truncation is not. These limits do not alter Equation 2, but they prevent claims of real-data competitiveness.

The confidence-threshold MIA is weaker than LiRA [Carlini et al. 2022] and its unlearning-specific extension [Hayes et al. 2025]; the counter certificate is stronger for WiSARD, while stronger attacks may expose more approximate-method residue. Timings use sequential forest implementations and one CPU thread; multi-core execution may narrow the wall-clock gap without changing algorithmic work.

7. Conclusion

This paper presented WiSARD counter subtraction as an exact unlearning mechanism, with a controlled benchmark and shared harness. The normalized RAM-address state matched retained-only training with maximum difference zero for every class; over 180 runs, mean deletion time was 39.8 ms and retained accuracy 0.921. WiSARD was the fastest exact method evaluated, while retraining and DaRE-greedy retained more accuracy and a plain forest used less memory. Deletion behavior therefore belongs beside accuracy and memory as a model-selection criterion.

Future work should test whether the deletion advantage persists on established real-world image and tabular benchmarks under both record- and class-deletion requests. It should also study how systems retain the labels and encoded contributions needed for reversal, reclaim memory after counters return to zero, and handle repeated or concurrent

requests. Finally, LiRA- and ULiRA-style reference-model attacks and an architecture-neutral protocol would strengthen privacy auditing and enable meaningful comparison with methods such as MUNKEY.

References

Aleksander, I., Thomas, W. V., and Bowden, P. A. (1984). WISARD: A radical step forward in image recognition. *Sensor Review*, 4(3):120–124.

Barthelmes, T. and Vidal, F. P. (2021). Where's wally? a machine learning approach. In *Computer Graphics and Visual Computing (CGVC)*, pages 28–31. The Eurographics Association.

Bledsoe, W. W. and Browning, I. (1959). Pattern recognition and reading by machine. In *Papers Presented at the Eastern Joint IRE-AIEE-ACM Computer Conference*, pages 225–232. ACM.

Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., Lie, D., and Papernot, N. (2021). Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE.

Brasil (2018). Lei n. 13.709/2018: Lei geral de proteção de dados pessoais (LGPD). Diário Oficial da União, Brasília, DF. Artigo 18: direito de eliminação.

Brophy, J. and Lowd, D. (2021). Machine unlearning for random forests. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 1092–1104. PMLR.

Cao, Y. and Yang, J. (2015). Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy (SP)*, pages 463–480. IEEE.

Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramèr, F. (2022). Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE.

Chen, M., Zhang, Z., Wang, T., Backes, M., Humbert, M., and Zhang, Y. (2021). When machine unlearning jeopardizes privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 896–911. ACM.

Constantinou, V. (2017). Hey, Waldo: Labeled images of the Where's Waldo puzzle. https://github.com/vc1492a/Hey-Waldo. GitHub repository.

European Parliament and Council of the European Union (2016). Regulation (EU) 2016/679 (general data protection regulation). Official Journal of the European Union, L 119. Article 17: Right to erasure ('right to be forgotten').

Ginart, A., Guan, M., Valiant, G., and Zou, J. (2019). Making AI forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Hayes, J., Shumailov, I., Triantafillou, E., Khalifa, A., and Papernot, N. (2025). Inexact unlearning needs more careful evaluations to avoid a false sense of privacy. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 497–519. IEEE.

Laguna, S., da Silva Gonçalves, J., Vandenhirtz, M., Ryser, A., Cannistraci, I., and Vogt, J. E. (2026). Rethinking machine unlearning: Models designed to forget via key deletion. In *ICLR 2026 Workshop on Test-Time Updates (TTU)*. Oral presentation.

Lima Filho, A. S., Guarisa, G. P., Lusquino Filho, L. A. D., Oliveira, L. F. R., França, F. M. G., and Lima, P. M. V. (2020). wisardpkg: A library for WiSARD-based models. *arXiv preprint arXiv:2005.00887*.

Lusquino Filho, L. A. D., Oliveira, L. F. R., Lima Filho, A. S., Guarisa, G. P., Felix, L. M., Lima, P. M. V., and França, F. M. G. (2020). Extending the weightless WiSARD classifier for regression. *Neurocomputing*, 416:280–291.

Mavrothalassitis, I., Puigdemont, P., Levi, N., and Cevher, V. (2025). Ascent fails to forget. In *Advances in Neural Information Processing Systems*, volume 38, pages 45330–45365. Curran Associates, Inc.

Nguyen, T. T., Huynh, T. T., Ren, Z., Nguyen, P. L., Liew, A. W.-C., Yin, H., and Nguyen, Q. V. H. (2025). A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, 16(5):108:1–108:46.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE.

Thudi, A., Jia, H., Shumailov, I., and Papernot, N. (2022). On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4007–4022. USENIX Association.

Xu, H., Zhu, T., Zhang, L., Zhou, W., and Yu, P. S. (2023). Machine unlearning: A survey. *ACM Computing Surveys*, 56(1):1–36.