Destructive Interference in Vector Retrieval: A Separation Result for Machine Unlearning

Nikit Phadke
nikitph@gmail.com

Abstract

We establish a fundamental separation between pointwise approximate nearest neighbor (ANN) indices and operator-based data structures with respect to machine unlearning. We prove that *no insertion-only operation* can remove an element from retrieval in selection-based indices without explicit deletion or query-time filtering. In contrast, we demonstrate that operator-based structures supporting signed superposition enable *constant-time unlearning* via destructive interference. Our experimental results on 100,000 vectors confirm that inserting negated vectors into FAISS-style indices leaves targets fully retrievable (rank 1, similarity 0.994), while operator-based field models achieve exact cancellation ($\phi \to 0$) in $O(1)$ time. This separation has immediate implications for GDPR/CCPA compliance, federated learning, and privacy-preserving retrieval systems. We argue that this constitutes a new computational primitive: *algebraic deletion*, which is fundamentally distinct from structural deletion in existing systems.

1 Introduction

The "Right to be Forgotten" under GDPR Article 17 and similar provisions in CCPA require that personal data be erasable upon request. For machine learning systems, this requirement has spawned the field of *machine unlearning* [2, 1], which seeks efficient methods to remove the influence of training data without full retraining.

Vector databases and retrieval-augmented generation (RAG) systems present a particularly acute challenge. These systems store embeddings of user data that can be retrieved via similarity search. Current approximate nearest neighbor (ANN) indices—including FAISS [7], ScaNN [6], and HNSW [9]—treat deletion as a structural operation requiring index modification or rebuild.

The core question. Can we achieve deletion semantics through *insertion alone*? Specifically, if we insert a "negative" or "anti-" vector, does this cancel the original vector's contribution to retrieval?

Our contribution. We prove that the answer is **no** for all ANN indices with pointwise selection semantics, and **yes** for operator-based data structures (OBDS) with signed superposition. This is not an implementation detail but a *semantic separation* arising from the fundamental difference between:

- **Selection operators**: $f(q) = \arg\max_{x_i \in X} s(q, x_i)$
- **Field operators**: $f(q) = \arg\max_x \phi(x)$ where $\phi = \sum_i w_i \mathcal{K}_{x_i}$

The key insight is that *deletion in selection-based systems is structural*, while *deletion in operator-based systems is algebraic*. This distinction enables constant-time unlearning as a first-class operation.

2 Background and Related Work

2.1 Approximate Nearest Neighbor Search

Given a dataset $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ and a query $q \in \mathbb{R}^d$, ANN search seeks:

$$\mathrm{ANN}(q) = \arg\max_{x_i \in X} s(q, x_i) \tag{1}$$

where $s(\cdot,\cdot)$ is a similarity function (inner product, cosine similarity, or negative Euclidean distance).

Modern ANN indices achieve sublinear query time through various techniques:

- **Inverted file indices (IVF)**: Partition space into Voronoi cells
- **Graph-based methods (HNSW)**: Navigate proximity graphs
- **Product quantization (PQ)**: Compress vectors for fast distance computation

All these methods share a common semantic property: retrieval selects from a discrete set of stored elements.

2.2 Machine Unlearning

Machine unlearning [2] aims to remove the influence of specific training data. Approaches include:

- **Exact unlearning**: Retrain from scratch (expensive)
- **SISA training** [1]: Partition data for efficient retraining
- **Influence functions** [8]: Approximate influence removal
- **Certified removal** [5]: Provide formal guarantees

For vector databases, unlearning typically requires index reconstruction, which is $\Omega(n)$ in the worst case and $\Omega(\log n)$ in the best case for hierarchical structures.

2.3 Kernel Methods and Operator Theory

Our approach draws on kernel methods [10] and reproducing kernel Hilbert spaces (RKHS). Given a positive-definite kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, the representer theorem states that solutions to regularized learning problems lie in the span of kernel evaluations at data points:

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x, x_i) \tag{2}$$

Crucially, *coefficients α_i can be negative*, enabling cancellation when $\alpha_i = -\alpha_j$ for $x_i = x_j$.

3 Theoretical Framework

3.1 Definitions

Definition 1 (Pointwise ANN Index). *A pointwise ANN index $\mathcal{I}$ over dataset X is a data structure supporting:*

- *$\textsc{Insert}(x)$: Add vector x to X*
- *$\textsc{Query}(q, k)$: Return* $\arg\max^{(k)}_{x_i \in X} s(q, x_i)$ *(top-k by similarity)*

where the query semantics are pointwise: each stored element is scored independently.

Definition 2 (Operator-Based Data Structure (OBDS)). *An operator-based data structure $\mathcal{F}$ maintains an information field:*

$$\phi(x) = \sum_{i=1}^{m} w_i K(x, c_i) \tag{3}$$

where K is a kernel function, $c_i \in \mathbb{R}^d$ are centers, and $w_i \in \mathbb{R}$ are signed weights. It supports:

- *$\textsc{Insert}(c, w)$: Add kernel centered at c with weight w*
- *$\textsc{Query}(q)$: Return local maximum of ϕ via gradient ascent from q*

Definition 3 (Retrievability). *An element x_j is retrievable from index $\mathcal{I}$ if there exists a query q such that $x_j \in \textsc{Query}(q, k)$ for some $k \geq 1$.*

Definition 4 (Attractor). *A point $x^* \in \mathbb{R}^d$ is an attractor of field ϕ if:*

1. *$\nabla\phi(x^*) = 0$ (critical point)*
2. *$\nabla^2\phi(x^*)$ is negative semi-definite (local maximum)*
3. *$\phi(x^*) > \tau$ for some threshold $\tau > 0$*

3.2 Main Results

Theorem 5 (Impossibility of Insertion-Based Deletion in ANN Indices)**.** *Let $\mathcal{I}$ be any pointwise ANN index with query semantics:*

$$\mathcal{I}(q) = \arg\max_{x_i \in X} s(q, x_i) \tag{4}$$

for similarity function $s : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$.

Then for any stored element $x_j \in X$, there exists no sequence of insertions $\{x_{n+1}, \ldots, x_{n+m}\}$ such that x_j becomes unretrievable, unless:

1. *x_j is explicitly removed from X, or*
2. *Query-time filtering excludes x_j.*

Proof. We proceed by contradiction. Suppose insertions $\{x_{n+1}, \ldots, x_{n+m}\}$ make x_j unretrievable.

Let $X' = X \cup \{x_{n+1}, \ldots, x_{n+m}\}$ be the augmented dataset. Consider the query $q = x_j$ (or $q = x_j + \epsilon$ for arbitrarily small ϵ if exact match is disallowed).

By pointwise semantics, the score $s(q, x_j)$ is computed independently of all other elements. Since $s(x_j, x_j)$ achieves maximum similarity (e.g., $s(x_j, x_j) = \|x_j\|^2$ for inner product, or $s(x_j, x_j) = 1$ for cosine similarity), we have:

$$x_j \in \arg\max_{x_i \in X'} s(q, x_i) \tag{5}$$

Thus x_j remains retrievable. Contradiction.

The only escape is if $s(q, x_k) > s(q, x_j)$ for some inserted x_k. But this requires x_k to be more similar to x_j than x_j is to itself, which is impossible for standard similarity functions. □ □

Remark 6. *The key observation is that ANN indices compute a selection over independent scores. Insertion adds candidates but cannot modify scores of existing elements.*

Theorem 7 (Constant-Time Unlearning via Signed Superposition)**.** *Let $\mathcal{F}$ be an OBDS with field:*

$$\phi(x) = \sum_{i=1}^{m} w_i K(x, c_i) \tag{6}$$

where K is a localized kernel (e.g., Gaussian $K(x, c) = \exp(-\|x - c\|^2/2\sigma^2)$).

Suppose c_j is an attractor with weight $w_j > 0$. Then inserting a kernel with weight $-w_j$ at center c_j removes the attractor. Specifically:

1. *The field value at c_j decreases: $\phi(c_j) \to \phi(c_j) - w_j K(c_j, c_j)$*
2. *For exact cancellation: if no other kernels overlap at c_j, then $\phi(c_j) = 0$*
3. *The operation is $O(1)$ time (append-only)*

Proof. Let ϕ be the original field and ϕ' be the field after insertion:

$$\phi'(x) = \phi(x) + (-w_j)K(x, c_j) = \phi(x) - w_j K(x, c_j) \tag{7}$$

At point c_j:

$$\begin{aligned} \phi'(c_j) &= \phi(c_j) - w_j K(c_j, c_j) && (8) \\ &= \sum_{i \neq j} w_i K(c_j, c_i) + w_j K(c_j, c_j) - w_j K(c_j, c_j) && (9) \\ &= \sum_{i \neq j} w_i K(c_j, c_i) && (10) \end{aligned}$$

If kernels are sufficiently localized (small σ) such that $K(c_j, c_i) \approx 0$ for $i \neq j$, then $\phi'(c_j) \approx 0$.

Since attractor status requires $\phi(c_j) > \tau$, the point c_j is no longer an attractor in ϕ'.

The insertion is $O(1)$ as it appends to the kernel list without modifying existing entries. □ □

Corollary 8 (Complexity Separation)**.** *Let n be the number of stored elements. Then:*

Operation	***ANN Index***	***OBDS***
Insertion	$O(\log n)$ *to* $O(n)$	$O(1)$
Deletion	$O(\log n)$ *to* $O(n)$	$O(1)$
Query	$O(\log n)$	$O(m \cdot d)$*

**Where m is the number of gradient steps. With spatial indexing, this improves to $O(\log n \cdot d)$.*

3.3 The Semantic Distinction

The fundamental difference can be summarized as:

Selection vs. Field Semantics

ANN Indices compute: $f(q) = \arg\max_{x_i \in X} s(q, x_i)$
This is a *selection operator* over discrete elements.

OBDS computes: $f(q) = \arg\max_{x \in \mathbb{R}^d} \phi(x)$ where $\phi = \sum_i w_i K(\cdot, c_i)$
This is an *extremum of a continuous field.*

The selection operator cannot implement cancellation because scores are computed independently. The field operator implements cancellation naturally through linear superposition.

4 Experimental Validation

4.1 Setup

We implement both a FAISS-style flat index and an OBDS field model in Rust for controlled comparison. Parameters:

- Dimension: $d = 128$
- Dataset size: $n = 100{,}000$ vectors
- Vectors: Random unit vectors (Gaussian, normalized)
- Similarity: Inner product (FAISS), Gaussian kernel with $\sigma = 0.1$ (OBDS)
- Query: Target vector plus small noise ($\epsilon = 0.01$)

4.2 Protocol

1. **Baseline**: Query for target vector v_{42}, verify retrievability
2. **Attempted deletion**: Insert negated vector $-v_{42}$
3. **Post-deletion query**: Repeat query, measure retrievability

For OBDS, retrieval uses gradient ascent with learning rate $\eta = 0.01$ and 100 steps.

4.3 Results

Table 1: Experimental results comparing FAISS and OBDS on the deletion task.

Metric	**FAISS**	**OBDS**
Target rank (before)	1	Attractor
Target similarity/field (before)	0.9939	1.0000
Target rank (after negative insertion)	1	Non-attractor
Target similarity/field (after)	0.9939	0.0000
Anti-vector in top-10	No	N/A
Target retrievable after deletion	**Yes**	**No**
Deletion operation cost	–	$O(1)$

FAISS results. After inserting the negated vector $-v_{42}$, the target remains at rank 1 with identical similarity score (0.9939). The anti-vector does not appear in the top-10 results because its inner product with the query is negative. *The negative vector is treated as an independent point with no effect on the original.*

OBDS results. After inserting a kernel with weight -1 at the target location:

- Field value at target: $1.0 \to 0.0$ (exact cancellation)
- Gradient ascent from query no longer converges to target
- Distance from converged point to target: 0.106 (vs. 0.0 before)
- *The target is provably unretrievable.*

4.4 Multi-Target Verification

We repeat the experiment for multiple target IDs to verify consistency:

Table 2: Field values before and after deletion for multiple targets.

Target ID	**FAISS Retrievable**	**OBDS ϕ (before)**	**OBDS ϕ (after)**
42	Yes	1.0000	0.0000
1000	Yes	1.0000	0.0000
5000	Yes	1.0000	0.0000

In all cases, FAISS targets remain retrievable while OBDS achieves exact cancellation.

5 Discussion

5.1 Implications for Machine Unlearning

Our results establish that *constant-time machine unlearning is impossible* in selection-based indices but *trivial* in operator-based structures. This has immediate practical implications:

GDPR/CCPA Compliance. Current vector databases cannot guarantee data erasure without index reconstruction. OBDS provides cryptographically verifiable deletion: the negative weight can be logged as proof that cancellation was applied.

Federated Learning. In federated settings, clients may need to withdraw their data. OBDS allows this without coordinator involvement—clients simply broadcast deletion kernels.

RAG Systems. Retrieval-augmented generation systems can implement "context amnesia" for sensitive queries by maintaining deletion masks as negative kernels.

5.2 Why This Is a New Primitive

We argue that algebraic deletion is not merely an optimization but a *new computational primitive.* The distinction is categorical:

- **Structural deletion**: Modify the data structure (remove pointers, rebuild indices)
- **Algebraic deletion**: Add an inverse element (append-only, no modification)

Algebraic deletion enables:

- **Versioned memory**: All operations are logged; any state can be reconstructed
- **Reversible unlearning**: Deletion can be undone by removing the negative kernel
- **Audit trails**: Compliance officers can verify deletion without accessing raw data
- **Concurrent operations**: No locks required; append-only is naturally concurrent

5.3 Limitations

Query complexity. OBDS queries require gradient computation over all kernels, giving $O(m \cdot d)$ complexity per step. For large m, this can be mitigated via spatial indexing (k-d trees, ball trees) or hierarchical kernel approximations.

Approximate cancellation. If the deletion kernel is not placed at exactly the original location, cancellation is imperfect. However, for GDPR compliance, approximate unlearning may suffice [5].

Kernel bandwidth. The effectiveness of cancellation depends on kernel locality (σ). Small σ gives precise cancellation but requires exact positioning; large σ is robust but may affect neighbors.

5.4 Hybrid Architectures

A practical deployment might combine both approaches:

1. Use FAISS for high-throughput approximate search
2. Maintain an OBDS "deletion mask" of negative kernels
3. At query time, filter FAISS results through the deletion mask

This achieves the performance of FAISS with the unlearning guarantees of OBDS.

6 Related Work

Machine unlearning. Cao and Yang [2] introduced machine unlearning for statistical query learning. Bourtoule et al. [1] proposed SISA training for efficient retraining. Our approach is orthogonal: we target retrieval systems rather than model parameters.

Differential privacy. DP provides privacy guarantees but does not enable point deletion [3]. OBDS complements DP by enabling removal of specific contributions.

Bloom filters. Counting Bloom filters [4] support deletion via decrements. Our OBDS can be viewed as a continuous generalization: a "kernel Bloom filter" over $\mathbb{R}^d$.

Soft attention. Transformer attention [11] computes weighted sums over values, similar to OBDS field evaluation. However, attention weights are query-dependent and non-negative, precluding cancellation.

7 Conclusion

We have established a fundamental separation between pointwise ANN indices and operator-based data structures with respect to deletion semantics. Our main results are:

1. **Theorem 5**: No insertion sequence can make an element unretrievable in selection-based indices.
2. **Theorem 7**: Signed superposition enables $O(1)$ deletion in operator-based structures.
3. **Experimental validation**: FAISS targets remain at rank 1 after negative insertion; OBDS achieves exact field cancellation.

The key insight is that *deletion in selection-based systems is structural, while deletion in operator-based systems is algebraic.* This distinction enables constant-time machine unlearning as a first-class operation.

We believe this work opens a new direction for privacy-preserving retrieval systems. The principle of destructive interference, borrowed from physics and signal processing, provides a theoretically grounded and practically implementable solution to the machine unlearning problem in vector databases.

Reproducibility. Code is available at `https://github.com/nikitph/bloomin/tree/master/negative-vector-experim`

References

[1] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE Symposium on Security and Privacy (SP)*, pages 141–159, 2021.

[2] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE Symposium on Security and Privacy (SP)*, pages 463–480, 2015.

[3] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[4] Li Fan, Pei Cao, Jussara Almeida, and Andrei Z Broder. Summary cache: a scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking*, 8(3):281–293, 2000.

[5] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *International Conference on Machine Learning (ICML)*, pages 3832–3842, 2019.

[6] Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning (ICML)*, pages 3887–3896, 2020.

[7] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[8] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, pages 1885–1894, 2017.

[9] Yu A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2018.

[10] Bernhard Schölkopf and Alexander J Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.

A Proof Details

A.1 Proof of Theorem 5 (Extended)

We provide additional rigor for the impossibility result.

Lemma 9. *For any positive-definite similarity function s, we have $s(x,x) \geq s(x,y)$ for all $y \neq x$.*

Proof. For inner product: $s(x,x) = \|x\|^2$ and $s(x,y) = \langle x,y\rangle \leq \|x\|\|y\|$ by Cauchy-Schwarz, with equality iff $y = \alpha x$ for $\alpha > 0$.

For cosine similarity on unit vectors: $s(x,x) = 1$ and $s(x,y) \leq 1$ with equality iff $x = y$.

For negative Euclidean distance: $s(x,x) = 0$ and $s(x,y) = -\|x-y\| < 0$ for $y \neq x$. □

Corollary 10. *The query $q = x_j$ always retrieves x_j at rank 1 in any pointwise ANN index using a positive-definite similarity function.*

This establishes that exact retrieval of any stored element is always possible, regardless of what other elements are inserted.

A.2 Gradient Computation for OBDS

For a Gaussian kernel field:

$$\phi(x) = \sum_{i=1}^{m} w_i \exp\left(-\frac{\|x - c_i\|^2}{2\sigma^2}\right) \tag{11}$$

The gradient is:

$$\nabla\phi(x) = \sum_{i=1}^{m} w_i \cdot K(x, c_i) \cdot \frac{c_i - x}{\sigma^2} \tag{12}$$

where $K(x, c_i) = \exp(-\|x - c_i\|^2/2\sigma^2)$.

Gradient ascent update:

$$x_{t+1} = x_t + \eta\nabla\phi(x_t) \tag{13}$$

followed by normalization to the unit sphere if operating on normalized embeddings.

B Experimental Details

B.1 Implementation

Our Rust implementation uses:

- `rand` crate for random vector generation
- `rayon` crate for parallel similarity computation
- No external dependencies for OBDS (pure Rust)

B.2 Hardware

Experiments run on a single machine with:

- CPU: Apple M-series (ARM64)
- Memory: 16GB
- Dataset generation: 228ms for 100,000 vectors
- Field construction: 9.5ms for 100,000 kernels

B.3 Reproducibility Checklist

✓ Random seed fixed for reproducibility
✓ All hyperparameters reported ($d = 128$, $n = 100{,}000$, $\sigma = 0.1$, $\eta = 0.01$)
✓ Source code available
✓ No proprietary data or models used