UNACT: GRADIENT-FREE UNLEARNING VIA TARGETED ACTIVATION INTERVENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning seeks to remove the influence of designated training data from a trained model without retraining from scratch. Retrain-free methods such as Selective Synaptic Dampening (SSD) and its label-free variant LFSSD avoid full retraining but still require backpropagation and parameter importance computed over the entire dataset. We ask: *what happens when a deletion request arrives with only a few images of the class to be forgotten?* To answer this question, we introduce **UnAct**, a gradient-free class-unlearning method that needs only forward passes over the forget images. UnAct scores late-layer units by their responses, attenuates the most responsive connections, and repeats this for up to 20 rounds using no gradients, no labels, and no retained data. On ResNet-18 trained with CIFAR-10, CIFAR-20, and CIFAR-100, UnAct is competitive with SSD and LFSSD when forgetting entire classes and, unlike them, never collapses the network when forget data is scarce. On ResNet-18, across all tested sizes, UnAct's retain accuracy stays within 2.5 points of retraining, while SSD and LFSSD, at their full-class operating points, lose up to 86 points on some classes. With five forget images on CIFAR-10, UnAct's distance to retraining is 0.21 points, against 67 for LFSSD and 90 for SSD, and re-selecting SSD's threshold at each size with an oracle does not close the gap. In preliminary transfer to ViT-B/16, UnAct's distance to retraining is 11.5 against 33.7 for SSD, and a request is 19$\times$ faster than SSD when SSD computes its importance at request time. The code is available at `https://anonymous.4open.science/r/UnAct-D688/`.

1 INTRODUCTION

A trained model can retain traces of the examples it has seen. Simply deleting a record from the dataset does not erase its influence; the model may continue to encode sensitive information, leaving it exposed to attacks such as membership inference or model inversion (Shokri et al., 2017; Fredrikson et al., 2015). This challenge has sparked the field of machine unlearning, which aims to selectively remove the impact of designated training data while retaining useful knowledge from the rest (Nguyen et al., 2025). Beyond legal imperatives like the GDPR's right to erasure (Voigt & von dem Bussche, 2017), unlearning is crucial for practical reasons: models must be able to discard mislabeled, corrupted, outdated, or harmful data without incurring the prohibitive cost of retraining from scratch, especially when deletion requests are frequent or affect only a small fraction of the dataset.

Existing unlearning methods reduce this cost in different ways. Some retain information from the original training process so that the contribution of deleted data can later be reversed, while others perform additional optimization after training (Graves et al., 2021; Chundawat et al., 2023a). A particularly attractive alternative is *post-hoc, retrain-free unlearning*, which directly modifies a trained model without retraining on the retained data. Selective Synaptic Dampening (SSD) (Foster et al., 2024b) is a representative of this approach: it uses Fisher information to estimate how important each parameter is to the forget data and to the full training set, and dampens the parameters that are disproportionately important to the data to be forgotten. Its label-free variant, LFSSD (Foster et al., 2024c), replaces the Fisher information with the gradient of the model's squared output norm, so that no labels are needed. Both methods still calculate the gradients, and both require an importance estimate over the full training set, which can be computed once and stored. We argue that the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

model's current activations provide a more direct signal for targeted unlearning: activations show which internal units the model actually uses to respond to the data to be forgotten.

Based on this insight, we introduce *UnAct*, a forward-only method for post-hoc class-level unlearning built on four design choices. (i) A *classifier-relative scope*: UnAct edits only the late units that write to the representation the classifier reads. (ii) A *forget-only score* of how much signal each unit carries from the forget images to the classifier, computed from forward passes with no labels, gradients or retained data; on transformers this requires measuring a unit's contribution to the class logit rather than its activation. (iii) *Rank-based selection*, which fixes the size of every edit in advance, independently of how many forget images arrive or how the scores are scaled. (iv) *Iterative soft attenuation* of each selected unit's incoming and outgoing weights with re-scoring, so that later rounds reach units that earlier rounds missed instead of deepening the same edit. Activation-based pruning (Hu et al., 2016) also ranks units by activation, but to remove the ones a network uses least, once, for compression; UnAct differs in all four respects, and (iii) is what bounds its edit when only a handful of forget images arrive, the regime in which importance-ratio dampening can collapse the network.

We evaluate UnAct against SSD and LFSSD on ResNet-18 across CIFAR-10, CIFAR-20, and CIFAR-100, and against SSD on ViT-B/16. With the full forget set on ResNet-18, UnAct reduces forget accuracy to 0% while keeping retain accuracy within 1.15 points of retraining on every dataset, competitive with both baselines. With few forget images the methods separate: across all forget-set sizes, UnAct's retain accuracy on ResNet-18 stays within 2.45 points of retraining, whereas SSD and LFSSD lose up to 86 points on some classes, and with five images on CIFAR-10 UnAct's distance to retraining is 0.21 points, against 67.07 for LFSSD and 89.85 for SSD. A one-round configuration (not the selected 20-round one on CIFAR-10 and CIFAR-20) removes a class in under 0.6 s, 1.8–3.3$\times$ faster than SSD with its importance precomputed. On ViT-B/16, UnAct's distance to retraining is 11.49 against 33.74 for SSD, and a request is 19$\times$ faster than SSD from cold. As with existing methods, UnAct improves output-level unlearning without removing the class from the learned representation. Our contributions can be summarized as follows:

1. We introduce **UnAct**, a forward-only method for post-hoc class-level unlearning that reads no gradients, labels or retained data, and whose rank-based selection bounds the edit independently of the number of forget images.
2. We establish the effectiveness and efficiency of UnAct on convolutional networks, with a preliminary transfer to transformers. UnAct is competitive with SSD and LFSSD on full forget sets, remains robust where the baselines collapse, requires far fewer forget examples, and, like the baselines, admits operating-point selection without a retrained model.
3. We show that iterative activation re-ranking widens the set of attenuated units rather than repeatedly deepening the same edit. The resulting coverage of the layer organizes UnAct's hyperparameters, and the number of rounds trades request time against closeness to retraining (Section 3.4, Appendix N).
4. We analyze what the edit changes inside the network. Units selected by activation alone become increasingly class-selective as the threshold rises (Appendix J), and on ResNet-18 editing only the last residual block suffices to forget (Section 3.2, Table 2). We also calibrate the standard evaluation: a trivial logit mask is as close to retraining as any method, and no method, including ours, removes the class from the penultimate features (Section 4.2, Appendix G).

2 Related Work

Machine unlearning methods span a spectrum defined by how much of the original training process they revisit at forgetting time. *Retraining-based approaches* retrain the model fully or partially, *optimization-based approaches* fine-tune the trained model, and *retrain-free approaches* directly modify the parameters associated with the forget set. Below, we review these approaches, together with methods that remove knowledge by editing individual units, and contrast them with UnAct.

Retraining-based unlearning. Retraining the model from scratch on the retained data guarantees complete removal of the forget set, making it the oracle baseline (Ginart et al., 2019; Cao & Yang,

Table 1: Summarized Comparison of unlearning families upon deletion request

Method	Requirements	Limitations
Retraining-based (Ginart et al., 2019; Bourtoule et al., 2021; Graves et al., 2021)	Retained data and full or partial retraining; SISA and Amnesiac also need shards, checkpoints or logged updates	Compute and storage cost; SISA and Amnesiac must be adopted at training time
Optimization-based (Chundawat et al., 2023a; Kurmanji et al., 2023; Jia et al., 2023; Fan et al., 2024)	Labels and/or retained data, many gradient steps	Fine-tuning cost, data access at request time
SSD (Foster et al., 2024b)	Labels, backward passes, Fisher information over the full dataset	Approximate method, unreliable with few forget images
LFSSD (Foster et al., 2024c)	Backward passes, sensitivity over the full dataset	Approximate method
Unit pruning (Wang et al., 2022; Pochinkov & Schoots, 2024; Chang et al., 2024)	Retain or reference data, often fine-tuning	Data access, permanent removal of units
UnAct (ours)	Forward passes over the forget images	Approximate method

2015; Li et al., 2025). However, it is computationally prohibitive, introduces latency for each deletion request, and requires permanent storage of the retained data. SISA (Bourtoule et al., 2021) reduces this cost by splitting the data into shards, training on each separately, and retraining only the shard that contains the deleted samples. Amnesiac unlearning (Graves et al., 2021) instead logs the updates made during training and subtracts those that involved the forget samples. Both methods must be set up at training time and require storing checkpoints or updates. SISA loses accuracy when deletions are frequent or shards are small (Bourtoule et al., 2021), and Amnesiac unlearning can damage retained classes when forget and retain samples appear in the same batches (Graves et al., 2021). UnAct, by contrast, is applied to an already-trained model and needs no retained data, checkpoints, or logs. Unlike retraining, it is an approximate method, so we use retrained models as the reference throughout our evaluation.

Optimization-based unlearning. Most approximate methods fine-tune the trained model, for example with gradient ascent on the forget set, fine-tuning on the retained data, teacher–student objectives such as Bad Teacher (Chundawat et al., 2023a) and SCRUB (Kurmanji et al., 2023), or sparsity-regularized fine-tuning (Jia et al., 2023). SalUn (Fan et al., 2024) computes a weight saliency map from the gradient of the forgetting loss and fine-tunes only the most salient weights, relabeling forget samples randomly while training on the retained data. These methods need labels, the retained data, or both, as well as many optimization steps. UnAct targets the setting in which neither labels nor retained data are available when a deletion request arrives. We therefore compare it with methods that work under the same conditions, and we make no claim of superiority over optimization-based methods, which use more information. Several methods relax data access but still optimize with gradients: Boundary Unlearning (Chen et al., 2023) and JiT (Foster et al., 2024a) use only forget samples, zero-shot unlearning (Chundawat et al., 2023b) synthesizes proxy data, and few-shot unlearning (Yoon et al., 2022) recovers a proxy training set by model inversion. UnAct shares their goal of working from few forget samples but performs no optimization.

Retrain-free dampening. Selective Synaptic Dampening (SSD) (Foster et al., 2024b) computes the diagonal Fisher information of every parameter over the forget set and over the full training set, and dampens the parameters that are disproportionately important for the forget set. SSD is fast and retrain-free, but it needs labels and a backward pass for every request, and its authors note that it is sensitive to its hyperparameters. Loss-Free SSD (LFSSD) (Foster et al., 2024c) removes the need for labels by replacing the Fisher information with a label-free sensitivity measure, the gradient of

the squared output norm (Aljundi et al., 2018), but it still requires backward passes and sensitivity computed over the full dataset. ASSD (Schoepf et al., 2024) selects α adaptively. Because none of these methods retrains the model, SSD and LFSSD are our main baselines, and LFSSD, which also reads no labels, is the closest to UnAct. UnAct instead uses only the forward activations of the forget images, and, as we show in Section 4.2, it remains reliable when only a few forget images are available, where SSD does not.

Unit-level removal. A separate line of work removes knowledge by editing individual units rather than individual weights. Removing a small set of units in a scene classifier removes its ability to recognize particular classes (Bau et al., 2020). For unlearning, Wang et al. (2022) score channels by a TF-IDF measure of class discrimination, prune the most discriminative channels for the target class in a federated model, and then fine-tune; Pochinkov & Schoots (2024) prune language-model neurons that are much more active on the forget data than on the retained data; Chang et al. (2024) perturb neurons ranked by layer-wise relevance propagation; and knowledge-neuron methods edit the MLP neurons attributed to a fact (Dai et al., 2022). Activation- and importance-based pruning criteria for model compression (Hu et al., 2016; Molchanov et al., 2017; Li et al., 2017) follow a similar idea for efficiency, removing the units a network uses least. UnAct is closest to these methods but differs in four ways: it scores units from forward passes on the forget images alone, with no retained data and no relevance backward pass; it selects a fixed fraction of units by rank, so the size of its edit does not depend on the scores or on the number of forget images; it weakens units over several rounds with re-scoring instead of removing them at once; and it needs no fine-tuning afterwards.

Positioning of UnAct. UnAct combines the efficiency of retrain-free methods with a unit-level edit whose size is fixed in advance. However, like SSD and LFSSD, UnAct is an approximate method, and we measure its forgetting empirically against retrained models. Since accuracy-based metrics and simple membership attacks can overstate forgetting (Lynch et al., 2024; Hayes et al., 2024; Carlini et al., 2022), we also examine how deeply its edit removes the forget class (Appendix G). Table 1 summarizes the comparison of UnAct with the existing methods.

3 METHOD

UnAct edits a trained classifier by weakening the internal units that carry the forget class to its output. Each round (Figure 1) scores the units of a classifier-relative scope by forward passes on the forget images, selects a fixed fraction of them by rank, and attenuates their incoming and outgoing parameters; the edited model is then rescored. Unlike activation-based pruning (Hu et al., 2016), which removes weakly used units once to compress a model, UnAct targets the units the forget class uses most, never removes them, and re-ranks after every edit.

3.1 PROBLEM FORMULATION

Let f_θ be a classifier trained on a dataset $\mathcal{D}$. Given a target class c, let $\mathcal{D}_f$ be the available forget images of class c, either the whole class or a subset of it, and let $\mathcal{D}_r$ be the remaining training data. Class-level unlearning seeks parameters θ' with which the model no longer retains class c but behaves as before on the other classes. A model retrained from scratch on $\mathcal{D}_r$ is the reference for ideal unlearning. UnAct is post-hoc and uses only the n images in $\mathcal{D}_f$: it does not access $\mathcal{D}_r$, read labels, optimize a loss, or run a backward pass.

3.2 SCOPE AND UNITS

A *unit* is an internal feature with its own incoming parameters In_j, which compute it, and outgoing parameters Out_j, through which it reaches later layers, such as a convolutional channel or a hidden neuron. The *scope* is the set of layers whose units UnAct may edit. We choose it by a rule that refers to the classifier rather than to a layer type: the scope is the late layers whose units write to the representation the classifier reads. These features are the most class-specific, and earlier, shared features stay untouched. The rule can be stated for any architecture with a classifier head; only the number of late layers to include depends on the architecture.

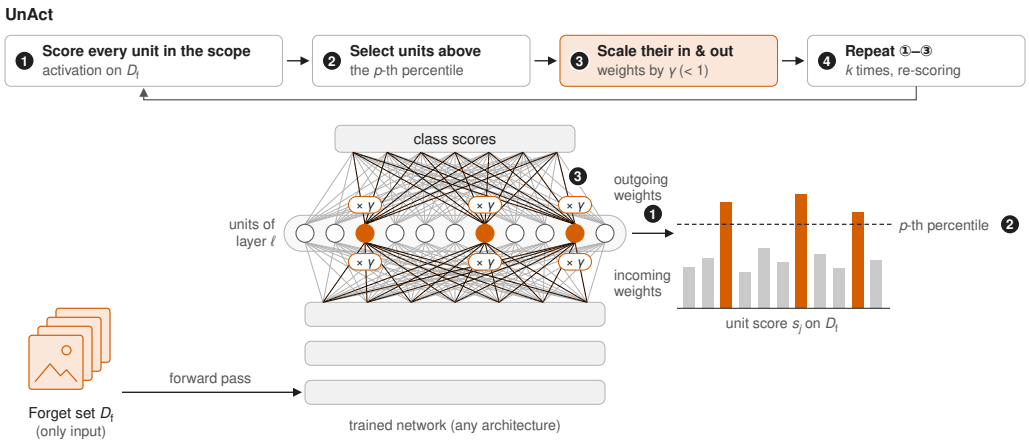

Figure 1: **Overview of UnAct.** Forward passes on the forget set $\mathcal{D}_f$ score every unit of the target layer (1). Units above the p-th percentile are selected (2), and their incoming and outgoing weights are multiplied by γ (3). The round is repeated k times, re-scoring the edited model each time (4). No labels, backward passes or retained data are used.

ResNet-18. The classifier reads the global average of the 512 output channels of the last residual block, so the scope is that block and each channel is a unit. In_j is the filter producing channel j with its BatchNorm scale and shift, and Out_j is the classifier's weight column for channel j.

ViT-B/16. The classifier reads the `[CLS]` token, to which every block writes through a shared residual stream. The units are the MLP hidden neurons of the last four blocks, which work better than the last two (Appendix L). In_j is the neuron's row of the first MLP projection with its bias, and Out_j is its column of the second. Residual-stream dimensions are not units, because every block writes to them.

3.3 Unit Scores

The score s_j estimates how much of the forget images' signal unit j carries to the classifier, that is, how strongly the model uses unit j when it sees class c. It needs only forward passes over $\mathcal{D}_f$ and is recomputed on the edited model every round. It does not compare the forget class with the retained ones, so a unit that responds to every class also scores high. In practice, the units above the high percentiles UnAct uses are strongly class-specific (Appendix J).

ResNet-18. The score is the channel's mean activation over the forget images, $s_j = \frac{1}{n}\sum_{x\in\mathcal{D}_f} a_j(x)$, where $a_j(x)$ is the post-ReLU output of channel j for image x, averaged over spatial positions, which is exactly the feature the classifier receives.

ViT-B/16. The same activation score, the signed mean GELU output, leaves the forget class almost intact on ViT-B/16 (Appendix L). GELU explains part of this. Most late-block neurons have a negative mean GELU output, so the selected set contains negative neurons whenever fewer than $(100-p)\%$ are positive; attenuation moves such a neuron toward zero and raises its score, so it is reselected every round. Scoring only the positive part of the GELU output avoids this but still fails ($\Delta = 95.41$, Table 17). The deeper cause is how a neuron reaches the classifier: only through its outgoing weights, which write into the shared residual stream that the final LayerNorm rescales. Its activation therefore does not show how much it contributes to class c. We instead score each neuron by its contribution to the class-c logit at `[CLS]`, $s_j = \frac{1}{n}\sum_{x\in\mathcal{D}_f} g_j(x)\,(\mathbf{w}_j^\top \mathbf{u}_c)/\sigma(x)$, where $g_j(x)$ is the neuron's GELU output at `[CLS]`, $\mathbf{w}_j$ its outgoing weights, $\mathbf{u}_c$ the direction of the class-c logit in the residual stream, and $\sigma(x)$ the final LayerNorm's scale. This is the neuron's direct logit attribution (Elhage et al., 2021), a first-order attribution that ignores paths through later blocks, and

Algorithm 1 UNACT

Require: model f_θ; forget images $\mathcal{D}_f$; scope; percentile p; factor γ; rounds k

```
set f_θ to inference mode                          ▷ BatchNorm statistics frozen, dropout off
for t = 1 to k do
    compute s_j for every unit in the scope        ▷ Section 3.3; forward passes only
    M_t ← { j : s_j ≥ Q_p(s) }                     ▷ ViT: top units by rank, ties by index
    In_j ← γ In_j,  Out_j ← γ Out_j   for all j ∈ M_t
end for
return f_θ'
```

it needs no backward pass (Appendix L). The class c is the one with the highest mean logit on $\mathcal{D}_f$ under the original model. No label is read, but this is effectively a pseudo-label that assumes a single-class forget set (Appendix I).

What carries over. Both scores measure the forget signal a unit sends to the classifier, so the choice depends on how units reach it. We expect the activation score to suit non-negative units the classifier reads directly and the class-token score to suit residual-stream transformers, but have tested one model of each kind.

3.4 SELECTION AND ITERATIVE ATTENUATION

Each round t selects the units whose scores lie at or above the p-th percentile of the scope's scores, $\mathcal{M}_t = \{\, j : s_j \geq Q_p(s) \,\}$, which is about the top $(100 - p)\%$ (ties: Appendix A). For every $j \in \mathcal{M}_t$, UnAct multiplies the parameters in In_j and Out_j by γ, with $0 < \gamma < 1$, weakening the unit without removing it. On ResNet-18 this is not a uniform rescaling of the channel, because BatchNorm's running statistics stay frozen (Appendix A). At a high percentile one round does not remove the class (Table 5), partly because on ResNet-18 the identity shortcut still carries part of a selected channel's signal (Appendix A); at lower percentiles one round can suffice, and CIFAR-100's selected configuration uses $k = 1$. UnAct therefore repeats scoring and attenuation for k rounds. An attenuated unit's score drops, so later rounds mostly select new units: iteration widens the edit rather than deepening it on a fixed set. By a union bound, the edited fraction of the scope after k rounds is at most k times the per-round fraction, and measured coverage stays close to this bound for small k (Appendix N). Thus (p, k) controls the breadth of the edit and γ its strength. Because selection is by percentile, these bounds hold whatever the number of forget images or the scale of the scores. SSD's threshold rule has no such bound: the number of parameters with $[\mathcal{D}_f]_i > \alpha[\mathcal{D}]_i$ depends on a statistic estimated from the forget images.

Algorithm and cost. Algorithm 1 summarizes the procedure. A request costs k forward passes over the n forget images and no backward pass, optimizer state or retained data, so its cost grows with $|\mathcal{D}_f|$ rather than with the training set. Given the model, $\mathcal{D}_f$ and (p, γ, k), the procedure is deterministic.

4 RESULTS

4.1 EXPERIMENTAL SETUP

Models and data. We train ResNet-18 (He et al., 2016) with the 3×3 CIFAR stem from scratch on CIFAR-10, CIFAR-20 (CIFAR-100 with its 20 superclass labels) and CIFAR-100 (Krizhevsky, 2009), reaching 94.97%, 85.34% and 77.35% test accuracy. We also fine-tune ViT-B/16 (Dosovitskiy et al., 2021), pre-trained on ImageNet-1k, on CIFAR-10 at 224×224 (Appendix B). Each request removes one class, and unless stated otherwise $\mathcal{D}_f$ contains all training images from that class. We fix five forget classes per dataset before any experiment and report averages over them; for each, a model retrained from scratch on the retain set with the same recipe is the reference.

Metric. A_r and A_f denote test accuracy on the retained and forgotten classes, respectively, following SSD and LFSSD (Foster et al., 2024b;c); they are measured on the test split and are distinct

Table 2: Removing one class with the full forget set, mean over the five forget classes. A_r, A_f: test accuracy on the retained and forgotten classes; Δ: distance to the retrained model (lower is better; best method in bold). Each method runs at its selected operating point; n/r: not run. ResNet-18 standard deviations, MIA, Avg. Gap and modified-parameter fractions are in Table 8; ViT-B/16 details are in Table 16.

	ResNet-18									ViT-B/16		
	CIFAR-10			CIFAR-20			CIFAR-100			CIFAR-10		
Method	A_r	A_f	$\Delta\downarrow$	A_r	A_f	$\Delta\downarrow$	A_r	A_f	$\Delta\downarrow$	A_r	A_f	$\Delta\downarrow$
Original model	95.16	93.28	93.72	85.15	88.96	89.50	77.31	81.00	81.39	98.33	97.92	98.11
Retrain	95.60	0.00	—	85.69	0.00	—	77.71	0.00	—	98.46	0.00	—
SSD	95.44	0.00	0.29	85.43	0.00	0.27	77.20	0.00	0.50	78.89	14.16	33.74
LFSSD	95.66	0.00	0.11	85.45	0.00	**0.24**	77.33	0.00	**0.38**	n/r	n/r	n/r
UnAct (ours)	95.59	0.00	**0.08**	84.90	0.00	0.80	76.56	0.00	1.15	91.93	4.96	**11.49**

from the training sets $\mathcal{D}_r$ and $\mathcal{D}_f$. Following prior work that reports gaps relative to a retrained model (Jia et al., 2023; Fan et al., 2024; Huang et al., 2024), we summarize each method using $\Delta = |A_r - A_r^{\text{gold}}| + |A_f - A_f^{\text{gold}}|$, measured in percentage points. We compute Δ separately for each forget class and then average. $\Delta = 0$ only when the edited model matches the retrained model on both terms, so it penalizes lost retain accuracy as well as excessive deviation on the forget classes. It is a two-term analogue of the Avg. Gap (Fan et al., 2024) and, like ToW (Zhao et al., 2024), is also used for hyperparameter selection. We additionally report an entropy-based membership inference attack (MIA) following SSD (Appendix B).

Baselines and tuning. We compare UnAct with two retrain-free methods closely related to our setting: SSD (Foster et al., 2024b) and its label-free variant LFSSD (Foster et al., 2024c), implemented according to their respective papers. SSD uses labels, gradients and a parameter importance over the full training set; LFSSD drops the labels but keeps the rest. On ResNet-18, each method has one operating point per dataset: the configuration with the lowest mean Δ over the five forget classes on a grid fixed before evaluation (100 configurations for UnAct and 90 each for SSD and LFSSD, with SSD's selected α restricted to the range reported in its paper; Appendix C). *All subsequent experiments reuse these operating points without re-tuning.* The selected points also transfer to three held-out classes on which SSD's original paper reports failures (Appendix E). In addition, a selection rule that does not require a retrained model selects the same operating point in 8 of 9 method–dataset pairs (Appendix D). On ViT-B/16, SSD uses the best point from an α–λ grid, while UnAct's configuration is selected using one forget class (Appendix L). We did not run LFSSD on ViT-B/16, whose importance pass over all 50,000 training images at 224×224 takes 1,248 s for SSD; SSD's ViT grid fixes the importance batch at 128. Accuracies are evaluated in fp32 (bfloat16 for the ViT-B/16 relearning curves and first selection stage; Appendices L and M) on NVIDIA L4 or A10 GPUs; the two GPUs agree to within one test image on the same checkpoint (Appendix B). All timing measurements use one L4 GPU. All experiments use seed 42.

Questions. We evaluate four progressively harder settings. (Q1) Given all training images of the forget class, how closely does each method approach retraining? (Q2) How many forget images does each method require, and does performance remain stable when only a few are available? (Q3) Do requested classes remain forgotten when requests are applied sequentially? (Q4) What is the computational cost of each request? We evaluate these questions on both architectures.

4.2 Findings

(Q1) With the full forget set, UnAct is within about one point of retraining on ResNet-18 and is clearly closest on ViT-B/16. On ResNet-18, all three methods drive forget accuracy to 0.00% on every dataset and stay within about one point of the retrained model (Table 2). UnAct is the closest on CIFAR-10 ($\Delta = 0.08$, against 0.11 for LFSSD and 0.29 for SSD), and LFSSD on CIFAR-20 and CIFAR-100 (0.24 and 0.38, against UnAct's 0.80 and 1.15); UnAct reaches this regime without labels, gradients, retained data or a pass over the training set. On ViT-B/16, where the scope, the

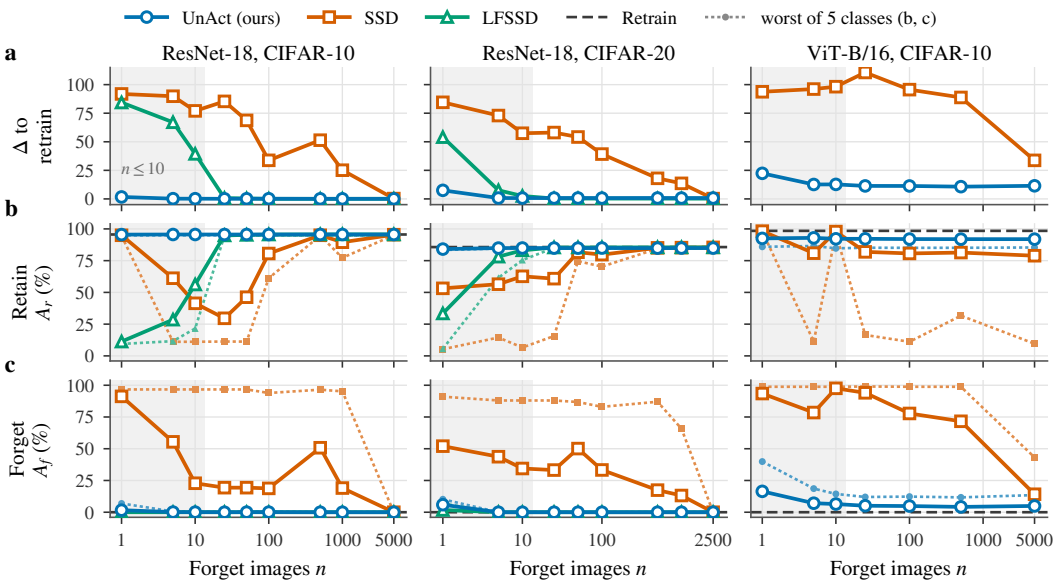

Figure 2: **Forgetting from few images.** Each method runs at its selected configuration at every forget-set size n (log scale; the last n is the whole class). (a) Δ, mean over the five forget classes. (b) Retain and (c) forget accuracy: solid, mean; dotted, the worst of the five classes (lowest A_r, highest A_f); dashed, the retrained models. Read (b) and (c) together: a collapsed network also reaches $A_f = 0$. The shaded band marks $n \leq 10$. CIFAR-100 and a log-scale Δ are in Figure 5.

score (with a pseudo-label) and the tie-breaking are adapted (Sections 3.2–3.4), UnAct is 2.9 times closer to retraining than SSD ($\Delta = 11.49$ against 33.74) and closer on 3 of the five classes; on the four classes its configuration was not chosen on, Δ is 13.60 against 35.71. The gap comes from safety: UnAct's lowest per-class retain accuracy is 85.33%, whereas SSD collapses one class to 9.57% and leaves another largely unforgotten (Tables 16 and 17). The same procedure thus carries over from a convolutional network to a transformer, although UnAct's mean retain accuracy on ViT-B/16 remains 6.5 points below the retrained model's. The membership attack does not separate the methods: every method drives it to at most 0.9% on ResNet-18, below the retrained models' rates, so we do not use it to rank them (Table 8).

(Q2) With few forget images, UnAct stays safe where the baselines collapse. We vary the number n of forget images each method receives, from one to the whole class, and still evaluate the removal of the whole class (Figure 2). UnAct uses only the n images, whereas SSD and LFSSD also receive, at every n, a parameter importance computed from gradients over the entire training set, as their papers prescribe, so the comparison gives the baselines strictly more information than UnAct. Even so, over all n, forget classes and ResNet-18 datasets (125 runs), UnAct's retain accuracy is never more than 2.45 points below the retrained model's. SSD and LFSSD, at the operating points that serve them well on the full class, damage the network when given few images: on CIFAR-10 their per-class retain accuracy falls by up to 85.34 and 86.09 points, and their mean retain accuracy drops to 29.60% and 11.26%, against UnAct's lowest mean of 95.51%. On ViT-B/16, UnAct's largest per-class drop is 13.7 points against 88.5 for SSD. Row (c) shows the complementary failure: on the classes it does not collapse, SSD leaves the forget class largely unforgotten until it sees most of the class, whereas UnAct's forget accuracy on ResNet-18 is near zero from $n = 5$.

Five images are enough for UnAct. With $n = 5$, UnAct's Δ is already 0.21, 0.77 and 1.37 on the three ResNet-18 datasets and 12.63 on ViT-B/16, within about one point of its full-class value there. On CIFAR-10 and CIFAR-20 it is the closest method at every $n \leq 10$; at $n = 5$ it is 321 and 10 times closer than LFSSD, and 430 and 94 times closer than SSD. SSD needs the entire class to come within a factor of two of its best Δ on ResNet-18 (Appendix H), and on ViT-B/16 its Δ is at least 88.78 at every $n \leq 500$ (Figure 9). LFSSD, drawing on its full-training-set gradients, reaches

a lower Δ than UnAct on CIFAR-100 from $n = 5$ (0.44 against 1.37), on CIFAR-20 from $n = 25$ and on CIFAR-10 at $n = 100$ (0.09 against 0.17), but with few images it collapses some classes on CIFAR-10 and CIFAR-20, as SSD does. Re-selecting SSD's α at every n within the range its paper reports, with the retrained model as an oracle, still leaves its Δ at $n = 5$ at 87.5, 64.6 and 46.1, far above UnAct's untuned values (Table 11).

(Q3) Sequential requests. Applied five times in sequence to the already-edited ResNet-18, UnAct keeps every requested class forgotten on CIFAR-10 and CIFAR-20 (forget accuracy at most 2.7% over five class orders), whereas SSD's forget accuracy on the requested classes exceeds 50% by the fifth request. UnAct's retain accuracy erodes on CIFAR-20 after the third request, to 70.1% after the fifth against 85.7% for SSD, and on CIFAR-100 SSD is the better choice (Appendix I).

(Q4) Cost. UnAct's cost is k forward passes over the forget images; it never reads the training set and stores nothing between requests. On ResNet-18, the best one-round configuration is within 0.06, 0.05 and 0.00 of the selected Δ and removes a whole class in 0.58, 0.38 and 0.12 s: 3.3, 3.0 and 1.8 times faster than SSD with its importance precomputed, and 35–215 times faster from cold (Appendix K). On ViT-B/16 a full-class request takes UnAct 73 s against SSD's 1,373 s, of which 1,248 s is the importance pass (19$\times$ faster from cold, 1.7$\times$ with the importance precomputed), and five images take 0.1 s. LFSSD has the same two-pass structure as SSD.

What Δ certifies. A trivial baseline that zeroes the class logit reaches $\Delta = 0.09$, 0.25 and 0.33 on ResNet-18, as close to retraining as any method, and a linear probe recovers the forget class from the penultimate features with AUROC of at least 0.98 after every method, including ours (Table 9). Δ therefore certifies that a class is no longer predicted, not that it has been removed, and we use it only in that sense.

5 Discussion and conclusion

We asked how little information a retrain-free method needs to remove a class from a trained classifier. UnAct answers with a forward-only edit: it scores late-layer units by the forget signal they send to the classifier, selects a fixed fraction of them by rank, attenuates them softly and rescores the edited model, reading no gradients, labels or retained data.

Robustness when forget data is scarce. With the whole class, UnAct is competitive with SSD and LFSSD on ResNet-18 while using strictly less information. The forget-set-size experiments are where the methods separate. There the baselines receive more information, an importance over the full training set at every n, but keep their full-class operating points; only SSD is also re-tuned per n. Even so, on ResNet-18 UnAct is the only method that is safe at every forget-set size, and with at most ten images it is the closest method to retraining on CIFAR-10 and CIFAR-20, a gap that oracle re-selection of SSD's threshold does not close. As n grows, LFSSD becomes comparable to UnAct and is closer on CIFAR-100 from $n = 5$, on CIFAR-20 from $n = 25$ and on CIFAR-10 at $n = 100$, while SSD needs the entire class to approach its best value. On ViT-B/16, UnAct is closer to retraining than SSD at every forget-set size.

Limitations. UnAct suppresses the forget class rather than erasing it: a few fine-tuning steps on a handful of forget images restore much of the class, as they do for SSD. It is designed for one class per request, since its scoring presumes that the forget images come from a single class. With the whole class available, it is slightly less close to retraining than SSD and LFSSD on CIFAR-20 and CIFAR-100 and edits more parameters than either, and on ViT-B/16 its retain accuracy remains below that of the retrained model. Finally, the score must be adapted to how units reach the classifier (Section 3.3). Our evidence covers CIFAR-scale data, two architectures and one training seed, with operating points selected on the test split for all methods.

AI use statement

In this work, we used generative AI tools to debug code, to format LaTeX tables and the algorithm, to identify related literature, and to review and edit the manuscript text, including the proper wording and framing of results the authors had already obtained. We have not used generative AI tools

to propose the method, formulate hypotheses, or generate or clean datasets, and synthetic data generation, translation, and qualitative data analysis are not applicable to this work. We have reviewed all AI-assisted work: AI-suggested code fixes were reviewed and tested by the authors, the baseline was cross-checked manually against the reference implementation, all results come from code written and executed by the authors, and every edited claim and added citation was checked by the authors against the results files and the cited papers. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

Section 3 and Algorithm 1 describe UnAct, and Appendix A gives further implementation details. Section 4.1 and Appendix B describe the datasets, forget classes, training setups, evaluation protocol, hardware and baseline implementations. Appendix C lists the hyperparameter grids and the selected operating points. Source code, experiment scripts and results files are provided in the supplementary material and at the link in the abstract.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations, Workshop Track*, 2017.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *European conference on computer vision*, pp. 144–161. Springer, 2018.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, 2020.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pp. 141–159. IEEE, 2021.

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

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *IEEE Symposium on Security and Privacy*, pp. 1897–1914, 2022.

Wenhan Chang, Tianqing Zhu, Ping Xiong, Yufeng Wu, Faqian Guan, and Wanlei Zhou. Zero-shot class unlearning via layer-wise relevance analysis and neuronal path perturbation. *arXiv preprint arXiv:2410.23693*, 2024.

Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7766–7775, 2023.

Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7210–7217, 2023a.

Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 18:2345–2354, 2023b.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 8493–8502, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2010.11929.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *International Conference on Learning Representations*, volume 2024, pp. 53643–53673, 2024.

Jack Foster, Kyle Fogarty, Stefan Schoepf, Cengiz Öztireli, and Alexandra Brintrup. Zero-shot machine unlearning at scale via Lipschitz regularization. *arXiv preprint arXiv:2402.01401*, 2024a.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 12043–12051, 2024b.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Loss-free machine unlearning. In *The Second Tiny Papers Track at ICLR 2024*, 2024c. arXiv:2402.19308.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1322–1333, 2015.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Zou. Making ai forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11516–11524, 2021.

Jamie Hayes, Ilia Shumailov, Eleni Triantafillou, Amr Khalifa, and Nicolas Papernot. Inexact unlearning needs more careful evaluations to avoid a false sense of privacy. *arXiv preprint arXiv:2403.01218*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016.

Zhehao Huang, Xinwen Cheng, JingHao Zheng, Haoran Wang, Zhengbao He, Tao Li, and Xiaolin Huang. Unified gradient-based machine unlearning with remain geometry enhancement. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2409.19732.

Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model sparsity can simplify machine unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2304.04934.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In *Advances in Neural Information Processing Systems*, 2023.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient ConvNets. In *International Conference on Learning Representations (ICLR)*, 2017. arXiv:1608.08710.

Na Li, Chunyi Zhou, Yansong Gao, Hui Chen, Zhi Zhang, Boyu Kuang, and Anmin Fu. Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 36(8):13709–13729, 2025.

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in LLMs. *arXiv preprint arXiv:2402.16835*, 2024.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations (ICLR)*, 2017. arXiv:1611.06440.

Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–46, 2025.

Nicholas Pochinkov and Nandi Schoots. Dissecting language models: Machine unlearning via selective pruning. *arXiv preprint arXiv:2403.01267*, 2024.

Stefan Schoepf, Jack Foster, and Alexandra Brintrup. Parameter-tuning-free data entry error unlearning with adaptive selective synaptic dampening. *arXiv preprint arXiv:2402.10098*, 2024.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Paul Voigt and Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer, 2017.

Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Federated unlearning via class-discriminative pruning. In *Proceedings of the ACM Web Conference 2022*, pp. 622–632, 2022.

Y. Yoon, J. Nam, H. Yun, J. Lee, D. Kim, and J. Ok. Few-shot unlearning by model inversion. *arXiv preprint arXiv:2205.15567*, 2022.

Kairan Zhao, Meghdad Kurmanji, George-Octavian Bărbulescu, Eleni Triantafillou, and Peter Triantafillou. What makes unlearning hard and what to do about it. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2406.01257.

APPENDIX

A METHOD DETAILS

A.1 THE IDENTITY PATH IN RESIDUAL BLOCKS

In ResNet-18 the score is measured on the block output, which includes the identity shortcut, while the update edits only the residual branch. A selected channel therefore still receives signal through the shortcut, and one round does not suppress it completely. This is one reason several rounds help on ResNet-18. Fully connected layers have no such path.

A.2 WHAT ATTENUATION DOES ON RESNET-18

In_j is the channel's `conv2` filter together with its BatchNorm scale w and shift b, and all three are multiplied by γ. The running statistics μ and σ stay frozen, so the channel's pre-activation $w(z-\mu)/\sigma + b$ becomes $\gamma[w(\gamma z-\mu)/\sigma + b]$, not γ times its original value. The classifier column Out_j is scaled exactly, and the classifier bias is not edited.

A.3 SELECTION COUNT AND TIES

With Q_p the linearly interpolated percentile, the selection rule of Section 3.4 selects $N - \lceil (N-1)\,p/100 \rceil$ of N units when the scores are distinct, which is slightly more than $N(1-p/100)$. When many scores are tied, the rule selects every tied unit. On ViT, many scores are exactly zero at the class token, so for the class-token score (Section 3.3) we select the same number of units by rank, breaking ties by index.

B TRAINING AND EVALUATION DETAILS

ResNet-18. We use the CIFAR stem, a 3×3 stride-1 first convolution with no max-pool; the stock ImageNet stem would downsample 32×32 inputs to 8×8 before the first residual block. We train for 100 epochs with SGD, momentum 0.9, weight decay 5×10^{-4}, batch size 256, and a cosine schedule peaking at a learning rate of 0.2 after 5 linear warm-up epochs, with random crops (padding 4) and horizontal flips. Training uses bfloat16 autocast; every reported accuracy is computed in fp32. Each retrained model uses the same recipe on the retain set alone.

ViT-B/16. We fine-tune torchvision's `vit_b_16` with `IMAGENET1K_V1` weights, pre-trained on ImageNet-1k only. The SSD paper uses a ViT-B/16 pre-trained on ImageNet-21k, so our ViT numbers are not a reproduction of its ViT results. Images are upsampled from 32×32 to 224×224 after the same crop and flip augmentation as for ResNet-18. We train for 8 epochs with SGD, momentum 0.9, no weight decay, learning rate 0.01 at batch size 128, gradient clipping at norm 1.0, and a linear warm-up over 6% of the steps followed by cosine decay, under bfloat16 autocast. The five retrained ViT models use the same recipe on the retain set. ViT training is not bit-reproducible, because the attention backward pass has no deterministic kernel; all unlearning and evaluation passes are forward-only and deterministic.

Label-free check. For forget class 3 of each dataset we run each method twice, once with the true forget-set labels and once with every label replaced by a constant, and compare the edited weights. UnAct runs at its selected configurations; SSD and LFSSD at $\alpha = 10$, $\lambda = 1$ and importance batch 256. UnAct's and LFSSD's weights are bit-identical in every case, and SSD's differ in 10.9–11.0 million parameters.

Forget classes and evaluation. The forget classes, fixed before any experiment, are $\{0, 2, 3, 5, 8\}$ for CIFAR-10 (also used for ViT-B/16), $\{3, 4, 10, 14, 19\}$ for CIFAR-20 and $\{3, 20, 51, 69, 85\}$ for CIFAR-100. Evaluation uses fp32, deterministic kernels and a fixed batch size. Rows were evaluated on an NVIDIA L4 or A10; on the same checkpoint the two devices differ by at most one test image (about 0.01 points of A_r, and 0.1, 0.2 and 1.0 points of A_f on CIFAR-10, CIFAR-20 and CIFAR-100), and the tables that mix devices say so. The edit itself is recomputed on the GPU that runs each experiment, so full-class Δ values taken from the A10 runs (Tables 3, 5, 6, 9, 10, 11 and 15) can differ from the L4 run of Table 2 by 0.01, for example 0.51 against 0.50 for SSD and 1.14 against 1.15 for UnAct on CIFAR-100.

Membership inference. As in SSD, a logistic regression on prediction entropy is trained to separate retain-set training images from test images; MIA is the fraction of forget-set training images it labels as members.

Baselines. SSD and LFSSD compute their importance over the full training set once per model and reuse it for every forget-set size. Their grids include the importance batch size because the importance is averaged over batches, so the meaning of α depends on it. Our LFSSD takes the absolute value of the batch-mean gradient of $\|f(x)\|_2^2$, as the authors' reference code does, rather than averaging per-sample absolute gradients as in the LFSSD paper's equation; within-batch cancellation therefore makes its importance depend on the batch size, which is one axis of its grid. On ViT-B/16, SSD's grid is $\alpha \in \{2, 5, 6, 7, 8, 9, 10, 15, 25, 50\}$ and $\lambda \in \{0.1, 0.5, 1\}$ at importance batch 128, selected on all five classes with bfloat16 evaluation on 2,000 retain test images and re-evaluated at fp32 on the full split.

Table 3: Complete SSD grid, Δ averaged over the five forget classes, at each dataset's selected λ (rows: α, columns: importance batch size). The selected operating point is in bold; the full λ axis is in the results file. $\alpha < 5$ lies outside the range SSD's paper reports and was never selected.

CIFAR-10 ($\lambda = 1$)

α	64	256	512
2	32.75	60.06	75.43
4	18.41	20.66	28.09
6	2.99	8.53	11.87
8	31.23	**0.29**	0.61
10	90.96	93.49	93.36
12	93.72	93.73	93.74
15	93.74	93.70	93.74
20	93.74	93.71	93.72
30	93.72	93.72	93.72
50	93.72	93.72	93.72

CIFAR-20 ($\lambda = 1$)

α	64	256	512
2	3.59	12.05	26.45
4	1.02	3.62	7.58
6	0.54	1.64	3.08
8	0.34	0.83	1.67
10	4.76	0.45	0.91
12	11.98	0.32	0.44
15	30.12	5.16	**0.27**
20	89.30	88.79	89.68
30	89.50	89.50	89.50
50	89.50	89.50	89.50

CIFAR-100 ($\lambda = 0.5$)

α	64	256	512
2	50.46	75.42	76.10
4	23.09	67.60	74.74
6	13.01	53.62	71.85
8	8.72	33.43	63.89
10	5.89	24.32	54.21
12	4.21	18.74	42.15
15	2.29	11.16	29.62
20	1.40	8.09	17.90
30	0.86	2.50	7.43
50	**0.51**	0.75	1.74

Table 4: Complete LFSSD grid, Δ averaged over the five forget classes, at each dataset's selected λ (rows: α, columns: importance batch size). The selected operating point is in bold; the full λ axis is in the results file. $\alpha < 5$ lies outside the range SSD's paper reports and was never selected.

CIFAR-10 ($\lambda = 0.1$)

α	64	256	512
2	83.98	84.22	84.49
4	10.54	54.27	74.67
6	**0.11**	0.78	6.54
8	3.84	8.29	7.31
10	93.72	93.72	93.72
12	93.72	93.72	93.72
15	93.72	93.72	93.72
20	93.72	93.72	93.72
30	93.72	93.72	93.72
50	93.72	93.72	93.72

CIFAR-20 ($\lambda = 0.1$)

α	64	256	512
2	76.19	80.00	80.05
4	11.48	51.54	69.57
6	1.18	10.87	22.37
8	0.45	1.33	4.69
10	**0.24**	0.41	0.85
12	0.31	0.31	0.36
15	3.90	5.70	4.57
20	89.49	89.49	89.49
30	89.49	89.49	89.49
50	89.49	89.49	89.49

CIFAR-100 ($\lambda = 0.5$)

α	64	256	512
2	76.21	76.66	76.69
4	48.37	74.96	76.33
6	20.58	51.70	69.22
8	3.61	27.89	43.92
10	1.77	13.10	26.73
12	1.24	6.22	16.03
15	0.86	1.71	5.56
20	0.56	0.84	1.14
30	**0.38**	0.50	0.48
50	7.77	37.15	32.77

C HYPERPARAMETER GRIDS AND SENSITIVITY

Tables 3–5 and Figure 3 report each method's full grid. Every method needs per-dataset tuning: SSD's selected α moves from 8 to 15 to 50 across the three datasets, LFSSD's from 6 to 10 to 30, and UnAct's (p, γ, k) from $(99, 0.01, 20)$ and $(99, 0.1, 20)$ to $(90, 0.3, 1)$.

Figure 4 sweeps each method along its own axes. UnAct is insensitive to γ below a threshold, and its (p, k) trade off through the coverage Γ_k of Eq. (1) (Appendix N).

D SELECTION WITHOUT A RETRAINED MODEL

Every operating point in the paper is chosen by Δ, which needs a model retrained without the forget class. Table 6 asks how much this matters. For class removal the retrained model never predicts the removed class, so its forget accuracy is 0 in every run we trained, and $\Delta = |A_r - A_r^{\text{gold}}| + A_f$. The only term that needs retraining is A_r^{gold}. The retrain-free rule therefore requires forget accuracy 0 and maximises retain accuracy; it can differ from Δ-selection when a configuration overshoots the retrained model's retain accuracy, or when the Δ-optimal configuration leaves a small nonzero forget accuracy. It picks the Δ-optimal configuration in 8 of 9 cases; the exception is UnAct on CIFAR-10, where its choice is only 0.03 worse. The rule still reads a labelled retain evaluation set, here the same test split we report. A practitioner would use a held-out validation split, so this shows that retraining is unnecessary, not that the selection is held-out.

Transfer across datasets separates the methods more clearly. A configuration chosen on the other two datasets, or one configuration shared by all three, keeps UnAct within $\Delta \leq 3.46$ on every dataset for a single request (the shared configuration is (99, 0.01, 20)). SSD's and LFSSD's transferred configurations fail on some dataset (Δ up to 32.42 and 93.72), because their selected α grows five-fold (LFSSD) and more than six-fold (SSD) from CIFAR-10 to CIFAR-100: an importance ratio does not have the same meaning on every model, whereas a percentile and an attenuation factor do.

Table 5: Complete UnAct grid, Δ averaged over the five forget classes, at each dataset's selected γ (rows: p, columns: rounds k). Selected operating point in bold. The usable region runs diagonally in (p, k): raising p and raising k trade off at roughly constant coverage (Eq. 1). The full γ axis is in the results file.

CIFAR-10 ($\gamma = 0.01$)

p	rounds k			
	1	5	10	20
50	0.58	5.90	7.10	7.22
75	0.15	0.36	0.48	0.46
90	61.56	0.12	0.15	0.15
95	80.79	0.14	0.12	0.11
99	89.96	78.66	52.65	**0.08**

CIFAR-20 ($\gamma = 0.1$)

p	rounds k			
	1	5	10	20
50	3.47	27.78	74.17	77.69
75	0.84	5.40	13.04	20.75
90	25.52	1.84	3.59	5.68
95	60.71	0.82	1.64	3.00
99	82.38	55.14	18.33	**0.80**

CIFAR-100 ($\gamma = 0.3$)

p	rounds k			
	1	5	10	20
50	17.15	60.81	73.59	75.04
75	4.17	33.87	57.07	66.56
90	**1.14**	9.54	24.42	47.96
95	11.62	3.21	9.49	23.83
99	58.57	6.00	1.32	2.85

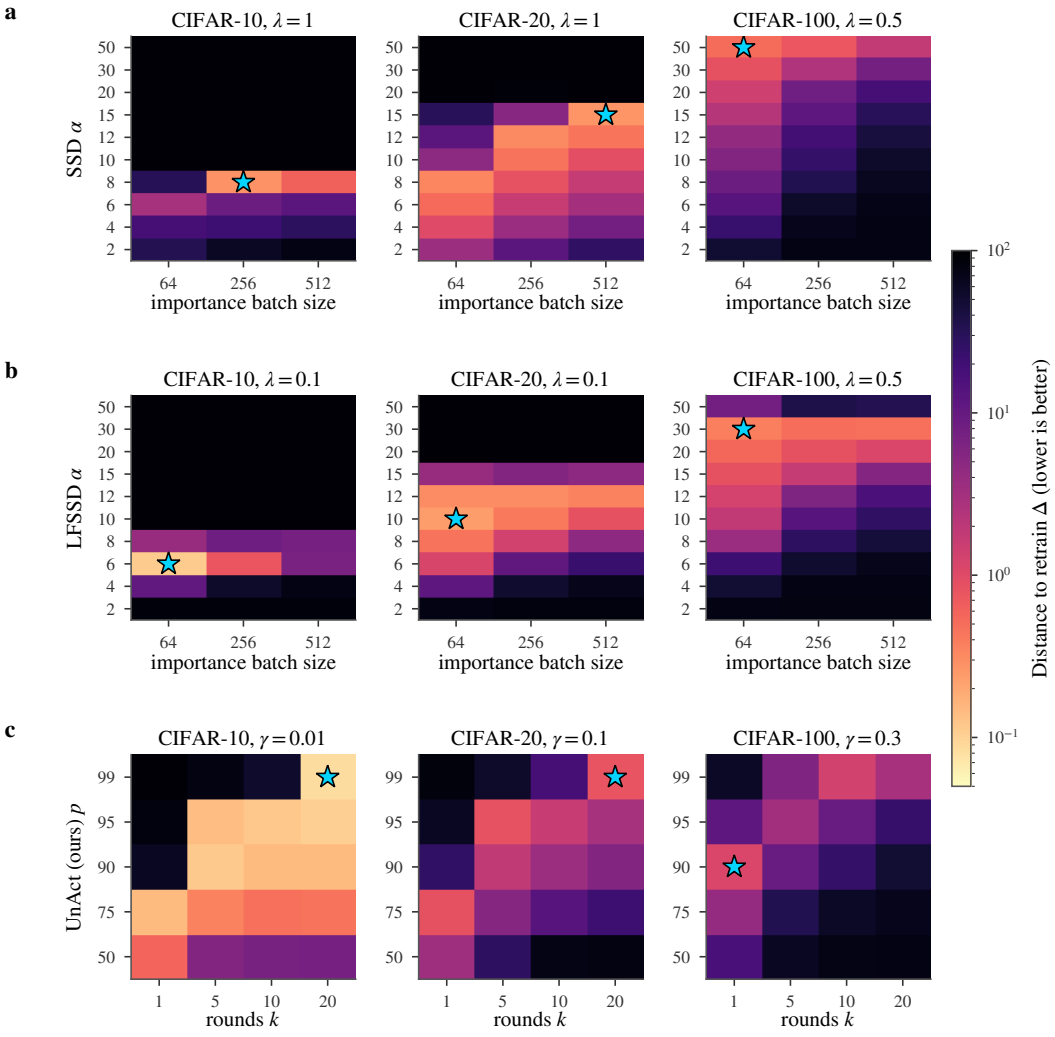

Figure 3: Complete search grids, Δ averaged over the five forget classes, with stars on the selected configurations. (a) SSD and (b) LFSSD at their selected λ; (c) UnAct at its selected γ. UnAct's usable region runs diagonally, as Eq. (1) predicts: raising p and raising k trade off against one another at fixed coverage.

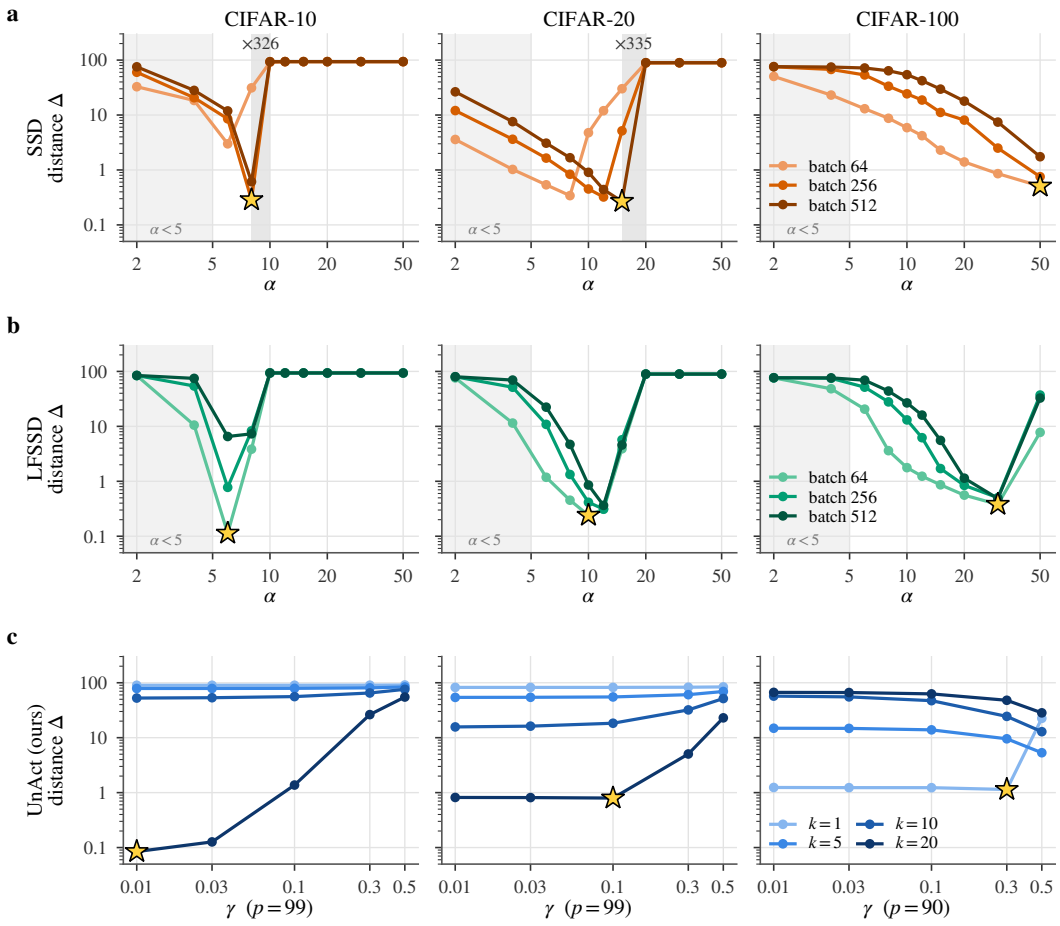

Figure 4: One-dimensional sweeps along each method's own axes, Δ averaged over the five forget classes, log scale; stars mark the selected configurations. (a) SSD and (b) LFSSD against α, one line per importance batch size; the shaded region $\alpha < 5$ lies outside the range the SSD paper reports. (c) UnAct against the attenuation factor γ at its selected p, one line per number of rounds k: flat below a threshold and degrading above it.

E Held-out classes

The SSD paper reports full-class failures on CIFAR-100 *baby* and *lamp* and on the CIFAR-20 superclass *household electrical devices*. We keep these classes out of every five-class mean, although two of the randomly drawn CIFAR-20 class orders of Appendix I include *household electrical devices*, and run each method at the configuration it selected on the five main classes (Table 7). All three methods transfer: each forgets the three classes and stays close to retraining, which indicates that the baselines are not weakened by our selection. At SSD's published default ($\alpha = 10$, $\lambda = 1$) the reported failure reproduces at some importance batch sizes.

F Full-class results

Table 8 adds standard deviations, the Avg. Gap and the fraction of modified parameters to Table 2. UnAct is not sparse: at its selected configurations it modifies 4.80%, 4.95% and 2.18% of the parameters, against 3.63%, 0.60% and 0.23% for SSD and 3.09%, 0.79% and 0.12% for LFSSD.

Table 6: Choosing the operating point without a retrained model. Δ (mean over the five forget classes) of the configuration each rule picks from the method's full grid. For class removal the retrained model's forget accuracy is 0 by construction, so the retrain-free rule keeps the configurations with mean forget accuracy 0 and picks the one with the highest retain accuracy; it uses a labelled retain evaluation set (here the test split) but no retrained model. Transfer: the configuration with the lowest mean Δ on the other two datasets. One configuration: the lowest mean Δ over all three datasets.

Method	Selection rule	CIFAR-10	CIFAR-20	CIFAR-100
SSD	Oracle ($\min \Delta$, needs retraining)	0.29	0.27	0.51
	Retrain-free (A_f=0, $\max A_r$)	0.29	0.27	0.51
	Transfer from the other two datasets	31.23	0.54	32.42
	One configuration for all three	2.99	0.54	12.29
LFSSD	Oracle ($\min \Delta$, needs retraining)	0.11	0.24	0.38
	Retrain-free (A_f=0, $\max A_r$)	0.11	0.24	0.38
	Transfer from the other two datasets	93.72	0.45	20.58
	One configuration for all three	3.84	0.45	3.81
UnAct	Oracle ($\min \Delta$, needs retraining)	0.08	0.80	1.14
	Retrain-free (A_f=0, $\max A_r$)	0.11	0.80	1.14
	Transfer from the other two datasets	1.37	0.82	3.46
	One configuration for all three	0.08	0.82	3.46

Table 7: Held-out classes on which SSD's own paper reports failures. Each method runs at the configuration it selected on the five main forget classes (transfer, the headline), with A_r/A_f and Δ; "oracle" is its best Δ over its whole grid on that class. These classes were chosen because a paper reports SSD failing on them, so they are kept out of every five-class mean. At SSD's published default ($\alpha = 10$, $\lambda = 1$) the failure reproduces at some importance batch sizes: baby, batch 64: A_r 72.8; baby, batch 256: A_r 6.1; baby, batch 512: A_r 2.0; lamp, batch 64: A_r 73.8; lamp, batch 256: A_r 54.9; lamp, batch 512: A_r 34.2; electrical dev., batch 64: A_r 80.3; electrical dev., batch 256: A_r 85.7; electrical dev., batch 512: A_r 85.6.

	Retrain	SSD			LFSSD			UnAct (ours)		
Class	A_r	A_r/A_f	Δ	oracle	A_r/A_f	Δ	oracle	A_r/A_f	Δ	oracle
CIFAR-100 baby	77.79	77.28/0.00	0.51	0.51	77.39/0.00	**0.39**	0.22	76.54/0.00	1.25	0.74
CIFAR-100 lamp	77.37	77.32/0.00	0.05	0.04	77.45/0.00	0.08	0.01	77.33/0.00	**0.04**	0.04
CIFAR-20 electrical dev.	86.18	85.80/0.00	**0.38**	0.29	85.78/0.00	0.40	0.35	85.19/0.20	1.19	0.98

G WHAT DISTANCE TO RETRAINING SHOWS, AND WHAT IT MISSES

Every comparison in Section 4 is in Δ, which is computed from output accuracies. Here we ask what it certifies, with a trivial baseline and three measurements beyond the output (Table 9).

A trivial logit mask matches Δ. A baseline that removes the class logit and changes nothing else reaches $\Delta = 0.09$, 0.25 and 0.33. This is comparable to SSD and LFSSD, and lower than UnAct's on CIFAR-20 and CIFAR-100. It is also unaffected by the forget-set size, since it needs the class index and no images at all. Δ therefore shows that a class is no longer predicted and that retain accuracy is preserved. For none of the methods does it show that the class has been removed, and throughout this paper we use it only in that sense.

UnAct's edit is not simply a classifier edit. Replaying UnAct's per-round channel masks on the classifier columns alone nearly forgets the class on CIFAR-10 (0.6% forget accuracy). On CIFAR-20 and CIFAR-100 it leaves forget accuracy at 20.2% and 41.4%, and the incoming weights alone leave 20.6–51.7%. Only the combination forgets on every dataset.

No method removes the class from the features. A linear probe (Alain & Bengio, 2017) trained on each model's penultimate features recovers the forget class with test AUROC of at least 0.98 after every method. The probe cannot rank the methods, because it recovers the class almost as well from

Table 8: Full version of Table 2: mean ± s.d. over the five forget classes. Avg. Gap follows Fan et al. (2024) over the three terms we record, $|\Delta A_f|$, $|\Delta A_r|$ and $|\Delta\mathrm{MIA}|$ with MIA in points; it omits their training-set retain accuracy. Because every unlearning method drives MIA to about 0 against a retrained model at 7–30, the MIA term dominates Avg. Gap. "Params" is the percentage of parameters modified. LFSSD rows on CIFAR-20 and CIFAR-100 were evaluated on an A10 rather than the L4; the two devices differ by at most one test image on the same checkpoint.

	Method	Configuration	A_r	A_f	MIA	$\Delta\downarrow$	Avg. Gap↓	Params (%)	Eval
CIFAR-10	Original model	---	95.16 ± 0.37	93.28 ± 3.32	90.4 ± 2.2	93.72 ± 3.10	51.25	—	L4
	Retrain	100 epochs	95.60 ± 0.60	0.00 ± 0.00	30.4 ± 10.1	—	—	—	L4
	SSD	a8.0, l1.0, b256	95.44 ± 0.38	0.00 ± 0.00	0.9 ± 1.8	0.29 ± 0.47	9.94	3.63	L4
	LFSSD	a6.0, l0.1, b64	95.66 ± 0.52	0.00 ± 0.00	0.0 ± 0.1	0.11 ± 0.06	10.17	3.09	L4
	UnAct (ours)	p99.0, g0.01, k20	95.59 ± 0.56	0.00 ± 0.00	0.0 ± 0.0	0.08 ± 0.07	10.17	4.80	L4
CIFAR-20	Original model	---	85.15 ± 0.17	88.96 ± 3.19	94.6 ± 1.2	89.50 ± 3.08	55.64	—	L4
	Retrain	100 epochs	85.69 ± 0.38	0.00 ± 0.00	17.1 ± 8.1	—	—	—	L4
	SSD	a15.0, l1.0, b512	85.43 ± 0.25	0.00 ± 0.00	0.2 ± 0.3	0.27 ± 0.22	5.72	0.60	L4
	LFSSD	a10.0, l0.1, b64	85.45 ± 0.28	0.00 ± 0.00	0.2 ± 0.2	0.24 ± 0.19	5.74	0.79	A10
	UnAct (ours)	p99.0, g0.1, k20	84.90 ± 0.22	0.00 ± 0.00	0.0 ± 0.0	0.80 ± 0.38	5.98	4.95	L4
CIFAR-100	Original model	---	77.31 ± 0.10	81.00 ± 10.00	94.4 ± 2.6	81.39 ± 10.04	56.18	—	L4
	Retrain	100 epochs	77.71 ± 0.19	0.00 ± 0.00	7.2 ± 3.8	—	—	—	L4
	SSD	a50.0, l0.5, b64	77.20 ± 0.15	0.00 ± 0.00	0.0 ± 0.0	0.50 ± 0.20	2.57	0.23	L4
	LFSSD	a30.0, l0.5, b64	77.33 ± 0.17	0.00 ± 0.00	0.0 ± 0.1	0.38 ± 0.20	2.51	0.12	A10
	UnAct (ours)	p90.0, g0.3, k1	76.56 ± 0.27	0.00 ± 0.00	0.0 ± 0.0	1.15 ± 0.35	2.78	2.18	L4

Table 9: What Δ measures (ResNet-18, full forget set, mean over the five forget classes, selected configurations). A_f: forget test accuracy. Agr_f: top-1 agreement with the retrained model on forget-class test images, i.e. whether the forgotten images go to the same classes as under retraining. Probe: test AUROC of a class-balanced linear probe for the forget class, trained on the model's own penultimate training features. The logit mask zeroes the classifier row of the class and sets its bias to -10^4. The last two rows replay UnAct's per-round channel masks on only the incoming (In_j: last block's `conv2` filters and BatchNorm affine) or only the outgoing (Out_j: classifier columns) parameters. Evaluated on the A10.

	CIFAR-10				CIFAR-20				CIFAR-100			
Method	A_f	Δ	Agr_f	Probe	A_f	Δ	Agr_f	Probe	A_f	Δ	Agr_f	Probe
Original model	93.3	93.73	5.3	0.992	89.0	89.49	7.2	0.991	81.0	81.39	10.0	0.981
Retrain	0.0	—	100.0	0.977	0.0	—	100.0	0.972	0.0	—	100.0	0.956
Logit mask (trivial)	0.0	0.09	62.4	0.992	0.0	0.25	46.0	0.991	0.0	0.33	45.0	0.981
SSD	0.0	0.29	65.5	0.992	0.0	0.26	48.8	0.990	0.0	0.51	39.6	0.979
LFSSD	0.0	0.11	62.5	0.991	0.0	0.24	47.4	0.990	0.0	0.38	43.2	0.979
UnAct	0.0	0.08	31.4	0.991	0.0	0.79	28.4	0.989	0.0	1.14	27.8	0.985
In_j only	51.7	51.82	25.5	0.991	21.2	21.89	28.8	0.989	20.6	21.49	29.6	0.985
Out_j only	0.6	0.67	31.7	0.992	20.2	20.85	28.5	0.991	41.4	42.11	25.8	0.981

the retrained model (AUROC ≥ 0.96), which never saw it: features learned from the other classes already separate it linearly. Scaling a unit by γ instead of zeroing it also leaves its signal linearly recoverable, which is consistent with the fast relearning of every method (Appendix M).

UnAct sends forgotten images to different classes than retraining does. SSD, LFSSD and the logit mask send 65.5%, 62.5% and 62.4% of the forget images on CIFAR-10 to the class the retrained model predicts for them, against 31.4% for UnAct, and the pattern is the same on CIFAR-20 and CIFAR-100. UnAct's attenuation, with BatchNorm statistics frozen, moves forget images away from the retrained model's predictions. This counts against UnAct, and Δ does not show it.

Details. Table 9 was evaluated on the A10 GPU; recomputed there, the SSD, LFSSD and UnAct rows reproduce the Δ of Table 2 to within 0.01. The logit mask zeroes the classifier row of the forget class and sets its bias to -10^4. The in/out rows replay the channel masks recorded in each round of the full UnAct run on only In_j (the last block's final convolution filters and BatchNorm affine parameters) or only Out_j (the classifier columns), so all three UnAct rows edit exactly the same channels. Agreement is the top-1 agreement with the retrained model on forget-class test images.

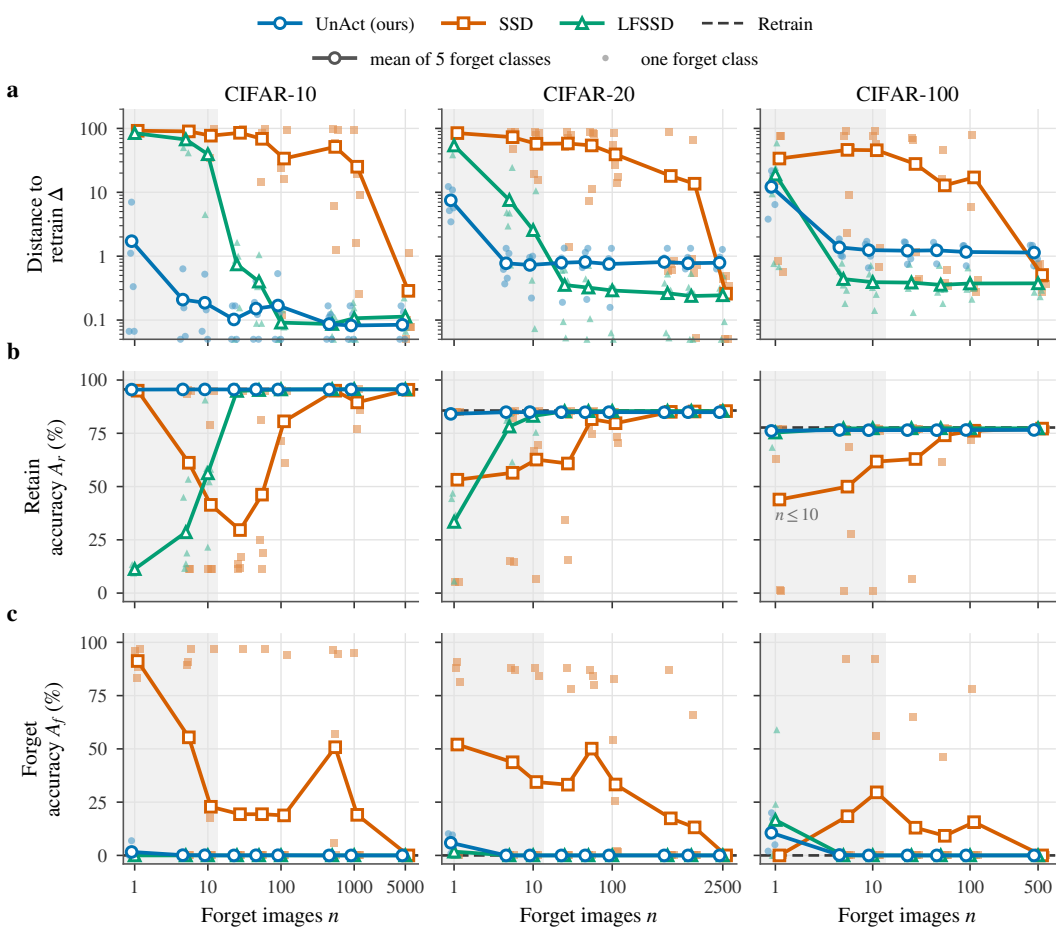

Figure 5: As Figure 2, for the three ResNet-18 datasets, with Δ on a log scale (a) so that differences below one point are visible. Lines: mean over the five forget classes; small faint markers: each class (slightly offset in n for legibility). SSD and LFSSD are bimodal at small n: on some classes retain accuracy collapses and forget accuracy falls with it, while others are barely changed. Values of Δ below 0.05 are drawn at 0.05.

The probe standardises the 512 pooled features with training-set statistics and fits an ℓ_2-regularised logistic regression (forget class against the rest) by full-batch L-BFGS on the model's own training features, with the positive class re-weighted to balance the classes; we report its AUROC on the test features. Because it is refit on each model's features, it measures whether the class is still linearly decodable, not whether the original readout still works. Standardisation undoes a uniform rescaling of a feature, so attenuation can lower the probe's accuracy only through the non-uniform BatchNorm rescaling of Appendix A and the ReLU and shortcut that follow the edited branch. The logit mask and the Out_j-only replay leave the features unchanged, so their probe values equal the original model's.

H FORGET-SET SIZE

Figure 5 shows every run of the forget-set-size sweep and Table 10 every mean. The smallest n at which a method comes within a factor of two of its own best Δ is $n^\star = 25$, 5 and 5 for UnAct, 100, 25 and 5 for LFSSD, and the whole class (5000, 2500 and 500) for SSD. Because $n^\star$ is relative to each method's own best, it should be read together with the absolute values in Table 10.

Table 11 separates two explanations of SSD's failure at small n: that SSD cannot work from a few images, and that its full-class α does not transfer to them. With α re-selected at each n, SSD

Table 10: Distance to the retrained model Δ against the number of forget images n each method is shown; the best method in each cell group is in bold. Every method runs at its single selected configuration at every n. The class being removed and the retain set are fixed, and SSD and LFSSD always receive their importance pass over the full training set, so n measures only how much evidence a method needs to locate the class. The last n of each dataset is the whole class. $n^\star$ is the smallest n at which a method comes within $2\times$ of its own best Δ.

	CIFAR-10			CIFAR-20			CIFAR-100		
n	SSD	LFSSD	UnAct	SSD	LFSSD	UnAct	SSD	LFSSD	UnAct
1	91.85	84.34	**1.72**	84.53	54.05	**7.50**	33.79	18.97	**12.19**
5	89.85	67.07	**0.21**	72.95	7.55	**0.77**	46.14	**0.44**	1.37
10	77.02	39.40	**0.19**	57.46	2.56	**0.73**	45.57	**0.39**	1.24
25	85.35	0.74	**0.10**	58.06	**0.35**	0.79	27.77	**0.39**	1.22
50	68.73	0.40	**0.15**	54.16	**0.32**	0.81	12.85	**0.36**	1.23
100	33.76	**0.09**	0.17	39.23	**0.29**	0.75	17.10	**0.37**	1.16
500	51.39	**0.09**	**0.09**	18.02	**0.26**	0.81	0.51	**0.38**	1.14
1000	25.12	0.11	**0.08**	13.61	**0.24**	0.77	—	—	—
2500	—	—	—	0.26	**0.24**	0.79	—	—	—
5000	0.29	0.11	**0.08**	—	—	—	—	—	—
$n^\star$	5000	100	25	2500	25	5	500	5	5

Table 11: SSD with α re-selected at every forget-set size (ResNet-18, Δ, mean over the five forget classes). UnAct and SSD run at their full-class configurations, as in Figure 2. SSD-n picks, at each n separately, the α in the range reported by the SSD paper ($\alpha \in [5, 50]$; value in parentheses) with the lowest mean Δ, which requires a retrained model at every n; λ and the importance batch stay at their selected values. UnAct receives no per-n tuning. From the existing forget-set-size runs; "all" is the whole class (500 images on CIFAR-100).

	CIFAR-10			CIFAR-20			CIFAR-100		
n	UnAct	SSD	SSD-n	UnAct	SSD	SSD-n	UnAct	SSD	SSD-n
1	1.72	91.85	89.26 (6)	7.50	84.53	83.68 (6)	12.19	33.79	33.79 (50)
5	0.21	89.85	87.47 (6)	0.77	72.95	64.62 (50)	1.37	46.14	46.14 (50)
10	0.19	77.02	77.02 (8)	0.73	57.46	55.91 (20)	1.24	45.57	25.17 (15)
25	0.10	85.35	77.46 (20)	0.79	58.06	57.13 (6)	1.22	27.77	16.04 (30)
50	0.15	68.73	65.45 (10)	0.81	54.16	38.42 (12)	1.23	12.85	7.39 (30)
100	0.17	33.76	31.20 (10)	0.75	39.23	22.48 (10)	1.16	17.10	5.31 (30)
500	0.09	51.39	8.91 (6)	0.81	18.02	1.83 (10)			
1000	0.08	25.12	23.03 (6)	0.77	13.61	0.72 (12)			
all	0.08	0.29	0.29 (8)	0.79	0.26	0.26 (15)	1.14	0.51	0.51 (50)

improves substantially at several sizes, most clearly at $n = 500$ (8.91 against 51.39 on CIFAR-10 and 1.83 against 18.02 on CIFAR-20), but not uniformly: on CIFAR-10 at $n = 1000$ it stays at 23.03 against 25.12. At $n \leq 10$, though, its Δ stays at 25 or above on every dataset, while UnAct's single configuration stays near its full-class value. Within the published range, SSD's small-n failure is therefore not only a calibration artefact of our protocol. The oracle is available only for SSD, because the forget-set-size runs replayed its α grid but not LFSSD's.

I Several classes

Figure 6 applies five one-class requests in sequence, each to the already-edited model, for five class orders per dataset drawn at random before running. Each method runs at its selected single-class configuration, and SSD recomputes its importance on the current model at every request. LFSSD was not run in this experiment.

On CIFAR-10 and CIFAR-20, UnAct keeps every requested class forgotten; SSD does not. In every class order, SSD's forget accuracy on the requested classes is above 50% by the fifth request:

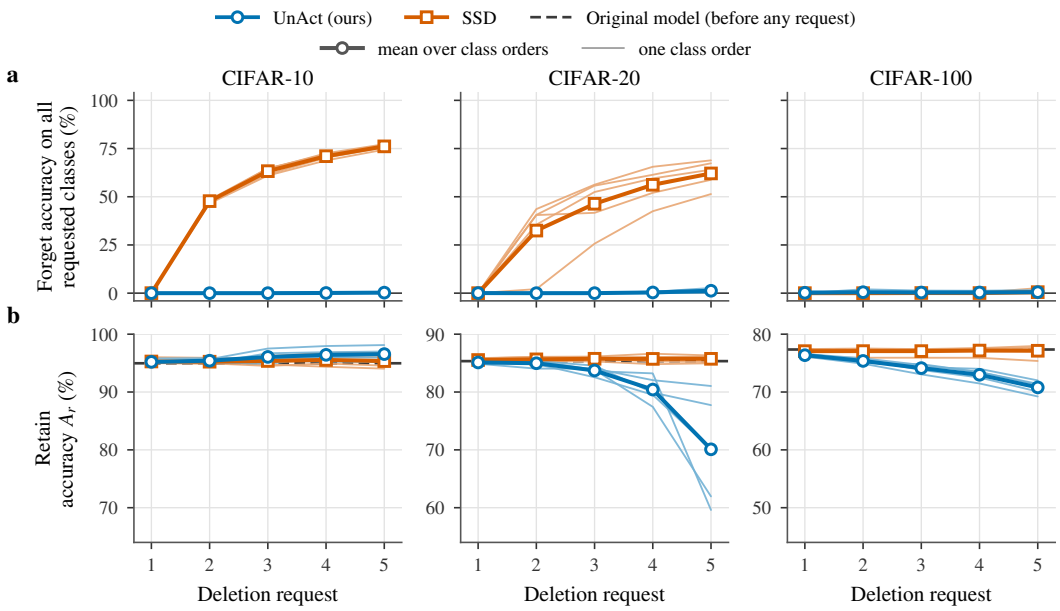

Figure 6: **Sequential deletion requests.** Five one-class requests applied one after another to the same model, each method at its selected single-class configuration with no re-tuning. (a) Forget accuracy on all classes requested so far. (b) Retain accuracy. Thick lines: mean over five class orders; thin lines: individual orders. The dashed line is the original model's test accuracy before any request, not a retrained model: most prefixes have no retrained reference. Per-request values are in Table 12.

47.7% and 32.5% after the second request and 76.1% and 62.1% after the fifth (between 51.4% and 68.9% across CIFAR-20 orders). Across all orders and requests, UnAct's forget accuracy on the requested classes never exceeds 2.7%. On CIFAR-10, UnAct's retain accuracy does not erode. It rises from 95.2% to 96.6% as classes are removed, as a retrained model's does. On CIFAR-20 it is stable for about three requests (85.1% to 83.7%) and then erodes, to 70.1% after the fifth (worst order 59.6%), while SSD's stays at 85.7%.

On CIFAR-100, SSD is the better choice. SSD keeps forgetting (forget accuracy at most 0.5% on average) and keeps its retain accuracy at 77.1%. UnAct also forgets but loses about 1.4 points of retain accuracy per request (76.4% to 70.8%). Two measurements point to the cause. First, a single UnAct request on CIFAR-100 already costs 1.15 points of retain accuracy (Table 2), and these costs add up over requests, while SSD edits roughly ten times fewer parameters (0.23% against 2.18%). Second, the more classes there are, the more of them share each of the 512 channels of the last residual block, so the channels most active for one class are less specific to it. At a fixed percentile, the ratio of forget-set to retain-set activation carried by the selected channels falls from 4.38 on CIFAR-10 to 2.54 on CIFAR-100 (Appendix J). Removing several classes in a single request, at the single-class settings, fails for both methods (Table 13); UnAct is designed for one class per request.

Table 12 gives the per-request values behind Figure 6. For the first class order, retrained models exist for most prefixes, so Table 13 also reports Δ; after the fifth request on CIFAR-10 the retrained model's retain accuracy is 97.92%, which is why UnAct's rising retain accuracy there is expected. Table 13 also removes q classes in a single request at the single-class settings. Both methods fail at this task for $q > 1$, UnAct less badly than SSD (Δ of 84.76, 60.51 and 43.66 against 94.54, 89.97 and 81.29 at $q = 5$). Neither was tuned for it.

Table 12: Sequential deletion requests over five class orders per dataset, drawn at random before running (one class per request, each applied to the already-edited model). Mean ± sd over the class orders of the forget accuracy on *all* classes requested so far (A_f) and the retain accuracy (A_r), both in %. Both methods run at their selected single-class configuration, with no re-tuning; SSD recomputes its importances on the current model at every request. Most prefixes have no retrained model, so the reference is the original model's test accuracy over all classes ("orig."); its accuracy on the remaining classes shifts as classes are removed. The first order is the one of Table 13 (evaluated on an A10); the others were evaluated on an L4. Seed 42.

		SSD		UnAct (ours)	
Dataset	Request	A_f	A_r	A_f	A_r
CIFAR-10 5 orders orig. 94.97	1	0.0 ± 0.0	95.3 ± 0.4	0.0 ± 0.0	95.2 ± 0.3
	2	47.7 ± 0.7	95.2 ± 0.4	0.0 ± 0.0	95.4 ± 0.4
	3	63.3 ± 1.8	95.4 ± 0.7	0.0 ± 0.0	96.1 ± 1.0
	4	71.0 ± 1.5	95.5 ± 0.9	0.1 ± 0.1	96.4 ± 1.0
	5	76.1 ± 1.1	95.3 ± 1.1	0.3 ± 0.3	96.6 ± 1.1
orders: (0,2,3,5,8); (5,6,1,2,0); (8,7,1,5,6); (6,0,3,7,8); (0,4,9,6,7)					
CIFAR-20 5 orders orig. 85.34	1	0.0 ± 0.0	85.5 ± 0.3	0.0 ± 0.1	85.1 ± 0.2
	2	32.5 ± 17.2	85.6 ± 0.4	0.0 ± 0.0	85.0 ± 0.6
	3	46.4 ± 13.0	85.7 ± 0.3	0.0 ± 0.0	83.7 ± 0.7
	4	56.2 ± 9.1	85.7 ± 0.7	0.4 ± 0.4	80.4 ± 2.3
	5	62.1 ± 7.1	85.7 ± 0.6	1.2 ± 1.0	70.1 ± 9.4
orders: (3,4,10,14,19); (5,13,2,11,19); (8,1,19,0,10); (6,12,3,7,1); (10,14,15,2,5)					
CIFAR-100 5 orders orig. 77.35	1	0.0 ± 0.0	77.1 ± 0.6	0.2 ± 0.4	76.4 ± 0.2
	2	0.0 ± 0.0	77.1 ± 0.7	0.5 ± 0.9	75.4 ± 0.4
	3	0.1 ± 0.1	77.1 ± 0.6	0.3 ± 0.6	74.1 ± 0.7
	4	0.1 ± 0.1	77.2 ± 0.7	0.3 ± 0.5	73.0 ± 1.0
	5	0.5 ± 1.2	77.1 ± 1.0	0.5 ± 0.6	70.8 ± 1.1
orders: (3,20,51,69,85); (45,15,90,32,35); (48,97,1,81,90); (86,42,89,92,3); (30,13,49,91,37)					

Table 13: Removing several classes, with each method at its selected single-class configuration (no re-tuning for the task). *Sequential*: one class per request, applied to the already-edited model; the column is the number of requests so far. *Simultaneous*: all q classes in one request. Δ is measured against a model retrained without all q classes; four-class retrained models were not trained, so $q = 4$ is omitted. The last column of each method is A_r/A_f after five classes. Sequential: the first of the five class orders of Table 12, the only one with retrained references. Seed 42.

		SSD					UnAct (ours)				
	Dataset	q=1	2	3	5	A_r/A_f	q=1	2	3	5	A_r/A_f
sequential	CIFAR-10	0.00	46.67	62.21	75.60	96.8/74.4	0.00	0.16	0.13	0.60	98.1/0.4
	CIFAR-20	0.43	43.73	56.37	69.07	85.0/68.9	0.63	0.39	2.84	15.21	70.1/0.2
	CIFAR-100	0.72	0.46	0.76	2.79	77.3/2.6	1.48	4.59	4.17	7.63	71.4/1.6
simultaneous	CIFAR-10	0.00	94.80	93.67	94.54	96.7/93.3	0.00	55.29	72.41	84.76	98.4/84.3
	CIFAR-20	0.43	86.83	88.74	89.97	84.1/89.0	0.63	13.64	31.10	60.51	85.3/60.4
	CIFAR-100	0.72	46.24	77.98	81.29	77.2/81.0	1.48	10.97	21.44	43.66	77.0/43.2

J CLASS SELECTIVITY

For each of the 512 output channels of the last residual block we compute $\rho = (a_f - a_r)/(a_f + a_r)$, where a_f and a_r are the channel's mean post-ReLU activations over the forget-class and retain training images; no weight is modified. Selectivity lies in $[-1, 1]$: $\rho = 1$ means the channel responds to the forget class alone, and $\rho = 0$ means it responds equally to both, so attenuating it removes as much retain signal as forget signal. We estimate a_r on a sample of the retain training set and have not quantified the resulting sampling error.

Selection by activation alone lands on class-specific channels: the selected channels become sharply more selective as the percentile p rises, although the rule never consults the retain set (Figure 7a). At a fixed $p = 75$, selectivity falls as the same 512 channels are shared by more classes (Table 14): the selected channels capture less of the forget set's activation and more of the retain set's, and the ratio of the two falls from 4.38 to 3.19 to 2.54. This is the measurement Appendix I uses. It does not predict the selected percentiles, which are 99, 99 and 90: p and k trade off, so the amount of the layer that is edited is better described by the coverage Γ_k (Appendix N).

Table 14: Class selectivity of the selected channels at a fixed $p = 75$, measured on the 512-channel output of the last residual block. These are forward-pass statistics and no weight is modified. Per channel, $\rho = (a_f - a_r)/(a_f + a_r)$, where a_f and a_r are mean activations on the forget and retain sets. Forget and retain mass are the fractions of each set's total activation that the selected channels carry. Mean $\pm$ s.d. over the five forget classes.

Dataset	Classes	Selectivity ρ	Forget mass (%)	Retain mass (%)	Ratio
CIFAR-10	10	0.574 ± 0.053	83.7 ± 3.5	19.1 ± 0.5	4.38 ± 0.24
CIFAR-20	20	0.414 ± 0.052	74.6 ± 0.5	23.4 ± 0.4	3.19 ± 0.03
CIFAR-100	100	0.298 ± 0.066	64.0 ± 4.6	25.2 ± 0.3	2.54 ± 0.16

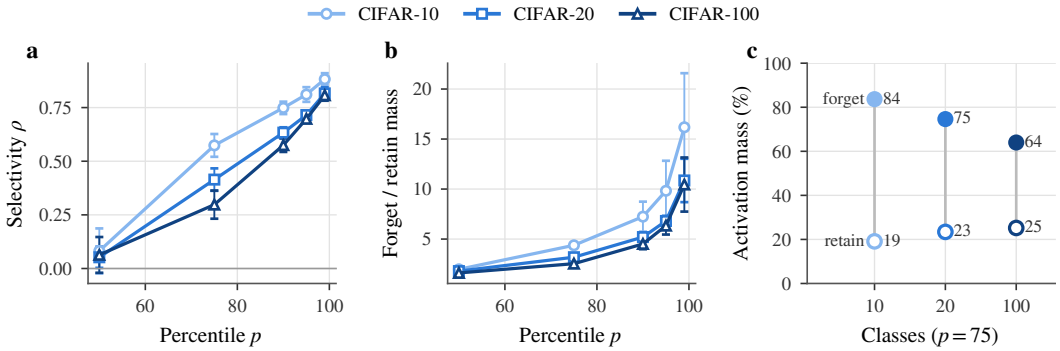

Figure 7: Class selectivity of the selected channels, from forward passes only. Error bars are one standard deviation over the five forget classes. (a) Selectivity rises with the percentile p. (b) The forget-to-retain activation mass ratio of the selected channels. (c) At fixed $p = 75$, as more classes share the same 512 channels, the selected channels carry less forget mass and more retain mass.

K Cost

Figure 8 plots the trade-off and Table 15 lists the timings. Each UnAct setting was timed in the same session as SSD, so each ratio is paired. SSD's time for identical work varies between sessions, and the $k = 1$ session recorded the fastest SSD, so the one-round ratios quoted in Section 4.2 are conservative. At $k = 5$ on every dataset, and at the selected $k = 20$ on CIFAR-10 and CIFAR-20, UnAct is slower than SSD with precomputed importance and faster than SSD from cold.

L Vision transformer details

Table 17 compares UnAct's ViT score and scope variants, each with its configuration chosen on class 3 and then run on all five classes. The reported variant scores the MLP neurons of the last four blocks with the class-token score of Section 3.3. Restricting it to the last two blocks raises Δ to 44.24, and scoring the same neurons by the positive part of their activation at the class token, the closest analogue of the ResNet score, fails ($\Delta = 95.41$).

The class-token score of Section 3.3 is computed in the forward pass, but it reads the outgoing weights and the classifier row, much as a gradient-times-activation score would. On ViT-B/16, UnAct is therefore gradient-free in the sense of needing no backward pass, not in the sense of using activations alone.

M Relearning

A relearning probe fine-tunes each unlearned model on m forget images per step, mixed 4:1 with retain images so that recovery cannot come from collapsing onto the forget class, and tracks forget accuracy (Table 18, Figure 10). Neither method deletes the class in a strong sense: both recover it far faster than a retrained model learns it from the same images. The order between the two is not

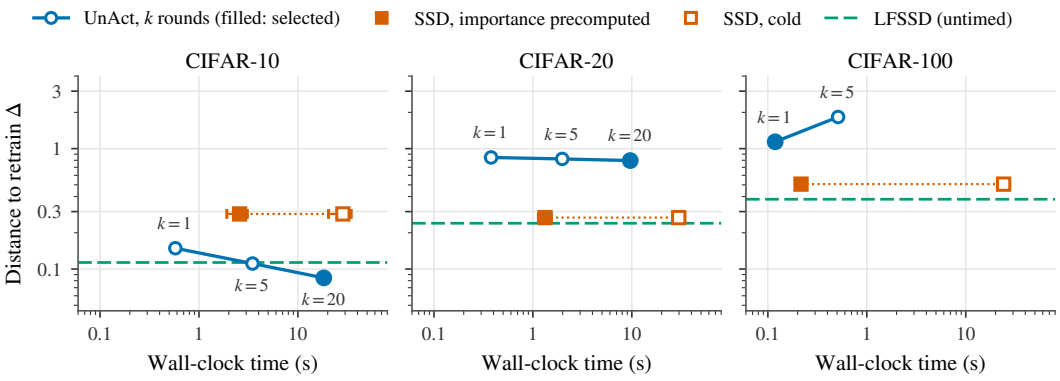

Figure 8: **Cost of one request** (full forget class, class 3, one L4 GPU). Distance to retraining against wall-clock time. UnAct runs the best configuration of its grid for $k = 1$ and $k = 5$ rounds and its selected configuration (filled). SSD runs its selected configuration with its full-training-set importance precomputed (filled) or computed at request time (cold, open); whiskers span the timing sessions. LFSSD's Δ is drawn as a line; it was not timed and has SSD's two-pass structure. Paired per-session timings are in Table 15.

Table 15: Wall-clock cost on one L4, serialised, full forget class (class 3), warm-up repeat discarded, mean of the remaining repeats. For each k, UnAct runs the best configuration of its grid with that number of rounds (†: the selected configuration); its Δ is the five-class grid value. SSD runs its selected configuration in the same session, so each ratio is paired. "Amortised" credits SSD's importance pass over the full training set as precomputed and reusable, which its paper allows; that pass is the listed share of SSD's cold cost. Ratios above 1 mean UnAct is faster; bold marks the settings where it is faster even than amortised SSD.

				UnAct	SSD (s)		full-D	UnAct speed-up	
Dataset	k	config	Δ	(s)	amortised	cold	share (%)	vs. amort.	vs. cold
CIFAR-10	1	`p75, γ0.01`	0.15	0.58	1.91	20.4	91	**3.31**×	35×
	5	`p90, γ0.03`	0.11	3.48	2.72	30.5	91	0.78×	9×
	20†	`p99, γ0.01`	0.08	18.27	3.08	34.5	91	0.17×	2×
CIFAR-20	1	`p75, γ0.1`	0.84	0.38	1.14	26.3	96	**3.01**×	69×
	5	`p95, γ0.1`	0.82	1.98	1.31	29.2	95	0.66×	15×
	20†	`p99, γ0.1`	0.80	9.64	1.53	33.9	95	0.16×	4×
CIFAR-100	1†	`p90, γ0.3`	1.14	0.12	0.22	25.5	99	**1.83**×	215×
	5	`p99, γ0.03`	1.83	0.51	0.22	22.8	99	0.42×	45×

consistent. At $m = 5$, UnAct recovers more slowly than SSD on CIFAR-10 and CIFAR-20 and faster on CIFAR-100, and at $m = 25$ it recovers faster on CIFAR-20 as well. At $m = 1$ the probe itself degrades the retrained model, whose retain accuracy after 50 steps is 40.6%, 35.8% and 4.4%, so we draw no conclusion from $m = 1$.

On ViT-B/16 we use a learning rate of 5×10^{-4}, at which the retrained model's retain accuracy stays high (Figure 11). After five steps at $m = 5$, UnAct's forget accuracy is back to 87.3%, against 33.6% for SSD and 0.0% for the retrained model, averaged over the five forget classes.

N Coverage

If a round selects m of the scope's N units, the fraction of the scope edited after k rounds satisfies the elementary union bound

$$\Gamma_k = \frac{1}{N}\Big|\bigcup_{t=1}^{k} \mathcal{M}_t\Big| \le \min\Big(1, \frac{k\,m}{N}\Big), \tag{1}$$

Table 16: Removing one class from ViT-B/16 on CIFAR-10, mean over the five forget classes. UnAct uses the class-token score (Section 3.3; $p = 85$, $\gamma = 0.1$, $k = 5$, chosen on class 3); SSD uses its best grid setting ($\alpha = 5$, $\lambda = 0.1$, importance batch 128). Worst A_r: lowest retain accuracy over the five classes; closer: classes on which the method has the smaller Δ. Per-class results are in Table 17.

Method	A_r	A_f	$\Delta\downarrow$	worst A_r	closer	Params (%)
Original model	98.33	97.92	98.11	98.2	—	—
Retrain	98.46	0.00	—	98.1	—	—
SSD	78.89	14.16	33.74	9.6	2/5	6.44
UnAct (ours)	91.93	4.96	**11.49**	85.3	3/5	9.75

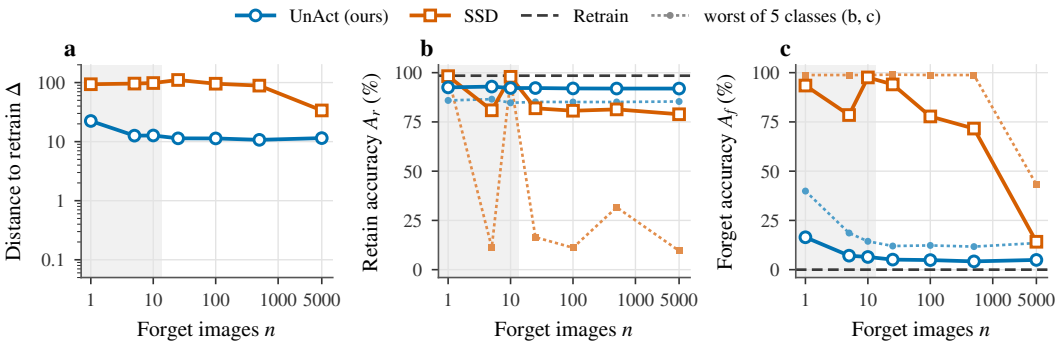

Figure 9: Forget-set size on ViT-B/16 (CIFAR-10), UnAct and SSD at their selected ViT configurations at every n, fp32 on the full test split. (a) Δ, mean over the five forget classes. (b) Retain accuracy: solid, mean; dotted, the lowest of the five classes; dashed, the retrained models. (c) Forget accuracy: solid, mean; dotted, the highest of the five classes; dashed, the retrained models. Below the whole class, SSD leaves its worst class almost entirely unforgotten, while UnAct's mean forget accuracy is below 10% from $n = 5$.

with equality when successive rounds select disjoint sets. The bound itself is trivial. The empirical point is that measured coverage stays close to it for small k, so the percentile p understates the size of the edit by up to a factor of k.

Figure 12 measures the coverage Γ_k of Eq. (1) over UnAct's whole grid. The bound holds in every run, and the usable configurations occupy a band of coverage that moves to smaller values as the number of classes grows.

Table 17: ViT-B/16 score variants. Each UnAct variant's configuration was chosen on class 3 (bf16, 2,000 retain images) and then run on all five classes at fp32 on the full test split; the rows here are that second stage. Columns 7–11 are Δ per forget class. Scoring by the positive part of the activation at the class token fails (row 3), which is the reason for the logit score of Section 3.3. SSD per class (A_r/A_f): 0: 98.1/43.2, 2: 98.2/1.6, 3: 96.1/23.1, 5: 92.5/2.9, 8: 9.6/0.0.

	Units and score	Config	A_r	A_f	Δ	c0	c2	c3	c5	c8	Params (%)
UnAct	MLP, blocks 9–12, class-token logit score (reported)	p85, γ0.1, k5	91.93	4.96	11.49	4.6	13.1	3.0	16.1	20.7	9.753
UnAct	MLP, blocks 11–12, class-token logit score	p75, γ0.1, k5	95.45	41.22	44.24	50.7	33.1	22.5	63.7	51.1	5.558
UnAct	MLP, blocks 9–12, class-token positive activation	p90, γ0.1, k10	94.48	91.42	95.41	102.9	99.6	66.7	106.1	101.8	7.210
SSD	importance-weighted dampening	α5, λ0.1, b128	78.89	14.16	33.74	43.4	1.8	25.8	9.1	88.5	6.437

Table 18: Forget accuracy (%) after 10 relearning steps, each on m forget images mixed 4:1 with retain images (SGD, lr 0.01), mean over the five forget classes. The retrained model is the reference: it never saw the class, so any forget accuracy it gains is learned from the m images alone. "Ret. A_r" is the retrained model's own retain accuracy at step 10; at $m = 1$ the probe degrades it substantially, so small-m curves mix relearning with damage (Figure 10).

	$m = 1$				$m = 5$				$m = 25$			
Dataset	Retrain	SSD	UnAct	Ret. A_r	Retrain	SSD	UnAct	Ret. A_r	Retrain	SSD	UnAct	Ret. A_r
CIFAR-10	0.0	47.2	25.9	68.4	0.0	68.1	36.8	87.5	0.0	87.4	47.5	92.7
CIFAR-20	0.0	28.3	12.0	67.4	0.0	41.3	33.2	79.6	0.0	52.2	59.7	83.5
CIFAR-100	1.0	53.2	54.0	63.8	1.2	40.2	83.6	71.8	6.6	52.2	95.2	75.2

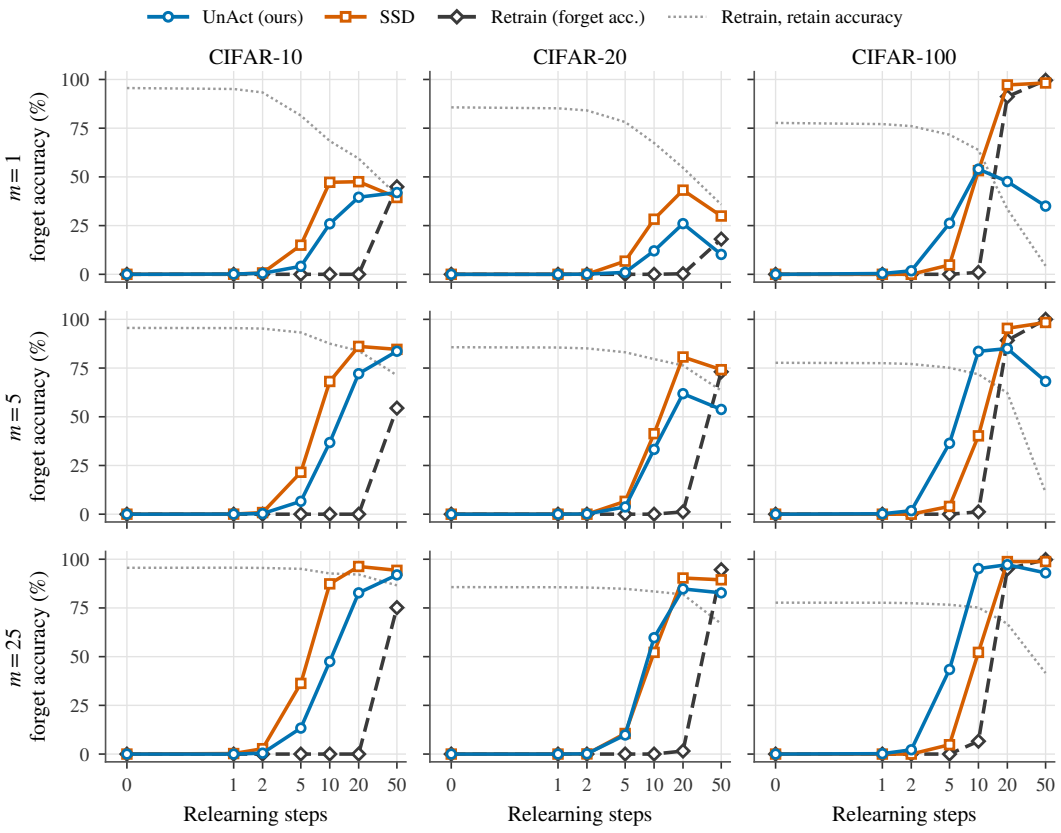

Figure 10: Forget accuracy against gradient steps of relearning on m forget images per step, mixed 4:1 with retain images (SGD, learning rate 0.01), ResNet-18. The dotted line is the retrained model's own retain accuracy: at $m = 1$ the probe degrades every model, so the late steps at $m = 1$ mix relearning with damage. Both unlearned models regain much of the class within 10 to 20 steps, while the retrained model is still at or near zero forget accuracy at step 10.

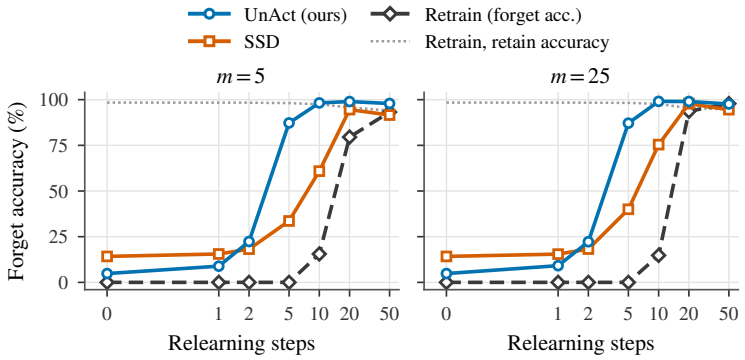

Figure 11: Relearning on ViT-B/16 (CIFAR-10), learning rate 5×10^{-4}, m forget images per step mixed 4:1 with retain images, evaluated in bfloat16, with retain accuracy on 2,000 retain test images and forget accuracy on the full forget test split. Mean over the five forget classes. UnAct's edit is undone within a few steps, faster than SSD's; the retrained model stays at zero forget accuracy for the first five steps.

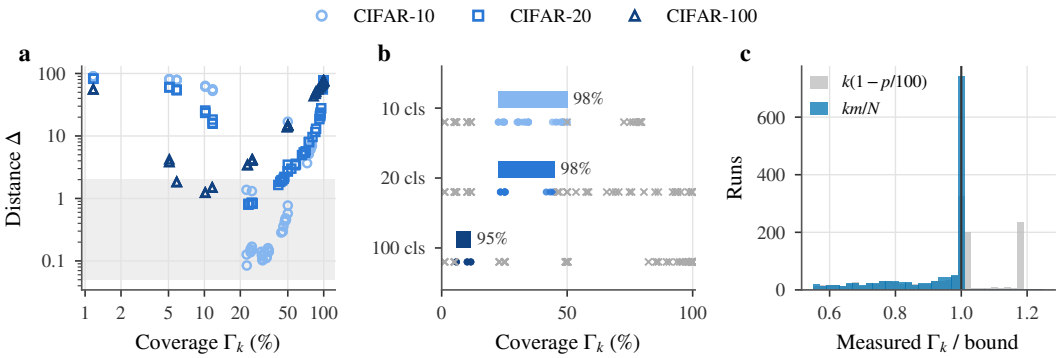

Figure 12: Cumulative coverage Γ_k organises UnAct's hyperparameters. (a) Across the grid at $\gamma \leq 0.1$, Δ is U-shaped in coverage. (b) A coverage window per dataset (bar) against configurations with $\Delta \leq 2$ (dots) and above (crosses); the percentage is the share of configurations the window classifies correctly. The windows were fitted on this grid, so they describe it rather than forecast a held-out one. (c) Measured coverage divided by two bounds: the naive $\min(1, k(1 - p/100))$ is exceeded in 489 of 1500 runs, because selection uses $\geq$ against an interpolated percentile and so keeps slightly more than a $(1 - p/100)$ share of units, while the bound km/N of Eq. (1) is exceeded in 0 runs.