FORGETTING MUST IGNITE: OVERCOMING UNDER-FORGETTING IN MACHINE UNLEARNING WITH PAIR-FREE CAUCHY–SCHWARZ ANGULAR REPULSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Approximate machine unlearning seeks to remove the influence of a forget set at substantially lower cost than full retraining. Yet strong baselines such as fine-tuning often under-forget: they preserve utility while leaving forget-set representations close to their original, memorized state. We address this failure mode with CSD-Unlearning, a feature-level objective based on Cauchy–Schwarz divergence that preserves retain-set representations while selectively repelling forget-set features from the original model. We show that the resulting objective admits a geometric interpretation as a negative log-cosine divergence between kernel mean embeddings. This analysis reveals that the repulsive gradient vanishes for deterministic feature extraction when the current and reference representations coincide. A capped one-dimensional surrogate further exhibits a sharp transition between locally suppressing and amplifying nonzero displacements, providing a mechanistic explanation for the threshold-like behavior observed in several training sweeps. Across CIFAR-10, CIFAR-100, and Tiny ImageNet with ResNet-18 and vision transformers, CSD-Unlearning consistently reduces the gap to retraining when added to fine-tuning. We also characterize its regime of effectiveness: the method is most useful for under-forgetting baselines, provides limited gains once a baseline already over-forgets, and degrades in selectivity at large forget ratios or outside a narrow stability range on vision transformers.

1 INTRODUCTION

Data governance and privacy regulations increasingly require models to remove the influence of data that can no longer be retained or used. Similar deletion needs also arise from poisoned training samples or licensing disputes. Machine unlearning studies how to fulfill such requests by making a trained model behave as if the designated data had never been used for training (Cao & Yang, 2015; Bourtoule et al., 2021; Nguyen et al., 2025). The gold-standard solution is retraining from scratch on the retained data, but its cost can be prohibitive at modern scale. Approximate unlearning therefore seeks to directly edit the trained model so that its behavior approaches that of the retrained reference model (Golatkar et al., 2020; Graves et al., 2021; Kurmanji et al., 2023; Jia et al., 2023; Fan et al., 2024).

Approximate unlearning is evaluated by how closely the edited model matches the retrained reference across utility and privacy metrics (Jia et al., 2023; Thudi et al., 2022). Under this protocol, fine-tuning on the retained data can under-forget: it preserves utility while leaving forget-set behavior close to its original state. Near the original solution, retain-only gradients can be too weak to erase this behavior. CSD does not itself break exact symmetry at θ_0 in the matched deterministic limit. Rather, once the base loss or optimization stochasticity produces a nonzero displacement, the repulsive term can amplify it. This motivates studying when such displacements are suppressed or activated.

Recent work supplies that push in representation space. CoUn (Khalil et al., 2025) uses contrastive learning on retained data to indirectly reorganize forget-set representations, while an earlier line directly contrasts forget embeddings against retained ones (Lee et al., 2025). These methods inherit the machinery of contrastive learning, including positive-pair construction through augmentation

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.

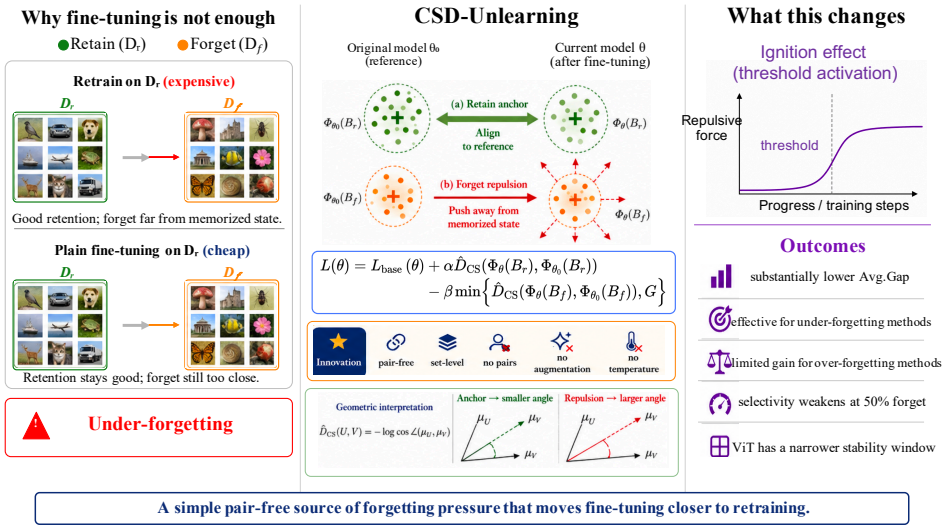


Figure 1: The framework of CSD-Unlearning.

and careful temperature tuning (Oord et al., 2018; Chen et al., 2020). This raises a natural question: *can a plain distribution-level divergence between feature sets, without pairing or augmentation, provide the required repulsive signal, and under what conditions does it fail?*

We answer with the Cauchy–Schwarz divergence (CSD), a classical quantity from information-theoretic learning with a closed-form kernel estimator (Jenssen et al., 2006; Príncipe, 2010). CS divergence can efficiently measure the similarity between two probability distributions based solely on samples without knowing the specific form of the distribution. However, its role in machine unlearning has not been fully explored.

To address this problem, we propose **CSD-Unlearning**. As illustrated in Fig. 1, the method augments fine-tuning with a distribution-level feature objective that anchors retain-set representations to the original model while repelling forget-set representations from their memorized state. The objective admits a simple geometric interpretation: using the RKHS mean-embedding representation (Gretton et al., 2012), its empirical Cauchy–Schwarz divergence becomes the negative log-cosine of the angle between the two mean embeddings. This geometry also exposes a fundamental optimization obstacle. We show that the repulsive gradient vanishes when the current and reference representations coincide, causing the forgetting signal to remain inactive at initialization. A capped one-dimensional surrogate further exhibits a critical repulsion strength: below the threshold, nonzero displacements are locally suppressed, whereas above it they are amplified toward the surrogate's saturation region. This analysis provides a mechanistic model for the threshold-like sensitivity observed empirically, rather than a guarantee for the full network dynamics. These results directly guide the design of CSD-Unlearning by identifying when repulsion can activate and how its strength should be coupled with the optimization scale. The resulting objective provides a source of forgetting pressure while preserving an explicit retain anchor without explicit pair construction. Empirically, this design consistently narrows the gap to retraining in the under-forgetting regime, while also revealing clear limits: its benefit diminishes for already over-forgetting methods, selectivity weakens at large forget ratios, and the stable operating range becomes narrower on vision transformers.

Our contributions are as follows.

- We propose CSD-Unlearning, a distribution-level machine unlearning method that jointly anchors retain-set representations to the original model and repels forget-set representations from their memorized state using Cauchy–Schwarz divergence. The method requires neither sample pairing nor data augmentation and incorporates a stability-oriented training design based on feature normalization, adaptive bandwidth selection, capped repulsion, and batch-norm recalibration (Section 4).

- We develop a theoretical characterization of CSD-Unlearning. We establish the population divergence and its RKHS geometric interpretation, show that the repulsive gradient vanishes when the current and reference representations coincide, derive an intrinsic bound under feature normalization, and identify a sharp transition in a capped one-dimensional surrogate that provides a mechanistic account of the threshold-like optimization behavior observed in several settings (Section 3).
- We systematically evaluate CSD-Unlearning across datasets, architectures, and forgetting regimes under a controlled protocol with method-specific hyperparameter tuning and retrained reference models. The results show that CSD-Unlearning reduces Avg.Gap in the evaluated under-forgetting regimes while exposing clear utility–forgetting trade-offs and architecture-dependent operating limits (Section 5).

2 Related work

Machine unlearning. Exact unlearning restructures training so that deletions are cheap, as in sharded retraining (Bourtoule et al., 2021) and data-deletion mechanisms for specific model classes (Ginart et al., 2019; Guo et al., 2020; Sekhari et al., 2021). Approximate unlearning edits a trained network directly. Representative approaches include retain-set fine-tuning, gradient-based methods such as NegGrad+ (Kurmanji et al., 2023), teacher–student methods such as SCRUB (Kurmanji et al., 2023) and incompetent-teacher unlearning (Chundawat et al., 2023), parameter-space scrubbing (Golatkar et al., 2020), sparsity- and saliency-based methods (Jia et al., 2023; Fan et al., 2024), and direct parameter editing (Graves et al., 2021).Evaluation typically measures the gap to a retrained reference on accuracy metrics and membership inference (Shokri et al., 2017; Song & Mittal, 2021; Jia et al., 2023), a protocol whose auditability has itself been examined (Thudi et al., 2022).

Representation-space unlearning. CoUn (Khalil et al., 2025) is closest to our work. It observes that a retrained model classifies forget data by semantic similarity to retained data and emulates this with a contrastive module trained on retain data only, which also lifts existing baselines when attached to them. Contrastive unlearning (Lee et al., 2025) pushes forget embeddings away from their class clusters directly. Both build on the InfoNCE apparatus (Oord et al., 2018; Chen et al., 2020). Our method instead uses a set-level divergence, avoiding positive-pair construction, augmentation policies, and temperature tuning. This trades contrastive engineering for frozen-reference forward passes and quadratic mini-batch kernel computations.

Cauchy–Schwarz divergence. The CSD measures the angular separation between two densities in $L^2(\mathbb{R}^d)$ and admits a closed-form Parzen estimator (Jenssen et al., 2006; Príncipe, 2010). Through the kernel trick the estimator is a function of mean embeddings in a reproducing kernel Hilbert space, the same objects underlying the maximum mean discrepancy (Gretton et al., 2012). To our knowledge CSD has not previously been used to drive machine unlearning.

3 Theoretical analysis

This section analyzes the CSD estimator and the forget-side repulsive component used by our study. All proofs are short and given in Appendix A. The central result is a one-dimensional surrogate analysis, stated in Theorem 1, that provides a mechanistic explanation for the threshold-like behavior observed in several experiments in Section 5.

Proposition 1 (Divergence properties)**.** *For densities $p, q \in L^2(\mathbb{R}^d)$, $D_{\mathrm{CS}}(p, q) \geq 0$, with equality if and only if $p = q$ almost everywhere.*

Proposition 2 (RKHS geometry of the estimator)**.** *Let $\mathcal{H}$ be the RKHS of k_σ with feature map φ, and let $\mu_U = \frac{1}{n}\sum_i \varphi(u_i)$ and $\mu_V = \frac{1}{m}\sum_j \varphi(v_j)$ be the empirical mean embeddings. Then*

$$\widehat{D}_{\mathrm{CS}}(U, V) \;=\; -\log \frac{\langle \mu_U, \mu_V \rangle_{\mathcal{H}}}{\|\mu_U\|_{\mathcal{H}} \, \|\mu_V\|_{\mathcal{H}}} \;=\; -\log \cos \angle(\mu_U, \mu_V), \tag{1}$$

which is finite and nonnegative, and equals zero if and only if μ_U and μ_V are positively collinear.

Minimizing $\widehat{D}_{\mathrm{CS}}$ therefore aligns the current embedding with the reference in angle, and maximizing it rotates the two apart. The next result identifies the fixed-point structure at initialization.

Proposition 3 (No repulsion at the fixed point)**.** *Assume that the feature extractor is deterministic for a fixed batch and that the current and reference networks use the same normalization and stochastic-layer modes. If $\theta = \theta_0$, then*

$$\nabla_\theta \widehat{D}_{\mathrm{CS}}(\Phi_\theta(B_f), \Phi_{\theta_0}(B_f))\Big|_{\theta=\theta_0} = 0. \tag{2}$$

Under the same matched deterministic conditions, the retain-side CSD anchor is stationary at θ_0 as well. Hence, neither CSD component supplies the initial displacement. The activation must begin from the base loss or from implementation-induced perturbations. Thus, the exact feature coincidence is a stationary point of the repulsive term. Proposition 3 therefore describes the matched deterministic limit rather than every executed training step.

Proposition 4 (Intrinsic bound under normalization)**.** *If all features are ℓ_2-normalized, then $\widehat{D}_{\mathrm{CS}}(U, V) \le 2/\sigma^2$.*

For any fixed $\sigma > 0$, Proposition 4 gives a finite bound. In our normalized runs, the median-heuristic bandwidths lead to order-one converged CSD values. Proposition 4 does not predict their numerical plateau.

Theorem 1 (Sharp ignition threshold)**.** *Let $\lambda > 0$, $\sigma > 0$, $G > 0$, and $\beta \ge 0$. Define*

$$L_\beta(x) = \frac{\lambda}{2}x^2 - \beta \min\left\{\frac{x^2}{2\sigma^2}, G\right\}, \qquad x \ge 0,$$

with $x_G = \sigma\sqrt{2G}$ and $\beta^\star = \lambda\sigma^2$. Then:

(i) If $\beta < \beta^\star$, the unique global minimizer is $x = 0$.

(ii) If $\beta > \beta^\star$, the unique global minimizer is $x = x_G$.

(iii) If $\beta = \beta^\star$, every $x \in [0, x_G]$ is a global minimizer.

Consequently, for $\beta \ne \beta^\star$, there is no stationary point in $(0, x_G)$, and the unique minimizer switches from 0 to x_G as β crosses $\beta^\star$.

Interpretation. Theorem 1 predicts a threshold-like switch in the one-dimensional surrogate: below $\beta^\star$, the zero-displacement solution is favored, whereas above $\beta^\star$, the surrogate favors the boundary solution. This qualitatively mirrors the abrupt responses observed in several hyperparameter sweeps, but is not a quantitative characterization of full-network optimization.

Remark 1 (Scope of Theorem 1)**.** *Theorem 1 assumes fixed σ and λ, and Algorithm 1 uses a per-batch adaptive bandwidth and the explicit cap is typically inactive. Thus, $\beta^\star = \lambda\sigma^2$ characterizes the surrogate qualitatively rather than predicting the numerical threshold of full-network training.*

Remark 2 (Learning-rate coupling)**.** *The learning-rate–β interaction is empirical: learning rate affects the optimization trajectory, but not the surrogate threshold in Theorem 1.*

Remark 3 (Implementation-induced perturbations)**.** *Proposition 3 describes matched deterministic feature evaluation. In practice, dropout and train/eval BatchNorm asymmetry can break exact feature coincidence at θ_0 and provide initial perturbations. These mechanisms may contribute to seed-dependent activation, although we do not isolate either as the unique cause.*

4 METHOD

4.1 SETUP AND EVALUATION PROTOCOL

Let $\mathcal{D} = \mathcal{D}_r \cup \mathcal{D}_f$ be the training set, split into retained data $\mathcal{D}_r$ and forget data $\mathcal{D}_f$. A model f_{θ_0} with feature extractor Φ_{θ_0} (the penultimate representation) has been trained to convergence on $\mathcal{D}$. An unlearning algorithm produces θ_u from θ_0 using $\mathcal{D}_r$ and $\mathcal{D}_f$, and is judged against a reference θ_r retrained from scratch on $\mathcal{D}_r$ alone.

We follow the established protocol (Jia et al., 2023; Khalil et al., 2025) and report retain accuracy (RA), test accuracy (TA), unlearning accuracy (UA), and MIA. Here, MIA is the SVC predicted-nonmember rate on the forget set under the confidence-based attack used by the project pipeline,

rather than attack accuracy or attack AUC. All accuracy-based quantities are reported in percentage points:

$$\mathrm{UA}(f_\theta; D_f) := 100 - \mathrm{Acc}(f_\theta; D_f). \tag{3}$$

Thus, forget-set accuracy and UA are complementary quantities: higher UA means that more forget samples are misclassified (Song & Mittal, 2021; Jia et al., 2023). For every metric m the target is the retrained value m^*, not an extreme. The headline number is

$$\mathrm{Avg.Gap} \;=\; \tfrac{1}{4}\big(|\mathrm{RA} - \mathrm{RA}^*| + |\mathrm{UA} - \mathrm{UA}^*| + |\mathrm{TA} - \mathrm{TA}^*| + |\mathrm{MIA} - \mathrm{MIA}^*|\big), \tag{4}$$

where lower is better. Avg.Gap is computed per seed and then averaged across seeds. Hence, its reported mean need not equal the gap computed from separately averaged metrics. A value of zero denotes an exact match on the four reported scalar metrics, not functional or distributional equivalence to the retrained model.

4.2 Cauchy–Schwarz divergence and its empirical estimator

Definition 1 (Cauchy–Schwarz divergence)**.** *For probability densities $p, q \in L^2(\mathbb{R}^d)$,*

$$D_{\mathrm{CS}}(p, q) \;=\; -\log \frac{\int p(x)q(x)\,dx}{\|p\|_2 \|q\|_2} \in [0, +\infty], \tag{5}$$

Given feature samples $U = \{u_i\}_{i=1}^n$ and $V = \{v_j\}_{j=1}^m$ and a Gaussian kernel $k_\sigma(x, y) = \exp\big(-\|x - y\|^2/(2\sigma^2)\big)$, the standard Parzen plug-in (Jenssen et al., 2006) gives the estimator

$$\widehat{D}_{\mathrm{CS}}(U, V) \;=\; -\log \frac{\frac{1}{nm}\sum_{i,j} k_\sigma(u_i, v_j)}{\sqrt{\frac{1}{n^2}\sum_{i,i'} k_\sigma(u_i, u_{i'}) \;\cdot\; \frac{1}{m^2}\sum_{j,j'} k_\sigma(v_j, v_{j'})}}. \tag{6}$$

Section 3 shows that this estimator is exactly the negative log-cosine of the angle between kernel mean embeddings, which makes its optimization geometry transparent.

4.3 The unlearning objective

Let L_{base} be any base unlearning loss, for instance the cross-entropy of fine-tuning on retain batches. For a retain batch $B_r \subset \mathcal{D}_r$ and a forget batch $B_f \subset \mathcal{D}_f$ we add an anchor and a capped repulsor,

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{base}}(\theta) + \alpha\, \widehat{D}_{\mathrm{CS}}(\Phi_\theta(B_r), \Phi_{\theta_0}(B_r)) - \beta\, \min\Big\{ \widehat{D}_{\mathrm{CS}}(\Phi_\theta(B_f), \Phi_{\theta_0}(B_f)), G \Big\}, \tag{7}$$

where Φ_{θ_0} is the frozen original extractor, α keeps retain features aligned with the original model, β pushes forget features away from their memorized configuration, and the cap G bounds the repulsive incentive. The rationale is the observation behind representation-space unlearning (Khalil et al., 2025). A retrained model keeps retain representations essentially unchanged while forget representations are re-encoded from the retained structure. The anchor enforces the first property directly. The repulsor supplies the active push that fine-tuning lacks, at the level of feature distributions rather than individual pairs.

4.4 Implementation

Alg. 1 summarizes CSD-Unlearning. Starting from the original model, we freeze a reference copy. Each epoch is indexed by the retain loader, with one forget minibatch paired with each retain update. The forget iterator is restarted when exhausted. Current and reference features are extracted, ℓ_2-normalized, and compared using Cauchy–Schwarz divergence with a median-heuristic Gaussian bandwidth. The retain term anchors representations to the original model, while the capped forget term repels memorized features. Their weighted combination augments the base unlearning loss for gradient-based optimization. Before evaluation, batch-normalization statistics are recalibrated using only retain data.

Algorithm 1 CSD-Unlearning

Require: Original model f_{θ_0}; retain set D_r; forget set D_f; base unlearning loss $\mathcal{L}_{\text{base}}$; anchor weight α; repulsion weight β; cap G

Ensure: Unlearned model f_θ

1: Initialize $\theta \leftarrow \theta_0$; freeze the reference model θ_0 and keep it in evaluation mode throughout unlearning

2: **for** each unlearning step **do**

3: Sample retain batch $B_r \subset D_r$ and forget batch $B_f \subset D_f$

4: Extract current and reference features

$$Z_r = \Phi_\theta(B_r), \quad Z_r^0 = \text{stopgrad}(\Phi_{\theta_0}(B_r)),$$

$$Z_f = \Phi_\theta(B_f), \quad Z_f^0 = \text{stopgrad}(\Phi_{\theta_0}(B_f)).$$

5: ℓ_2-normalize all feature vectors in Z_r, Z_r^0, Z_f, Z_f^0

6: Set the Gaussian-kernel bandwidth for each CSD term by the per-batch median heuristic

7: Compute the retain anchor

$$\mathcal{L}_r = \widehat{D}_{\text{CS}}(Z_r, Z_r^0)$$

8: Compute the capped forget repulsion

$$\mathcal{L}_f = \min\{\widehat{D}_{\text{CS}}(Z_f, Z_f^0),\, G\}$$

9: Form the unlearning objective

$$\mathcal{L} = \mathcal{L}_{\text{base}} + \alpha\mathcal{L}_r - \beta\mathcal{L}_f$$

10: Update θ by gradient descent on $\mathcal{L}$

11: **end for**

12: Before evaluation, reset BN running statistics and re-estimate them using D_r only, without gradient updates

13: **return** θ

5 Experiments

5.1 Protocol

Benchmarks. We evaluate random-subset forgetting on CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) and Tiny ImageNet (Le & Yang, 2015), with ResNet-18 (He et al., 2016) and a vision transformer for small datasets (Dosovitskiy et al., 2021), at forget ratios of 10 and 50 percent. All pipelines build on the public CoUn codebase (Khalil et al., 2025).

Original and retrained models. ResNet originals follow the CoUn recipe, 182 epochs of SGD. The same recipe severely underfits the transformer, so ViT originals and retrained references use AdamW (Loshchilov & Hutter, 2019) with learning rate 10^{-3}, weight decay 0.05, 15 warmup epochs, and cosine decay (Loshchilov & Hutter, 2017). Retrained reference values for ResNet follow the CoUn protocol (Khalil et al., 2025). For ViT no published references exist, so we retrain them ourselves on D_r, one run per setting, and use the resulting project-pipeline reference values throughout the corresponding unlearning experiments.

Unlearning. Unless otherwise stated, all methods run 50 epochs with cosine decay to 10^{-4}, batch size 256, and forget batch size 64. The learning rate is tuned per method and per setting over $\{0.01, 0.05, 0.1\}$, and β over a log grid. Unless otherwise stated, headline unlearning results report mean and standard deviation over five seeds. The LiRA-style audit in Table 3 uses the designated split described in Appendix B. Hyperparameter sweeps are single-seed and are marked as such. Runs that diverge are reported, never silently dropped. No final five-seed configuration diverged.

5.2 Main results

Table 1 summarizes the six headline settings, all of which show a lower mean Avg.Gap with CSD. CIFAR-10/ViT is reported separately as an additional stress test in Appendix C.2–C.3. The gains are

Table 1: Main results. Average gap to the retrained reference (lower is better), mean ± std over five seeds, 50 unlearning epochs. Results use the configurations selected for five-seed evaluation.

Setting	FT (plain)		FT + CSD (ours)		Reduction
	config	Avg.Gap	config	Avg.Gap	
CIFAR-10, RN18, 10%	lr 0.1	0.61 ± 0.12	lr 0.1, β=0.15	$\mathbf{0.39 \pm 0.15}$	36%
CIFAR-100, RN18, 10%	lr 0.1	3.30 ± 0.18	lr 0.05, β=3	$\mathbf{1.48 \pm 0.21}$	55%
CIFAR-100, RN18, 50%	lr 0.1	6.93 ± 0.26	lr 0.1, β=1	$\mathbf{5.41 \pm 0.18}$	22%
Tiny ImageNet, RN18, 10%	lr 0.05	3.57 ± 0.20	lr 0.05, β=3	$\mathbf{1.78 \pm 0.25}$	50%
Tiny ImageNet, ViT, 10%	lr 0.05	6.47 ± 0.31	lr 0.05, β=3	$\mathbf{5.01 \pm 1.90}$	23%
CIFAR-100, ViT, 10%	lr 0.05	24.52 ± 0.14	lr 0.05, β=3	$\mathbf{16.92 \pm 2.39}$	31%

Table 2: Per-metric results on two headline benchmarks over five seeds. UA and MIA measure forgetting, with all metrics evaluated against the retrained reference. CSD-Unlearning moves the under-forgotten metrics closer to that reference.

Setting	Method	RA	UA	TA	MIA	Avg.Gap
CIFAR-100	Retrain (reference)	99.98	24.26	75.56	48.44	0
RN18, 10%	FT (plain)	99.98	16.31±0.29	76.52±0.34	44.16±0.50	3.30±0.18
	FT + CSD	99.83±0.05	**24.68±0.94**	72.24±0.44	**49.06±1.89**	**1.48±0.21**
Tiny ImageNet	Retrain (reference)	99.98	36.16	63.82	63.73	0
RN18, 10%	FT (plain)	99.98	28.09±0.31	65.10±0.47	58.80±0.75	3.57±0.20
	FT + CSD	99.89±0.02	**36.07±1.01**	61.68±0.41	**59.59±0.62**	**1.78±0.25**

largest exactly where fine-tuning under-forgets the most. On CIFAR-100 with ResNet-18 the gap drops by 55 percent, and on Tiny ImageNet by 50 percent. Table 2 shows where the gains come from on the two headline benchmarks. Plain fine-tuning keeps utility but stalls on the forgetting metrics. CSD primarily moves the under-forgotten UA and MIA metrics toward their retrained targets, with an accompanying test-accuracy trade-off. On CIFAR-100 the unlearning accuracy rises from 16.31 to 24.68 against a reference of 24.26, and the MIA score rises from 44.16 to 49.06 against 48.44. On Tiny ImageNet the unlearning accuracy lands at 36.07 against a reference of 36.16, a gap of 0.09 points. The improvement in forgetting comes with a utility trade-off: test accuracy decreases by approximately 4.3 percentage points on CIFAR-100 and 3.4 points on Tiny ImageNet, which is explicitly penalized by Avg.Gap.

Matched learning rate. At the shared learning rate 0.1 on CIFAR-100, plain fine-tuning reaches 3.30 ± 0.18 and the CSD plug-in with $\beta = 1$ reaches 2.39 ± 0.38, a 28 percent reduction under an identical optimizer. Tuning both knobs jointly then yields the 1.48 of Table 1. The gain is therefore not an artifact of learning-rate choice.

Stronger membership inference. We further evaluate a LiRA-style per-sample attack (Carlini et al., 2022) with the RULI logit-scaled confidence statistic (Naderloui et al., 2025). Under the same attack pipeline, Retrain, FT, and CSD-Unlearning achieve AUCs of 0.504, 0.571, and 0.499, respectively; CSD is therefore much closer to Retrain (0.005 vs. 0.067 AUC gap). Their TPRs at $1\%/0.1\%$ FPR are $0.8/0.0\%$, $2.5/0.9\%$, and $0.0/0.0\%$, respectively. This supports closer matching to retraining under this attack, not a formal privacy guarantee.

Contextual Comparison with CoUn. Table 4 compares CSD-Unlearning with an independent five-run reproduction of CoUn and two recent strong unlearning baselines, CT and MSG, on ResNet-18 at a 10% forgetting ratio across CIFAR-10, CIFAR-100, and Tiny ImageNet. CoUn (Khalil et al., 2025) is reproduced using the public implementation and its archived evaluation configuration, whereas CT (Cadet et al., 2025), MSG (Triantafillou et al., 2024), and CSD-Unlearning are evaluated with the audited pipeline described above. We therefore emphasize the directly observed RA, UA, TA, and MIA values rather than use Avg.Gap to impose a strict ranking across evaluation pipelines.

Table 3: Stronger per-sample membership inference on CIFAR-100/ResNet-18 with 10% forgetting. Retrain, FT, and CSD-Unlearning are evaluated on the same 1,000 forgotten members and 1,000 test nonmembers using the same 64-shadow likelihood-ratio attack. TPR@ α is the maximum TPR subject to $\mathrm{FPR} \leq \alpha$. The evaluation target is the retrained reference rather than monotone minimization of these attack statistics.

Method	Attack AUC	TPR@1% FPR	TPR@0.1% FPR
Retrain	0.504	0.8%	0.0%
FT	0.571	2.5%	0.9%
CSD-Unlearning	0.499	0.0%	0.0%

Table 4: Comparison with CoUn, CT, and MSG on ResNet-18 at 10% forgetting. Values are mean $\pm$ standard deviation over five runs. CoUn is independently reproduced using its archived evaluation configuration, whereas CT (Cadet et al., 2025), MSG, and CSD-Unlearning use the audited pipeline of this paper.

Setting	Method	RA	UA	TA	MIA
CIFAR-10 RN18, 10%	CoUn reproduction	100.00±0.00	4.54±0.38	95.20±0.09	16.18±0.49
	CT	100.00±0.00	4.68±0.25	94.39±0.19	11.16±0.33
	MSG	98.54±0.18	2.76±0.21	92.80±0.16	8.52±0.96
	CSD-Unlearning	99.99±0.00	3.88±0.20	94.53±0.20	10.74±0.58
CIFAR-100 RN18, 10%	CoUn reproduction	99.98±0.00	19.86±0.62	77.88±0.16	52.91±0.77
	CT	99.97±0.00	21.75±0.55	76.55±0.44	44.88±0.29
	MSG	92.87±0.16	17.21±0.50	71.65±0.47	25.95±1.11
	CSD-Unlearning	99.83±0.05	24.68±0.94	72.24±0.44	49.06±1.89
Tiny ImageNet RN18, 10%	CoUn reproduction	99.98±0.00	40.09±0.43	60.27±0.32	67.67±0.39
	CT	99.98±0.00	28.63±0.27	64.85±0.06	57.07±0.51
	MSG	97.91±0.00	13.88±0.18	63.70±0.33	25.93±0.67
	CSD-Unlearning	99.89±0.02	36.07±1.01	61.68±0.41	59.59±0.62

Additional shared-protocol comparisons with SCRUB (Kurmanji et al., 2023) and SalUn (Fan et al., 2024) are reported in Appendix C.9.

Component Ablation. A controlled component ablation in Appendix C.5 shows that the retain anchor alone remains close to plain fine-tuning, whereas repulsion supplies the missing forgetting pressure but produces an imbalanced metric profile. Combining both components gives the approximation to retraining.

The methods exhibit markedly different metric profiles. On CIFAR-10, CT most closely matches the retrained reference across the four reported metrics, while CSD-Unlearning remains competitive. On CIFAR-100, CSD-Unlearning brings both UA and MIA substantially closer to their retrained targets, whereas CoUn and CT retain higher test accuracy. On Tiny ImageNet, CSD-Unlearning nearly matches the retrained UA while maintaining competitive test accuracy; CT preserves stronger test accuracy but under-forgets, and MSG deviates substantially from the retrained targets on several metrics. Overall, these results suggest that CSD-Unlearning provides a balanced retrain-matching trade-off rather than uniformly dominating every individual metric.

5.3 The stability window

Figure 2 shows the β sweeps. Several empirical patterns are qualitatively consistent with the surrogate analysis of Section 3.

Threshold-like response. Several settings exhibit a sharp transition as β increases, rather than a uniformly smooth response. On CIFAR-100 at learning rate 0.05, $\beta \leq 1$ leaves every metric at the plain value, and $\beta = 3$ ignites and lands on the reference. At 50 percent forgetting, $\beta = 1$ moves little, $\beta = 3$ diverges, and $\beta = 6$ is unstable across repeated runs, diverging in one run and not in another. In this particular sweep, we do not observe a stable intermediate regime, qualitatively matching the threshold behavior of the surrogate in Theorem 1.

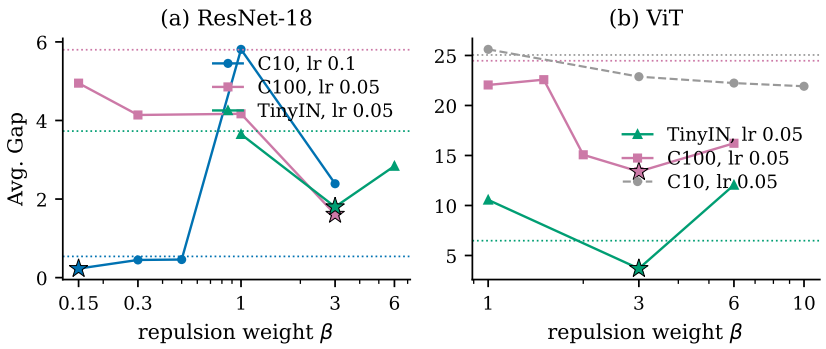


Figure 2: Single-seed β sweeps at 50 epochs. Dotted lines denote plain baselines and stars denote five-seed configurations. ResNet-18 has a broader stable window, with $\beta = 3$ transferring across harder settings, while ViT shows a narrower, more irregular response.

Saturation. The converged forget-side divergence is setting dependent. For the two reported ResNet headline diagnostics it concentrates near 0.69–0.72, whereas the corresponding ViT diagnostics settle at lower values. We therefore interpret saturation qualitatively rather than as a universal plateau value. Proposition 4 guarantees that the normalized CSD is bounded, but does not predict the precise value of this empirical plateau.

Learning-rate coupling. The best β shrinks as the learning rate grows, from $\beta = 3$ at rate 0.05 to $\beta = 1$ at rate 0.1 on CIFAR-100, matching Remark 2. A practical recipe follows. Sweep β on a log grid, expect a threshold rather than a smooth optimum, and pair large learning rates with small β.

Additional single-seed sensitivity experiments on the fixed anchor weight α and forget-side cap G are reported in Appendix C.3. The default $\alpha = 1$ gives the best balance among the tested values, while performance is unchanged for $G \in \{2, 5, 10\}$.

5.4 The tested ViT exhibits a narrower stability window

For the reported ViT configurations, the Avg.Gap is reduced by 23 percent on Tiny ImageNet and 31 percent on CIFAR-100. On CIFAR-100, the unlearning accuracy increases from 1.31 ± 0.09 to 37.04 ± 9.41. Two symptoms nevertheless distinguish the transformer from the CNN.

First, seed variance is an order of magnitude larger. The unlearning accuracy of the ignited runs fluctuates with a standard deviation of 9.41 on CIFAR-100 and 6.78 on Tiny ImageNet, against roughly 1 on ResNet. This observation is compatible with the stochastic-perturbation mechanism discussed in Remark 3, since the ViT trains with dropout and the ignition threshold becomes stochastic. Second, the ViT sweeps are more irregular (Figure 2b), with MIA substantially overshooting its retrained target on CIFAR-100 (99.62% vs. 56.86%), indicating poor retrain matching.

6 Conclusion

We have shown that Cauchy–Schwarz divergence provides a simple pair-free source of feature-level forgetting pressure that can substantially improve tuned fine-tuning in the evaluated under-forgetting regimes. We further identified a zero-gradient fixed point at initialization and used a capped one-dimensional surrogate to characterize threshold-like sensitivity to the repulsion strength. Just as importantly, we map where the approach helps and where it does not, across base methods, forget ratios, and architectures. The results support CSD as a simple, pair-free source of forgetting pressure while showing that its benefit depends on the residual failure mode of machine unlearning.

AI USE STATEMENT

The research design and conceptual development of this work are led by the human authors. Generative AI tools are used to assist with language polishing and programming. We have also used AI-based simulated peer-review tools to help identify aspects of the work that could be further improved.

ETHICS STATEMENT

Our research primarily focuses on data compliance and data privacy. All experiments use publicly available academic datasets, including CIFAR-10, CIFAR-100, and Tiny ImageNet.

REPRODUCIBILITY STATEMENT

Key implementation details, training recipes, reference values, and batch-norm recalibration procedures are specified in Section 5.1 and Appendices B to C. Headline results report mean and standard deviation over five unlearning seeds, while hyperparameter sweeps are explicitly single-seed and diverged runs are retained. Hyperparameters are selected using the validation split only. The held-out test set is used only for reporting the final five-seed evaluation.

REFERENCES

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, 2021.

Xavier F Cadet, Anastasia Borovykh, Mohammad Malekzadeh, Sara Ahmadi-Abhari, and Hamed Haddadi. Deep unlearn: Benchmarking machine unlearning for image classification. In *2025 IEEE 10th European Symposium on Security and Privacy (EuroS&P)*, pp. 939–962, 2025.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pp. 463–480, 2015.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, pp. 1897–1914, 2022.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607, 2020.

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

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.

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 (ICLR)*, 2024.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. In *NeurIPS*, 2019.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9301–9309, 2020.

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.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13:723–773, 2012.

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)*, pp. 3832–3842, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Robert Jenssen, Jose C Principe, Deniz Erdogmus, and Torbjørn Eltoft. The cauchy–schwarz divergence and parzen windowing: Connections to graph theory and mercer kernels. *Journal of the Franklin Institute*, 343(6):614–629, 2006.

Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model sparsity can simplify machine unlearning. In *NeurIPS*, pp. 51584–51605, 2023.

Yasser Khalil, Mehdi Setayesh, and Hongliang Li. Coun: Empowering machine unlearning via contrastive learning. In *NeurIPS*, pp. 131254–131283, 2025.

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 *NeurIPS*, pp. 1957–1987, 2023.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. Technical report, Stanford University, 2015.

Hong-Kyu Lee, Qiuchen Zhang, Carl Yang, Jian Lou, and Li Xiong. Contrastive unlearning: A contrastive approach to machine unlearning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 7464–7472, 2025.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Nima Naderloui, Shenao Yan, Binghui Wang, Jie Fu, Wendy Hui Wang, Weiran Liu, and Yuan Hong. Rectifying privacy and efficacy measurements in machine unlearning: A new inference attack perspective. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 5545–5564, 2025.

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.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

José C. Príncipe. *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. Springer, 2010.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *NeurIPS*, pp. 18075–18086, 2021.

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, 2017.

Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX security symposium (USENIX security 21)*, pp. 2615–2632, 2021.

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX security symposium (USENIX Security 22)*, pp. 4007–4022, 2022.

Eleni Triantafillou, Peter Kairouz, Fabian Pedregosa, Jamie Hayes, Meghdad Kurmanji, Kairan Zhao, Vincent Dumoulin, Julio Jacques Junior, Ioannis Mitliagkas, Jun Wan, et al. Are we making progress in unlearning? findings from the first neurips unlearning competition. *arXiv preprint arXiv:2406.09073*, 2024.

A Proofs

A.1 Proof of Proposition 1

Proof.

$$0 \le \frac{\int pq}{\|p\|_2 \|q\|_2} \le 1.$$

If $\int pq = 0$, then $D_{\mathrm{CS}}(p, q) = +\infty$. Otherwise the ratio lies in $(0, 1]$, implying $D_{\mathrm{CS}}(p, q) \ge 0$. Equality holds iff $p = cq$ almost everywhere; normalization gives $c = 1$. □

A.2 Proof of Proposition 2

Proof. By the reproducing property, $\langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}} = k_\sigma(x, y)$. Expanding the empirical mean embeddings,

$$\langle \mu_U, \mu_V \rangle = \tfrac{1}{nm} \textstyle\sum_{i,j} k_\sigma(u_i, v_j),$$
$$\|\mu_U\|^2 = \tfrac{1}{n^2} \textstyle\sum_{i,i'} k_\sigma(u_i, u_{i'}), \qquad \|\mu_V\|^2 = \tfrac{1}{m^2} \textstyle\sum_{j,j'} k_\sigma(v_j, v_{j'}),$$

which are exactly the three sums in Eq. (6), so $\widehat{D}_{\mathrm{CS}} = -\log\big(\langle \mu_U, \mu_V \rangle / (\|\mu_U\| \|\mu_V\|)\big)$. The Gaussian kernel is strictly positive, so $\langle \mu_U, \mu_V \rangle > 0$, and by Cauchy–Schwarz in $\mathcal{H}$ the cosine lies in $(0, 1]$. Hence $\widehat{D}_{\mathrm{CS}} \in [0, \infty)$ is finite, and it vanishes exactly at cosine one, that is when μ_U and μ_V are positively collinear. □

A.3 Proof of Proposition 3

Proof. Fix the batch B_f and write $F(\theta) = \widehat{D}_{\mathrm{CS}}(\Phi_\theta(B_f), \Phi_{\theta_0}(B_f))$. At $\theta = \theta_0$ the two sample sets coincide element-wise, so $\mu_U = \mu_V$ and $F(\theta_0) = 0$ by Proposition 2. The same proposition gives $F(\theta) \ge 0$ for all θ, so θ_0 is a global minimizer of F. The map F is differentiable at θ_0, being a composition of the network (differentiable in θ), smooth kernel sums, a quotient with strictly positive denominator and strictly positive numerator in a neighborhood of θ_0, and the logarithm. A differentiable function is stationary at an interior global minimum, so $\nabla_\theta F(\theta_0) = 0$. □

A.4 Proof of Proposition 4

Proof. On the unit sphere, $\|u - v\|^2 = 2 - 2\langle u, v \rangle \le 4$ for any pair, so every cross-kernel value satisfies $k_\sigma(u_i, v_j) \ge e^{-2/\sigma^2}$ and hence $\langle \mu_U, \mu_V \rangle \ge e^{-2/\sigma^2}$. Every kernel value is at most one, so $\|\mu_U\| \le 1$ and $\|\mu_V\| \le 1$. The cosine is therefore at least e^{-2/σ^2} and $\widehat{D}_{\mathrm{CS}} \le 2/\sigma^2$. □

A.5 Proof of Theorem 1

To obtain an analytically unconstrained scalar surrogate, we temporarily drop the unit-sphere constraint and set $u = xe$, $v = 0$.

Proof. Under $\lambda > 0$, $\sigma > 0$, $G > 0$, and $\beta \geq 0$, we have $x_G = \sigma\sqrt{2G} > 0$. First note that the uncapped single-sample divergence along a line is exact. With $n = m = 1$, $u = xe$ for a unit vector e and $v = 0$, Eq. (6) reads

$$\widehat{D}_{\mathrm{CS}} = -\log \frac{k_\sigma(u,v)}{\sqrt{k_\sigma(u,u)\,k_\sigma(v,v)}} = -\log e^{-x^2/(2\sigma^2)} = \frac{x^2}{2\sigma^2},$$

so $g(x) = \min\{x^2/(2\sigma^2), G\}$ is the clamped objective with cap G, reached at $x_G = \sigma\sqrt{2G}$.

On $[0, x_G]$ the total objective is $L_\beta(x) = \frac{1}{2}(\lambda - \beta/\sigma^2)\,x^2$, and on $[x_G, \infty)$ it is $L_\beta(x) = \frac{\lambda}{2}x^2 - \beta G$ with derivative $\lambda x > 0$, so L_β is always strictly increasing beyond the cap.

(i) If $\beta < \lambda\sigma^2$, the quadratic coefficient is positive, so L_β is strictly increasing on $(0, x_G]$ and, by the previous paragraph, on $[x_G, \infty)$ as well. Continuity at x_G makes L_β strictly increasing on $(0, \infty)$ with unique global minimizer $x = 0$.

(ii) If $\beta > \lambda\sigma^2$, the coefficient is negative, so L_β strictly decreases on $[0, x_G]$ from 0 to $L_\beta(x_G) = \frac{\lambda}{2} \cdot 2G\sigma^2 - \beta G = G(\lambda\sigma^2 - \beta) < 0$, then strictly increases. The unique global minimizer is x_G.

(iii) For $x \in (0, x_G)$,

$$L'_\beta(x) = \left(\lambda - \frac{\beta}{\sigma^2}\right) x.$$

If $\beta \neq \beta^\star = \lambda\sigma^2$, this derivative cannot vanish for any $x \in (0, x_G)$. Hence there is no interior stationary point.

At the critical value $\beta = \beta^\star$, however,

$$L_{\beta^\star}(x) = \frac{1}{2}\left(\lambda - \frac{\beta^\star}{\sigma^2}\right) x^2 = 0, \qquad x \in [0, x_G].$$

Thus every point in $[0, x_G]$ is a global minimizer at the critical value. For $\beta < \beta^\star$ the unique minimizer is 0, whereas for $\beta > \beta^\star$ it is x_G. □

B Implementation details

Feature extraction. Features are taken at the penultimate layer, the pooled 512-dimensional output for ResNet-18 and the pre-head token representation for ViT, and are ℓ_2-normalized before the kernel computation. The reference features $\Phi_{\theta_0}(B)$ are computed with the frozen original model on the same batch, with gradients detached.

Feature normalization. All features are ℓ_2-normalized before entering Eq. (6). On the unit sphere the divergence is intrinsically bounded (Proposition 4), which prevents unbounded repulsive gradients.

Kernel and bandwidth. For each CSD term, let S denote the concatenated current and reference feature set. We use

$$\sigma = \max\left\{10^{-3}, \mathrm{median}_{i<j,\,\|s_i - s_j\|_2 > 0}\,\|s_i - s_j\|_2\right\}.$$

The bandwidth is computed from detached features and is treated as constant during backpropagation. Kernel sums are evaluated in double precision.

Cap and weights. The forget term is clamped at $G = 5$. We fix $\alpha = 1$ throughout and tune β only. The forget batch size is 64 against a retain batch size of 256.

Batch-norm recalibration. Before evaluation, BN running statistics are reset and recomputed from 50 retain-only batches using the evaluation transform and cumulative averaging, without gradient updates. The original momentum values are then restored, and evaluation is performed with the model in evaluation mode.

Optimization. Unlearning uses SGD with momentum 0.9 and weight decay 5×10^{-4}, cosine decay to 10^{-4} over 50 epochs, batch size 256. Original and retrained ViT models use AdamW with learning rate 10^{-3}, weight decay 0.05, 15 warmup epochs, and cosine decay over 182 epochs.

Original and retrained ResNet models follow the CoUn recipe with SGD at learning rate 0.1 over 182 epochs.

MIA. The confidence-based membership inference attack follows the SVC protocol of (Jia et al., 2023). The attack model separates members from nonmembers using prediction confidence, and the reported MIA value is the fraction of forget-set examples predicted as nonmembers. Its target is the corresponding retrained value.

LiRA-style attack protocol. For the stronger per-sample attack in Table 3, we use 1,000 forgotten CIFAR-100 training examples as pre-unlearning members and 1,000 CIFAR-100 test examples as nonmembers. We train 64 shadow ResNet-18 models for 50 epochs. For every target sample, exactly half of the shadows include the sample and half exclude it. The attack statistic is the logit-scaled true-class confidence used by RULI,

$$s(x, y) = \log p_\theta(y \mid x) - \log \sum_{j \neq y} p_\theta(j \mid x).$$

We fit sample-specific Gaussian KDEs to the IN and OUT shadow-score distributions and use a monotone transform of their likelihood ratio as the membership score. Retrain, FT, and FT+CSD share the same target pool and saved shadow distributions. AUC is computed from the resulting membership scores. For low-FPR reporting, TPR@ α is defined as $\max_{\mathrm{FPR} \leq \alpha} \mathrm{TPR}$.

Diagnostics. We log $\widehat{D}_{\mathrm{CS}}$ on retain and forget features at convergence for every run, denoted CSD_r and CSD_f in the sweep tables, which underlie the selectivity analysis of Figure 4.

C Additional Experimental Results

C.1 Retrained reference values

Table 5: Retrained reference values used throughout. ResNet references follow the CoUn protocol. ViT references are retrained by us on $\mathcal{D}_r$ with the AdamW recipe, one run each.

Setting	RA	UA	TA	MIA
CIFAR-10, RN18, 10%	100.00	4.81	94.67	11.02
CIFAR-100, RN18, 10%	99.98	24.26	75.56	48.44
CIFAR-100, RN18, 50%	99.98	31.41	68.41	58.35
Tiny ImageNet, RN18, 10%	99.98	36.16	63.82	63.73
Tiny ImageNet, ViT, 10%	99.99	43.81	44.11	73.19
CIFAR-100, ViT, 10%	99.98	61.56	61.40	56.86
CIFAR-10, ViT, 10%	99.99	12.18	87.39	19.00

Metric convention. For every setting, unlearning accuracy is the classification error on the forget set:

$$\mathrm{UA} = 100 - \mathrm{Acc}(D_f).$$

The reference tables therefore report forget-set error rather than forget-set classification accuracy.

CIFAR-10/ViT reference verification. A deterministic reevaluation of the retrained CIFAR-10/ViT checkpoint gives a forget-set accuracy of 87.82%, corresponding to $\mathrm{UA} = 12.18\%$ reported in Table 5. The remaining reference metrics retain the values used by the original project evaluation pipeline.

C.2 Per-metric results for all settings

Table 6 reports the per-metric results for all evaluated settings. CSD-Unlearning generally moves under-forgotten UA toward its retrained target, while the MIA statistic can overshoot on ViT. The ViT settings also exhibit substantially larger variance.

Table 6: Per-metric results, mean ± std over five seeds, 50 epochs.

Setting	Method	RA	UA	TA	MIA	Avg.Gap
CIFAR-10 RN18, 10%	FT	100.00	3.78±0.19	94.71±0.08	9.67±0.41	0.61±0.12
	FT+CSD	99.99	3.88±0.20	94.53±0.20	10.74±0.58	0.39±0.15
CIFAR-100 RN18, 10%	FT	99.98	16.31±0.29	76.52±0.34	44.16±0.50	3.30±0.18
	FT+CSD	99.83±0.05	24.68±0.94	72.24±0.44	49.06±1.89	1.48±0.21
CIFAR-100 RN18, 50%	FT	99.98	17.51±0.25	74.28±0.57	50.40±0.40	6.93±0.26
	FT+CSD	99.98	19.10±0.35	73.29±0.51	53.91±0.30	5.41±0.18
Tiny ImageNet RN18, 10%	FT	99.98	28.09±0.31	65.10±0.47	58.80±0.75	3.57±0.20
	FT+CSD	99.89±0.02	36.07±1.01	61.68±0.41	59.59±0.62	1.78±0.25
Tiny ImageNet ViT, 10%	FT	99.98	28.43±0.58	46.55±0.72	65.13±0.43	6.47±0.31
	FT+CSD	99.98±0.01	39.94±6.78	45.40±0.57	83.60±12.67	5.01±1.90
CIFAR-100 ViT, 10%	FT	99.97	1.31±0.09	63.40±0.39	21.03±0.82	24.52±0.14
	FT+CSD	99.95	37.04±9.41	61.02±0.25	99.62±0.09	16.92±2.39
CIFAR-10, ViT, 10%	FT	99.96±0.01	0.80±0.12	87.83±0.19	5.40±0.43	6.36±0.14
	FT+CSD	99.93±0.02	2.16±0.43	87.65±0.14	13.18±0.90	4.04±0.22
CIFAR-100, RN18, 10%, matched lr 0.1:						
	FT+CSD (β=1)	99.96±0.01	24.63±1.01	74.69±0.34	56.47±0.59	2.39±0.38

Table 7: CIFAR-100, RN18, 10%. Avg.Gap over the learning-rate and β grid. Plain FT per learning rate: 15.51 (0.01), 5.80 (0.05), 3.46 (0.1).

	β=0.15	β=0.3	β=1	β=3
lr 0.01	15.07	14.92	12.86	6.17
lr 0.05	4.95	4.14	4.17	**0.92**
lr 0.1	2.81	2.47	**2.07**	2.77

C.3 Hyperparameter Sensitivity

All hyperparameter studies in this appendix are single-seed explorations with 50 unlearning epochs. The learning-rate–β sweeps are used to characterize the optimization regime and select the configurations evaluated in the main experiments, while the additional α and G ablations examine the sensitivity to the fixed CSD design parameters. Final configurations reported in the main tables are evaluated over five independent seeds. Consequently, individual sweep entries are not expected to match the corresponding five-seed means exactly. For example, plain fine-tuning on CIFAR-100 at learning rate 0.1 gives an Avg.Gap of 3.46 in the single-seed sweep and 3.30 ± 0.18 over five seeds.

Sensitivity to the learning rate and β. Table 7 shows a clear interaction between the learning rate and the repulsion weight β. At small learning rates, weak repulsion barely changes the fine-tuning solution: for $\mathrm{lr} = 0.01$, Avg.Gap remains above 12 for $\beta \leq 1$, but drops sharply to 6.17 at $\beta = 3$. A similar transition occurs at $\mathrm{lr} = 0.05$, where increasing β from 1 to 3 reduces Avg.Gap from 4.17 to 0.92. At the larger learning rate $\mathrm{lr} = 0.1$, however, the best value shifts to the smaller $\beta = 1$. Further increasing β to 3 degrades Avg.Gap from 2.07 to 2.77. The preferred β shifts with learning rate in these sweeps. Because the surrogate threshold in Theorem 1 contains no learning-rate term, we treat this interaction as an empirical optimization effect rather than a theoretical prediction. Practically, β should be retuned when the learning rate changes.

Sensitivity to α and G. Table 8 examines the two fixed CSD design parameters while keeping the learning rate and repulsion weight at $\mathrm{lr} = 0.05$ and $\beta = 3$. The anchor weight exhibits a clear intermediate optimum. Removing retain-side anchoring ($\alpha = 0$) increases Avg.Gap from 0.92 to 2.25, whereas weakening it to $\alpha = 0.5$ gives an intermediate gap of 1.30. Increasing the anchor

Table 8: Sensitivity to the retain-side weight α and the forget-side cap G on CIFAR-100 with ResNet-18 at 10% forgetting. All runs use a fixed seed, 50 unlearning epochs, $\mathrm{lr} = 0.05$, and $\beta = 3$. Lower Avg.Gap is better.

Setting	RA	UA	TA	MIA	Avg.Gap	D^f_{CS}
Anchor-weight sensitivity ($G = 5$)						
$\alpha = 0$	99.96	21.30	75.01	53.92	2.25	0.7010
$\alpha = 0.5$	99.95	22.72	74.21	50.74	1.30	0.6934
$\alpha = 1$	99.84	24.28	72.47	48.88	**0.92**	0.6876
$\alpha = 2$	98.58	29.78	69.06	49.88	3.71	0.6679
Cap sensitivity ($\alpha = 1$)						
$G = 2$	99.84	24.28	72.47	48.88	0.92	0.6876
$G = 5$	99.84	24.28	72.47	48.88	0.92	0.6876
$G = 10$	99.84	24.28	72.47	48.88	0.92	0.6876

weight further to $\alpha = 2$ is also detrimental, raising Avg.Gap to 3.71 and reducing both retain and test accuracy. Thus, the default $\alpha = 1$ provides the best utility–forgetting balance among the tested values.

In contrast, varying the forget-side cap over $G \in \{2, 5, 10\}$ produces identical outcomes in this setting. The converged forget-side divergence is only 0.688, below even the smallest tested cap. This is consistent with the intrinsic boundedness induced by feature normalization: the explicit cap acts primarily as a numerical safeguard and is inactive in this normalized regime.

Table 9: Remaining single-seed sweeps, reported in Avg.Gap. Diverged runs are marked. For CIFAR-10/ViT, Avg.Gap values are recomputed using the verified retrained reference $\mathrm{UA} = 12.18$. The stability description for CIFAR-100 with a 50% forget ratio is discussed in the text.

Setting (lr)	β grid	Avg.Gap
CIFAR-10, RN18 ($\mathrm{lr} = 0.1$), plain 0.54	0.15, 0.3, 0.5, 1, 3	**0.23**, 0.45, 0.46, 5.81, 2.39
Tiny ImageNet, RN18 ($\mathrm{lr} = 0.05$), plain 3.73	1, 3, 6	3.65, **1.80**, 2.84
CIFAR-100, RN18, 50% ($\mathrm{lr} = 0.1$), plain 7.12	0.3, 1, 3, 6	6.42, **6.39**, diverged, unstable
CIFAR-100, RN18, 50% ($\mathrm{lr} = 0.05$), plain 11.28	0.3, 1, 3, 6	10.47, 10.09, 10.04, diverged
Tiny ImageNet, ViT ($\mathrm{lr} = 0.05$), plain 6.48	1, 3, 6	10.56, **3.69**, 12.05
CIFAR-100, ViT ($\mathrm{lr} = 0.05$), plain 24.47	1, 1.5, 2, 3, 6	22.05, 22.58, 15.07, **13.39**, 16.22
CIFAR-100, ViT ($\mathrm{lr} = 0.03$)	3	23.16
CIFAR-10, ViT ($\mathrm{lr} = 0.05$), plain 6.14	1, 3, 6, 10	6.70, 3.97, 3.33, 3.01
CIFAR-10, ViT ($\mathrm{lr} = 0.1$), plain-FT stress	–	2.18 (UA 7.28)

The NegGrad+ coefficient used below is selected from the single-seed sweep in Table 10.

C.4 Convergence diagnostics

Converged feature divergences for the ignited five-seed configurations, supporting the saturation and selectivity analyses. For CIFAR-100/RN18 at 10% forgetting, we obtain $\mathrm{CSD}_r = 0.298 \pm 0.036$ and $\mathrm{CSD}_f = 0.689 \pm 0.007$. For Tiny ImageNet/RN18, the corresponding values are $\mathrm{CSD}_r = 0.404 \pm 0.036$ and $\mathrm{CSD}_f = 0.721 \pm 0.010$. For Tiny ImageNet/ViT and CIFAR-100/ViT, the forget-side divergences are $\mathrm{CSD}_f = 0.496 \pm 0.013$ and $\mathrm{CSD}_f = 0.567 \pm 0.005$, respectively. For CIFAR-10/ViT at $\beta = 10$, $\mathrm{CSD}_f = 0.671$. Finally, for CIFAR-100/RN18 with 50% forgetting at $\mathrm{lr} = 0.05$ and $\beta = 3$, $\mathrm{CSD}_r = 1.024$ and $\mathrm{CSD}_f = 1.014$, illustrating the selectivity collapse shown in Figure 4.

C.5 Ablation Study

Figure 3 depicts the visual comparison of Avg.Gap to the retrained reference across six settings. Results are averaged over five seeds. CSD-Unlearning reduces Avg.Gap relative to tuned fine-tuning by 22%–55% across these settings.

Table 10: NegGrad+ tuning on CIFAR-100, RN18, 10% at 50 epochs and learning rate 0.1, single seed. The method is acutely sensitive to β_{ng}, and the best setting slightly over-forgets.

β_{ng}	0.95	0.99	0.995
Avg.Gap	23.10	7.07	**2.45**
UA	74.68	42.96	31.36

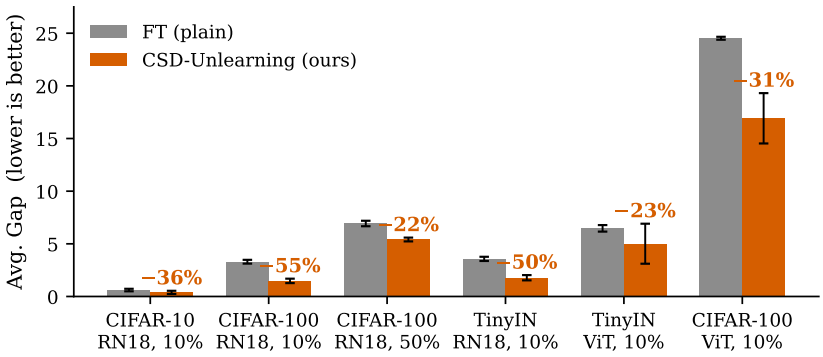


Figure 3: Average gap to the retrained reference across the six headline settings, five seeds each. The CSD objective improves tuned fine-tuning everywhere, by 22 to 55 percent.

Table 11: Component ablation on CIFAR-100/ResNet-18 with 10% forgetting. All variants use one fixed seed, 50 epochs, learning rate 0.05, and $G = 5$. The target for every metric is the retrained value rather than an extreme.

Variant	α	β	RA	UA	TA	MIA	Avg.Gap
Retrain reference	–	–	99.98	24.26	75.56	48.44	0.00
Plain FT	0	0	99.98	9.96	77.09	41.78	5.62
Anchor only	1	0	99.97	9.90	76.90	42.48	5.42
Repulsion only	0	3	99.96	21.30	75.01	53.92	2.25
CSD-Unlearning	1	3	99.84	24.28	72.47	48.88	**0.92**

Table 11 isolates the retain anchor and forget repulsion on CIFAR-100/ResNet-18. All variants use one fixed seed, 50 epochs, learning rate 0.05, and $G = 5$. Anchor-only training remains nearly indistinguishable from plain fine-tuning: its Avg.Gap changes only from 5.62 to 5.42, while UA remains near 10 against a retrained target of 24.26. Thus, the anchor alone does not provide the active pressure required to overcome under-forgetting. Forget repulsion supplies this missing pressure, reducing the gap to 2.25 and moving UA from 9.96 to 21.30. However, repulsion-only produces an imbalanced metric profile: UA still undershoots its retrained target, whereas MIA overshoots from 48.44 to 53.92. Combining the two components brings UA and MIA to 24.28 and 48.88, respectively, and achieves the lowest overall gap of 0.92. These results support complementary roles: repulsion initiates forgetting, while the anchor regulates the coupled optimization trajectory toward a more balanced match to retraining.

C.6 Computational overhead

Table 12 reports the runtime and memory cost of CSD on CIFAR-100 with ResNet-18. Adding CSD increases wall-clock time from 601.3 to 907.2 seconds over 50 epochs (+50.9%), while peak GPU memory increases by only 118 MB (+4.5%). Thus, the additional cost is primarily computational rather than memory.

Table 12: Computational overhead on CIFAR-100 with ResNet-18 at 10% forgetting. Results are averaged over five runs of 50 unlearning epochs. GPU memory reports the peak device-memory increase during unlearning.

Method	Time (s)	Time / epoch (s)	Peak GPU memory (MB)
FT	601.3 ± 5.7	12.03 ± 0.11	2612 ± 0
CSD-Unlearning	907.2 ± 11.0	18.14 ± 0.22	2730 ± 0

Table 13: Baseline compatibility on CIFAR-100, RN18, 10%, single-seed exploration at 50 epochs. The CSD objective adds forgetting pressure, so it transforms an under-forgetting baseline and barely moves a well-tuned over-forgetting one. The sparsity baseline is reported separately and exhibits a distinct retain-accuracy failure mode.

Base method	Variant	Avg.Gap	Note
FT (under-forgets)	plain	3.46	UA 8.2 below reference
	+ CSD (β=1)	**2.07**	forgetting supplied
NegGrad+ (over-forgets)	plain, tuned (β_{ng}=.995)	2.45	UA 7.1 above reference
	+ CSD (β=0.15)	2.34	marginal, 4.5%
	+ CSD retain-only (α=3)	2.32	marginal, 5.3%
ℓ_1-sparse	γ=10^{-4}	17.11	no forgetting, UA 0.04
	γ=10^{-3}	9.59	RA collapses to 85.13
	γ=10^{-2}	diverged	

C.7 Baseline compatibility

Table 13 tests CSD as a plug-in for FT and NegGrad+ and reports the ℓ_1-sparse method separately as a parameter-space baseline. Fine-tuning under-forgets, and the CSD objective supplies exactly the missing ingredient. Once tuned, NegGrad+ (Kurmanji et al., 2023) slightly over-forgets, with UA about 7 points above the retrained reference, and neither CSD variant provides more than a marginal improvement, whether using forget-side repulsion or retain-side anchoring alone. The ℓ_1-sparse baseline (Jia et al., 2023) fails along a third axis. Aggressive weight sparsification damages retain accuracy itself, a deficit that lives in parameter space. The lesson is a matching principle. In a single-seed CIFAR-100/RN18 diagnostic, the plug-in provides substantial benefit when attached to an under-forgetting FT baseline, but only marginal benefit when attached to the tested over-forgetting NegGrad+ configuration.

C.8 Operating Boundaries

The gains above have a clear boundary, and the boundary teaches as much as the gains. **Selectivity collapses at high forget ratios.** Figure 4 tracks the converged divergences of retain and forget features from the original model. At 10% forgetting, the forget-side divergence is 2.31 times the retain-side divergence and Avg.Gap drops by 55%. At 50%, the two divergences nearly equalize (1.02 vs. 1.01) and the gain shrinks to 22%, indicating substantially weaker selectivity.

C.9 Additional Comparison with SCRUB and SalUn

All models in Table 14, including the retrained references, are reevaluated with the same retain-only BN recalibration using 50 batches of size 256. UA denotes forget-set classification error; MIA denotes the SVC predicted-nonmember rate on the forget set, not attack AUC. Both are interpreted relative to retraining rather than an extreme value. Because this audit uses split-matched retrained references and a common BN recalibration for every method, its FT/CSD values are reported as a separate shared-protocol reevaluation and need not numerically coincide with the legacy project-pipeline values in Tables 1.

For SCRUB (Kurmanji et al., 2023), the selected configuration uses learning rate 10^{-3}, retain distillation weight 0.5, temperature 4, and 20 retain epochs, with forget-maximization phases in the

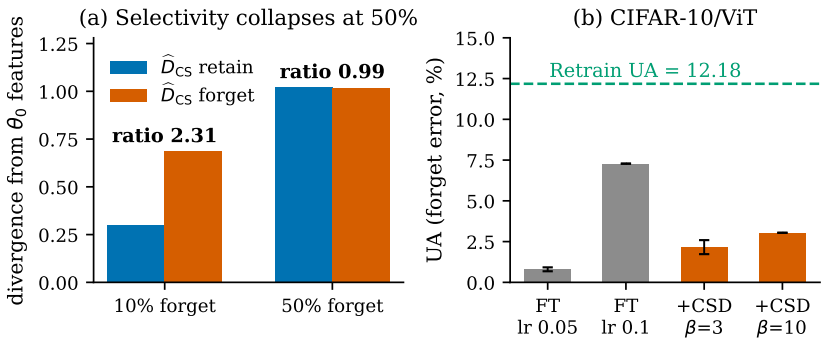


Figure 4: Operating-boundary diagnostics. (a) Converged retain- and forget-side CSD on CIFAR-100/ResNet-18, showing loss of selectivity as the forget ratio increases from 10% to 50%. (b) CIFAR-10/ViT UA under FT and two CSD settings, illustrating the narrow operating window relative to the retrained UA target.

Table 14: Additional comparison on CIFAR-100/ResNet-18 with 10% random-subset forgetting under a shared evaluation protocol. Values are mean ± sample standard deviation over five seed-dependent deletion splits. SCRUB denotes the final checkpoint, while SCRUB (rewind) uses the validation-based rewind checkpoint from the same training trajectory. Avg.Gap is computed separately for each run against its split-matched retrained reference and then averaged. † denotes our reproduction under the reported finite hyperparameter search.

Method	RA	UA	TA	MIA	Avg.Gap ↓
Retrain	99.974 ± 0.002	24.616 ± 0.608	75.610 ± 0.353	49.260 ± 0.612	0
FT	99.973 ± 0.005	16.512 ± 0.490	76.742 ± 0.454	45.224 ± 0.451	3.320 ± 0.262
FT+CSD	99.831 ± 0.039	24.648 ± 0.983	72.193 ± 0.507	48.884 ± 1.664	$\mathbf{1.496 \pm 0.283}$
SCRUB†	99.976 ± 0.003	0.096 ± 0.026	76.713 ± 0.466	6.828 ± 0.470	17.015 ± 0.214
rewound SCRUB	99.922 ± 0.027	1.008 ± 0.059	75.645 ± 0.605	12.964 ± 0.882	15.092 ± 0.292
SalUn†	99.840 ± 0.028	21.368 ± 0.757	71.462 ± 0.471	88.528 ± 0.511	11.699 ± 0.319

first 5 epochs. For SalUn (Fan et al., 2024), the selected configuration uses learning rate 0.0065, a saliency-mask fraction of 0.5, and 10 unlearning epochs. Each configuration is selected on the designated development split and then fixed across the five deletion splits. These results characterize our reproductions under the reported finite searches rather than the best attainable performance of either baseline.

Under this shared evaluator, FT+CSD reduces Avg.Gap from 3.320 ± 0.262 for tuned FT to 1.496 ± 0.283. SCRUB and rewound SCRUB remain strongly under-forgotten in the selected configurations. SalUn moves UA closer to the retrained target, but its confidence-based SVC statistic substantially overshoots the retrained reference, yielding an Avg.Gap of 11.699 ± 0.319. The comparison therefore highlights that improving a single forgetting metric does not necessarily imply a balanced match to retraining across the full metric profile.

C.10 Comparison with Maximum Mean Discrepancy

We further compare CSD with maximum mean discrepancy (MMD) (Gretton et al., 2012), a standard kernel-based discrepancy between sample distributions. MMD is a natural controlled alternative because it operates on the same feature batches and Gaussian kernel as CSD, while measuring Euclidean separation between kernel mean embeddings rather than their angular discrepancy.

To isolate the effect of the discrepancy itself, we keep the feature representation, normalization, Gaussian kernel, per-batch median bandwidth, optimizer, batch sizes, and evaluation protocol unchanged. Since CSD and MMD have different numerical scales, their regularization weights are tuned independently under the same search budget, $\alpha \in \{0.3, 1, 3\}$ and $\beta \in \{1, 3, 10\}$. All exper-

Table 15: Controlled comparison between CSD and MMD on CIFAR-100/ResNet-18 with 10% forgetting. Both discrepancies use the same training and evaluation pipeline. Results are from one fixed seed.

Discrepancy	α	β	RA	UA	TA	MIA	Avg.Gap
Retrain reference	–	–	99.98	24.26	75.56	48.44	0.00
MMD	1	10	99.72	26.90	72.75	49.28	1.64
CSD	1	3	99.84	24.28	72.47	48.88	**0.92**

iments use one fixed seed on CIFAR-100/ResNet-18 with 10% forgetting, learning rate 0.05, and 50 unlearning epochs. Table 15 compares MMD and CSD as unlearning regularizers. The best MMD configuration occurs at the upper edge of the initial repulsion grid, $\beta = 10$. We therefore additionally evaluate $\beta \in \{20, 30\}$ with $\alpha = 1$. Both runs diverge, indicating that further increasing the MMD repulsion enters an unstable regime rather than continuing the improvement. Under the same search budget, CSD achieves a lower Avg.Gap than MMD, 0.92 versus 1.64. The difference is mainly reflected in the forgetting metrics: CSD nearly matches the retrained UA (24.28 versus 24.26) and remains close in MIA (48.88 versus 48.44), whereas the best MMD configuration overshoots both targets. These results suggest that the benefit is not solely due to introducing an arbitrary kernel discrepancy, while we treat this comparison as a controlled single-seed ablation rather than a statistically significant ranking.

D TAKEAWAYS

1. **A set-level divergence suffices.** Without pair construction, augmentation, or temperature tuning, the CSD objective reduces the gap to retraining by 22–55% across the six headline settings.
2. **Match the tool to the failure mode.** CSD substantially helps the tested under-forgetting FT configuration and adds little to the tested over-forgetting NegGrad+ configuration. The ℓ_1-sparse baseline exhibits a separate utility failure mode.
3. **Expect threshold-like sensitivity.** Several settings exhibit a sharp response to the repulsion weight, motivating logarithmic rather than purely local sweeps over β.
4. **Selectivity can weaken at large forget ratios.** On CIFAR-100/ResNet-18, increasing the forget ratio from 10% to 50% largely eliminates the separation between retain- and forget-side feature divergences and reduces the gain over fine-tuning.
5. **The operating window is architecture- and optimization-dependent.** The mean Avg.Gap is lower for CSD-Unlearning in all six reported settings. The improvement is particularly clear on the ResNet settings, whereas the Tiny ImageNet/ViT result exhibits substantially larger seed variability.

E LIMITATIONS

Our scope is random-subset forgetting in image classification with two model families, and class-wise forgetting, larger scales, and other modalities remain untested. The retrained references for ViT are self-trained with one training seed per setting, so their values carry uncertainty due to retraining randomness. Hyperparameter sweeps and the baseline-compatibility study are single-seed explorations, clearly marked, while every headline unlearning result is evaluated over five seeds. The CoUn comparison uses results obtained under a different evaluation pipeline and is therefore indicative only. Theorem 1 analyzes a one-dimensional capped surrogate and should be interpreted as a mechanistic explanation rather than a guarantee for the full network. Finally, our proposed method requires one additional frozen forward pass and a kernel computation per batch, and its cost remains untested at foundation-model scale.

For ViT, the original and retrained models are trained with AdamW, whereas the unlearning runs use the common SGD protocol adopted for the cross-architecture comparison. Consequently, the narrower stability window observed for ViT may reflect an interaction between architecture and optimizer choice rather than architecture alone.