LACS: DISTRIBUTION-SENSITIVE LABEL-AGNOSTIC CONFIDENCE SELECTION FOR DYNAMIC DATA PRUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Training on large datasets spends much of its computation on examples whose value changes as the model learns: confidently learned examples contribute little further gradient signal, while confident mistakes can dominate the next fitting step. Dynamic data pruning reduces this cost, but a fixed quota decides how much to prune before seeing the current score distribution, and an absolute threshold is brittle to the scale and drift of model confidence.

We introduce LACS (Label-Agnostic Confidence Selector), a training-time subset selector whose threshold and retained count are both determined by the current confidence distribution. LACS reads only each example's maximum predictive confidence—never its label—and removes an example when its confidence exceeds a log-mean-exp reference of the current population plus a margin. At a fixed mean confidence this reference moves with dispersion and upper-tail shape, and it admits an exact no-pruning criterion: nothing is removed when too much confidence mass lies near the current maximum. The pruned fraction never exceeds a user-set bound. On real training batches the rule behaves as the analysis predicts: the pruning rate rises from about 11% early in CIFAR-10 training to 43% in mid-training and returns to 2% once confidences become uniformly high, the no-pruning condition matches the observed decision on every logged batch, and batches with equal mean confidence but different shape receive different retained fractions.

We evaluate LACS in seven settings spanning neural, language-model, long-tailed, and semi-supervised training and three repeated-fitting (boosting) regimes. In neural training LACS removes 10–35% of examples from backward propagation with accuracy within seed noise of full training on CIFAR-10 and CIFAR-100; at a matched backward budget and matched number of optimizer updates it outperforms small-batch training by 1.0 pp and random selection by 1.2 pp over five seeds, and it has the higher mean than every work-matched dynamic-pruning baseline we ran. In repeated fitting the selected support improves prediction: across 14 tabular datasets, LACS lowers the mean test error of a quadratic boosting learner from 23.36% to 22.96% while fitting about 30% fewer rows over full replays, and it beats a count-matched random support on 13 of 14 datasets. In fitting-dominated RBF-SVM boosting the saved work becomes wall-clock savings of 30–94% in seven of eight dataset/split cells, and none where the rule prunes nothing. Executed work and wall-clock time are reported separately: the neural implementation adds a scoring forward and does not yet convert its sparsity into faster training, and cached scores alone do not change that; the conversion is a systems question, not a property of the rule.

1 INTRODUCTION

Modern learning increasingly relies on datasets whose scale makes data efficiency a first-order computational concern (Qin et al., 2024; Zhou et al., 2025; Raju et al., 2021). The training value of an example is not fixed, however. A confidently correct example has little remaining cross-entropy gradient, while a confidently wrong example carries a large derivative and can dominate the next fit.

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.

Pruning can therefore serve two purposes at once: skip repeated work on redundant examples, and limit the influence of unusually influential ones. Neither purpose requires knowing *why* an example is confidently wrong; it need not be mislabeled or anomalous, and confidence alone does not say.

We study *training-time instance subset selection*: at each selection event a rule chooses which candidates enter the expensive part of the update, backward propagation or weak-learner fitting. Our primary efficiency quantity is this executed work, counted in examples; wall-clock time depends on architecture, libraries, and scoring overhead, so we measure it separately (Section 4.6). This separates two questions: does selection reduce expensive work without hurting prediction, and when does that reduction become acceleration?

The selection decision itself is delicate because pruning changes the empirical objective and the subsequent trajectory. A fixed quota fixes the amount of pruning before seeing whether the current scores support it. An absolute threshold is brittle to the scale and drift of model confidence. Population-relative rules improve on both: InfoBatch, for example, uses the current mean loss to define a low-loss pool and then thins that pool with a configured probability (Qin et al., 2024). We ask a further question: can the *shape* of the current confidence distribution determine both where to cut and how much to keep?

LACS, the *Label-Agnostic Confidence Selector*, answers this with a single scalar rule. At each selection event it reads only the unsigned maximum predictive confidence of each candidate—never the label—compares its exponential with the population mean of exponentials, and assigns an exact zero weight to examples that are sufficiently extreme relative to the current population. Neither the threshold nor the retained count is prespecified; both follow from the score geometry. The threshold is a log-mean-exp reference plus a margin. Where an arithmetic mean fixes its boundary from the first moment, the log-mean-exp reference is Schur-convex and, at the same mean, rises with dispersion and upper-tail mass; its maximum-centered form gives an exact no-pruning criterion in terms of how much confidence mass lies near the current maximum. In a two-cluster illustration the upper cluster is removed only when it is both sufficiently separated *and* sufficiently small: holding confidence fixed gives a critical mass, holding mass fixed gives a critical confidence (Figure 1; full phase diagram and proofs in Appendix A). These are not only properties of a toy model: on logged CIFAR-10 batches the retained fraction moves from .89 to .58 and back to .98 as the confidence population spreads and then collapses, the no-pruning condition coincides with every observed no-pruning event, and same-mean batches of different shape receive different supports (Section 4.5).

We evaluate the same rule in seven settings: (i) supervised neural training, (ii) Transformer language-model training, (iii) long-tailed and (iv) semi-supervised variants, and repeated fitting with (v) compact nonlinear, (vi) tree, and (vii) kernel-SVM weak learners. The main text reports the neural, compact, tree, and SVM results with the systems accounting; the tree experiments use the fixed-threshold companion LACS-T, kept distinct from the normalized rule throughout. The long-tailed and semi-supervised boundary studies, extended baselines, null results, and row-level mechanism diagnostics (wrong removals, re-entry, label-flip enrichment, one-sided controls, an InfoBatch-style boosting adaptation) are in the supplement. Boosting is especially informative because the support is recomputed before every weak-learner fit.

Contributions. (1) A label-agnostic, distribution-sensitive selector with an $O(n)$ scalar implementation, whose threshold and retained count are outcomes of the current confidence population rather than a fixed quota. (2) An analytical characterization of that cutoff, verified on real trajectories: same-mean populations can induce different thresholds, the maximum-centered mass gives the exact no-pruning condition $M_\tau(c) > \rho n$ and the bound $|D|/n \leq \rho$, and the two-cluster model yields closed-form mass/gap boundaries. (3) A seven-setting evaluation that separates prediction, statistical uncertainty, executed work, and wall time: fewer backward or fitting rows at similar prediction, a controlled neural result in which LACS beats update-matched small-batch and random training by about 1 pp, statistically supported gains in several repeated-fitting settings, and fitting-dominated SVM boosting in which the saved work becomes 30–94% wall-clock savings. Section 5 states the limits.

2 LACS: DISTRIBUTION-SENSITIVE INSTANCE SELECTION

2.1 CONFIDENCE, RELATIVE SCORES, AND SPARSE SUPPORT

Let $\mathcal{B}_t = \{x_i\}_{i=1}^n$ be the candidate population at selection event t, and let $p_{\theta_t}(k \mid x_i)$ be the model's predictive probability. We use standard maximum confidence

$$c_{t,i} = \max_k p_{\theta_t}(k \mid x_i), \qquad q_{t,i} = \frac{\exp(c_{t,i}/\tau)}{n^{-1}\sum_j \exp(c_{t,j}/\tau)}, \qquad u_{t,i} = [1 - \rho q_{t,i}]_+. \tag{1}$$

Here $[a]_+ = \max(a, 0)$, $\tau > 0$, and $\rho > 0$. The active set and count are

$$S_t = \{i : u_{t,i} > 0\}, \qquad K_t = |S_t|. \tag{2}$$

Selection is *label-agnostic*: Eq. (2) never reads y_i (the downstream learner and the hyperparameter search remain supervised). Subtracting $\max_i c_{t,i}$ inside both exponentials gives a numerically stable implementation.

Two downstream contracts are used. *Hard support* passes S_t to the fitter with its ordinary fitting weights. *Continuous weighting* uses $w_{t,i} = u_{t,i}/\bar{u}_t$, where $\bar{u}_t = n^{-1}\sum_j u_{t,j}$; neural experiments detach these coefficients. The analysis below uses $0 < \rho < 1$, which guarantees $\bar{u}_t > 0$. Historical searches over a wider domain are identified in Appendix B.3.

For detached coefficients and sample-independent computations, exact physical omission targets the dense weighted update

$$\nabla_\theta \left[\frac{1}{n}\sum_i \mathrm{stopgrad}(w_i)\ell_i(\theta)\right] = \frac{1}{n}\sum_{i\in S} w_i \nabla_\theta \ell_i(\theta). \tag{3}$$

Here $\mathrm{stopgrad}(w_i)$ holds the coefficients fixed. A zero-weight term contributes no gradient, so omitting its computation reproduces the frozen weighted update whenever retained activations, Jacobians, the original denominator, and stochastic realizations are unchanged; training-mode BatchNorm violates this because batch statistics couple examples (Ioffe & Szegedy, 2015). This is an identity for the LACS-weighted surrogate, not an equality to unweighted CE (conditions and checks in Appendix A).

Given scalar confidences, selection needs $O(n)$ arithmetic and no sorting or subset solver. Computing these confidences can require additional model execution. Equal support also need not yield equal updates: survivor weights, normalization, and boosting coefficients are part of the downstream contract.

2.2 A THRESHOLD DETERMINED BY THE CURRENT DISTRIBUTION

For $0 < \rho < 1$, write $\kappa = 1/\rho > 1$. An example is removed exactly when

$$c_i \geq T(c), \qquad T(c) = \tau \log\left(\frac{1}{n}\sum_j e^{c_j/\tau}\right) + \tau \log \kappa. \tag{4}$$

Thus LACS is a thresholding rule, but not an absolute-confidence rule: its threshold moves with the whole confidence population.

Why log-mean-exp is more than a mean. Set $\beta = 1/\tau$ and write

$$\mathrm{LME}_\beta(c) = \frac{1}{\beta}\log\left(\frac{1}{n}\sum_i e^{\beta c_i}\right), \qquad T(c) = \mathrm{LME}_\beta(c) + \frac{\log\kappa}{\beta}.$$

Fact: same-mean dispersion sensitivity. For every $\beta > 0$, LME_β is symmetric and convex, hence Schur-convex: among score populations with the same mean, a more dispersed vector in the majorization order cannot lower the reference. Around $\beta = 0$,

$$\mathrm{LME}_\beta(c) = \mu + \frac{\beta}{2}\sigma^2 + \frac{\beta^2}{6}\kappa_3 + O(\beta^3), \tag{5}$$

where κ_3 is the empirical third cumulant. The reference also satisfies $\mu \le \mathrm{LME}_\beta(c) \le c_{\max}$, tending to μ as $\beta \to 0$ and to $c_{\max}$ as $\beta \to \infty$. A boundary using only the arithmetic mean of the *same score* is identical for same-mean populations; the LACS boundary separates them through spread and upper-tail shape: concentration near the mean pulls the reference toward μ, whereas dispersion and, locally, positive upper-tail skew raise it. These reference limits do not apply to the full cutoff: at fixed $\kappa > 1$, the added $\log \kappa/\beta$ diverges as $\beta \to 0$. A higher reference also need not imply fewer removals, because the scores can move with the boundary. Appendix A gives the proof and same-mean examples.

Let $c_{\max} = \max_i c_i$ and define the maximum-centered mass

$$M_\tau(c) = \sum_i e^{(c_i - c_{\max})/\tau}. \tag{6}$$

Then

$$\boxed{D = \varnothing \iff M_\tau(c) > \rho n}, \qquad |D|/n \le \rho. \tag{7}$$

Each term in M_τ lies in $(0, 1]$: examples near the maximum count almost fully and distant examples count less. Hence a homogeneous population is never pruned, whether its common confidence is low or high. The criterion is about how much mass lies near the current upper end, not about total confidence ($\sum_i q_i = n$ is fixed). Proofs and boundary cases appear in Appendix A.

2.3 Two-cluster illustration: mass and confidence play different roles

The rule in Eq. (7) applies to arbitrary score distributions. To make its geometry interpretable, consider an illustrative distribution with lower-cluster confidence $c_L = a$ and upper-cluster confidence $c_H = a + \Delta$, where the upper cluster has mass r. The normalized near-upper mass is

$$m_{\mathrm{eff}}(r, \Delta) = r + (1 - r)e^{-\Delta/\tau} = M_\tau(c)/n. \tag{8}$$

The complete tied upper cluster is pruned iff $m_{\mathrm{eff}} \le \rho$, equivalently

$$r < \rho, \qquad \Delta \ge \Delta_{\mathrm{crit}}(r) = \tau \log \frac{1 - r}{\rho - r}. \tag{9}$$

The raw upper mass r is therefore only part of the criterion. A small gap makes lower-cluster points contribute to the effective mass; a large gap discounts them. Increasing upper-cluster mass raises the threshold. At fixed $r < \rho$, a larger gap moves c_H faster than the threshold and can trigger removal; bounded confidence may limit the reachable gap.

Relationship to quantiles. At the realized K_t, hard LACS support is exactly the K_t lowest-confidence examples, so LACS is a *gap- and mass-sensitive quantile level*, not a new ranking: a fixed quantile chooses the count first, LACS makes it an outcome of the scores. Common additive shifts leave the rule unchanged; arbitrary monotone recalibration need not.

2.4 Applications and variants

A generic interface is $S_t = \mathrm{Select}(\mathcal{B}_t, p_{\theta_t})$ followed by an application-specific update on S_t: neural selective execution applies continuous coefficients and preserves the original denominator when skipping rows; repeated fitting passes the selected rows to a tree, SVM, or compact nonlinear weak learner before every round.

The tree experiments use a companion fixed unsigned-margin rule $S_t^T = \{i : |F_t(x_i)| \le \vartheta\}$, denoted LACS-T. Because $\sigma(|F_t|)$ is the confidence of a binary logistic prediction, LACS-T is an absolute-confidence variant, not the distribution-sensitive rule of Eq. (1); its results support unsigned-confidence selection in general but do not test the normalized cutoff, and we label them accordingly. Complete score, weighting, parameter, and denominator contracts are listed in Appendix B.3.

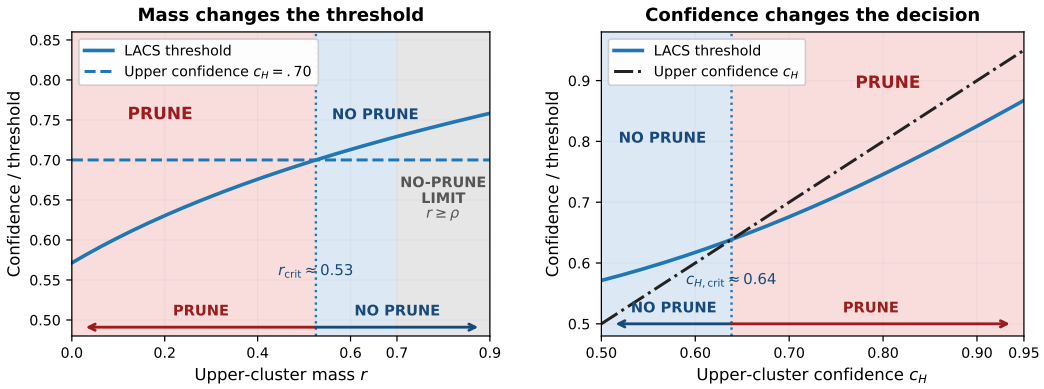


Figure 1: **Two one-dimensional slices of the illustrative two-cluster geometry. Left: fixed confidences, varying upper-cluster mass.** With $c_L = .50$, $c_H = .70$, $\tau = .20$, and $\rho = .7$, the adaptive threshold rises with mass and crosses the fixed upper confidence at $r_{\text{crit}} \approx .53$; the upper cluster is pruned at or left of the crossing and retained to the right. **Right: fixed mass, varying upper-cluster confidence.** With $r = .40$, the threshold and c_H cross at $c_{H,\text{crit}} \approx .64$; insufficient separation gives no pruning; at or above the crossing the upper cluster is pruned. Background colors indicate the cluster decision in each slice. These are illustrative slices, not assumptions made by LACS; $\tau = .20$ is chosen here to keep the transition away from the plot boundary, whereas experiments use the protocol-specific frozen or selected τ. Appendix A gives the full (r, Δ) phase diagram.

3 Interpreting the Unsigned Confidence Rule

The same unsigned score can have different update-level meanings. For binary cross-entropy, write $m_i = y_i F_i$, $y_i \in \{-1, +1\}$, and $c_i = \sigma(|F_i|)$. Then

$$\left| \frac{\partial \ell_i}{\partial F_i} \right| = \begin{cases} 1 - c_i, & m_i > 0, \\ c_i, & m_i < 0. \end{cases} \tag{10}$$

A confidently correct row has a small output derivative and is often redundant for the current update; a confidently wrong row keeps a large one and can dominate the next fit. The same unsigned score therefore serves *redundancy removal* in the first case and *influence trimming* in the second, without reading the label; Appendix G bounds the resulting gradient residual by $\sum_{D_+} L_i(1-c_i) + \sum_{D_-} L_i c_i$ over the two removed tails.

Which tail matters depends on the setting: at the inspected neural checkpoints all removed rows are correct and the gradient residual is 0.4–3.6%, whereas in tree trajectories confidently wrong rows are removed and later re-enter. Removing only the correct members fails severely on two datasets when the confident errors left behind dominate the next fit; the unified rule avoids those failures, although the aggregate one-sided contrast is unresolved (Appendix H).

4 Experiments: Sparse Work, Prediction, and Realized Speed

4.1 Overview

The seven settings of Section 1 share one selection rule but differ in what the retained set is used for; tree results use the companion LACS-T. The main text reports the settings that most directly answer the paper's two questions, how much expensive work is avoided and what happens to prediction: compact repeated fitting (4.2), tree boosting (4.3), a direct neural comparison with dynamic-pruning baselines (4.4), the rule's behavior on real trajectories (4.5), and the accounting that separates work from time (4.6); Table 1 previews them. The long-tailed and semi-supervised settings are boundary studies and are in the supplement with seed-level values, additional baselines, null results, and protocol details.

Table 1: **Four representative results from seven evaluated settings.** Work counters are reported separately from wall time. Multi-dataset SD is across dataset means; single-run cases report no SD. †: corrected $p < .05$ in the stated comparison family.

Setting	Work retained / saved	Reference → selected method	Statistical scope
Quadratic boosting, R2	replay rows ≈ 70% of full	error 23.36 ± 11.46 → 22.96 ± 11.27	$\Delta = -.401$ pp, CI $[-.721, -.081]$, post-hoc Holm $p \approx .008^{\dagger}$
Tree boosting (LACS-T), depth 2	same per-round K_t as GOSS	error 20.61 ± 11.00 → 20.27 ± 11.01	$\Delta = -.346$ pp, CI $[-.604, -.088]$, new-block Holm $p = .0147^{\dagger}$; LACS-T variant
RBF-SVM boosting	fit rows 3.000 → 2.121 M	error 24.936 → 24.702; wall 3400 → 2136 s	one common-endpoint replay; no multi-seed significance claim
CIFAR-10 neural	backward rows 2.496 → 1.776 M (28.9% fewer)	accuracy $89.82 \pm .20$ → $89.77 \pm .51$	three seeds; CI crosses zero; current two-pass wall is slower

Table 2: **Quadratic repeated fitting across 14 datasets.** Error is mean $\pm \mathrm{SD}_D$. † marks $p < .05$ under the reported post-hoc Holm correction of the six full-data comparisons; intervals are nominal, not simultaneous. The supplement reports all random-support controls and per-dataset results.

Weak learner	Full error	LACS error	Δ_{err}	95% t CI	$p_{\mathrm{H,full}}$
Quad R1	23.64 ± 11.30	23.34 ± 11.19	-0.294	$[-0.623, +0.035]$	0.281
Quad R2	23.36 ± 11.46	22.96 ± 11.27	$-0.401^{\dagger}$	$[-0.721, -0.081]$	0.008
Quad R4	23.19 ± 11.38	22.81 ± 11.23	$-0.374^{\dagger}$	$[-0.664, -0.084]$	0.014

Uncertainty conventions. For a single dataset, $\pm$ is the sample SD over seeds; for multi-dataset aggregates it is the SD across dataset means after averaging resplits (task heterogeneity, not seed noise). Paired intervals are nominal 95% t intervals for the mean difference with the dataset (or seed) as the unit. Sign-flip tests and Holm families are used only where recorded or explicitly recomputed (Appendix B); an interval that crosses zero is reported as unresolved, not as equivalence.

4.2 Selected fitting can improve prediction

The clearest result for the normalized rule comes from *compact repeated fitting*: boosting with small nonlinear weak learners on 14 tabular datasets with three resplits. A rank-R quadratic learner has raw predictor $b + w^{\top}x + \sum_{j=1}^{R} a_j (u_j^{\top} x)^2$ followed by a bounded output (denoted R1, R2, R4); one-hidden-layer MLP learners (H1, H2, H4) and all six families are reported in Appendix D. Before each round LACS selects the support from the current ensemble confidence; ordinary logistic fitting weights are kept on that support, and the boosting coefficient is estimated on it. Full-data and LACS procedures are tuned independently on validation data, so this tests the complete selected-training procedure rather than a membership-only intervention.

Quadratic R2 is the strongest full-data result (Table 2): mean error falls from 23.36 to 22.96%, a paired difference of $-.401$ pp with nominal CI $[-.721, -.081]$. R4 also improves under the same post-hoc correction; the MLP versus full-data comparisons are unresolved. The per-round count control is an exact-count random support at the LACS configuration, which removes the same number of rows per round: R2 wins on 13/14 datasets and passes the original six-learner Holm family ($p = .0012$), and H2, R1, and R4 pass it as well. This supports the complete LACS selected-fitting procedure against that random control. It does not isolate membership alone: the stage coefficient is estimated on each support, and each arm chooses its own validation stopping round.

A *replay* is the full recorded run; its *selected prefix* ends at the validation-chosen checkpoint. Over complete replays R2 fits about .70 of the full-training rows; at the selected predictor the pooled

Table 3: **Neural comparisons.** (a) Update-matched controls on CIFAR-10, five seeds, one host per seed: small-batch CE uses batch 91 for 19,764 steps (about the same examples and steps as LACS); random drop executes a uniformly random subset of LACS's per-batch size with LACS's weights. (b) Dynamic-pruning baselines on CIFAR-10, three seeds (Full CE and LACS values from that panel). (c) The same baselines on CIFAR-100, three seeds; every non-LACS arm in (c) is work-matched to LACS's backward fraction except InfoBatch-V. V: validation-selected; B/N: backward-matched or near-budget. Contrasts are LACS minus the row, paired by seed, nominal 95% t intervals.

Panel	Method	Test acc. (%)	Bwd. frac.	LACS − row [95% CI]
(a) C10, 5 seeds	Full CE	$89.66 \pm .29$	.998	$+.29\ [-.40, +.99]$
	LACS	$89.95 \pm .44$	.709	—
	Small-batch CE	$88.95 \pm .28$	.719	$\mathbf{+1.01}\ [\mathbf{+.57}, \mathbf{+1.44}]$
	Random drop	$88.77 \pm .28$	.693	$\mathbf{+1.19}\ [\mathbf{+.67}, \mathbf{+1.71}]$
(b) C10, 3 seeds	Full CE	$89.82 \pm .20$	.998	$-.05\ [-1.80, +1.70]$
	LACS	$89.77 \pm .51$	.711	—
	InfoBatch (V)	$89.75 \pm .21$	.814	$+.02\ [-1.34, +1.38]$
	InfoBatch (B)	$89.06 \pm .33$	.711	$+.71\ [-.79, +2.21]$
	RePB (N)	$89.46 \pm .05$	.731	$+.31\ [-.95, +1.57]$
	Selective-Backprop	$89.55 \pm .14$	.719	$+.22\ [-1.37, +1.81]$
(c) C100, 3 seeds	Full CE	$62.41 \pm .31$	.998	$-.17\ [-1.80, +1.46]$
	LACS	$62.24 \pm .46$	.870	—
	InfoBatch (V)	$61.74 \pm .67$	.925	$+.50\ [-2.29, +3.29]$
	InfoBatch (B)	$61.77 \pm .25$	.868	$+.47\ [-1.13, +2.07]$
	RePB (N)	$61.25 \pm .73$	.864	$+.99\ [-.46, +2.44]$
	Selective-Backprop	$61.76 \pm .80$	.863	$+.48\ [-2.29, +3.24]$
	RS2	$61.27 \pm .48$	.863	$\mathbf{+.96}\ [\mathbf{+.46}, \mathbf{+1.46}]$

prefix ratio is .641 but the mean of per-run ratios is .936, because stopping rounds and dataset sizes differ. Prefix wall time was not recorded, so we make no time-to-equal-quality claim (Appendix D).

4.3 Boosting as repeated subset selection

The tree campaign compares the fixed-threshold companion LACS-T (Section 2) with budget-matched GOSS-style selection and full XGBoost in one harness (Chen & Guestrin, 2016; Ke et al., 2017); it evaluates the companion, not the normalized cutoff.

At depth 2, budget-matched GOSS has mean error $20.61 \pm 11.00\%$ and LACS-T $20.27 \pm 11.01\%$, a paired change of $-.346$ pp with nominal CI $[-.604, -.088]$; on the newly added blocks the comparison survives the prespecified three-depth Holm correction ($p = .0147$). An independently tuned GOSS arm gives a difference of $-.386$ pp (nominal sign-flip $p = .031$) and is reported as a secondary comparison. Depths 1 and 3, full-data arms, a ten-resplit depth-one study, and per-dataset results are in Appendix E; the point is that repeated confidence selection can improve prediction at a controlled fitting budget.

4.4 Neural training: controls and dynamic-pruning baselines

Two questions are separated here. First, is the accuracy of LACS-trained networks due to *which* examples it keeps, or merely to the smaller effective batch and the number of optimizer updates? Second, how does LACS compare with recent dynamic-pruning methods under one training recipe? All runs use ResNet-32 with GroupNorm for 50 epochs on an 8,000-image test partition; Table 3 collects the three panels, and Appendix C gives seed values, paired intervals, scoring forwards, steps, and same-host wall times.

Selection, not update count. Panel (a) removes the confound that a per-batch selector keeps every optimizer step while dataset-level pruners take fewer. At about the same backward budget and the same number of updates, LACS is $+1.01$ pp above small-batch CE (95% CI $[+.57, +1.44]$) and $+1.19$ pp above a random subset of the same size trained with the same weights ($[+.67, +1.71]$),

positive in all five seeds; small-batch CE is itself .71 pp below full-batch CE. The accuracy of LACS-trained networks therefore comes from which rows are kept, not from more or smaller updates. Against full CE, LACS is $+.29$ pp with an interval that includes zero.

Baselines. On CIFAR-10 (panel b) LACS removes 29% of backward rows with accuracy within seed noise of full CE; it matches validation-selected InfoBatch while executing fewer backward rows and has the highest mean among the budget-matched arms, but none of those differences is resolved at three seeds. On CIFAR-100 (panel c) LACS removes 13% of backward rows at an accuracy within noise of full CE, has the highest mean among all budget-matched pruning arms, and is $+.96$ pp above RS2 ($[+.46, +1.46]$); the differences to work-matched InfoBatch ($+.47$), RePB ($+.99$) and Selective-Backprop ($+.48$) are positive but unresolved. Across both benchmarks LACS has the higher mean in every matched-work comparison with InfoBatch and Selective-Backprop, and we state exactly that: a consistently favorable point estimate, not statistical superiority. LACS pays a scoring forward in every neural run (Section 4.6).

Which removals carry the accuracy? A label-aware diagnostic separates the two tails of the LACS set (Section 3). In a separate same-host panel (Appendix C.2; its wall times are not comparable with Table 4), with backward budgets within 1.5% of each other, correct-only LACS (C-LACS, which removes only the currently correct members of the LACS set) reaches $90.15 \pm .28\%$, unified LACS $89.74 \pm .46\%$, and work-matched InfoBatch $89.25 \pm .23\%$; C-LACS exceeds InfoBatch by $+.896$ pp (95% CI $[+.119, +1.672]$, positive in all three seeds), while C-LACS and LACS are within seed noise of each other. C-LACS reads the label and is not the proposed method; the result is mechanism evidence that deterministic removal of confidently correct examples can beat stochastic low-loss thinning at an equal backward budget, not a superiority claim for LACS, and InfoBatch remains faster on this host (139 s versus 221–226 s; Section 4.6).

4.5 The rule on real training trajectories

Section 2 predicts three checkable behaviors: the retained count follows the shape of the confidence population rather than a schedule, the exact condition $M_\tau(c) > \rho n$ coincides with every no-pruning event, and same-mean batches of different shape receive different supports. All three hold on the CIFAR-10 trajectories (three seeds; per-epoch values in Appendix C.2). The mean retained fraction K_t/n starts near .89, falls to a minimum of .57 at epoch 39, and returns to .98 at epoch 50, while the fraction of batches with no pruning is zero through epoch 40 and .93 at epoch 50: LACS prunes most while the confidence population is spread out and stops pruning once it collapses into a flat high-confidence mass. Across all 480 logged raw batches, exactly the 108 batches with $M_\tau(c)/(\rho n) > 1$ are the no-pruning batches (agreement 1.000); the same identity holds for all 171 no-pruning rounds on the *pol* boosting trajectory and the single no-pruning round on *electricity*. Finally, two epoch-50 batches with mean confidences .9395 and .9411 have thresholds .9979 and 1.0002 and retained fractions .625 and 1.000: the arithmetic mean does not determine the decision, the shape does.

The late-training behavior is a deliberate conservatism with a cost: a homogeneous confident population is never pruned, which protects the data once the model is uniformly confident but forgoes savings exactly when most examples are redundant; a confidence floor or schedule could recover them (future work).

4.6 Work reduction versus wall-clock speed

Executed work and elapsed time can move in opposite directions. If a dense step costs a forward F plus a backward B, and an exact-current-batch two-pass implementation first scores the full batch and then trains a retained fraction r, then

$$C_{\text{full}} = F + B, \qquad C_{\text{2pass}} \approx F + r(F + B) + H, \tag{11}$$

where H collects selection and packing overhead. Under the illustrative dense-layer rule of thumb $B \approx 2F$ and $H = 0$, the two-pass path saves arithmetic only when $r < 2/3$; the CIFAR-10 run retains $r \approx .71$, so this model predicts about 4.5% *more* arithmetic than full CE. What this costs in time depends on the host (Table 4): on the RTX 3090 panel LACS is 27% slower than full CE, whereas on the RTX 4090 panel of Table 3(a) it is within noise of full CE (280 versus 289 s), because at this model size the step is not dominated by arithmetic. Executed rows and measured time are therefore reported separately.

Table 4: **Same rule, different systems outcomes.** Neural LACS removes backward-active examples but adds a full scoring forward; whether that costs time depends on the host. RBF-SVM fitting is expensive enough that fewer fit rows translate directly into elapsed-time savings; the replication runs eight dataset/split cells at frozen configurations (Appendix C.6).

Setting	Task metric: full → LACS	Expensive work: full → LACS	Wall: full → LACS
CIFAR-10 / ResNet-32 GN, RTX 3090, 3 seeds	acc. $89.82 \pm .20 \rightarrow 89.77 \pm .51$	bwd. $2.496 \rightarrow 1.776$ M	$312.9 \rightarrow 396.8$ s
CIFAR-10 / ResNet-32 GN, RTX 4090, 5 seeds	acc. $89.66 \pm .29 \rightarrow 89.95 \pm .44$	bwd. $.998 \rightarrow .709$ (fraction)	$289 \rightarrow 280$ s
Credit / RBF-SVM boosting (original case)	err. $24.936 \rightarrow 24.702$	fit rows $3.000 \rightarrow 2.121$ M	$3400 \rightarrow 2136$ s
RBF-SVM replication, 7 pruning cells	err. within $[-1.96, +.57]$ pp of full	fit rows .06–.63 of full	-30% to -94%

Why the neural implementation uses two forwards, and why removing them is not enough. To decide from the *current* batch confidence and physically omit backward rows, our implementation runs a no-gradient scoring forward, forms the support, then runs a gradient-enabled forward/backward on the retained examples only; masking losses on a dense autograd graph would not remove the dense backward kernels. The second forward is an execution choice, not a requirement of the rule, so we tested the obvious alternative: confidences cached from each example's previous visit, with no scoring forward. It does not help. Over five seeds on the 4090 host, cached scores do not reduce wall time (341–352 s versus 280 s for two-pass LACS), and a purely cached rule loses accuracy ($-.89$ pp, 95% CI $[-1.38, -.41]$) while an EMA-cached rule stays within noise only by retaining 84% of rows (Appendix C.4). At this scale the scoring forward is not the bottleneck; converting backward sparsity into time requires fused selective execution, which we have not built. A seed-0 pilot that keeps InfoBatch's dataset-level execution and replaces only its eligibility rule by the LACS cutoff runs with no scoring forward (Appendix C.7); its full seeds are pending.

A measured speedup without systems optimization. Repeated RBF-SVM fitting is expensive enough that the support reduction dominates selection overhead. In the original credit replay LACS fits 29.3% fewer rows and reduces end-to-end time by 37.2% while slightly lowering final error. A replication over eight dataset/split cells (bank-marketing, california, credit; frozen per-split configurations; Appendix C.6) reproduces the pattern: in the seven cells where LACS prunes, wall time falls by 30–94% relative to full fitting (mean -65%) with final error within $[-1.96, +.57]$ pp of full; in the one cell where the no-pruning condition holds throughout (california, split 0), LACS is a no-op and saves nothing. A random support of the same per-round size is as fast (mean -3% relative to LACS) but has higher error in five of seven cells, so the speed comes from the row reduction and the accuracy from which rows are kept.

4.7 Breadth and boundaries

The remaining settings test whether the rule transfers beyond the headline cases. In a neural execution matrix over eight dataset/architecture/schedule configurations, exact zeros remove 9.9–35.1% of backward rows while reproducing the dense LACS-weighted same-state gradient to fp64 precision for GroupNorm/LayerNorm architectures; BatchNorm is the deliberate counterexample. With CE and InfoBatch baselines now run for the completed configurations (three seeds), LACS is within $[-.44, +.13]$ pp of CE on SVHN, CIFAR-100, and CIFAR-100-LT (all intervals include zero), and above InfoBatch at its default pruning ratio on every completed configuration: $+.68$ pp on CIFAR-10/ViT-Tiny ($[+.32, +1.04]$), $+2.2$ pp on CIFAR-100 and $+3.5$ pp on CIFAR-100-LT (intervals exclude zero). InfoBatch there executes 11–16% less backward work than LACS and was not tuned per configuration, so these are frontier comparisons rather than matched-work ones (Appendix F). In Transformer language-model training (800 steps, three seeds), selective execution removes 11.5% of backward sequences at a validation NLL of 1.569 versus 1.567 for dense LACS and 1.520 for CE, and remains 4.5% slower than the optimized dense implementation. Long-tailed, semi-supervised, historical InfoBatch, label-noise, and one-sided studies are in Appendices F–I; none is pooled with the CIFAR-10 comparison as a performance-preservation result.

5 Related dynamic pruning and scope

Dynamic example selection. Our direct neural baselines are Selective-Backprop, RS2, RCAP, InfoBatch, and RePB (Jiang et al., 2019; Okanovic et al., 2024; Hassan et al., 2025; Qin et al.,

2024; Zhou et al., 2026a); Raju et al., SeTa, BLS, and OrderDP give further context (Raju et al., 2021; Zhou et al., 2025; 2026b; Jin et al., 2026), and DropSample, calibration-oriented pruning, and Online Batch Selection establish earlier confidence- or loss-based reselection (Yang et al., 2016; Patra et al., 2023; Loshchilov & Hutter, 2015). InfoBatch, the closest comparator, defines eligibility by a mean of cached losses, thins the eligible pool stochastically, rescales survivors, and returns to full data at the end; SeTa uses loss clusters and a scheduled window. LACS differs in its score (label-agnostic maximum confidence), its reference (an exponential quasi-arithmetic mean plus a margin, which responds to shape at a fixed mean confidence), deterministic removal, and a pruned fraction that is an outcome rather than a setting. In probability coordinates InfoBatch's mean loss is a geometric mean, so the two references weight opposite tails (Appendix K); empirical comparisons are in Appendix C.

Boosting and influence control. Weight trimming and influence trimming are direct precedents for omitting examples during weak-learner fitting (Friedman et al., 2000; Friedman, 2001). In particular, LogitBoost weights $p_i(1-p_i)$ do *not* require the class label at selection time: they are small for either confidently correct or confidently wrong predictions. LogitBoost trims a prescribed fraction of cumulative working-weight mass, whereas LACS thresholds each exponential-confidence contribution relative to its population average; the cutoff policy, not the unsigned ranking, is what is new here.

RHO-LOSS selects using reducible holdout loss (Mindermann et al., 2022); OHEM, Focal Loss, and Selective-Backprop emphasize hard examples (Shrivastava et al., 2016; Lin et al., 2017; Jiang et al., 2019). RobustBoost and ArchBoost limit extreme-error influence through modified potentials or losses (Freund, 2009; Li & Bradic, 2018). These methods motivate distinct comparisons rather than interchangeable guarantees. Appendix K also covers static pruning, gradient matching, and importance sampling.

What is new. LACS is neither the first dynamic pruning method nor the first to use a data-dependent population statistic, and label-free trimming has a LogitBoost precedent. Its distinctive object is a confidence cutoff whose *threshold and selected count move together* with the current near-maximum confidence mass; that structure yields the closed-form no-pruning condition, the mass/confidence geometry of Figure 1, and the trajectory behavior of Section 4.5. At an identical realized count the hard support coincides with a confidence quantile, so the contribution is the cutoff policy, not a new ranking.

Limitations. Confidence alone cannot identify label noise or semantic atypicality, so LACS is not an outlier detector, and the two-cluster analysis is explanatory rather than a generative assumption. The neural baseline comparisons rest on three seeds per benchmark and are unresolved against InfoBatch, RePB, and Selective-Backprop; the resolved neural results are the update-matched controls (five seeds) and the RS2 and default-ratio InfoBatch comparisons. The two-pass implementation is slower than InfoBatch on every host, and cached scores do not recover the time. The tree headline uses the fixed-threshold LACS-T, and the compact-learner search includes $\rho \geq 1$, outside the domain of the nonempty-support guarantee. A tuned fixed budget beats the normalized cutoff on one dataset, the single RBF-SVM timing case needs replication, and the rule forgoes savings late in training by design (Section 4.5). These limits call for fused selective execution, more seeds on the baseline panels, normalized-rule tree experiments, and a multi-dataset adaptive-versus-fixed comparison.

6 Conclusion

LACS turns label-agnostic confidence into a distribution-sensitive sparse subset whose threshold and size adapt jointly to score separation and upper-end mass, with an exact no-pruning condition and a bounded pruned fraction, and it behaves on real trajectories as the analysis predicts. Across seven settings it reduces backward or fitting work by 10–35%, keeps CIFAR-10 and CIFAR-100 accuracy within seed noise of full training while beating update-matched small-batch and random training by about 1 pp, and improves prediction in several repeated-fitting settings. Whether saved work becomes saved time is a systems question: yes when fitting dominates overhead, as in RBF-SVM boosting; not yet in our neural implementation, with or without cached scores.

AI USE STATEMENT

In this work, we used generative AI tools for several research and writing tasks. For tasks requiring disclosure under the ICLR 2027 policy, the tools assisted with developing and refining conceptual and theoretical framing; formulating and checking mathematical claims and proof steps; proposing and refining hypotheses; designing and reviewing experimental methodology; implementing and reviewing analysis and training code; translating between Korean and English; interpreting experimental results; and creating small synthetic diagnostic examples used for mechanism checks. We also used generative AI for literature search and summarization, identifying related work and research gaps, suggesting experimental parameters, drafting and editing manuscript text, organizing the paper, formatting references and tables, and creating or modifying scientific figures. We did not use generative AI to generate or alter the benchmark datasets or their labels, nor as a substitute for the reported empirical measurements. AI-assisted code, mathematical arguments, citations, tables, figures, and numerical claims were checked against executable code, experiment logs, retained source data, and primary literature before inclusion. The authors reviewed all AI-assisted work and take responsibility for the final content of this submission.

REPRODUCIBILITY STATEMENT

The appendix specifies selection contracts, parameter ranges, evaluation protocols, uncertainty calculations, and cost horizons. Numerical tables and plot assets are supplied in the anonymous manuscript-source package. These materials reproduce the document, not all training experiments; quantities absent from retained records are identified explicitly.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. URL https://arxiv.org/abs/1607.06450.

Joseph K. Bradley and Robert E. Schapire. Filterboost: Regression and classification on large datasets. In *Advances in Neural Information Processing Systems*, volume 20, 2007. URL https://proceedings.neurips.cc/paper/2007/hash/072b030ba126b2f4b2374f342be9ed44-Abstract.html.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/621461af90cadfdaf0e8d4cc25129f91-Abstract.html.

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/2f37d10131f2a483a8dd005b3d14b0d9-Abstract.html.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. doi: 10.1145/2939672.2939785. URL https://arxiv.org/abs/1603.02754.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30, 2006. URL https://www.jmlr.org/papers/v7/demsar06a.html.

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*, 2021. URL https://arxiv.org/abs/2010.11929.

Yoav Freund. A more robust boosting algorithm. *arXiv preprint arXiv:0905.2138*, 2009. URL https://arxiv.org/abs/0905.2138.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. doi: 10.1006/jcss.1997.1504. URL https://doi.org/10.1006/jcss.1997.1504.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 28(2):337–407, 2000.

Jerome H. Friedman. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. doi: 10.1214/aos/1013203451.

Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38 (4):367–378, 2002. doi: 10.1016/S0167-9473(01)00065-2. URL https://doi.org/10.1016/S0167-9473(01)00065-2.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems*, volume 35, pp. 507–520, 2022. URL https://papers.neurips.cc/paper_files/paper/2022/file/0378c7692da36807bdec87ab043cdadc-Paper-Datasets_and_Benchmarks.pdf.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Atif Hassan, Swanand Khare, and Jiaul H. Paik. RCAP: Robust, class-aware, probabilistic dynamic dataset pruning. In *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pp. 1648–1662. PMLR, 2025. URL https://proceedings.mlr.press/v286/hassan25a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. URL https://arxiv.org/abs/1512.03385.

Bulat Ibragimov and Gleb Gusev. Minimal variance sampling in stochastic gradient boosting. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.neurips.cc/paper/9645-minimal-variance-sampling-in-stochastic-gradient-boosting.pdf.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456. PMLR, 2015. URL https://proceedings.mlr.press/v37/ioffe15.html.

Angela H. Jiang, Daniel L.-K. Wong, Giulio Zhou, David G. Andersen, Jeffrey Dean, Gregory R. Ganger, Gauri Joshi, Michael Kaminsky, Michael Kozuch, Zachary C. Lipton, and Padmanabhan Pillai. Accelerating deep learning by focusing on the biggest losers. *arXiv preprint arXiv:1910.00762*, 2019. URL https://arxiv.org/abs/1910.00762.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 2304–2313, 2018. URL https://proceedings.mlr.press/v80/jiang18c.html.

Chenhan Jin, Shengze Xu, Qingsong Wang, Fan Jia, Dingshuo Chen, and Tieyong Zeng. Orderdp: A theoretically guaranteed lossless dynamic data pruning framework. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=e77QyyRQPz.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2525–2534. PMLR, 2018. URL https://proceedings.mlr.press/v80/katharopoulos18a.html.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

Krishnateja Killamsetty, Durga S, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. GRAD-MATCH: Gradient matching based data subset selection for efficient deep model training. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5464–5474. PMLR, 2021a. URL https://proceedings.mlr.press/v139/killamsetty21a.html.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. GLISTER: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8110–8118, 2021b. URL https://arxiv.org/abs/2012.10630.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://cave.cs.toronto.edu/kriz/cifar.html.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23, pp. 1189–1197, 2010. URL https://papers.nips.cc/paper_files/paper/2010/hash/e57c6b956a6521b28495f2886ca0977a-Abstract.html.

Yann LeCun, L'eon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. URL https://doi.org/10.1109/5.726791.

Alexander Hanbo Li and Jelena Bradic. Boosting in the presence of outliers: Adaptive classification with nonconvex loss functions. *Journal of the American Statistical Association*, 113(522):660–674, 2018. doi: 10.1080/01621459.2016.1273116. URL https://doi.org/10.1080/01621459.2016.1273116.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017. doi: 10.1109/ICCV.2017.324. URL https://openaccess.thecvf.com/content_iccv_2017/html/Lin_Focal_Loss_for_ICCV_2017_paper.html.

Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. *arXiv preprint arXiv:1511.06343*, 2015. URL https://arxiv.org/abs/1511.06343.

Andr'e Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1614–1623. PMLR, 2016. URL https://proceedings.mlr.press/v48/martins16.html.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Soren Mindermann, Jan M. Brauner, Muhammed T. Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Holtgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15630–15649. PMLR, 2022.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6950–6960. PMLR, 2020. URL https://proceedings.mlr.press/v119/mirzasoleiman20a.html.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. URL https://research.google/pubs/reading-digits-in-natural-images-with-unsupervised-feature-learning/.

Patrik Okanovic, Roger Waleffe, Vasilis Mageirakos, Konstantinos E. Nikolakakis, Amin Karbasi, Dionysios Kalogerias, Nezihe Merve Gürel, and Theodoros Rekatsinas. Repeated random sampling for minimizing the time-to-accuracy of learning. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JnRStoIuTe.

Thomas Parnell, Andreea Anghel, Malgorzata Lazuka, Nikolas Ioannou, Sebastian Kurella, Peshal Agarwal, Nikolaos Papandreou, and Haralampos Pozidis. SnapBoost: A heterogeneous boosting machine. *arXiv preprint arXiv:2006.09745*, 2020. URL https://arxiv.org/abs/2006.09745.

Rishabh Patra, Ramya Hebbalaguppe, Tirtharaj Dash, Gautam Shroff, and Lovekesh Vig. Calibrating deep neural networks using explicit regularisation and dynamic data pruning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1541–1549, 2023. URL https://openaccess.thecvf.com/content/WACV2023/html/Patra_Calibrating_Deep_Neural_Networks_Using_Explicit_Regularisation_and_Dynamic_Data_WACV_2023_paper.html.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2107.07075.

Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Zhaopan Xu, Daquan Zhou, Lei Shang, Baigui Sun, Xuansong Xie, and Yang You. InfoBatch: Lossless training speed up by unbiased dynamic data pruning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/ad1d7a4df30a9c0c46b387815a774a84-Abstract-Conference.html.

Ziheng Qin, Hailun Xu, Wei Chee Yew, Qi Jia, Yang Luo, Kanchan Sarkar, Danhui Guan, Kai Wang, and Yang You. Info-coevolution: An efficient framework for data model coevolution. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50384–50396. PMLR, 2025. URL https://proceedings.mlr.press/v267/qin25h.html.

Ravi S. Raju, Kyle Daruwalla, and Mikko Lipasti. Accelerating deep learning with dynamic data pruning. *arXiv preprint arXiv:2111.12621*, 2021. URL https://arxiv.org/abs/2111.12621.

Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/2ba61cc3a8f44143e1f2f13b2b729ab3-Abstract.html.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 761–769, 2016. doi: 10.1109/CVPR.2016.89. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Shrivastava_Training_Region-Based_Object_CVPR_2016_paper.html.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2206.14486.

Mariya Toneva, Alessandro Sordoni, R'emi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1812.05159.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.03762.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision*, 2018. doi: 10.1007/978-3-030-01261-8_1. URL https://arxiv.org/abs/1803.08494.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. URL https://arxiv.org/abs/1708.07747.

Weixin Yang, Lianwen Jin, Dacheng Tao, Zecheng Xie, and Ziyong Feng. Dropsample: A new training method to enhance deep convolutional neural networks for large-scale unconstrained handwritten chinese character recognition. *Pattern Recognition*, 58:190–203, 2016. doi: 10.1016/j.patcog.2016.04.007. URL https://doi.org/10.1016/j.patcog.2016.04.007.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference*, pp. 87.1–87.12, 2016. doi: 10.5244/C.30.87. URL https://bmva-archive.org.uk/bmvc/2016/papers/paper087/index.html.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. FlexMatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2110.08263.

Qing Zhou, Junyu Gao, and Qi Wang. Scale efficient training for large datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20458–20467, 2025. URL https://arxiv.org/abs/2503.13385.

Qing Zhou, Tao Yang, Bingxuan Zhao, Hongyuan Zhang, Junyu Gao, and Qi Wang. Inconsistency biases in dynamic data pruning. In *International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=Zw1Uw7u6Su.

Qing Zhou, Bingxuan Zhao, Tao Yang, Hongyuan Zhang, Junyu Gao, and Qi Wang. Batch loss score for dynamic data pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026b. URL https://arxiv.org/abs/2604.04681.

A MATHEMATICAL PROPERTIES AND IMPLEMENTATION DETAILS

A.1 THRESHOLD, CARDINALITY, AND NO-PRUNING REGION

For $q_i = e^{c_i/\tau}/\overline{e^{c/\tau}}$ and $\kappa = 1/\rho$, $u_i = 0$ is equivalent to $q_i \geq \kappa$. Taking logarithms gives Eq. (4). If D is the removed set, $n = \sum_i q_i \geq \sum_{i \in D} q_i \geq \kappa |D|$, hence $|D|/n \leq \rho$. For $0 < \rho < 1$, at least one example remains; moreover,

$$\bar{u} = \frac{1}{n} \sum_i [1 - \rho q_i]_+ \geq 1 - \rho > 0,$$

so the continuous-weight denominator is nonzero. All retained coefficients are nonnegative. By Jensen, $\tau \log \overline{e^{c/\tau}} \geq \bar{c}$; therefore $\max c - \bar{c} < \tau \log \kappa$ implies $\max c < T(c)$ and no pruning. The stronger-to-check range condition $\max c - \min c < \tau \log \kappa$ is also sufficient. Equal scores give $q_i = 1$ and $u_i = 1 - \rho$, hence $w_i = 1$.

These are conditional statements about the score operator. For $\rho \geq 1$, a flat population has an empty support. Nonflat populations can still have nonempty support; a failed run cannot be inferred from its parameter alone. Some historical boosting searches deliberately included this larger domain and stopped on degenerate support rather than replacing it with uniform weights. Their performance does not validate the conservative-domain guarantee.

A.2 EXACT MASS CRITERION: WHAT MAKES PRUNING STOP?

Let $\pi_i = e^{c_i/\tau} / \sum_j e^{c_j/\tau}$ denote the normalized exponential mass *across examples*, not a predicted class probability. Since $q_i = n\pi_i$ and $\max_i q_i = n/M_\tau$, the following conditions are equivalent:

$$D = \varnothing \iff \max_i q_i < \kappa \iff \max_i \pi_i < \frac{\kappa}{n} \iff M_\tau > \frac{n}{\kappa} = \rho n. \tag{12}$$

The inequality is strict because equality produces a zero coefficient and removes at least one maximum. The total $\sum_i \pi_i = 1$ (or $\sum_i q_i = n$) cannot itself diagnose concentration. An unnormalized total is not a meaningful absolute trigger either: replacing c_i by $c_i + a$ changes it while leaving the mask unchanged.

For a particular example, the same rule can be written as

$$i \in D \iff \sum_j e^{(c_j - c_i)/\tau} \leq \rho n.$$

Thus removal depends on its exponential gaps to *all* candidates. The criterion is closest to multiplicative exceedance of an empirical exponential mean, or equivalently a softmax-concentration threshold. It is not a density-based outlier test, a calibrated tail probability, or a false-discovery-rate guarantee. No inference about label correctness follows from crossing it.

Table 5: Exact illustrative decisions for $n = 100$, $\rho = .7$, $\tau = .05$. No-pruning requires $M_\tau > 70$. These constructed distributions are not additional training evidence. The fixed-quantile comparator always removes exactly ten rows.

Confidence population	M_τ	LACS removed	Top-10%	$c > .74$
100 at .95	100.00	0	10	100
90 at .70; 10 at .71	83.69	0	10	0
90 at .70; 10 at .75	43.11	10	10	10
40 at .70; 60 at .75	74.72	0	10	60
40 at .70; 60 at .80	65.41	60	10	60
20 at .70; 80 at .90	80.37	0	10	80

The second and third rows preserve confidence ordering but change the gaps. The third and fourth use the same two values but change the upper-cluster mass. The fourth and fifth show why a common high-confidence group is not categorically protected: sufficient separation can still trigger removal below the cardinality ceiling. A mass such as M_τ is an algebraic control variable, not evidence that the resulting subset is optimal.

Same mean, different shape. Let $\beta = 1/\tau$. The empirical log-mean-exp is the scaled cumulant-generating function

$$\mathrm{LME}_\beta(c) = \frac{1}{\beta} \log \left(\frac{1}{n} \sum_i e^{\beta c_i} \right).$$

Because log-sum-exp is symmetric and convex, it is Schur-convex: if two finite score vectors have the same arithmetic mean and c majorizes c', then $\mathrm{LME}_\beta(c) \geq \mathrm{LME}_\beta(c')$ for every $\beta > 0$. Thus a pure mean boundary is identical for same-mean populations, while the LACS boundary can rise when the population becomes more dispersed. Expanding the empirical cumulant-generating function around $\beta = 0$ gives

$$\mathrm{LME}_\beta(c) = \mu + \frac{\beta}{2}\sigma^2 + \frac{\beta^2}{6}\kappa_3 + \frac{\beta^3}{24}\kappa_4 + O(\beta^4),$$

where κ_m are empirical cumulants. The variance term is always nonnegative; positive third cumulant gives an additional local upper-tail correction. This is an interpretation of the boundary, not the rule used numerically.

For example, at $\tau = .2$, the populations $(.5, .5, .5, .5)$ and $(.2, .4, .6, .8)$ both have mean $.5$, but their LME values are $.500$ and $.611$. A deterministic cutoff based only on the mean places the boundary at the same location for both, while the LACS population reference changes. This should not be paraphrased as "pointiness": concentration at the mean lowers LME toward the mean, whereas dispersion and upper-tail shape can raise it. Nor does a higher LME alone imply less pruning, because the upper scores themselves can move relative to the adaptive boundary; Eq. (13) gives the exact two-cluster decision.

A.3 Invariances, sensitivity, and quantile equivalence

A common additive score shift multiplies numerator and denominator by the same constant, leaving q unchanged. Replicating every score the same number of times also preserves q and the selected fraction. Neither property means that scores may be arbitrarily recalibrated: a nonlinear monotone map changes gaps and generally changes the cutoff. A confidence quantile, in contrast, depends only on ordering (and a specified tie rule).

For $n > 1$, q_i is strictly increasing in c_i for fixed other scores, so S is a confidence lower-level set. If the realized count is K and no score is tied across the boundary, S is precisely the set of the K lowest scores. Thus a matched-K confidence-ranker experiment cannot distinguish LACS's hard support from quantile selection. It is a valid identity check, not an independent accuracy baseline. A fixed quantile over time differs because it sets K in advance.

The threshold is 1-Lipschitz in the infinity norm: if $\|c - c'\|_\infty \leq \epsilon$, bounding the exponentials by $e^{\pm\epsilon/\tau}$ yields $|T(c) - T(c')| \leq \epsilon$. This bounds threshold motion, not the number of membership flips: many scores can lie close to the boundary. Also,

$$\frac{\partial T}{\partial c_j} = \frac{e^{c_j/\tau}}{\sum_i e^{c_i/\tau}} > 0.$$

Raising one extreme score raises the cutoff for the other examples and cannot newly prune an unchanged survivor. It can instead prevent other high-score examples from being pruned. This masking effect is a reason not to claim robust outlier detection from self-normalization.

A.4 Separation, mass, and parameter roles

For scores a and $a + \Delta$, with the latter occurring at fraction r, factoring out $e^{(a+\Delta)/\tau}$ gives

$$q_H = \frac{1}{r + (1-r)e^{-\Delta/\tau}}, \qquad q_H \geq \kappa \iff r < \rho \text{ and } \frac{\Delta}{\tau} \geq \log \frac{1-r}{\rho - r}. \tag{13}$$

The low group has $q_L = [(1-r) + re^{\Delta/\tau}]^{-1} \leq 1$, so it is retained when $\kappa > 1$. The high group is removed if and only if $r + (1-r)e^{-\Delta/\tau} \leq \rho$. This is possible for finite positive gaps exactly when $r < \rho$, and rearrangement yields the stated boundary.

Table 6: Two-level examples at $\rho = .7$. These are exact operator calculations, not training measurements. A quantile is assumed to remove exactly 10% with random tie-breaking.

High-group fraction r	Minimum Δ/τ	LACS removal above boundary	Quantile recall within high group
.01	.361	1%	100%
.10	.405	10%	100%
.40	.693	40%	25%
.60	1.386	60%	16.7%
.70 or more	no finite boundary	0% (two-level case)	$.1/r$

At $r < .1$, a fixed 10% quota also removes some bulk; its bulk removal rate is $(.1 - r)/(1 - r)$. At $r > .1$, it removes only a fraction $.1/r$ of the high group. These formulas avoid treating a tied group's presence in the selected set as 100% recall.

Increasing κ at fixed scores makes pruning more conservative and lowers the ceiling $1/\kappa$. Temperature changes the concentration of normalized exponentials. As $\tau \to \infty$, $q_i \to 1$ and the conservative-domain selector retains all examples. As $\tau \to 0$, a unique maximum absorbs the normalized mass; with $n > \kappa$, eventually that maximum alone is removed. Intermediate temperatures can therefore remove more examples than either limiting case. A Gaussian-score calculation illustrates, rather than proves universality of, this nonmonotonicity: for an idealized unbounded score population with $a = \sigma/\tau$,

$$\Pr\{c \geq T\} = 1 - \Phi\left(\frac{a}{2} + \frac{\log \kappa}{a}\right),$$

which is maximized at $a = \sqrt{2 \log \kappa}$. At $\kappa = 1/.7$ the maximum is approximately .199. Predictive probabilities are bounded, so this Gaussian model is only a shape calculation, not a calibrated probability model or a universal pruning-rate limit.

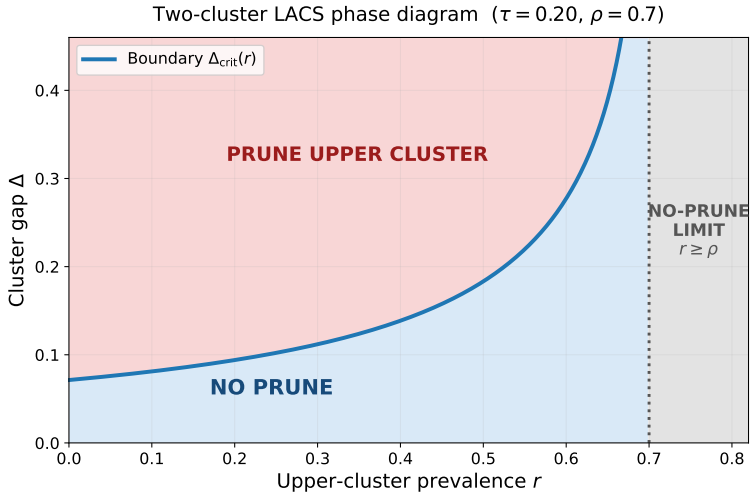


Figure 2: **Illustrative two-cluster phase diagram.** Above the boundary, LACS prunes the upper cluster; below it, LACS does not prune. For $r \geq \rho$, the two-cluster model lies in a no-prune limit because a tied upper group of that mass cannot cross the finite-gap cutoff. The region labels use the same parameters as Figure 1.

A.5 Exact selective execution and its limits

At a fixed state, detaching w gives Eq. (3) by dropping zero terms from a finite sum. Its implementation requires unchanged retained activations and Jacobians, original candidate normalization, and matched stochastic realizations. Batch-independent normalization such as LayerNorm or GroupNorm permits this decomposition; BatchNorm changes a retained example's computation when

other examples are removed. Augmentation, dropout, contrastive coupling, and batch-level losses require separate care. A fresh scoring forward can also change the state if running statistics or other mutable buffers are updated.

This is an exact identity for the frozen weighted surrogate $\widetilde{L}_t(\theta) = n^{-1} \sum_i w_i(\theta_t)\ell_i(\theta)$. Recomputing and detaching weights every step does not in general equal differentiating $n^{-1} \sum_i w_i(\theta)\ell_i(\theta)$, which includes derivatives through w. The neural experiments test the former, not a claimed globally conservative vector field. Different floating-point reduction orders can perturb multi-step trajectories even when same-state gradients agree to numerical precision.

B Evaluation protocol and inference

B.1 Reporting means and standard deviations

For a single-dataset experiment with n_s seeds, we report $\bar{a} \pm s_a$, with the unbiased sample SD s_a (denominator $n_s - 1$). Paired inference first forms $d_s = a_s - b_s$ and uses $\bar{d} \pm t_{.975,n_s-1} s_d/\sqrt{n_s}$. The SD of paired differences is not the difference of the two marginal SDs. Neither $\bar{a} \pm s_a$ nor its overlap with another method's interval is a significance test.

For multi-dataset studies, let d_{js} be the paired difference in dataset j and split s. We first average splits to obtain $\bar{d}_j$, then compute a macro mean and inference across datasets. In the compact study the independent aggregation unit is 14 dataset means, not 42 resplits; in the widened tree study it is also 14 means, not 140. Raw macro-score SDs describe variation across task difficulty. Appendix tables with one dataset per row instead display within-dataset split SDs. We make no probabilistic claim that this finite collection is a random sample of all tasks. Dataset-level aggregation and multiple-comparison issues have long been emphasized in classifier evaluation (Demšar, 2006); our particular t and sign-flip procedures are specified here rather than attributed to that reference.

The exact sign-flip statistic enumerates all 2^{14} sign assignments and compares the absolute mean with the observed absolute mean. It requires the corresponding null symmetry/exchangeability assumption. The t procedure instead relies on a reasonable model for the mean paired difference. The two can disagree; both are retained where available rather than selecting the more favorable test. The $n = 3$ neural exact sign-flip test has coarse resolution and cannot establish a two-sided $p < .05$ even with all signs favorable.

B.2 Comparison families and significance markers

The original compact primary family consists of six LACS–random comparisons. Its Holm-adjusted permutation values are reported with $*$. Full-data contrasts were secondary; the added † uses a *post-hoc* Holm correction across the six full-data permutation values. That correction is calculated from the source's four-decimal p values, so its displayed precision is limited. It does not retroactively make the full-data family primary or adjust all historical research choices. Nominal 95% t intervals remain nominal after a p-value correction.

The tree expansion used a pilot-triggered design. Its recorded correction is across three depths on new comparison blocks only. Pooled results that include the initial screen are descriptive; the new-only depth-2 result passes its corrected threshold. Independently tuned GOSS comparisons and expanded-resplit AdaBoost comparisons are secondary. The ten-resplit study expanded from five after intermediate values existed, and prior datasets were reused. Its uncertainty is therefore not described as a fully untouched confirmatory design.

All principal neural comparisons have only three seeds. We do not equate failure to reject a difference with evidence of negligible degradation. A future noninferiority claim would need a prespecified practical margin, appropriate test, and sufficient independent replication. We do not add such a margin after observing these results.

Dataset and architecture sources. CIFAR-10/100 originate from Krizhevsky (2009); MNIST, Fashion-MNIST, and SVHN are attributed to LeCun et al. (1998); Xiao et al. (2017); Netzer et al. (2011). The residual, wide-residual, attention, and vision-transformer components follow the architecture families of He et al. (2016); Zagoruyko & Komodakis (2016); Vaswani et al. (2017);

Dosovitskiy et al. (2021). LayerNorm and GroupNorm are attributed to Ba et al. (2016); Wu & He (2018). These citations identify data/model origins, not identical preprocessing, pretrained weights, or full original recipes; our variants and splits are specified below.

B.3 DATA, HYPERPARAMETERS, AND REUSE

Table 7: **Which implementation does each result test?** A common confidence-selection principle does not imply identical coefficients or cutoff policies. Guarantees for $0 < \rho < 1$ apply only in that domain.

Setting	Support rule	Retained weights / domain
midrule Neural selective training	LACS, minibatch scores	Continuous detached weights; $\rho = .7$
Compact weak learners	LACS, ensemble confidence	Ordinary logistic fitting weights; searched $0 < \rho < 4$
Tree/GOSS comparisons	LACS-T, fixed $\lvert F \rvert$ cutoff	Unit fitting weights; not the normalized cutoff
RBF-SVM case	LACS, ensemble confidence	Native LACS weights; fixed $\rho \simeq .864$
Semi-supervised learning	LACS, cached unlabeled pool	FlexMatch loss; admissible HPO $0 < \rho < 1$
bottomrule		

Neural baseline panel. CIFAR-10 uses ResNet-32 with GroupNorm(8), 50 epochs, batch 128, SGD with learning rate .1, momentum .9 and weight decay 5×10^{-4}, five warmup epochs followed by cosine decay. Validation consists of 2,000 stratified official-test images, and the remaining 8,000 form the reporting test partition. Seed zero is used for validation selection and seeds 1–3 for the frozen panel. Source JSONs also contain intermediate test diagnostics at epochs 10 and 25. The reported endpoint is the fixed final epoch; this is not a claim that the test set was accessed only once. The benchmark is therefore not directly comparable with published 10,000-image CIFAR-10 scores.

LACS fixes $\rho = .7, \tau = .05$. RS2 searches retained fractions $\{.3, .5, .712\}$ and compresses the host recipe's learning-rate schedule over executed steps. RCAP searches the same fractions and $\beta \in \{1, 3\}$; the port follows the paper's class allocation and uses loss clipping at the maximum epoch-zero loss. Selective-Backprop searches $\beta \in \{.4, 1, 2, 3\}$, starts with a full epoch, and keys learning rate to cumulative backward examples. Frozen choices are RS2 .712, RCAP .712/$\beta = 3$, and SB $\beta = .4$. These adaptations are stated so that method labels do not imply unmodified upstream training pipelines. InfoBatch uses the same base trainer, a seed-zero validation search over retained-easy-pool parameters $\{.3, .5, .7\}$, and a frozen choice of .7 with a final full-data phase ($\delta = .875$). Its realized nominal backward fraction is .8137, not .7. The three InfoBatch seeds were retrieved after the original baseline panel; all full-CE replay values and the available LACS replay values reproduce the earlier frozen accuracy records. The same-host three-seed LACS and CE replays now reproduce the original prediction values. Their completed timing series and separate work-calibrated InfoBatch setting are reported in Appendix C.9. RePB comparisons use the same data/model/optimizer recipe but a different execution host, as detailed there.

Tree panels. The tree cache contains numeric binary derivatives of 14 tabular tasks associated with the benchmark studied by Grinsztajn et al. (2022). The producer reads cached arrays and uses a binary target conversion. Training size is $\lfloor \min(.7n, 10000) \rfloor$; 30% of the remaining rows are validation and the rest test. These are cached derivatives, not a claim to reproduce every original task's published preprocessing or split protocol. Cached transformations and examples may have been accessed before the present split. In the resplit study the same underlying examples are repartitioned, so frozen hyperparameters do not imply an untouched holdout population.

The original tree/GOSS campaign uses 200 shared configuration draws and a 500-round cap. The separate widened depth-one campaign uses 200 draws and a 2,000-round cap. Its recorded shared search is learning rate log-uniform $[10^{-3}, 3]$, row and feature subsampling uniform $[.5, 1]$, minimum child weight rounded log-uniform $[1, 100]$, and leaf regularization log-uniform $[1, 4]$. LACS-T searches ϑ log-uniform $[.05, 20]$; tuned random searches its retained fraction uniformly in $[.1, 1]$. The standard full-data arm may itself use XGBoost's built-in subsampling; "full" means no additional LACS filter. Configuration and round are validation-selected. The widened resplit configurations and round counts are then frozen. The AdaBoost family (Freund & Schapire, 1997) supplies

the external stump baseline: the implemented SAMME variant uses stump learners, learning rates $\{.1, .5, 1\}$ and a 2,000-estimator cap, not the same 200-draw search budget.

Compact nonlinear fitting. The 14 datasets are credit, california, bank-marketing, heloc, MiniBooNE-10k, HIGGS-20k, Diabetes130US, electricity, MagicTelescope, house-16H, pol, eye-movements, jannis, and credit-card default. This is not identical to the tree panel: the high-dimensional Bioresponse task is omitted for runtime and HIGGS is included. Size duplicates are not counted as different tasks. Stage one uses split zero; the final panel uses resplits 1–3. Six learner configurations are promoted; MLP width eight was screened but not promoted under the recorded rule.

MLP raw output is $a^\top \tanh(Wx + b) + c$, with $dH + 2H + 1$ parameters. Quadratic raw output is $b + w^\top x + \sum_{r=1}^{R} a_r (u_r^\top x)^2$, with $d + Rd + R + 1$ parameters. Both are followed by $\tanh$. Full-batch Adam fits weighted logistic loss with gradient clipping at 10. The post-smoke grid is learning rate $\{.003, .01, .03\}$, weight decay $\{0, 10^{-4}, 10^{-3}\}$, and 100 or 300 optimizer steps. The initial 10/30-step smoke undertrained MLPs; this grid was amended before screening. Each cell evaluates 32 paired candidates and at most 100 boosting rounds, with shared initialization streams.

In historical experiment logs, the sign convention is $\lambda := -\rho$; the current manuscript uses $\rho > 0$ for the normalized rule. The inherited selector search draws λ uniformly in $(-4, 0)$, τ log-uniform in $[10^{-3}, 200]$, shrinkage log-uniform in $[10^{-3}, 1]$, and warmup from $\{0, 2, 5, 10, 20\}$. Thus some candidate selectors are outside $0 < \rho < 1$. Empty supports and degenerate fits are stopped and recorded, not silently changed to all-row fitting. Ordinary logistic fitting weights are normalized over retained rows. The stage coefficient uses the retained-set weighted sign error; it is not asserted to be an exact line search for a real-valued $\tanh$ weak learner. LACS and full training choose their own configuration and round on validation. The random control uses LACS's configuration and per-round counts, but chooses its own validation round. This changes both the fitted weak learner and its coefficient and may yield different selected-prefix budgets.

C Neural results and baseline coverage

C.1 Raw confirmation and work accounting

Table 8: **CIFAR-10/GN direct pruning baselines, three seeds.** Accuracy and available wall times are mean $\pm \mathrm{SD}_s$. V: validation-selected; B: backward-matched; N: near-backward budget. Wall times are from the four-arm InfoBatch host only; a dash means no comparable three-seed wall series.

Method	Test acc. ↑	Bwd. (M)	Extra fwd. (M)	Steps	Wall (s) ↓
Full CE	89.82 ± 0.20	2.496	0.000	19,500	312.9 ± 0.5
LACS	89.77 ± 0.51	1.776	2.496	19,500	396.8 ± 3.4
InfoBatch (V)	89.75 ± 0.21	2.034	0.000	15,893	271.6 ± 1.8
InfoBatch (B)	89.06 ± 0.33	1.776	0.000	13,876	244.5 ± 0.8
RePB (V)	89.33 ± 0.19	2.041	0.000	15,948	—
RePB (N)	89.46 ± 0.05	1.827	0.000	14,274	—
RCAP	89.11 ± 0.08	1.779	0.050	13,900	—
RS2	88.90 ± 0.12	1.779	0.000	13,900	—
Selective-Backprop	89.55 ± 0.14	1.795	2.446	14,047	—

Table 9 preserves all three-seed prediction series. Positive signs across three seeds do not by themselves establish superiority. The new comparisons extend the earlier frozen neural panel; they are not additional independent LACS seeds.

The nominal stream has 50,000 images over 50 epochs (2.5 M rows), but dropping incomplete *training* batches yields 2.496 M full-training rows. LACS's nominal backward fraction .7102 is .7114 relative to actual full training. Absolute counters avoid this ambiguity. LACS has 2.496 M extra scoring-forward rows; RCAP has .050 M, RS2 zero, and SB 2.446 M. Training-forward rows equal backward rows in the four-arm InfoBatch panel. Matching backward counts does not match these forwards or optimizer steps.

Table 9: CIFAR-10 final test accuracy by seed, fixed validation/test partition and network recipe. V/B/N are defined in Table 8. RePB reuses the InfoBatch-host LACS/CE series; exact cross-host prediction reproduction is checked only at seed 1.

Method	Seed 1	Seed 2	Seed 3
Full CE	89.7750	89.6500	90.0375
LACS	89.9000	90.2000	89.2125
InfoBatch (V)	89.9875	89.5875	89.6750
InfoBatch (B)	88.6875	89.3250	89.1750
RePB (V)	89.5375	89.1750	89.2875
RePB (N)	89.4000	89.5000	89.4750
RCAP	89.0625	89.2000	89.0625
RS2	88.8000	89.0250	88.8625
Selective-Backprop	89.5125	89.4375	89.7000

Table 10: Paired accuracy uncertainty for the main CIFAR-10 baseline table. Every row is LACS minus the comparator; positive favors LACS. These nominal intervals are kept in the supplement because all include zero.

Comparator	Mean $\pm$ SD	95% t CI	p_t	p_{sf}
Full CE	-0.050 ± 0.704	$[-1.799, +1.699]$	0.913	1.000
InfoBatch (V)	$+0.021 \pm 0.546$	$[-1.335, +1.376]$	0.953	1.000
InfoBatch (B)	$+0.708 \pm 0.605$	$[-0.794, +2.211]$	0.180	0.250
RePB (V)	$+0.438 \pm 0.554$	$[-0.938, +1.813]$	0.305	0.500
RePB (N)	$+0.313 \pm 0.508$	$[-0.949, +1.574]$	0.398	0.500
RCAP	$+0.663 \pm 0.451$	$[-0.458, +1.783]$	0.126	0.250
RS2	$+0.875 \pm 0.456$	$[-0.258, +2.008]$	0.080	0.250
Selective-Backprop	$+0.221 \pm 0.641$	$[-1.373, +1.814]$	0.611	0.750

Operating-point selection. The InfoBatch port uses the official class at commit `8bbb012`. Seed-0 validation selects easy-pool keep .7 from $\{.3, .5, .7\}$, with final full-data fraction governed by $\delta = .875$. A separate keep value .537 is calibrated using backward counts only, not accuracy, to target LACS's nominal .7102 fraction. At reporting seeds, it executes 1,776,085 backward rows on average versus LACS's 1,775,604, a difference of approximately .027% of LACS's count. This is approximate count matching, not exact per-step support replay.

RePB uses the upstream `repb.py` at commit `80e217d`, with $\alpha = B = 128$. Validation selects keep .7 from $\{.25, .4, .55, .7\}$; interpolation of pilot work selects .509 for the near-budget run. Its observed .7308 nominal fraction is above LACS's .7102; this is explicitly *not* exact matching. The on-host freeze file was not recovered; the choices were reconstructed from the seed-0 pilot JSONs and match those executed. Our port preserves the upstream code's first-epoch prefix, $(E + 1)/N$ weights, fixed-count resampling, dropped sub-window tail, and per-epoch pruning reseeding. It uses ResNet-32/GN, 50 epochs, the common SGD recipe, and epoch-progress LR rather than the original full model/optimizer/schedule. This is a code-faithful common-recipe comparison, not a complete reproduction of the published recipe.

C.2 Trajectory, exact-condition, and one-sided panel values

These values support Section 4.5; they come from the logged CIFAR-10 selection events of the three reporting seeds and from the boosting round logs.

- *Retained fraction.* The seed-mean K_t/n by epoch is .886 at epoch 1, .917 at its early maximum (epoch 2), .573 at its minimum (epoch 39), and .984 at epoch 50; the epoch median is .668. The fraction of batches with no pruning is 0 through epoch 40 and then .002, .006, .050, .141, .347, .622, .797, .880, .924, .934 for epochs 41–50.
- *Exact condition.* Of 480 logged raw batches (3 seeds $\times$ 40 batches $\times$ epochs 1, 10, 25, 50), 108 satisfy $M_\tau(c)/(\rho n) > 1$; exactly these 108, all at epoch 50, prune nothing (agreement 1.000). On the boosting trajectories (split 1, every round), all 171 no-pruning rounds on

pol and the single no-pruning round on *electricity* (round 1, a flat initial model) satisfy the condition, and no pruning round does.

- *Boosting retained fractions* (normalized LACS, depth 1, three splits): mean keep ranges from .626 (MiniBooNE) to .873 (pol); on *pol* the keep rises from .771 in the first quarter of rounds to 1.000 in the last, whereas on *electricity* it stays near .71–.75. Trajectories are available for 11 of the 14 datasets.
- *Same mean, different shape.* Among 480 CIFAR-10 batches there are 1,722 pairs with $|\Delta\text{mean}| \leq .002$. The largest retained-fraction gap: batches with mean confidence .9395 and .9411 have variance .0120 and .0143, LME references .9801 and .9824, thresholds .9979 and 1.0002, and retained fractions .625 and 1.000. Boosting rounds show the same effect (e.g., MiniBooNE round 113 and pol round 39: means .9152 and .9194, retained .609 and .727).
- *Correct-only panel* (Section 4.4; three seeds on one RTX 4090 host, torch 2.4, InfoBatch keep .556 calibrated on seed-0 backward counters): C-LACS $90.15 \pm .28\%$ (backward fraction .721, 226 s), unified LACS $89.74 \pm .46\%$ (.710, 221 s), work-matched InfoBatch $89.25 \pm .23\%$ (.722, 139 s). C-LACS minus InfoBatch $+.896$ pp, 95% t CI $[+.119, +1.672]$; C-LACS minus LACS $+.41$ $[-.98, +1.80]$; LACS minus InfoBatch $+.49$ $[-1.22, +2.20]$. Confidently wrong rows are only 1.7–1.9% of the LACS drop set on CIFAR-10, so C-LACS and LACS have nearly the same support by construction. This host differs from the timing host of Table 16; wall times are comparable only within the panel.

C.3 Update-matched controls on CIFAR-10

Five seeds (1–5); for each seed all arms run on one RTX 4090 host (torch 2.4), so wall times are paired within seed but not across seeds. Small-batch CE uses batch 91 for 36 epochs (19,764 steps, about 1.80 M examples; LACS: 19,500 steps, 1.78 M backward rows). Random drop executes, in every batch, a uniformly random subset of the size of LACS's support with LACS's continuous weights. Paired contrasts ($n = 5$): LACS

Table 11: Update-matched controls, CIFAR-10, five seeds. Test accuracy per seed, mean ± SD, validation mean, backward fraction, extra scoring-forward rows, steps, and same-host wall.

Arm	Test per seed (1–5)	Mean ± SD	Val.	Bwd.	Extra fwd. (M)	Steps	Wall (s)
Full CE	89.21 / 89.99 / 89.67 / 89.80 / 89.61	$89.66 \pm .29$	89.99	.998	0	19,500	289
LACS	90.14 / 89.85 / 89.24 / 90.36 / 90.16	$89.95 \pm .44$	89.58	.709	2.50	19,500	280
Small-batch CE (91)	88.66 / 89.06 / 88.67 / 89.31 / 89.01	$88.95 \pm .28$	88.91	.719	0	19,764	288
Random drop	88.97 / 89.10 / 88.41 / 88.61 / 88.72	$88.77 \pm .28$	87.97	.693	0	19,500	381

− full CE $+.292$ $[-.404, +.989]$ (per seed $+.93, -.14, -.44, +.56, +.55$); LACS − small-batch CE $+1.005$ $[+.572, +1.438]$ $(+1.48, +.79, +.56, +1.05, +1.15)$; LACS − random drop $+1.185$ $[+.665, +1.705]$ $(+1.16, +.75, +.82, +1.75, +1.44)$; small-batch CE − full CE $-.713$ $[-1.002, -.423]$. The three-seed values of LACS here (seeds 1–3) are the same runs as the correct-only panel above.

C.4 Cached (lagged) confidences: a negative result

Same hosts and seeds as Appendix C.3. Lagged arms use the confidence cached at each example's previous visit and run no scoring forward: *pure* uses the cached value as is; *EMA* smooths it across visits; *refresh* periodically refreshes the cache. A seed-0 pilot (validation and work counters only) rejected *refresh* (-1.60 pp on validation); its seed 1–5 runs are exploratory. Same-host paired wall

Table 12: Cached-confidence variants, CIFAR-10, five seeds.

Arm	Test per seed (1–5)	Mean ± SD	Bwd.	Extra fwd. (M)	Wall (s)	Contrast vs. LACS [95% CI]
LACS (two-pass)	90.14 / 89.85 / 89.24 / 90.36 / 90.16	$89.95 \pm .44$	.709	2.50	280	—
Lagged, EMA	89.17 / 89.88 / 89.84 / 89.54 / 89.76	$89.64 \pm .29$	.839	0	352	$-.31$ $[-1.11, +.48]$
Lagged, pure	88.88 / 88.96 / 88.99 / 89.42 / 89.04	$89.06 \pm .21$	.647	0	341	$-.89$ $[-1.38, -.41]$
Lagged, refresh (exploratory)	88.76 / 89.50 / 89.46 / 89.78 / 89.53	$89.41 \pm .38$	.830	0	352	$-.55$ $[-1.26, +.17]$

differences are $+72$ s (EMA) and $+61$ s (pure) relative to two-pass LACS, with intervals of about ± 190 s because hosts differ across seeds; in no seed is a cached variant faster. The correlation between cached and fresh confidence on the forwarded rows in the last epoch is .70 (EMA) and .59–.60 (pure), and most cache entries are one visit old. Removing the scoring forward therefore neither saves time at this scale nor preserves the two-pass rule's accuracy; the pure variant also prunes more (.647) because stale scores are more dispersed.

C.5 CIFAR-100 BASELINE PANEL

CIFAR-100, ResNet-32 GN, 50 epochs, seeds 1–3, one RTX 4090 host per seed (torch 2.4), 8,000-image test partition and 2,000-image validation carve. Recipes were frozen from seed-0 validation and work counters only. InfoBatch's validation-best keep (.85) lies at the edge of its extended grid. Paired contrasts (LACS minus row, $n = 3$): CE $-.17$ $[-1.80, +1.46]$; InfoBatch val.-best $+.50$ $[-2.29, +3.29]$; InfoBatch work-matched $+.47$ $[-1.13, +2.07]$; RePB val.-best $+.13$ $[-.69, +.95]$; RePB near-budget $+.99$ $[-.46, +2.44]$; Selective-Backprop $+.48$ $[-2.29, +3.24]$; RS2 $+.96$ $[+.46, +1.46]$.

Table 13: CIFAR-100 direct comparison, three seeds.

Arm	Test per seed	Mean ± SD	Bwd.	Extra fwd. (M)	Steps	Wall (s)
Full CE	62.76 / 62.30 / 62.16	$62.41 \pm .31$	.998	0	19,500	203
LACS	61.91 / 62.76 / 62.04	$62.24 \pm .46$	.870	2.50	19,500	264
InfoBatch, val.-best keep .85	62.26 / 60.99 / 61.96	$61.74 \pm .67$	.925	0	18,057	202
InfoBatch, work-matched keep .738	61.69 / 61.56 / 62.05	$61.77 \pm .25$	.868	0	16,948	190
RePB, val.-best keep .85	61.49 / 62.56 / 62.26	$62.10 \pm .55$	.914	0	17,845	224
RePB, near-budget keep .743	61.48 / 61.84 / 60.44	$61.25 \pm .73$	.864	0	16,873	212
Selective-Backprop, $\beta = .16$	62.67 / 61.38 / 61.23	$61.76 \pm .80$	.863	2.45	16,871	254
RS2, keep .865	60.79 / 61.74 / 61.30	$61.27 \pm .48$	.863	0	16,850	179

C.6 RBF-SVM REPLICATION OVER EIGHT CELLS

Three datasets (bank-marketing, california, credit; the third chosen before any test evaluation from train-side pruning counts), 300 rounds, each (dataset, split) at that split's panel-selected LACS configuration with all arms at the same configuration. All 24 jobs ran concurrently on one 32-core CPU host (one thread each), so absolute wall times include contention and only within-cell ratios are meaningful. California split 1 has no panel cell and was not run. Over the seven pruning cells,

Table 14: RBF-SVM replication. Rows entering fits (M), wall seconds for full / LACS / same-count random, and test error at round 300. "Gate" records whether the LACS replay reproduced the stored panel validation prefix on this host.

Dataset / split	LACS rows vs. full (M)	Wall full / LACS / random (s)	LACS vs. full	LACS vs. random	Test err. full / LACS / random	Gate
bank-marketing / 0	.178 vs 1.918	1296 / 76 / 111	-94.2%	-32.1%	20.75 / 20.43 / 20.66	pass
bank-marketing / 1	.996 vs 2.221	1448 / 419 / 495	-71.1%	-15.3%	20.84 / 20.93 / 20.84	pass
bank-marketing / 2	.276 vs 2.221	1584 / 141 / 174	-91.1%	-18.5%	22.41 / 22.59 / 22.68	pass
california / 0	3.000 vs 3.000 (no-op)	1840 / 1743 / 1785	-5.3%	-2.4%	12.72 / 12.68 / 12.72	pass
california / 2	1.877 vs 3.000	1558 / 1088 / 860	-30.2%	$+26.5\%$	13.21 / 13.04 / 13.29	pass
credit / 0	2.121 vs 3.000	3108 / 1941 / 1822	-37.6%	$+6.5\%$	24.94 / 24.70 / 25.19	fail
credit / 1	1.984 vs 3.000	3159 / 1728 / 1733	-45.3%	-0.2%	24.55 / 25.13 / 24.98	pass
credit / 2	.351 vs 3.000	2556 / 281 / 248	-89.0%	$+13.1\%$	26.66 / 24.70 / 26.91	pass

LACS versus full has mean wall change -65.5% and LACS versus same-count random -2.9%; LACS minus random error is ≤ 0 in five of seven cells, and LACS minus full ranges from -1.96 to $+.57$ pp. The credit split-0 replay does not reproduce its stored validation prefix on this host (it did on the original host), a numeric host difference; the cell is retained and flagged. The california split-0 no-op is the no-pruning condition holding at every round.

C.7 INFOBATCH EXECUTION WITH THE LACS CUTOFF: SEED-0 PILOT

To test whether the scoring forward is intrinsic to the selector, a subclass of the official InfoBatch implementation keeps its dataset-level sampler, per-epoch pruning call, and final full-data phase,

and replaces only the eligibility rule: c_i is the max-softmax confidence cached from the ordinary training forward (unseen rows are always kept), q_i is normalized over the seen rows with $\rho = .7$, $\tau = .05$, and rows with $q_i \geq 1/\rho$ form the eligible set, which is either thinned at random with probability p and rescaled by $1/(1-p)$ or removed deterministically. Seed-0 validation values (CIFAR-10, one RTX 4090 host): official InfoBatch work-matched (keep .537) 89.40 at backward fraction .710; official InfoBatch keep .7: 88.55 at .813; LACS-eligibility with $p = .5$: 89.85 at .851; deterministic LACS-eligibility: 88.90 at .683. No arm runs a scoring forward. The LACS-eligible pool covers 28–49% of rows in mid-training, so thinning it to InfoBatch's work target would require $p \approx .97$, above the preregistered cap of .9: the distribution-sensitive eligibility is markedly more conservative than InfoBatch's below-mean pool. Seeds 1–3 were interrupted before completion and are not reported; this pilot is a diagnostic, not a result.

C.8 Execution matrix with CE and InfoBatch baselines

Seeds 1–3, same recipes as Table 27. InfoBatch is the official implementation at its default pruning ratio .5 and $\delta = .875$ without per-configuration tuning; LACS is the selective arm with frozen $\rho = .7$, $\tau = .05$. Because InfoBatch's default ratio executes less backward work than LACS in every configuration, the LACS−InfoBatch column compares different budgets; CE is the quality reference. Cells with fewer than three completed seeds are marked. The CIFAR-100 ResNet configuration appears both here and in Appendix C.5; the two are separate campaigns on different hosts (CE 62.78 versus 62.41, LACS 62.38 versus 62.24) and are not pooled.

Table 15: Execution-matrix configurations with baselines, three seeds unless marked.

Configuration	CE	InfoBatch (bwd.)	LACS (bwd.)	LACS − CE [95% CI]	LACS − InfoBatch [95% CI]
FashionMNIST / MLP	$90.63 \pm .11$ (2/3)	— (0/3)	$90.48 \pm .09$ (.649)	—	—
SVHN / ResNet-32 GN	$96.56 \pm .07$	96.68 (1/3; .629)	$96.68 \pm .08$ (.859)	$+.13$ $[-.22, +.47]$	—
CIFAR-10 / ViT-Tiny	$82.92 \pm .16$ (2/3)	$82.32 \pm .20$ (.691)	$83.00 \pm .15$ (.800)	—	$+.68$ $[+.32, +1.04]$
CIFAR-100 / ResNet-32 GN	$62.78 \pm .26$	60.17 ± 1.07 (.745)	$62.38 \pm .56$ (.871)	$-.40$ $[-2.41, +1.62]$	$+2.21$ $[+.73, +3.70]$
CIFAR-100-LT / ResNet-32 GN	24.92 ± 1.73	21.00 ± 1.89 (.754)	$24.48 \pm .79$ (.901)	$-.44$ $[-3.20, +2.33]$	$+3.48$ $[+.74, +6.22]$
CIFAR-100 / ViT-Tiny	$57.24 \pm .02$ (2/3)	$56.08 \pm .62$ (.705)	$56.48 \pm .36$ (.871)	—	$+.40$ $[-1.93, +2.73]$

All four InfoBatch-panel methods run on one host with two RTX 3090 GPUs, one job per GPU at a time. The GPUs run concurrently and share host resources; run order and initial warm-up can affect elapsed time. We retain the complete measured series (Table 16) without converting backward savings into total-FLOP savings. Ratios of mean wall time are 1.268 for LACS/full CE, 1.461 for LACS/InfoBatch-V, and 1.623 for LACS/InfoBatch-B. These results do not support a LACS speedup in this setting.

C.9 Completed same-host runtime comparisons

Table 16: InfoBatch-host wall times (seconds). Means and sample SD use the three measured runs; this series replaces the incomplete timing snapshot in the earlier manuscript.

Method	Seed 1	Seed 2	Seed 3	Mean ± SD
Full CE	312.7	313.5	312.5	312.9 ± 0.5
LACS	397.8	393.0	399.5	396.8 ± 3.4
InfoBatch (V)	271.4	269.9	273.5	271.6 ± 1.8
InfoBatch (B)	244.2	245.4	244.0	244.5 ± 0.8

RePB runs on a separate two-RTX-3090 host. Its three-seed accuracy comparisons reuse LACS and CE from the InfoBatch host. Seed-1 accuracy is reproduced exactly for both reference methods, but no analogous seed-2/3 check is recorded. Only seed 1 has the complete same-host reference timing panel below. It is not pooled with Table 16, and no single-run SD is fabricated.

The eight neural accuracy contrasts in Table 10 are nominal paired intervals with the dataset fixed. The original five-contrast post-hoc Holm values remain in the accompanying statistical export. The three newly added contrasts (InfoBatch-B, RePB-V, RePB-N) are exploratory and carry no retro-

Table 17: Separate RePB-host timing panel: seed 1 only, same CIFAR-10 recipe. This is a measured case, not a three-seed timing comparison.

Method	Seed-1 wall time (s)
Full CE	343.1
LACS	397.1
RePB (V)	306.7
RePB (N)	274.5

spective preregistration. All eight nominal intervals include zero; neither extra comparisons nor multiple-testing corrections create a supported superiority claim.

C.10 InfoBatch comparisons already performed

A historical balanced CIFAR-100/BatchNorm panel gives the single-run values in Table 18. It is retained as evidence rather than replaced by a positive long-tail comparison. Its older LACS implementation differs from the newer CIFAR-10/GN panel. A separate CIFAR-100-LT experiment

Table 18: Historical balanced CIFAR-100 comparison, one seed. Different network/training protocol from Table 8. No SD or significance is available. Runtimes are the recorded panel values, not an optimized universal method comparison.

Method	Test accuracy (%)	Recorded wall (s)
Full CE	78.77	2608.30
Native InfoBatch-CE	78.30	2121.12
Older LACS subset-BN	78.06	3280.53
Selective-Backprop	78.22	3358.99

uses Focal loss and a final 25-epoch full-data phase. The six paired seed values produce Table 19; the source reports a LACS–InfoBatch difference of $+1.237$ pp with 95% CI $[-.037, 2.510]$, and LACS–dense Focal -1.085 pp $[-1.909, -.261]$. LACS uses more rows, updates, and time than InfoBatch. It cannot substantiate simultaneous accuracy and efficiency superiority. Small differences between a mean recomputed from two-decimal seed values and the higher-precision reported mean are rounding, not extra runs. The existing implementation fidelity checks compare InfoBatch sam-

Table 19: Six-seed Focal comparison: mean $\pm \mathrm{SD}_s$ from the recorded per-seed values. The distinct Focal objective prevents merging this table with CE results.

Method	Test acc.	Bwd. rows (M)	Updates	Wall (s)
Cached LACS + full tail	39.50 ± 0.79	1.562	17,000	756
InfoBatch-Focal	38.26 ± 0.66	1.515	11,934	636
Random + full tail	39.06 ± 0.44	1.562	17,000	767
Dense Focal	40.59 ± 0.31	2.169	17,000	736

pling indices, retained-example compensation weights, losses, gradients, and the transition to final full-data training against the reference package. Passing those gates verifies the tested implementation, not every later port or an accuracy advantage. The new CIFAR-10 comparison in Table 8 is an additional same-harness test, not a replacement for the historical evidence above. It now includes both InfoBatch operating points, the same-host rerun times, and completed RePB common-recipe comparisons. Their protocol limitations remain attached; completion does not establish a statistical advantage. The historical panels above are retained under their original losses, model choices, and cost definitions.

Table 20: All six promoted compact learners against random supports at the LACS configuration. Error is mean $\pm \mathrm{SD}_D$; LACS errors are in Table 2. $*$ denotes the recorded primary-family Holm correction $p < .05$. Random replays per-round counts but chooses its own validation round.

Learner	Random error	Δ_{err}	95% t CI	p_{sf}	p_{H}	Wins
MLP H1	24.15 ± 11.30	-0.169	$[-0.357, +0.020]$	0.052	0.104	10/14
MLP H2	22.95 ± 11.40	-0.223^*	$[-0.418, -0.028]$	0.005	0.025	11/14
MLP H4	22.24 ± 11.42	-0.119	$[-0.271, +0.032]$	0.112	0.112	8/14
Quad R1	23.94 ± 11.31	-0.601^*	$[-1.097, -0.104]$	0.011	0.037	10/14
Quad R2	23.58 ± 11.27	-0.621^*	$[-1.065, -0.176]$	$< .001$	0.001	13/14
Quad R4	23.40 ± 11.12	-0.584^*	$[-1.135, -0.033]$	0.009	0.037	9/14

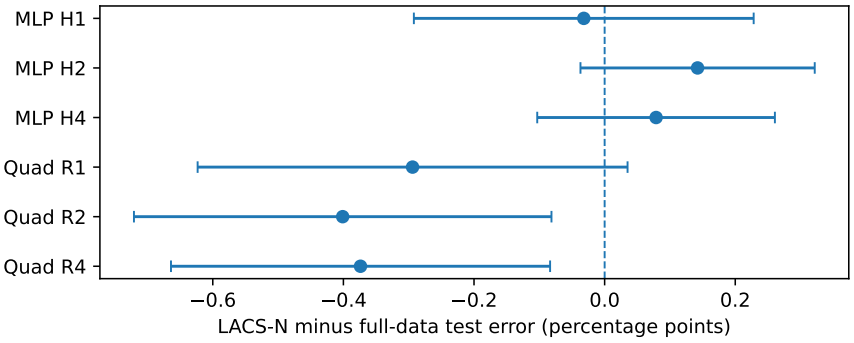


Figure 3: Paired macro error differences against full-data fitting. Bars are nominal 95% t confidence intervals across 14 dataset means, *not SDs*. Both positive and unresolved learner configurations are retained.

D Compact weak learners: full results and cost horizons

Averaging ratios can be dominated by runs where the reference chooses a very early checkpoint. Pooling instead weights expensive or large runs more heavily. Neither is automatically the right single headline; both are reported. For R2, selected rounds average 58.2 for full, 54.2 for LACS, and 53.1 for random. Random and LACS choose the same round in only 19% of runs. The originally reported 30% row reduction and .941 fit-time ratio describe full replays, not identical selected endpoints. We do not estimate missing prefix timing by multiplying total runtime by the selected-round fraction.

E Tree comparisons and resplit evaluation

E.1 GOSS semantics and budget controls

Matched GOSS selects approximately two thirds of K_t by the pinned implementation's $|g_i h_i|$ ranking and draws the remainder uniformly, scaling sampled gradients and Hessians by $(n - a_t)/(K_t - a_t)$. Here a_t is the top-group size, not a global pruning fraction. It uses exactly the LACS replay's per-round counts, omits an independent warmup, and executes a global rather than thread-block selection. These are declared adaptations for the matched-support experiment. The all-row identity gate checks that $K_t = n$ reproduces the all-row model; the half-row perturbation must change it. Replays are recomputed together when host-dependent differences affect the LACS count trajectory.

Independently tuned GOSS instead uses top-rate $\{.05, .1, .2, .3\}$ and other-rate $\{.05, .1, .2\}$, 200 shared configuration candidates and its own validation-selected round, with $\lfloor 1/\eta \rfloor$ warmup rounds. Official LightGBM GBDT and GOSS references are reported separately because library execution, defaults, and sampling differ. The pinned gradient–Hessian product need not behave like raw-gradient magnitude; in particular, small Hessians can suppress both extreme tails. The experiment is not evidence that every version of GOSS must keep every confident error.

Table 21: Quadratic R2, per-dataset test error mean ± SD across three resplits. The macro SD in the main table measures different variation. Neither the large random spread on california nor its favorable LACS difference establishes a universal mechanism.

Dataset	Full	LACS	Random	LACS − Full
Diabetes130US	39.02 ± 0.32	39.05 ± 0.26	39.39 ± 0.44	$+0.029$
MagicTelescope	16.52 ± 0.89	16.49 ± 1.07	16.69 ± 1.08	-0.036
MiniBooNE_10k	7.90 ± 0.88	7.92 ± 0.64	7.79 ± 0.71	$+0.016$
bank-marketing	23.22 ± 0.97	22.02 ± 0.52	22.71 ± 0.79	-1.200
california	15.26 ± 0.20	15.13 ± 0.36	18.03 ± 4.59	-0.130
credit	25.49 ± 0.06	25.07 ± 0.42	25.70 ± 0.19	-0.418
credit-card-default	29.51 ± 0.20	29.39 ± 0.23	29.73 ± 0.09	-0.120
electricity	24.70 ± 0.28	24.49 ± 0.44	24.74 ± 0.20	-0.205
eye_movements	43.15 ± 2.27	42.30 ± 1.38	42.67 ± 1.73	-0.855
heloc	28.19 ± 0.85	28.21 ± 1.18	28.35 ± 1.52	$+0.016$
higgs_20k	34.03 ± 0.42	32.19 ± 0.74	33.82 ± 0.99	-1.848
house_16H	14.23 ± 0.86	13.63 ± 0.53	13.98 ± 1.00	-0.600
jannis	23.80 ± 0.51	23.61 ± 0.57	24.06 ± 0.52	-0.186
pol	2.06 ± 0.14	1.98 ± 0.12	2.49 ± 0.41	-0.079

Table 22: **Do not mix cost horizons.** LACS/full ratios; "mean" is the mean of run ratios, "pooled" the ratio of summed work. Prefix ends at each method's selected predictor. Replay spans all recorded rounds. Selected-prefix time is unavailable for every learner.

	Selected-prefix rows		Replay rows		Replay fit time	
Learner	Mean	Pooled	Mean	Pooled	Mean	Pooled
MLP H1	4.827	.963	.887	.865	1.094	.937
MLP H2	1.097	.799	.821	.816	1.002	.807
MLP H4	1.372	.832	.788	.785	.817	.723
Quad R1	4.601	.776	.760	.750	1.086	.892
Quad R2	.936	.641	.700	.695	.941	.747
Quad R4	2.671	.760	.663	.666	.921	.678

Table 23: **Aggregate tree results across 14 datasets.** Error is mean $\pm \mathrm{SD}_D$. LACS-T is the historical fixed unsigned-margin variant, not normalized LACS.

Depth	Full (partner)	GOSS (own HP)	GOSS (matched K_t)	LACS-T
1	21.74 ± 10.47	21.94 ± 10.75	23.14 ± 10.60	21.50 ± 10.74
2	20.45 ± 10.99	20.66 ± 11.21	20.61 ± 11.00	20.27 ± 11.01
3	19.98 ± 10.79	20.16 ± 11.42	20.21 ± 10.81	19.99 ± 10.95

Table 24: **GOSS paired uncertainty and replication scope.** Negative differences favor LACS-T. New-only Holm values apply to the primary matched-GOSS contrasts across three depths.

Depth	Matched Δ	Pooled 95% CI	p_{sf}	New-only p_{H}	Own-HP Δ (p_{sf})
1	-1.643	$[-3.210, -.076]$	.0049	.1250	$-.442$ (.0665)
2	$-.346$	$[-.604, -.088]$	.0049	.0147*	$-.386$ (.0314)
3	$-.220$	$[-.396, -.045]$	.0068	.1250	$-.171$ (.6238)

The primary matched contrast is favorable at every depth in the pooled panel. The initial stage includes six datasets at depths one and three, so those pooled results partly reuse selection-triggering evidence. The new-only comparison has eight additional datasets at those depths and all 14 depth-two blocks. Its primary means are -2.021 pp at depth one (CI $[-4.990, .947]$), $-.346$ at depth two ($[-.604, -.087]$), and $-.307$ at depth three ($[-.617, .002]$). Holm-adjusted values are .125, .0147, .125. These are new experimental comparisons on existing data splits, not three independent new-data confirmations.

Table 25: Depth-2 raw errors on the 14 datasets. Each row uses one dataset-specific split; randomized controls average their replays. No per-dataset SD is invented for the deterministic single-split arms.

Dataset	LACS-T	Full	GOSS matched	GOSS own	Random
electricity	15.508	16.165	15.790	16.336	15.989
house_16H	11.154	11.401	11.218	11.543	11.295
pol	2.408	2.502	2.446	2.172	2.351
eye_movements	37.523	37.523	37.523	39.337	37.523
credit	22.702	22.979	22.877	22.532	22.911
MagicTelescope	13.843	14.235	14.285	14.093	14.292
bank-marketing	20.117	20.072	21.440	19.802	20.765
MiniBooNE	6.889	6.875	6.870	6.937	6.918
Bioresponse	23.130	23.130	23.130	24.654	23.130
credit-card-default	28.659	29.053	29.641	28.874	29.254
jannis	23.566	24.176	24.324	23.704	24.603
Diabetes130US	39.658	39.637	39.635	39.642	39.618
heloc	28.238	28.238	28.238	28.429	28.238
california	10.371	10.357	11.188	11.123	11.093

The independently tuned depth-two GOSS difference is $-.386$ pp, CI $[-.754, -.019]$, nominal sign-flip $p = .0314$. Its counterparts at depths one and three are unresolved. Against full training at the LACS configuration, differences are $-.239$, $-.184$, and $+.008$ pp. At depth two, pooled fitting rows fall to .767 of full, but pooled fitting time rises to 1.529. The tree evidence therefore does not establish that LACS is faster than an optimized XGBoost or LightGBM implementation.

E.2 Widened depth-one experiment with ten resplits

Table 26: Widened depth-one study: test error mean $\pm$ SD across ten resplits of each cached dataset. All hyperparameters and round counts are frozen from prior development. This is distinct from the 500-round GOSS experiment.

Dataset	LACS-T	Tuned random	Full
Bioresponse	25.22 ± 1.53	28.77 ± 1.35	28.03 ± 1.61
Diabetes130US	39.70 ± 0.23	39.56 ± 0.29	39.70 ± 0.23
MagicTelescope	17.62 ± 0.48	17.26 ± 0.33	17.62 ± 0.48
MiniBooNE	7.94 ± 0.11	7.89 ± 0.10	7.93 ± 0.10
bank-marketing	20.13 ± 0.49	21.40 ± 0.67	20.36 ± 0.78
california	11.37 ± 0.27	11.75 ± 0.35	11.55 ± 0.28
credit	23.99 ± 0.29	23.31 ± 0.57	23.51 ± 0.53
credit-card-default	29.48 ± 0.82	29.88 ± 0.68	29.45 ± 0.88
electricity	16.59 ± 0.26	17.48 ± 0.29	17.00 ± 0.30
eye_movements	39.82 ± 1.49	40.51 ± 1.04	41.46 ± 1.86
heloc	28.75 ± 1.15	28.60 ± 1.20	28.75 ± 1.15
house_16H	12.71 ± 0.27	12.97 ± 0.22	12.75 ± 0.31
jannis	24.75 ± 0.17	24.94 ± 0.11	24.98 ± 0.16
pol	4.25 ± 0.39	5.58 ± 0.42	5.98 ± 0.44

The macro errors are 21.595 ± 10.800 for LACS-T, 22.134 ± 10.782 for tuned random, and 22.078 ± 10.872 for full XGBoost, where these SDs are across dataset means. LACS–random is $-.540$ pp (95% t CI $[-1.142, .063]$, $p_t = .0750$, $p_{\text{sf}} = .0465$). LACS–full is $-.484$ pp ($[-1.011, .044]$, $p_t = .0693$, $p_{\text{sf}} = .0645$). A later AdaBoost-stump comparison gives $-.535$ pp ($[-1.028, -.041]$, $p_t = .0358$, $p_{\text{sf}} = .0251$), a nominal result rather than a family-adjusted confirmation. The five-to-ten expansion and overlap with development examples are disclosed in Appendix B. The main claim does not depend on choosing the one inferential procedure that excludes zero.

Table 27: Dense-weighted versus selective LACS execution. Accuracies are mean ± seed SD. Standard cells use seeds 1–3; the separate 200-epoch extension uses seeds 0–2. Work reductions are the recorded campaign aggregates; the standard aggregates also include development-seed counters. Time ratios compare selective with dense weighted LACS.

Dataset / epochs	Model	Dense LACS acc.	Selective acc.	Bwd. saved	Time ratio
CIFAR-10 / 50	ResNet-32 GN	89.64 ± 0.14	89.82 ± 0.04	28.8%	1.192
CIFAR-100 / 50	ResNet-32 GN	62.00 ± 0.41	62.38 ± 0.56	12.9%	1.105
CIFAR-100-LT / 50	ResNet-32 GN	24.48 ± 1.44	24.48 ± 0.79	9.9%	1.263
SVHN / 50	ResNet-32 GN	96.53 ± 0.07	96.68 ± 0.08	14.1%	1.123
CIFAR-10 / 50	ViT-Tiny LN	82.40 ± 0.10	83.00 ± 0.15	20.0%	1.149
CIFAR-100 / 50	ViT-Tiny LN	56.66 ± 0.47	56.48 ± 0.36	12.9%	1.161
FashionMNIST / 20	MLP LN	90.59 ± 0.20	90.48 ± 0.09	35.1%	1.040
CIFAR-100-LT / 200	ResNet-32 GN	31.94 ± 0.56	32.22 ± 0.38	25.5%	0.998

F Neural execution and the Transformer case

The comparison tests execution of one weighting rule; CE and InfoBatch baselines for these configurations were run in a later campaign and are reported in Appendix C.8. The matrix and the baseline panel of Table 8 are distinct campaigns reporting the same nominal CIFAR-10 architecture and epoch budget. Their selective-LACS means are 89.82 and 89.77%, respectively; aggregate values alone do not establish that the full execution recipes are identical or identify the source of the discrepancy. They are not pooled as paired repetitions. The standard matrix's maximum same-state fp64 relative discrepancy is 3.663×10^{-15}. Initial fp32 checks did not all pass the original narrow threshold; the protocol was amended to a fp64 identity gate, while fp32 training results remained separate. No claim of exact trajectory agreement in practical floating point follows. The 200-epoch local extension is not pooled as another independent architecture.

Table 28: Transformer decoder, native LayerNorm and optimized attention, 800 steps, seeds 11–13 on an RTX 4090. All entries are mean ± sample SD. Quality is validation NLL, not held-out test perplexity or a convergence result.

Implementation	Wall (s) ↓	Val. NLL ↓	Bwd. sequences
Optimized CE	103.91 ± 0.56	1.5200 ± 0.0093	102400.0 ± 0.0
Optimized dense LACS	105.56 ± 1.20	1.5666 ± 0.0092	102400.0 ± 0.0
Custom engine, keep all	118.67 ± 0.96	1.5670 ± 0.0073	102400.0 ± 0.0
Custom engine, selective	110.35 ± 0.71	1.5688 ± 0.0084	90657.0 ± 83.2

The model has four layers, hidden width 512, eight heads, batch 128 and sequence length 256. The corpus is a byte-level derivative of WikiText-2 (Merity et al., 2017); this is not the standard word-level benchmark. Optimized dense LACS and selected custom execution have mean NLL 1.5666 and 1.5688. Selected execution reduces backward sequence count by 11.47%, but its wall ratio to optimized dense is 1.0454. The ratio to the custom keep-all engine is .9299. The former is the practical same-objective comparison; the latter isolates the gain inside that engine. Earlier measurements using different timing and quality configurations are not substituted for this matched experiment.

SVM details. The recorded credit configuration is $C = 45.7802$, $\gamma = .106545$, shrinkage .0084634, no warmup, $\lambda = -.863959$, and $\tau = .0101386$. The final replay runs 300 rounds. LACS fitting weights differ from the logistic weights in the all-row and random arms; this is a combined-method case. The LACS replay records approximately 1151.5 seconds fitting, 983.3 seconds scoring, and .19 seconds in mask computation. Full pipeline cost is dominated by more than the mask. Summed support-vector incidences cannot be interpreted as the number of unique vectors stored in the final ensemble. Other explored configurations can prune no examples, so the one speed case is not a general SVM acceleration guarantee.

G GRADIENT FIDELITY, SIGN, AND CUTOFF CONTROLS

For binary cross-entropy, let $g_i = \nabla_\theta \ell_i$, $G = n^{-1} \sum_i g_i$, and $G_S = n^{-1} \sum_{i \in S} g_i$. If the removed correct and incorrect sets are D_+ and D_- and $\|\nabla_\theta F_i\| \le L_i$, then

$$\|G - G_S\| \le \frac{1}{n} \left[\sum_{i \in D_+} L_i (1 - c_i) + \sum_{i \in D_-} L_i c_i \right].$$

The first term can be small for confidently correct rows; the second makes clear that removing confident mistakes is an influence intervention rather than an approximation to their original contribution.

G.1 CHECKPOINT DESIGN AND SCALE CONVENTIONS

The FashionMNIST model is an MLP 784–512–256–10 with LayerNorm. One dense-LACS seed is retrained on CPU to save checkpoints at epochs 1, 10, and 20. Eight preregistered batches of 128 examples are analyzed per checkpoint, with 20 random supports per batch for the initial random baseline. The original CE gradient is evaluated at the LACS-trained model state, not along an independently CE-trained trajectory. This distinction matters when interpreting fidelity.

For hard support normalized over survivors, $\widehat{G}_S = K^{-1} \sum_{i \in S} g_i$ and $G_S = (K/n) \widehat{G}_S$. Direction, scale, and residual are separate metrics. A near-unit cosine with a norm ratio of 1.65 is consistent with a small original-denominator residual; it is not itself small raw vector error. All corrected residuals below were calculated batch by batch, not reconstructed from mean cosines and mean norm ratios.

Table 29: Dropped-gradient follow-up. Ranges span eight batches of one model/seed at each checkpoint; they are not confidence intervals. All early removed rows were correctly classified.

Epoch	Mean K/n	Full-n rel. residual	Cosine range	Wrong drops	Cancellation factor
1	.61	.015–.036	.9994–.9999	0	.26–.33
10	.61	.004–.014	.9999–1.0000	0	.30–.36
20	1.00	0	1	none	not applicable

Here cancellation factor is $\| \sum_{i \in D} g_i \| / \sum_{i \in D} \|g_i\|$. It shows that both small individual contributions and partial cancellation enter the measured residual. Continuous LACS adds a survivor-reweighting term:

$$G_w - G = \frac{1}{n} \sum_{i \in S} (w_i - 1) g_i - \frac{1}{n} \sum_{i \in D} g_i.$$

A small dropped-gradient term alone does not bound $G_w - G$. Indeed, at epoch 20 hard support is the full set, but continuous mean-one weights still change the update substantially.

G.2 WHAT THE GRADMATCH-STYLE DIAGNOSTIC DOES AND DOES NOT SHOW

The last-layer embedding $(p - e_y) \otimes h$ with bias coordinates has 2,570 dimensions and is checked against autograd. Nonnegative ridge OMP uses regularization .5, a maximum count equal to the LACS count, and no residual-tolerance stopping. The initial implementation stops on a repeated index, so its returned count is below the cap in all 24 batches. Its smaller support is not an equal-budget comparison.

A no-repeat extension reaches the LACS count and yields support Jaccard overlap .84–.96 at epoch 1 and .69–.90 at epoch 10. With the same offline refitting of weights, mean full-model residuals are .059 for both supports at epoch 1 and .147 for both at epoch 10; equal-count random supports give .429 and .553. These are supportive representation diagnostics, not evidence that deployed LACS runs the refit solver, that OMP is inferior, or that either produces better final trained models. Normalizing fitted weights to sum one changes their scale and does not establish a GradMatch failure. At epoch 20 the learning rate is zero, making one-step validation probes vacuous; earlier probes are also norm-confounded.

Table 30: Initial checkpoint diagnostic, means across eight batches. Residual is relative to full-model CE; native and fitted weight scales are deliberately separate. OMP is a diagnostic on last-layer gradients, not a complete training-method reproduction.

Epoch	Method	K	Relative L_2	Cosine	Norm ratio
1	LACS hard, survivor mean	77.63	.6488	.9997	1.6480
	LACS continuous	77.63	1.2447	.9912	2.2288
	Random K	77.63	.6526	.8370	1.1951
	OMP fitted scale	44.13	.0925	.9956	.9826
	OMP normalized weights	44.13	1.8165	.9956	2.8097
10	LACS hard, survivor mean	77.50	.6560	1.0000	1.6559
	LACS continuous	77.50	1.7416	.9819	2.7130
	Random K	77.50	.6932	.8305	1.2492
	OMP fitted scale	19.75	.2035	.9805	.9239
20	LACS hard	128	0	1	1
	LACS continuous	128	2.2194	.9800	3.1903
	OMP fitted scale	14.13	.2374	.9711	.9312

G.3 Signed and cutoff interventions

The exact $\lambda_{\text{neg}} = 0$ signed variant protects all confidently wrong rows before global normalization. It normalizes the positive population separately, so it changes more than a single binary decision. Over 14 datasets, signed-minus-unsigned error is $+.1297 \pm .4871$ pp (dataset SD), CI $[-.1515, .4109]$. Only four datasets favor protection. Earlier signed threshold, split-softmax, and global-softmax variants likewise did not establish a robust advantage. This supports a simpler unsigned design within these trials, not a universal statement about label information.

Table 31: Cutoff/score study, MagicTelescope, one fixed split, three training seeds. Accuracy metric is test error mean $\pm$ seed SD; all operating points are validation-selected. Thus rows and rounds need not be equal.

Variant	Selected setting	Test error	Active fraction	Fit rows (M)
Full	—	$17.058 \pm .054$	1.000	3.40
Native hard, max probability	$\tau = .1$	$17.177 \pm .196$	.677	1.92
Native hard, true-label probability	$\tau = .02$	18.316 ± 2.015	.682	2.78
Native hard, negative CE	$\tau = 2$	$17.058 \pm .054$	1.000	3.40
Native continuous	$\tau = .5$	$16.762 \pm .188$	.988	1.82
Fixed keep, max probability	keep .3	$16.690 \pm .094$	.300	.67
Fixed keep, true-label probability	keep .8	$17.177 \pm .563$	.800	3.27
Absolute max-probability threshold	cutoff .95	$16.809 \pm .307$	.772	1.33
Absolute true-label threshold	cutoff .8	$17.153 \pm .339$	.591	.97

The max-probability ranking's favorable comparison with true-label probability under the fixed-ratio *policy* uses different selected numeric ratios (.3 and .8); it is not a matched-ratio intervention. Native negative-CE, continuous, and true-label-ratio optima reach search-grid edges. This small exploratory study establishes neither optimal hyperparameters nor a universal ranking/cutoff hierarchy.

H One-sided diagnostic controls: definition and evidential scope

A one-sided intervention can ask whether the same unsigned high-confidence mask should be restricted to correct or incorrect predictions. Let $z_i = \mathbf{1}\{\arg\max_k p(k \mid x_i) = y_i\}$, using a fixed class tie convention, and compute q_i once over the full candidate population. With $d_i = \mathbf{1}\{q_i \geq \kappa\}$, define the hard keep masks

$$a_i^U = 1 - d_i, \qquad a_i^C = 1 - d_i z_i, \qquad a_i^W = 1 - d_i(1 - z_i), \qquad a_i^F = 1. \tag{14}$$

Here C removes only correct members of the unsigned removal set, W removes only wrong members, and U removes both. At a fixed model state, their removed sets form the partition $D = D_C \dot{\cup} D_W$. The score normalization is not recomputed within either group. The C/W controls are explicitly label-aware; they are not deployed versions that retain the label-agnostic guarantee.

The previous negative-side-zero study does not implement Eq. (14): it applies a separate positive-side normalizer and continuous weights, followed by global coefficient renormalization. Likewise, independent model trajectories diverge after the first update, so a fixed-state partition is not an identity between masks taken from different trained models.

The new global-normalizer campaign freezes one full-data depth-1 XGBoost configuration and round endpoint per dataset, then compares Full, unsigned U, correct-only C, wrong-only W, and exact-count random controls on six datasets with three fresh splits. This is a mechanism comparison under a common base configuration, not a comparison between separately tuned algorithms. Table 32 reports test error.

Table 32: **One-sided global-normalizer LACS ablation.** Test error (%, mean ± SD across three fresh splits). U removes the full high-confidence mask; C removes only currently correct members; W removes only currently wrong members.

Dataset	Full	U	C	W
bank-marketing	$20.36 \pm .76$	$20.13 \pm .88$	20.61 ± 1.29	20.61 ± 1.11
Diabetes130US	$39.76 \pm .11$	$39.66 \pm .27$	$55.61 \pm .41$	$39.74 \pm .08$
electricity	$18.46 \pm .28$	$18.20 \pm .21$	23.27 ± 2.18	$18.23 \pm .30$
california	$12.05 \pm .35$	$11.93 \pm .34$	$12.09 \pm .52$	$12.07 \pm .44$
MagicTelescope	$16.99 \pm .51$	$16.98 \pm .71$	$16.75 \pm .74$	$17.09 \pm .56$
credit	$22.76 \pm .71$	$22.89 \pm .89$	$22.96 \pm .88$	$22.58 \pm .42$

Using dataset as the statistical unit, U–C is -3.583 pp with 95% t-CI $[-10.274, 3.107]$ (permutation $p = .125$, Holm-adjusted $p = .25$), while U–W is $-.090$ pp with CI $[-.354, .173]$ (permutation $p = .469$). U–Full is $-.099$ pp with CI $[-.248, .051]$. Thus the aggregate contrasts are not statistically resolved. The diagnostic value is the pair of large C-only failures on Diabetes130US and electricity. A possible explanation is altered retained-set composition after many correct examples are removed; the experiment does not isolate that explanation from other changes along the training trajectory. Unified U removes both parts of the selected high-confidence tail and avoids those failures in these datasets. W alone is close to Full on average, so the experiment does not support a universal claim that pruning confident errors improves accuracy.

Confident errors are not merely a planted corner case in this campaign. Averaged over rounds, currently wrong rows account for about 4.8%, 30.3%, 3.8%, 1.6%, 3.8%, and 7.3% of the unified removal set on bank-marketing, Diabetes130US, electricity, california, MagicTelescope, and credit, respectively; a nonempty wrong subset occurs in 92–100% of rounds. These are current-model mistakes, not verified semantic outliers or corrupted labels. In contrast, the FashionMNIST same-state gradient diagnostic removes no wrong rows at the inspected checkpoints. The mechanism is therefore data- and state-dependent.

Common-checkpoint implementation checks on planted tails and 24 reused neural batches verify the set partition and unsigned label-swap invariance. In those real neural batches, D_W is empty, so $W = F$ and $C = U$; they cannot identify a benefit from removing wrong examples there.

Canonical churn and re-entry. A row-ID audit of unified LACS on Diabetes130US, california, and electricity shows that the selected support is not a permanent deletion. Across the three trajectories, 50–67% of rows are pruned at least once. Among ever-pruned rows, $26.5 \pm 10.5\%$ re-enter on the short 13-round Diabetes run, while $71.3 \pm 9.6\%$ and $68.1 \pm 2.0\%$ re-enter on california and electricity. Completed inactive spells have median 1.3–2 rounds and mean 1.8–9.3 rounds. Confidently wrong rows also re-enter: among rows ever entering D_W, the re-entry fraction is about 22.5%, 30.9%, and 44.9%, respectively. This dynamic behavior follows from recomputing confidence; it should not be described as a formal safety guarantee.

Known label flips are not preferentially selected by the label-agnostic mask. We inject 10% symmetric training-label corruption on Diabetes130US and electricity while keeping validation/test

clean and freeze the original configuration. The probability of entering the unified removal set is not enriched for corrupted rows: $P(D \mid \text{corrupt})/P(D \mid \text{clean}) = 1.05 \pm .09$ and $.89 \pm .09$, and confidence-decile-matched ratios are approximately one. This is structurally consistent with label-agnostic selection. The *wrong* portion D_W is different because correctness uses the observed label: corrupted examples comprise $75.5 \pm 1.7\%$ of D_W on electricity (10% base rate) and about 21% on Diabetes. Thus, when a selected row is both highly confident and wrong, label corruption can be enriched, but LACS does not directly target corruption and the experiment is not a robustness benchmark.

Historical versus canonical re-entry. An older cached long-tail selector also showed non-monotone support and about 11.5% explicit return paths, but that implementation includes stale/random re-admission. We therefore keep it separate from the canonical online churn statistics above.

InfoBatch-style boosting adaptation. To test whether the closest InfoBatch idea can be carried into repeated weak-learner fitting, we implement a small *InfoBatch-style* adaptation rather than calling it canonical InfoBatch: rows below the current mean logistic loss are eligible, eligible rows are thinned with a fixed probability calibrated only to LACS's fit-row budget, surviving eligible fitting weights are multiplied by $1/(1-p)$, and the final 12.5% of rounds use the full data. The original SGD expectation argument is not transferred to nonlinear weak-learner fitting. At the calibrated

Table 33: **InfoBatch-style boosting adaptation at a work-matched budget.** Error is mean $\pm$ SD across three frozen splits. The adaptation obtains similar errors with a very different support; this is evidence that InfoBatch-like ideas can be ported to boosting, not that the original InfoBatch algorithm or guarantee applies unchanged.

Dataset	Full	LACS-U	IB-style	Pruned-set Jaccard U/IB
Diabetes130US	$39.763 \pm .105$	$39.664 \pm .269$	$39.656 \pm .233$	$.083 \pm .009$
MagicTelescope	$16.987 \pm .506$	$16.975 \pm .714$	$17.307 \pm .623$	$.229 \pm .005$
electricity	$18.458 \pm .278$	$18.197 \pm .209$	$18.187 \pm .186$	$.231 \pm .001$

budgets, IB-style minus LACS-U is $-.008 \pm .257$, $+.332 \pm .445$, and $-.010 \pm .058$ pp on the three datasets. The small pruned-set Jaccards (.08–.23) show that comparable row budgets and errors can arise from quite different supports. Random thinning also makes re-entry far more frequent for the IB-style rule (about .72–.97) than for unified LACS (.27–.68 in the overlapping audited trajectories).

I Long-tail and semi-supervised boundaries

I.1 Ordinary CE and specialized losses

Table 34: CIFAR-100-LT ordinary-CE recipe, three seeds. The .703 figure for unsigned LACS is a *nonzero-weight fraction*; this arm performs full backward and is not a measured sparse-execution gain. Other control rows are not used for speed claims.

Method	Test accuracy	Nonzero fraction	Physical backward
Full CE	37.95 ± 0.83	1.000	full
Unsigned LACS weights	38.48 ± 0.38	0.703	full
Signed rescue	38.83 ± 0.54	0.697	not claimed
Random (signed partner)	39.23 ± 0.50	0.697	not claimed

Unsigned-minus-CE is approximately $+.53$ pp with CI $[-.60, 1.67]$, so the positive point estimate is unresolved. Signed rescue improves its point estimate further but does not outperform its own exact-count random control. That random control is partnered with the signed arm, not with unsigned LACS; it cannot isolate the unsigned support's identity effect. These are supporting supervised observations, not the central accuracy-efficiency result.

LDAM-DRW plus LACS weighting improves by .317 pp, CI $[-.342, .975]$; its selector improves by .121 pp, CI $[-.820, 1.061]$. Neither establishes an improvement. The cached selector also re-admits

Table 35: LDAM and LDAM-DRW combinations, mean ± SD across three seeds. The 8,000-image reporting test partition and training recipe differ from the Balanced Softmax and historical InfoBatch experiments.

Recipe	LACS/control arm	Test accuracy
LDAM	dense	40.39 ± 0.09
LDAM	weight	40.27 ± 0.44
LDAM	selector	39.77 ± 0.75
LDAM	random	39.33 ± 0.62
LDAM-DRW	dense	43.35 ± 0.50
LDAM-DRW	weight	43.67 ± 0.74
LDAM-DRW	selector	43.47 ± 0.70
LDAM-DRW	random	43.83 ± 0.77

unseen/old rows and a random fraction, and uses a detached CE probe head rather than the specialized loss's raw logits. It is not the pure online support in Eq. (2). Approximately .546 of candidate rows are retained, but recorded walls are confounded by concurrent jobs and are not evidence of a speedup. The probe uses a constant learning rate and some selected hyperparameters lie on a grid edge. Balanced Softmax's separate validation-frozen adaptation gives 45.57% versus 45.87% for the baseline and uses a fixed-ratio selector; it does not establish superiority of the normalized gate. The prior-work objectives are distinct from example selection (Cao et al., 2019; Ren et al., 2020).

I.2 Semi-supervised development search and evaluation audit

The FlexMatch development setting uses CIFAR-100, WRN-28-8, 2,000 labeled training images and an intended 500-image validation carve, with 10,000 iterations under a 50,000-iteration cosine schedule (Zhang et al., 2021). The following numbers are *historical recorded diagnostics under the audited loader*, not corrected full-validation or full-test benchmark estimates. Test diagnostics were computed during development; they are not used by the recorded selection rule. Earlier descriptions that test was never evaluated were inaccurate.

Preserved development observations. For canonical $\lambda = -.7, \tau = .05$, the recorded test-error differences against a count-schedule random replay are -1.152, $+.381$, $+2.144$, and $+3.105$ pp at 2,500, 5,000, 7,500, and 10,000 iterations. They describe one development seed and must not be interpreted as corrected multi-seed evidence. Table 36 preserves representative search values. The source-reconstructed evaluation counts below apply to the pinned loader; per-run evaluated-ID logs do not survive.

Table 36: **Historical SSL diagnostics; validation truncation confirmed.** Values are not verified on the intended complete validation carve. Stage-1 values and the reported warm-up winner are preserved rather than silently replaced. Backward work includes labeled examples and the two unlabeled views.

λ	τ	Warm-up	Recorded val. error (%)	Mean K/N	Bwd. (M)
$-.4$	.5	0	49.609	.9967	9.569
$-.1$	.2	0	49.609	.9984	9.585
$-.3$	.2	0	50.391	.9804	9.423
$-.3$	.1	0	51.172	.9365	9.025
$-.7$	.05	0	54.688	.6567	6.539
$-.1$	.5	5,000	48.438	1.0000	9.600

Search status and candidate ledger. The base grid contains 54 configurations. After the disclosed admissible-domain amendment, 42 complete in $-1 < \lambda < 0$, ten excluded configurations have observed empty-support initialization, and two lack failure receipts. All 24 warm-up cells subsequently complete. The original finalizer omitted the reused canonical P0 candidate and reported 65 candidates. A corrected ledger restores all $42 + 24 = 66$ and leaves the historical winner unchanged. This counting correction does not repair the truncated validation ranking.

Why the reported candidate cannot prune. The recorded accuracy and Pareto finalists coincide at $\lambda = -.1$, $\tau = .5$, warm-up 5,000. Weak-view maximum probability and the initialization score both lie in $[0, 1]$. For any such vector,

$$q_i = \frac{e^{c_i/.5}}{n^{-1}\sum_j e^{c_j/.5}} \le e^2 < 10, \qquad u_i = 1 - .1q_i \ge 1 - .1e^2 > 0. \tag{15}$$

Its hard support is therefore all rows for every admissible confidence distribution, not merely along the recorded trajectory. This proves a property of that parameter setting; it does not validate the search winner or imply a bitwise full-training trajectory when sampling order differs.

Evaluation truncation and required correction. The installed `semilearn 0.3.2` loader defaults to `drop_last=True`. The SSL trainer requests evaluation batches of 256 without overriding that flag. Under the pinned unmodified loader this evaluates only $\lfloor 500/256 \rfloor 256 = 256$ validation images and $\lfloor 10000/256 \rfloor 256 = 9{,}984$ test images. The observed validation granularity $100/256 = .390625$ pp is consistent with that code path. This is a source-level problem; complete remote evaluated-ID and package-identity records remain an author-side audit item, not an invented recovered set.

A new evaluation-only loader fix uses `drop_last=False` and passes complete-coverage gates. However, no trained checkpoints or per-example predictions survive, so the full-carve ranking cannot be recovered from aggregate scores. No corrected full-validation training comparison is complete. The SSL observations therefore remain a transparent development boundary and provide no positive sparse-efficiency claim.

Subsequent source audit and pool correction. The pinned-library audit confirms the evaluation path above. Class-ordered labeled indices, a sorted carve, and an unshuffled validation loader mean the evaluated 256 images cover classes 0–50 with five images each and one image of class 51; classes 52–99 are absent. Per-run evaluated-ID logs do not survive, so this class coverage is source-reconstructed. A new trainer version sets `drop_last=False` on evaluation loaders only and passes coverage checks on 500, 10,000, and 257 items. No trained checkpoints or per-example predictions survive from the old sweep, so its full-500 ranking cannot be repaired from summary metrics. Restoring the omitted canonical cell produces a 66-candidate ledger and leaves the historical winner unchanged, but does not repair its validation scope. A corrected rerun has not been completed. Test values were evaluated at intermediate checkpoints; the accurate claim is that test was not used for selection, not that it was never evaluated.

I.3 Separate extensions, retained only for interpretive completeness

An age-aware modification of InfoBatch initially gives $+.683$ pp on six CIFAR-100-LT seeds, but a four-seed replication gives $-.488$ pp. Pooled IF100 effect is $+.215$ pp with CI $[-.292, .722]$, while IF50 is negative. It is a separate temporal extension and supplies no positive evidence for the LACS confidence rule. Historical multiplicative-weighting label-noise experiments also give mixed or adverse results; they are not current hard-support robustness benchmarks. Neither branch justifies a general claim that confident mistakes are mislabeled or should always be removed.

J Reproducibility, scope, and responsible reporting

This supplement retains the definitions, protocols, uncertainty, and controls needed to interpret the completed evidence. The accompanying manuscript-source package contains the numerical LaTeX tables and plot assets used here, together with available analytical-figure scripts. It reproduces the document; it is not a complete training-code, raw-data, or statistical-recomputation release. Internal workspace paths, job logs, and author verification tasks are maintained separately, not as a paper bibliography. Unrecorded quantities, such as selected-prefix fitting time and single-run SD, remain unavailable rather than reconstructed without evidence.

Selection can disproportionately affect minority or hard examples even when it does not read labels. Accuracy aggregated over a task is therefore insufficient for deployment decisions. The long-tail class behavior, provisional semi-supervised observations, and evaluation audit motivate subgroup

checks, refreshed scores, complete evaluation sets, and realistic cost accounting. The paper makes no medical or other high-stakes deployment claim. Its principal empirical conclusions concern the reported finite benchmark panels and implementations, not a guarantee that sparse selection preserves performance in new settings.

K Extended related work: what is and is not new

Curricula, easy/hard emphasis, and robust influence. Self-paced learning explicitly changes which examples are presented as training progresses (Kumar et al., 2010); OHEM and Focal Loss instead emphasize hard examples and reduce the relative contribution of easy ones (Shrivastava et al., 2016; Lin et al., 2017). MentorNet learns a data-driven curriculum under corrupted labels (Jiang et al., 2018). These methods establish that example difficulty can be used to reshape optimization, but their direction differs from LACS on confidently wrong examples: hard-example methods retain or emphasize them. RobustBoost and ArchBoost occupy the complementary side, reducing the influence of persistently misclassified or outlying points through label-signed margins and robust nonconvex potentials/losses (Freund, 2009; Li & Bradic, 2018). LACS combines neither objective directly; its unsigned confidence mask can remove both confidently correct and confidently wrong rows without reading the label at the selection event.

Learning signals, static pruning, and dynamic reuse. Active Bias uses the history of true-class predictive probabilities to emphasize informative examples (Chang et al., 2017). Forgetting events provide another trajectory-based criterion (Toneva et al., 2019); Data Diet studies early gradient-norm and prediction-error scores (Paul et al., 2021). Pruning has also been studied through its effects on neural scaling, including scoring approaches that need not use class labels (Sorscher et al., 2022). These works preclude novelty claims for unequal example utility, label-unaware scoring alone, or improved accuracy after removal. LACS instead analyzes a particular current-population normalization and clipping rule. Dynamic pruning (Raju et al., 2021) and repeated random sampling (Okanovic et al., 2024) additionally make reselection and re-entry established baselines rather than LACS contributions.

Loss-based pruning and recent scoring alternatives. Selective-Backprop uses a recent-loss percentile to set acceptance probabilities (Jiang et al., 2019); it is already distribution-aware. InfoBatch removes a stochastic fraction of its low-loss population and compensates retained contributions (Qin et al., 2024). RCAP uses class-aware allocation and probabilistic selection (Hassan et al., 2025). SeTa selects through loss-based difficulty groups and a moving window (Zhou et al., 2025), while RePB addresses inconsistencies associated with score drift and training history (Zhou et al., 2026a). BLS constructs per-example importance proxies from an exponential moving average of observed batch losses (Zhou et al., 2026b); it is a scoring component that can be combined with dynamic-pruning frameworks, not evidence of a LACS comparison we have run. Against this literature, the relevant LACS question is the effect of an unsigned confidence cutoff that responds to score separation and mass. None of these relationships establishes superiority without a corresponding controlled comparison.

Arithmetic mean versus log-mean-exp. The arithmetic mean is not "blind" to mass: changing the fraction of high- or low-score points changes the mean. The difference is that a mean retains only the first moment. For the same confidence scores,

$$\mathrm{LME}_\tau(c) = \tau \log\left(\frac{1}{n}\sum_i e^{c_i/\tau}\right) \geq \bar{c},$$

with equality only for a constant population. More revealingly,

$$T(c) - c_{\max} = \tau \log \frac{M_\tau(c)}{\rho n}.$$

Thus LACS's boundary is controlled directly by the *near-maximum mass*: isolated upper scores keep M_τ small and can cross the boundary, whereas a broad concentration near the maximum raises the threshold. Two distributions can share the same arithmetic mean while having different LME because their upper-tail dispersion differs. This is the sense in which the LACS reference

Table 37: **How the selection boundary is determined.** "No-prune" describes the rule itself, not a guarantee that an implementation never samples zero rows by chance. InfoBatch details follow its ICLR 2024 mean-loss policy; SeTa details follow the CVPR 2025 method and official implementation.

Method	score	boundary / selection	cardinality / no-prune	retained-example treatment
LACS	max predictive confidence	deterministic upper tail above log-mean-exp + margin	endogenous; no prune iff $M_\tau > \rho n$	hard 0/1 support, or declared continuous u_i variant
InfoBatch	per-sample loss	loss below arithmetic mean is eligible; each eligible row is thinned with a configured stochastic pruning probability	realized count depends on both the below-mean population and random thinning	retained low-loss rows are rescaled to correct gradient expectation; final full-data phase
SeTa	per-sample loss	configured random reduction, 1D k-means difficulty groups, then an easy-to-hard sliding window over groups	controlled by keep ratio, window scale, schedule, and cluster sizes; no analogous closed-form distribution condition for nontrivial settings	selected rows are used as a hard subset
Fixed quantile	chosen scalar score	order statistic at a prescribed quota	quota fixed in advance (modulo ties)	hard subset
Absolute cutoff	chosen scalar score	fixed score value	data-dependent count; no prune if all scores stay below cutoff	hard subset

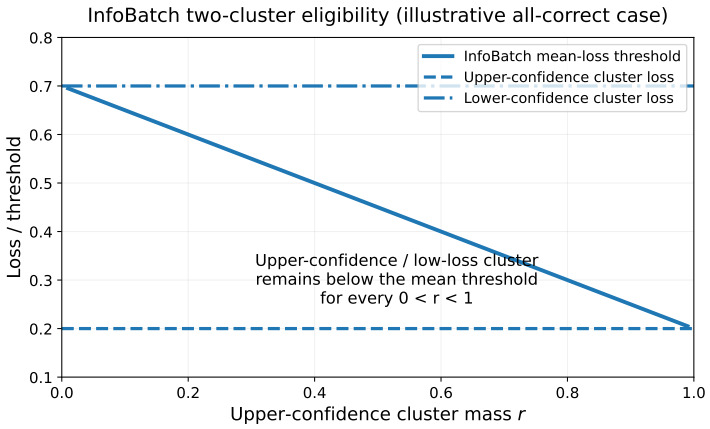


Figure 4: **InfoBatch two-cluster eligibility in an illustrative all-correct case.** If the upper-confidence cluster has lower loss than the lower-confidence cluster, the arithmetic mean loss lies strictly between the two cluster losses for every nontrivial mass $0 < r < 1$. The low-loss upper cluster is therefore always eligible for InfoBatch's configured stochastic thinning; its expected removed mass is governed by the thinning probability and eligible-pool mass rather than by a LACS-style gap-and-mass no-pruning boundary. This is a conceptual comparison of selection rules, not a claim that either rule is superior.

is distribution-sensitive beyond a first-moment threshold; it should not be described as recovering the full distribution.

InfoBatch is therefore closer to LACS than SeTa in one important sense: both compare a current per-example score to a population-relative reference. The references are nevertheless different. InfoBatch's arithmetic mean partitions a *loss* population into an eligible low-loss set and then prunes that set randomly; if two loss populations have the same arithmetic mean, the eligibility boundary itself is unchanged even though the composition below it can differ. LACS uses a log-mean-exp reference plus an explicit positive margin on *confidence*; even at a fixed mean confidence, its boundary can change with dispersion and upper-tail shape. It then deterministically zeros the sufficiently separated upper tail. InfoBatch's random rescaling is designed to recover the original gradient in expectation, whereas LACS does not impose that objective. SeTa is distribution-dependent through loss clustering, but its operative window is determined by the configured grouping/window schedule

rather than a closed-form score threshold. Thus saying that SeTa has "no threshold" is best understood as *no explicit scalar threshold of the LACS/InfoBatch form*; its cluster boundaries are still implicit, data-dependent boundaries. Likewise, SeTa does not literally prune under every possible parameter setting, so the useful contrast is the absence of a distribution-derived no-pruning criterion analogous to Eq. (7), not an unconditional statement that it must always discard data.

Coordinate-aware comparison of population means. For a CE score evaluated at a specified model state, let $v_i = p_\theta(y_i \mid x_i)$. Then

$$\bar{\ell} = -\frac{1}{n}\sum_i \log v_i = -\log\left[\left(\prod_i v_i\right)^{1/n}\right].$$

Consequently, $\ell_i < \bar{\ell}$ is equivalent to v_i exceeding their geometric mean. This transformed reference can change when probability dispersion changes even at a fixed arithmetic mean of v. InfoBatch is therefore not generally insensitive to probability-distribution shape. LACS instead applies an exponential quasi-arithmetic mean to $c_i = \max_k p_\theta(k \mid x_i)$, with an additional margin; $v_i = c_i$ only for correctly predicted rows. Actual InfoBatch scores may also be cached from different visits. The relevant distinction combines the score, its population transformation, deterministic versus stochastic removal, and survivor compensation. It is not a claim that an arithmetic loss mean uses less distributional information in every coordinate system.

Gradient approximation, importance sampling, and validation targets. CRAIG selects weighted coresets to approximate full-training gradients (Mirzasoleiman et al., 2020). GradMatch directly optimizes gradient discrepancy and supplies both budgeted and residual-tolerance selection formulations (Killamsetty et al., 2021a). GLISTER instead uses a validation-based bilevel criterion (Killamsetty et al., 2021b). Importance sampling can use a tractable gradient-norm bound rather than full per-example gradient vectors (Katharopoulos & Fleuret, 2018). Thus neither adaptive cardinality nor avoiding full gradient embeddings is unique to LACS. Our diagnostic asks whether the support of a different scalar rule happens to retain a useful gradient representation. Fitted versus normalized coefficients and hard versus continuous support remain distinct; a small checkpoint residual is not a full training reproduction of any of these methods.

Boosting and influence control. AdaBoost, stochastic gradient boosting, and FilterBoost established stagewise weighting or selection long before this work (Freund & Schapire, 1997; Friedman, 2002; Bradley & Schapire, 2007). GOSS and MVS study sampled tree-fitting statistics (Ke et al., 2017; Ibragimov & Gusev, 2019), whereas SnapBoost studies heterogeneous hypothesis classes (Parnell et al., 2020). Robust boosting also makes the treatment of difficult examples a prior research question (Freund, 2009). LACS does not introduce boosting by filtering, heterogeneous weak learners, or robustness through influence control. Its evaluated selection procedures use unsigned confidence with explicitly identified cutoff and fitting-weight contracts. Preserving a sampled gradient expectation does not generally reproduce a nonlinear fitting operation, and this distinction does not imply that an InfoBatch-inspired boosting adaptation is impossible.

The specific LogitBoost trimming precedent. Section 9 of Friedman et al. (2000) trims low working weights by cumulative weight mass and permits rows to re-enter at subsequent iterations. Its LogitBoost discussion explicitly distinguishes label-free $p(1-p)$ weights from other boosting weights that depend on the signed margin. Thus the absence of y_i from the selection score is not unique to LACS, and correct-only imbalance is a prior motivation as well as a useful present diagnostic. Our normalized cutoff controls exponential-confidence concentration, with the separate cardinality bound described above. Comparative accuracy requires a faithful implementation-level experiment.

Calibration, sparse mappings, and acquisition. Confidence need not estimate correctness (Guo et al., 2017); LACS temperature acts across examples, not class logits. Sparsemax projects scores onto the simplex to obtain sparse outputs (Martins & Astudillo, 2016); LACS instead clips weights of relatively high-confidence examples. Info-Coevolution addresses data acquisition and adaptive annotation (Qin et al., 2025), rather than skipped updates on a fixed labeled set. LDAM-DRW and Balanced Softmax modify the supervised long-tail loss (Cao et al., 2019; Ren et al., 2020);

FlexMatch modifies pseudo-label acceptance (Zhang et al., 2021). These objectives are related but not interchangeable with LACS selection.