Prune to Protect: Faster Training and Enhanced Privacy by Dynamic Data Pruning

Chinmay Joshi [1] **Advait Gadhikar** [2] **Celia-Rubio Madrigal** [3]
Aneet Kumar Dutta [3] **Mridula Singh** [3] **Rebekka Burkholz** [3]

Abstract

Deep networks memorize parts of their training data, exposing them to privacy attacks such as membership inference. The most vulnerable samples are often hard-to-learn tail examples: they are fitted late, carry high information for generalization, and are difficult to protect without sacrificing performance. We show that this privacy–utility tension can be mitigated by learning such samples earlier. To this end, we propose Weighted Loss InfoBatch (WLIB), which dynamically prunes easy samples while re-weighting hard ones according to sample-wise pruning ratios. By adjusting the influence of informative samples throughout training, WLIB jointly reduces memorization, improves robustness to membership inference attacks, and speeds up optimization. Extensive experiments demonstrate that WLIB achieves a favorable combination of privacy, utility, and training efficiency.

1. Introduction

Deep neural networks achieve state-of-the-art performance by training heavily overparameterized architectures on massive datasets (Zhang et al., 2017). This scaling results in *memorization* of some training examples (Feldman and Zhang, 2020). Such example-specific dependence creates a privacy risk such as Membership Inference Attacks (MIAs) (Shokri et al., 2017; Zarifzadeh et al., 2024; Carlini et al., 2022a). Consequently, highly memorized examples are especially vulnerable to MIAs; recent work shows that per-example memorization strongly predicts per-example MIA vulnerability (Choi et al., 2023).

Memorization is highly non-uniform. It concentrates on atypical, hard-to-learn examples in the tail of the data distribution. However, learning, and sometimes memorizing, these long-tail examples is often necessary for strong generalization (Feldman and Zhang, 2020; Feldman, 2020; Zhou et al., 2025; Brown et al., 2021). This creates a fundamental tension as the examples that support generalization also carry the highest privacy risk. We therefore ask whether a training-time procedure based on dynamic data pruning can reduce tail memorization, and hence privacy leakage, without sacrificing the generalization performance.

Dynamic data pruning provides a mechanism to control sample influence during training. InfoBatch (Qin et al., 2023) uses per-example loss at each epoch to prune low-loss easy examples while retaining high-loss hard examples followed by a small duration of training with the full-dataset. While designed for efficiency, this rule changes per-sample optimization dynamics as hard examples receive more gradient updates than easy examples. We investigate if this exposure asymmetry can reduce privacy-relevant memorization.

Our analysis yields three findings. First, InfoBatch reduces and redistributes hard-example memorization. Under standard training, hard examples form much of the high-memorization tail, while InfoBatch shifts their mass towards lower and intermediate scores. Second, this effect is phase-dependent: pruning delays the onset of memorization, but memorization rises sharply during the full-dataset training phase (annealing) of InfoBatch. This is driven mainly by hard examples whose memorization has remained low earlier in training. Third, the reduction in hard-example memorization improves robustness to MIAs: less-memorized hard examples have higher predictive entropy, weakening the member–non-member signal exploited by the attacker, especially in the low-false-positive-rate regime.

These findings show that InfoBatch can improve training efficiency while reducing privacy-relevant memorization, but they also reveal a late-stage vulnerability: when pruning is annealed and full-dataset exposure is restored, memorization rises sharply, mainly on hard examples. We address this

[1]Saarland University, Germany [2]Microsoft Research, India [3]CISPA Helmholtz Center for Information Security, Germany. Correspondence to: Chinmay Joshi <chjo00003@stud.uni-saarland.de>, Celia-Rubio Madrigal <celia.rubio-madrigal@cispa.de>.

Published at ICML 2026 Workshop on the Impact of Memorization on Trustworthy Foundation Models, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

with WLIB—**W**eighted **L**oss **I**nfo**B**atch—which modifies the annealing phase. At the start of annealing, WLIB computes each example's cumulative pruning rate, normalizes the rates to unit mean, and fixes them as multiplicative loss weights for the rest of training. This uses pruning history to reweight sample influence on the loss as frequently retained hard examples receive smaller weights, reducing their late-stage memorization increase, while frequently pruned easy examples receive larger weights, preserving the accuracy gains from annealing.

We summarize our contributions as follows. (i) We show that InfoBatch has privacy-relevant effects beyond efficiency: it reduces and redistributes hard-example memorization, shifting mass away from the high-memorization tail. (ii) We identify annealing as the memorization-critical phase of InfoBatch. As pruning is annealed, hard, low-pruning-rate examples drive a sharp late-stage increase in memorization. (iii) We propose WLIB, a weighted-loss variant of InfoBatch that uses pruning history to rebalance sample influence during annealing. WLIB improves MIA robustness over standard training and InfoBatch while preserving InfoBatch-level accuracy and training efficiency.

2. Related work

Memorization in deep networks. The current discussion of memorization in deep networks begins with the observation that overparameterized networks fit even random labels while still generalizing on natural data (Zhang et al., 2017; Arpit et al., 2017). (Feldman, 2021) argues that memorization can be necessary for near-optimal generalization on long-tailed distributions, and (Feldman and Zhang, 2020) formalize a per-example notion through a leave-one-out estimator that we adopt; subsequent work uses related estimators to study how memorization scales and shifts across training (Zhang et al., 2023; Biderman et al., 2023). Memorization is non-uniform across examples, scales, and architectures (Lukasik et al., 2023; Carlini et al., 2023), can be localized to a small subset of neurons rather than confined to final layers (Maini et al., 2023; Stoehr et al., 2024), and is closely tied to forgetting and training-order dynamics (Toneva et al., 2018; Jagielski et al., 2022). On the privacy side, (Carlini et al., 2022b) shows that the most vulnerable examples form an outer layer whose removal exposes new layers, and (Choi et al., 2023) links per-example memorization to membership-inference vulnerability. In this work, we empirically show that reduced memorization has privacy benefits.

Membership inference attacks. (Shokri et al., 2017) introduced membership inference for modern models, and the literature has progressively moved from confidence- and loss-based attacks to stronger statistical tests (Yeom et al., 2018; Sablayrolles et al., 2019; Ye et al., 2022). (Carlini et al., 2022a) introduced LiRA, emphasizing the low-FPR regime, and (Zarifzadeh et al., 2024) introduced RMIA, which uses pairwise likelihood-ratio tests and achieves stronger performance at low FPRs and under limited reference budgets. For our setting, RMIA is the most appropriate privacy lens because it provides the strongest and most stable auditing procedure among the attacks considered in prior work.

Data pruning. Static data pruning selects a subset before training using forgetting scores (Toneva et al., 2018), early gradient or error signals (Paul et al., 2023), generalization influence (Yang et al., 2022), coreset objectives (Mirzasoleiman et al., 2020; Sorscher et al., 2023), validation- or gradient-matching objectives (Killamsetty et al., 2021a;b), or proxy models (Coleman et al., 2020). Because example informativeness changes during training (Loshchilov and Hutter, 2015; Jiang et al., 2019), dynamic pruning methods re-evaluate utility over time (Raju et al., 2021; Mindermann et al., 2022; Qin et al., 2023), and recent work explores model-aware and diversified batch selection (Yu et al., 2024; Hong et al., 2024). Among these, InfoBatch is the cleanest substrate for our purpose because its rule stays close to standard SGD and changes sample exposure rather than the training objective. While the literature treats pruning as an efficiency mechanism; we study its effects on memorization.

Closest to our setting, (Ye et al., 2026) study static pretraining corpus pruning and show that it reshapes language-model memorization by changing retained-data frequency. In contrast, we study dynamic pruning and, to our knowledge, we are the first to analyze its joint effect on memorization and membership-inference robustness.

3. WLIB: Weighted Loss InfoBatch

We first introduce InfoBatch(Qin et al., 2023), a dynamic data pruning method. Each example $z = (x, y)$ maintains a score $H_t(z)$ initialized to a large value and updated to the current loss when the example is seen. At each epoch, examples whose score is below the epoch-wise mean $\bar{H}_t$ are considered easy and pruned with probability r; examples at or above the mean are always retained. Equivalently, the pruning probability, denoted $P_t(z)$, is

$$\begin{aligned} P_t(z) &= r \text{ if } H_t(z) < \bar{H}_t \text{ (\textit{easy} examples)}, \\ P_t(z) &= 0 \text{ if } H_t(z) \geq \bar{H}_t \text{ (\textit{hard} examples)}. \end{aligned} \tag{1}$$

When an easy example is retained, its loss is rescaled by the inverse retention probability $(1 - r)^{-1}$, so that the expected gradient contribution remains close to that of the full dataset. Pruned examples carry forward their last known score; retained examples update their score per epoch.

A final annealing phase covering the final $1 - \delta$ fraction of training trains with the full dataset and is crucial to maintain generalization performance. However, as we observe in our

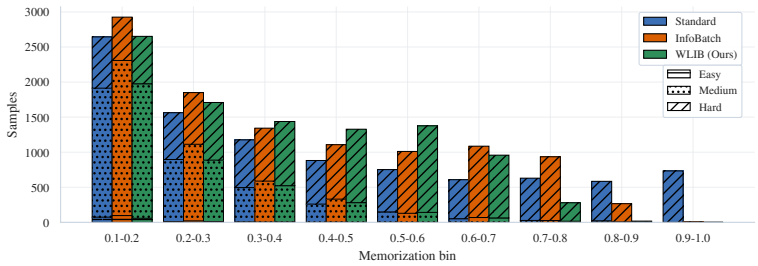


Figure 1. Distribution of final per-sample memorization scores on CIFAR-10 (scores ≥ 0.1), stacked by difficulty group via hatching. InfoBatch and WLIB progressively depopulate the extreme bins occupied by standard training.

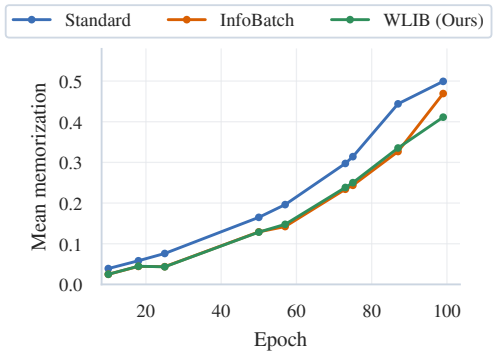


Figure 2. Mean memorization trajectory on CIFAR-10 within the hard group ($\rho \in [0.00, 0.21]$). InfoBatch and WLIB suppress memorization.

experiments, it is also the stage in which memorization rises sharply.

We now introduce WLIB, which leverages dynamic data pruning to prevent the model from memorizing the hard tail while maintaining overall accuracy. Memorization concentrates on atypical, hard-to-learn examples in the long tail of the training distribution (Feldman and Zhang, 2020). Standard training quickly learns easy examples and subsequently focuses its capacity on fitting these hard examples, which drives extreme memorization later in the training (Arpit et al., 2017). WLIB alters this dynamic.

It operates in two distinct phases to prevent tail memorization while maintaining accuracy. The initial pruning phase, inherited from InfoBatch, computes the mean training loss at each epoch, stochastically skips easy examples with probability r, and consistently retains hard examples.

Optimization eventually requires a full-dataset annealing phase to recover generalization, but restoring uniform exposure triggers a sharp memorization surge concentrated precisely on the previously protected hard examples (elaborated in 4). To prevent this surge, WLIB uses the pruning history to reweight the loss. Let $\rho_i = c_i/T_s$ be the cumulative pruning rate, where c_i is the total number of times the algorithm skipped example i over the T_s total epochs of the pruning phase. During the annealing phase, WLIB disables stochastic pruning and leverages the cumulative sample exposure to weigh the loss:

$$\mathcal{L}_{\text{WLIB}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left(\frac{\max(\rho_i, \rho_{\min})}{\frac{1}{N} \sum_{j=1}^{N} \max(\rho_j, \rho_{\min})} \right) \ell_i \quad (2)$$

where $\mathcal{B}$ is the minibatch, ℓ_i is the standard per-example loss, N is the dataset size, and $\rho_{\min}$ is a small constant. This formulation assigns small loss weights to hard examples to safely maintain their favorable pre-annealing memorization state, while easy examples receive large weights to reinstate their influence and successfully recover model accuracy.

4. Analysis and Results

We first study InfoBatch in two phases: the pruning phase and the annealing phase. We group examples into *easy*, *medium*, and *hard* by their cumulative pruning rate ρ: low ρ identifies samples whose loss persistently exceeds the epoch-wise mean and are therefore retained on most epochs. Memorization is estimated via the subset-sampling estimator of Feldman and Zhang (2020) on CIFAR-10 with ResNet-20 ($M = 400$ models per method); privacy is measured via RMIA (Zarifzadeh et al., 2024) on CIFAR-10, CIFAR-100, and ImageNet, reporting AUC and true-positive rate (TPR) at fixed low false-positive rates (FPR). Full setup is in Appendix A.

Pruning reduces hard-sample memorization despite increased exposure. Hard examples are retained more often during pruning and thus receive higher relative exposure than easy examples. Intuitively, higher relative exposure should increase memorization; instead, Figure 2 shows that InfoBatch keeps hard-group mean memorization below standard training throughout, converging to 0.4693 vs. 0.4993 (WLIB coincides with InfoBatch during pruning). Easy and medium groups stay near zero under all methods (Appendix C).

This counterintuitive decrease in memorization in InfoBatch is structural. Figure 1 shows the per-example memorization distribution at the end of training: under standard training, hard examples populate a tail extending to $[0.9, 1.0]$, while under InfoBatch, the $[0.9, 1.0]$ bin is empty and the $[0.8, 0.9]$ bin shrinks, with mass redistributed into intermediate bins. These extreme bins are the most vulnerable to membership inference (Appendix E), and this privacy-relevant tail is redistributed by the pruning phase.

One possible explanation for this redistribution is how memorization relates to how the loss distribution changes. In standard training, the model first learns easy samples and

Table 1. RMIA results and test accuracy across datasets and methods (%); best per dataset and column in bold. AUC and TPR columns are computed under the standard RMIA auditing protocol, where target and shadow models are trained on random 50% subsets of the dataset; Test accuracy reports a separately trained full-data model under the same training procedure. CIFAR-10 and CIFAR-100 use online RMIA with 64 shadow models, averaged over 20 target models; ImageNet uses offline RMIA with 2 shadow models with $a = 1.0$, averaged over 4 target models. All settings use $\gamma = 2.0$.

Dataset	Method	AUC ↓	TPR@0% FPR ↓	TPR@0.01% FPR ↓	TPR@1% FPR ↓	Test Acc. ↑
CIFAR-10 (ResNet-20)	Standard	71.82 ± 2.95	1.87 ± 1.08	3.15 ± 1.17	13.76 ± 2.87	**92.21**
	InfoBatch	66.69 ± 3.82	0.74 ± 0.55	1.25 ± 0.67	8.47 ± 2.88	91.74
	WLIB (Ours)	$\mathbf{61.43 \pm 4.61}$	$\mathbf{0.20 \pm 0.35}$	$\mathbf{0.40 \pm 0.43}$	$\mathbf{4.58 \pm 2.43}$	91.64
CIFAR-100 (WRN-28-2)	Standard	91.02 ± 0.25	4.89 ± 2.52	7.80 ± 2.03	34.74 ± 0.78	74.27
	InfoBatch	88.56 ± 0.28	1.81 ± 1.09	3.57 ± 1.39	25.20 ± 1.22	73.63
	WLIB (Ours)	$\mathbf{88.06 \pm 0.27}$	$\mathbf{1.22 \pm 0.70}$	$\mathbf{2.39 \pm 0.70}$	$\mathbf{21.59 \pm 1.06}$	**74.34**
ImageNet (ResNet-50)	Standard	73.7 ± 0.15	0.0	1.50 ± 1.7	17.4 ± 0.11	**76.53**
	InfoBatch	73.3 ± 0.04	0.0	1.13 ± 1.3	15.5 ± 0.05	75.95
	WLIB (Ours)	$\mathbf{72.9 \pm 0.16}$	0.0	$\mathbf{0.60 \pm 0.7}$	$\mathbf{12.9 \pm 0.13}$	75.85
ImageNet (DeiT-Small)	Standard	78.34 ± 0.72	0.0	0.0	19.4 ± 0.60	**72.9**
	InfoBatch	79.30 ± 0.09	0.0	0.0	19.5 ± 0.09	72.2
	WLIB (Ours)	$\mathbf{77.12 \pm 0.08}$	0.0	0.0	$\mathbf{17.3 \pm 0.14}$	72.5

focuses on learning and eventually memorizing the remaining hard examples in the long tail in later stages (Arpit et al., 2017). In contrast, training exclusively on hard samples creates a uniformly high loss across the data. Without the consistent gradient from easy samples, the gradient updates fluctuate, preventing the model from settling into the specific regions of the loss landscape required to memorize individual examples. InfoBatch induces a softer version of this regime: by stochastically pruning easy samples while retaining hard ones more often, it pushes the effective training distribution away from the long-tailed standard-training regime and toward a less concentrated hard-sample regime.

Memorization surges in the late phase, concentrated at low pruning rates. The pruning phase ends at epoch T_s. After this, InfoBatch trains on the full dataset with uniform per-example weights. We measure how the mean memorization changes once full exposure is restored. Between epochs 87 (start of the annealing) and 99 (last epoch) on CIFAR-10, the hard-group mean rises sharply ($+0.143$) under InfoBatch, as seen in Figure 7a in Appendix D. Easy and medium groups change negligibly. InfoBatch therefore *delays* hard-sample memorization: the pruning-phase reduction is partially undone once full exposure returns.

The surge is also non-uniform within the hard group. To characterize this, figure 3 plots the per-example memorization change against the pruning rate during annealing against ρ at epoch 87. The rise concentrates in the lower spectrum of the pruning rates WITHIN the hard example group. As observed, WLIB precisely downweights these samples and reduces the memorization accumulated here.

Privacy without compromising utility or efficiency. The redistributed memorization translates to membership-inference robustness, and WLIB achieves this without trading off either utility or training efficiency. Table 1 shows TPR at 1% FPR decreasing monotonically from standard to InfoBatch to WLIB across all settings: $13.76 \rightarrow 8.47 \rightarrow 4.58\%$ on CIFAR-10, $34.74 \rightarrow 25.20 \rightarrow 21.59\%$ on CIFAR-100, $17.4 \rightarrow 15.5 \rightarrow 12.9\%$ on ImageNet (ResNet-50), and $19.4 \rightarrow 19.5 \rightarrow 17.3\%$ on ImageNet (DeiT-Small). Simultaneously, it preserves utility: test accuracy under WLIB stays within 0.1 pp of InfoBatch on CIFAR-10 and ImageNet, and slightly exceeds standard training on CIFAR-100. Efficiency is inherited from InfoBatch as the pruning phase is unchanged, with $r = 0.5$ on the first $\delta = 0.875$ fraction of epochs yielding $\sim 35\%$ effective data reduction and $\sim 20\%$ wall-clock savings over standard training.

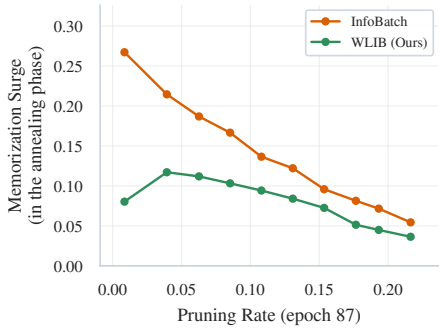


Figure 3. Mean memorization surge during the annealing phase within the CIFAR-10 hard examples, plotted against the cumulative pruning rate ρ anchored at epoch 87 (beginning of the annealing phase). The late-phase rise is largest at low ρ and is suppressed most strongly there by WLIB. Full range of pruning rate vs memorization surge during the annealing phase in Figure 7b in Appendix D

5. Conclusion

Dynamic data pruning has the ability to suppress memorization on the hard examples that carry the most privacy risk. For this to happen, those examples are exposed more often than the rest, as we see in InfoBatch. This effect lives almost entirely in the pruning phase, but the full-data annealing phase that follows reverses these benefits to a large extent. We introduce WLIB, which closes this gap with a pruning-history based loss re-weighting that holds the favorable memorization state from the pruning phase in place. Moreover, it does so without giving up the accuracy or the efficiency offered by InfoBatch. This reduced memorization further unlocks privacy benefits, as we observe in our experiments on membership-inference attacks across various image classification datasets.

References

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, 2017. URL https://arxiv.org/abs/1611.03530.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: discovering the long tail via influence estimation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models . In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, Los Alamitos, CA, USA, May 2017. IEEE Computer Society. doi: 10.1109/SP.2017.41. URL https://doi.ieeecomputersociety.org/10.1109/SP.2017.41.

Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles, 2022a. URL https://arxiv.org/abs/2112.03570.

Jihye Choi, Shruti Tople, Varun Chandrasekaran, and Somesh Jha. Why train more? effective and efficient membership inference via memorization. *ArXiv*, abs/2310.08015, 2023. URL https://api.semanticscholar.org/CorpusID:263909462.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 954–959, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369794. doi: 10.1145/3357713.3384290. URL https://doi.org/10.1145/3357713.3384290.

Mo Zhou, Haoyang Ma, and Rong Ge. Memorizing long-tail data can help generalization through composition, 2025. URL https://arxiv.org/abs/2510.16322.

Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. When is memorization of irrelevant training data necessary for high-accuracy learning? In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC '21, page 123–132. ACM, 2021. doi: 10.1145/3406325.3451131. URL http://dx.doi.org/10.1145/3406325.3451131.

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, 2023. URL https://arxiv.org/abs/2303.04947.

Devansh Arpit, Stanisław Jastrzbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 233–242. JMLR.org, 2017.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail, 2021. URL https://arxiv.org/abs/1906.05271.

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Counterfactual memorization in neural language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. Emergent and predictable memorization in large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Michal Lukasik, Aditya Krishna Menon, Ankit Singh Rawat, Vaishnavh Nagarajan, and Sanjiv Kumar. What do large

networks memorize? In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=QcA9iGaLpH4.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models, 2023. URL https://arxiv.org/abs/2202.07646.

Pratyush Maini, Michael C. Mozer, Hanie Sedghi, Zachary C. Lipton, J. Zico Kolter, and Chiyuan Zhang. Can neural network memorization be localized? In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Niklas Stoehr, Mitchell A. Gordon, Chi Zhang, and Owen Lewis. Localizing paragraph memorization in language models. *ArXiv*, abs/2403.19851, 2024. URL https://api.semanticscholar.org/CorpusID:268793619.

Mariya Toneva, Alessandro Sordoni, Rémi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. *ArXiv*, abs/1812.05159, 2018. URL https://api.semanticscholar.org/CorpusID:55481903.

Matthew Jagielski, Om Thakkar, Florian Tramèr, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Chiyuan Zhang. Measuring forgetting of memorized training examples. *ArXiv*, abs/2207.00099, 2022. URL https://api.semanticscholar.org/CorpusID:250243645.

Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. The privacy onion effect: Memorization is relative, 2022b. URL https://arxiv.org/abs/2206.10469.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting, 2018. URL https://arxiv.org/abs/1709.01604.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, Proceedings of Machine Learning Research, pages 5558–5567. PMLR, 2019. URL http://proceedings.mlr.press/v97/sablayrolles19a.html.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 3093–3106, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450394505. doi: 10.1145/3548606.3560675. URL https://doi.org/10.1145/3548606.3560675.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training, 2023. URL https://arxiv.org/abs/2107.07075.

Shuo Yang, Zeke Xie, Hanyu Peng, Minjing Xu, Mingming Sun, and P. Li. Dataset pruning: Reducing training data by examining generalization influence. *ArXiv*, abs/2205.09329, 2022. URL https://api.semanticscholar.org/CorpusID:248887235.

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*, ICML'20. JMLR.org, 2020.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: beating power law scaling via data pruning, 2023. URL https://arxiv.org/abs/2206.14486.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning, 2021a. URL https://arxiv.org/abs/2012.10630.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, Abir De, and Rishabh K. Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, 2021b. URL https://api.semanticscholar.org/CorpusID:235421747.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning, 2020. URL https://arxiv.org/abs/1906.11829.

Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. *ArXiv*, abs/1511.06343, 2015. URL https://api.semanticscholar.org/CorpusID:5324823.

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

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

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15630–15649. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/mindermann22a.html.

Zichun Yu, Spandan Das, and Chenyan Xiong. Mates: Model-aware data selection for efficient pretraining with data influence models, 2024. URL https://arxiv.org/abs/2406.06046.

Feng Hong, Yueming Lyu, Jiangchao Yao, Ya Zhang, Ivor W. Tsang, and Yanfeng Wang. Diversified batch selection for training acceleration, 2024. URL https://arxiv.org/abs/2406.04872.

Jiayuan Ye, Vitaly Feldman, and Kunal Talwar. Cram less to fit more: Training data pruning improves memorization of facts, 2026. URL https://arxiv.org/abs/2604.08519.

Sree Harsha Nelaturu, Advait Gadhikar, and Rebekka Burkholz. TurboPrune: High-Speed Distributed Lottery Ticket Training. URL https://github.com/nelaturuharsha/TurboPrune.

Sunny Sanyal, Hayden Prairie, Rudrajit Das, Ali Kavis, and Sujay Sanghavi. Upweighting easy samples in fine-tuning mitigates forgetting, 2025. URL https://arxiv.org/abs/2502.02797.

A. Experimental setup

This section specifies the training, memorization-estimation, and membership-inference setup used throughout the paper. The same protocols are applied uniformly across standard training, InfoBatch, and WLIB; only the training procedure itself varies between methods. For InfoBatch and WLIB, the pruning ratio is $r = 0.5$ and the annealing parameter is $\delta = 0.875$, so that the annealing phase begins at epoch $\lfloor \delta \cdot T \rfloor$ for total epoch budget T. WLIB additionally clamps the pruning-rate-based weights at $\rho_{\min} = 0.05$ to avoid ignoring contributions completely from any samples.

A.1. Training configuration

Per-dataset training configurations are summarized in Table 2. All runs use a triangular learning-rate schedule, and our training pipeline builds on the TurboPrune framework (Nelaturu et al.).

Table 2. Training configurations across datasets and architectures.

Setting	CIFAR-10 ResNet-20	CIFAR-100 WRN-28-2	ImageNet ResNet-50	ImageNet DeiT-Small
Optimizer	SGD	SGD	SGD	AdamW
Momentum	0.9	0.9	0.9	0.9
Learning rate	0.2	0.1	0.25	5×10^{-3}
Weight decay	5×10^{-4}	3×10^{-4}	5×10^{-5}	0.1
Batch size	512	128	1024	1024
Epochs	100	100	100	300
Warmup steps	20	20	10	10

Hardware. CIFAR-10 and CIFAR-100 experiments were conducted on a single NVIDIA A6000 (48 GB). ImageNet experiments, for both ResNet-50 and DeiT-Small, were conducted on $4\times$ NVIDIA A100 GPUs (40 GB).

A.2. Memorization estimation

We measure memorization using the estimator from Feldman and Zhang (2020). For a training algorithm A and dataset $S = \{(x_i, y_i)\}_{i=1}^n$, the memorization of example i is defined as

$$\mathrm{mem}(A, S, i) = \Pr_{h \leftarrow A(S)}[h(x_i) = y_i] - \Pr_{h \leftarrow A(S \setminus i)}[h(x_i) = y_i]. \tag{3}$$

This quantity measures how much the model's correctness on (x_i, y_i) depends on that example being included in the training set. Since direct leave-one-out estimation is computationally expensive, they estimate memorization by training models on overlapping random subsets of the data. For a model m trained on subset T_m, let $a(\theta_{T_m}; i)$ indicate whether it correctly classifies example i. Memorization is estimated as

$$\widehat{\mathrm{mem}}(i) = \frac{1}{|\mathcal{M}_{\mathrm{in}}(i)|} \sum_{m \in \mathcal{M}_{\mathrm{in}}(i)} a(\theta_{T_m}; i) - \frac{1}{|\mathcal{M}_{\mathrm{out}}(i)|} \sum_{m \in \mathcal{M}_{\mathrm{out}}(i)} a(\theta_{T_m}; i), \tag{4}$$

where $\mathcal{M}_{\mathrm{in}}(i) = \{m : i \in T_m\}$ and $\mathcal{M}_{\mathrm{out}}(i) = \{m : i \notin T_m\}$.

We use $M = 400$ subsets on CIFAR-10, with each model trained under the configuration in Table 2. Scores are estimated separately for standard training, InfoBatch, and WLIB, with all 400 runs of each method using that method's training procedure.

A.3. Membership inference attack

We evaluate privacy risk using RMIA (Zarifzadeh et al., 2024), a membership inference attack that frames the problem as a pairwise hypothesis test.

By grounding the test in population-relative likelihoods rather than a fixed null, RMIA achieves stronger attack power than prior methods. We report the Area Under the ROC Curve (AUC), and True Positive Rates (TPR) at fixed False Positive Rates (FPR) of 0%, 0.01%, and 1%, following standard evaluation practice (Carlini et al., 2022a).

We follow the RMIA auditing protocol of Zarifzadeh et al. (2024): target and shadow models are trained on independent random 50% subsets of the training data such that every example is in exactly half of the subsets. The membership signal, reference distributions, and pairwise test are computed exactly as specified in Zarifzadeh et al. (2024).

For a target model θ and target example z, RMIA computes a membership score by comparing z against reference points $z' \sim \pi$ drawn from the population:

$$\mathrm{Score}_{\mathrm{RMIA}}(z;\theta) \;=\; \Pr_{z'\sim\pi}\left(\frac{\Pr(z \mid \theta)/\Pr(z)}{\Pr(z' \mid \theta)/\Pr(z')} \geq \gamma\right), \tag{5}$$

we use $\gamma = 2.0$ throughout. All target and shadow models are trained under the configuration in Table 2.

Our attack is *defense-unaware*: shadow models are always trained with standard training, while only the target model varies between standard, InfoBatch, and WLIB. InfoBatch and WLIB are treated as defenses applied at the target side, and the attacker has no knowledge of which (if any) is in use.

CIFAR-10 and CIFAR-100. Online RMIA with 64 shadow models per target. AUC and TPR values are averaged over 20 target models.

ImageNet. Offline RMIA with 2 shadow models and reference ratio $a = 1.0$. The offline variant precomputes OUT-side likelihoods and avoids retraining shadow models per query, which would be prohibitive at this scale. AUC and TPR values are averaged over 4 target models for both ResNet-50 and DeiT-Small. We refer the reader to Zarifzadeh et al. (2024) for the offline-mode derivation.

Test accuracy is reported separately from that of a model trained on the full training set under the same configuration.

B. Pruning rate as a hardness proxy

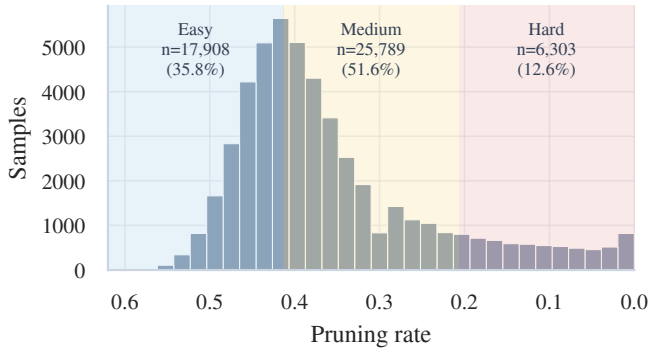


Figure 4. Distribution of ρ on CIFAR-10 under InfoBatch, partitioned into three equal-width bins defining easy (n=17,908), medium (n=25,789), and hard (n=6,303) groups. Lower ρ indicates samples persistently retained during training.

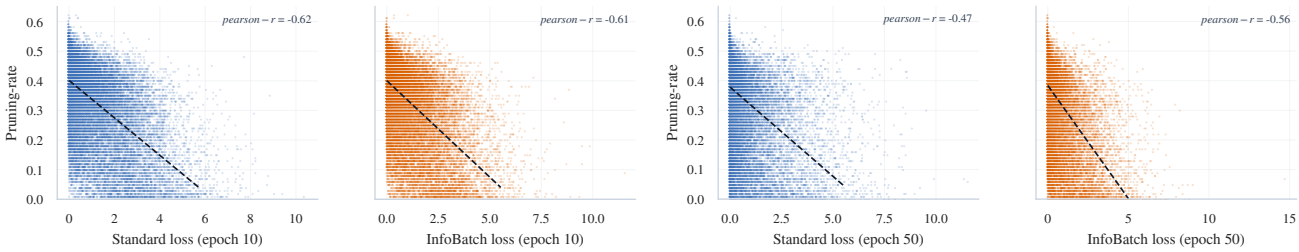
We use the cumulative pruning rate ρ_i (Eq. 6) as our aggregate measure of sample hardness. The choice is natural in our setting because ρ_i is computed from quantities InfoBatch already maintains, and it integrates relative difficulty over the entire training trajectory rather than a single-checkpoint snapshot.

A low pruning rate indicates that an example was persistently hard relative to the training set, while a high pruning rate indicates that it was consistently estimated to be easy, and therefore omitted often. We therefore divide examples into three pruning-rate tiers: the lowest third of the range is labeled *hard*, the middle third *medium*, and the highest third *easy*.

We define its cumulative pruning rate up to epoch T, $\rho_i(T)$, as

$$\rho_i(T) = \frac{1}{T}\sum_{t=1}^{T} p_i(t). \tag{6}$$

Per-sample loss is the established hardness proxy in data pruning and curriculum learning (Paul et al., 2023; Sanyal et al., 2025), and it is also the per-epoch quantity that drives InfoBatch's pruning decisions. To check that ρ behaves as a hardness signal in the same sense, we compare it to per-sample loss at two stages of training on CIFAR-10. Figure 5 shows the relationship at epoch 10 and epoch 50, for both standard training and InfoBatch.



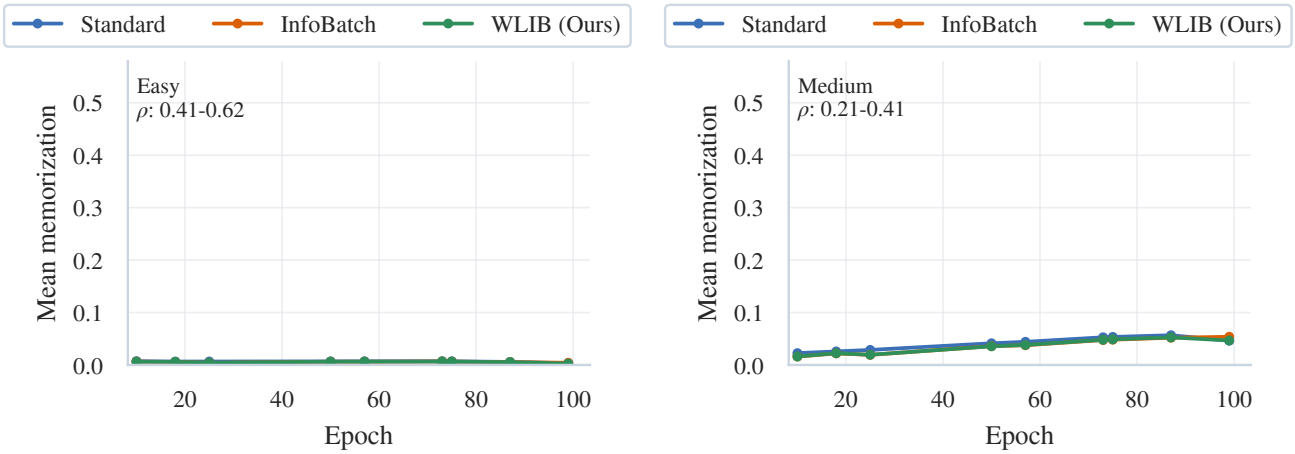
(a) Epoch 10 *(b)* Epoch 50

Figure 5. Pruning rate ρ vs. per-sample loss on CIFAR-10 at epochs 10 and 50, for standard training (blue) and InfoBatch (orange). Pearson coefficients between -0.47 and -0.62.

The relationship is monotone in both regimes: examples with high loss are pruned less often and accumulate low ρ, while examples with low loss are pruned more often and accumulate high ρ. The agreement also persists under standard training, where pruning decisions do not influence the measured loss, so the two signals agree on what they identify as "hard" even when computed independently. The advantage of ρ over a single-checkpoint loss is that the latter is informative only when the loss distribution is sufficiently spread; once a model has converged, training losses cluster near zero and lose discriminative power, while ρ continues to reflect the cumulative training history.

C. Easy and medium group memorization trajectories

The main paper reports the mean memorization trajectory only for the hard group, since this is the group on which the three methods diverge. Figure 6 shows the corresponding trajectories on CIFAR-10 for the easy ($\rho \in [0.41, 0.62]$) and medium ($\rho \in [0.21, 0.41]$) groups.



(a) Easy group, $\rho \in [0.41, 0.62]$. *(b)* Medium group, $\rho \in [0.21, 0.41]$.

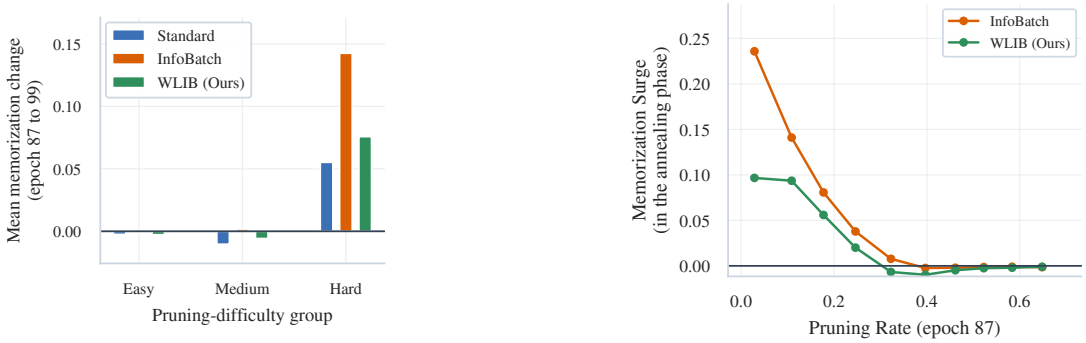
Figure 6. Mean memorization trajectories for the easy and medium groups on CIFAR-10. Standard training, InfoBatch, and WLIB are visually indistinguishable in both groups.

Both groups remain at low memorization throughout training under all three methods, and the curves overlap within each group. This is consistent with the observations from the main paper, i.e., dynamic pruning's effect on memorization is concentrated on the hard tail.

D. Memorization surge in the late phase

The pruning phase ends at T_s, after which InfoBatch trains on the full dataset with uniform per-example weights. We measure how memorization changes once full exposure is restored. Figure 7(a) shows the change in mean memorization between epoch 87 (start of annealing) and epoch 99 (last epoch) on CIFAR-10, separated by difficulty group. Easy and medium groups change negligibly under all three methods. The entire late-phase increase is concentrated in the hard group, where InfoBatch rises by $+0.143$, standard training by about a third of that, and WLIB by only $+0.076$.

Within the hard group, the surge is non-uniform. Figure 7(b) extends the within-hard-group view in the main paper to the full range of pruning rates. The surge is largest at low ρ and decays rapidly, becoming negligible once ρ exceeds roughly 0.3. Across the rest of the range, the InfoBatch and WLIB curves are essentially flat and overlapping. The late-phase memorization phenomenon is therefore a hard-sample effect, and WLIB's intervention does not perturb the memorization state of medium or easy examples; it acts where the surge actually occurs. This also justifies the asymmetric reweighting: because medium and easy examples are not at risk of late-phase memorization, they can be safely upweighted to restore a consistent gradient and close the performance gap left by pruning, without reintroducing the memorization that WLIB intends to suppress.



(a) Surge by difficulty group.

(b) Surge across full ρ range.

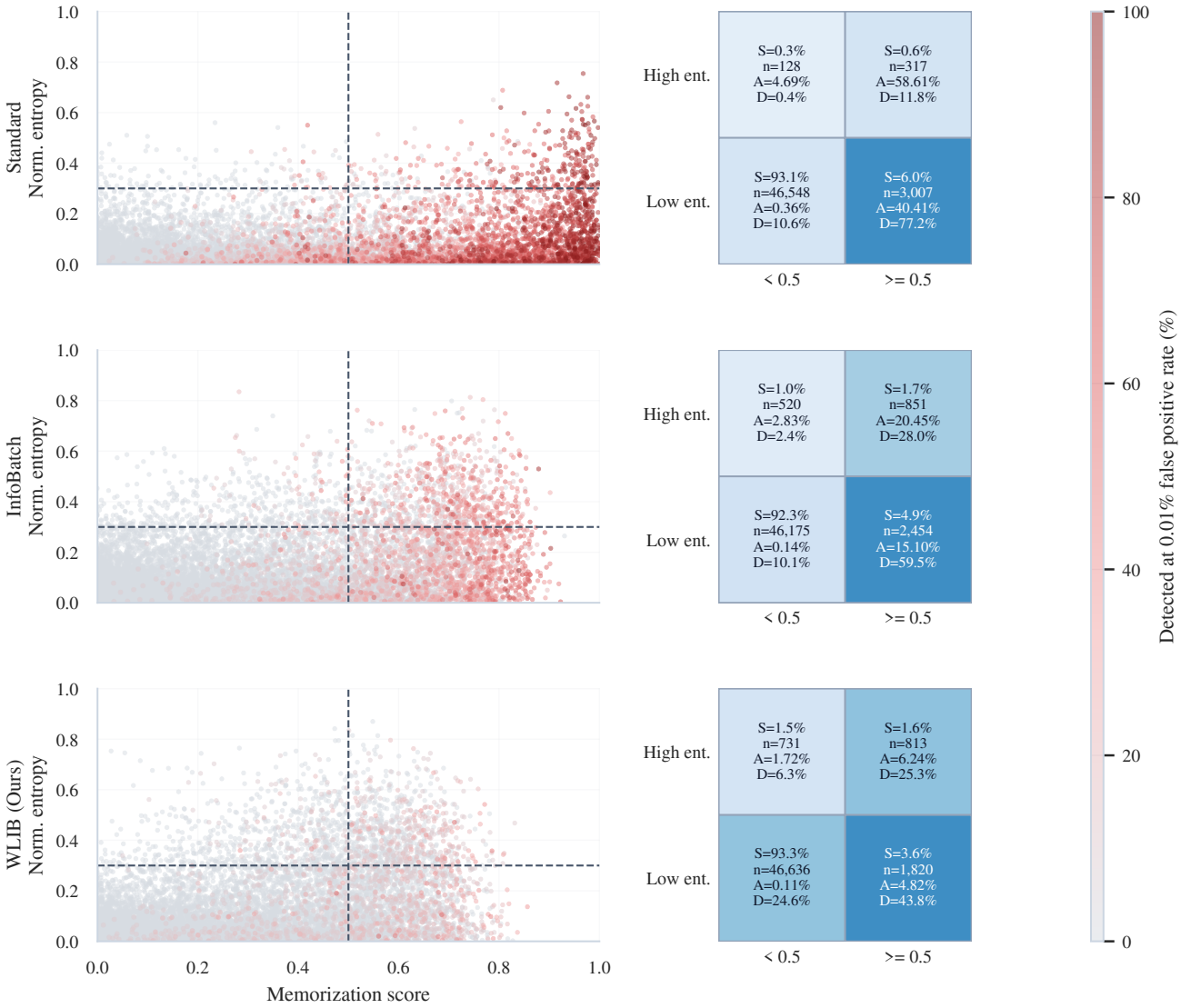
Figure 7. Late-phase memorization surge on CIFAR-10. (a) The rise is entirely a hard-sample phenomenon. (b) Within all samples, the rise concentrates at low ρ and decays past $\rho \approx 0.3$.

E. Memorization–entropy analysis of RMIA detections

For a trained model f_θ and input x_i, predictive entropy $H_\theta(x_i) = -\sum_{k=1}^{K} p_\theta(k \mid x_i) \ln p_\theta(k \mid x_i)$ measures the concentration of the output distribution: low entropy corresponds to a sharp, high-confidence prediction, and high entropy to a diffuse one. Since H_θ is bounded above by $\ln K$, we use the normalized form $\tilde{H}_\theta(x_i) = H_\theta(x_i)/\ln K \in [0, 1]$ for the analysis below.

The aggregate RMIA results in the main paper show that WLIB reduces leakage at all reported operating points. We now examine, on CIFAR-10, where in the per-sample landscape the detected members come from, and how that distribution shifts across methods. The analysis ties the aggregate numbers back to the memorization dynamics in the main paper: it shows that the samples RMIA detects are precisely the ones the redistribution of memorization acts on.

For each CIFAR-10 sample, we compute its memorization score (Appendix A.2) and its normalized predictive entropy on the trained model, and color each point by its RMIA detection rate at 0.01% FPR. To summarize the structure, we partition the plane into four quadrants at memorization 0.5 and normalized entropy 0.3, and annotate each quadrant with sample share S, count n, mean per-sample detection rate A, and the quadrant's share D of the total detected mass at 0.01% FPR.



MIA = membership-inference attack. S = sample share; A = average detection rate in quadrant; D = share of total detected mass at 0.01% FPR.

Figure 8. CIFAR-10 RMIA detection at 0.01% FPR in the memorization–entropy plane, partitioned into quadrants at memorization 0.5 and normalized entropy 0.3. Per-quadrant annotations: sample share S, count n, mean detection rate A, share D of total detected mass.

Detection at 0.01% FPR concentrates in the high-memorization, low-entropy quadrant: samples whose correctness depends strongly on training inclusion *and* on which the model is highly confident. This is the regime in which RMIA's pairwise likelihood ratio is most informative, since both factors of the test statistic are pushed toward detection.

The quadrant tells a consistent story across the three methods. Under standard training, it contains 6.0% of samples but accounts for $D = 77.2\%$ of detected mass, with within-quadrant rate $A = 40.4\%$. Under InfoBatch, D falls to 59.5% and A to 15.1%: the quadrant still dominates detection, but per-sample detection rate within it is reduced by more than a factor of two. Under WLIB, the trend continues to $D = 43.8\%$ and $A = 4.8\%$, an order-of-magnitude reduction in within-quadrant detection rate relative to standard training. The detected mass that remains is more spread out across the other three quadrants, all of which exhibit per-sample rates below 1%. Both inputs to RMIA's score, the absolute memorization level and the model's confidence, are therefore reduced jointly under WLIB on the samples that drive the attack, which explains the aggregate TPR reductions reported in Table 1.

Algorithm 1 WLIB: Weighted-Loss InfoBatch

Require: Dataset $\mathcal{D} = \{z_i\}_{i=1}^N$, model θ, total epochs T, pruning probability r, annealing parameter δ, weight clamp $\rho_{\min}$

Initialize score $\mathcal{H}_0(z_i) \leftarrow 3$ for all i
Initialize prune count $c_i \leftarrow 0$ for all i
$T_s \leftarrow \lfloor \delta \cdot T \rfloor$
for $t = 1$ to T **do**
  if $t \leq T_s$ **then**
    Pruning phase (identical to InfoBatch)
    $\bar{\mathcal{H}}_t \leftarrow \frac{1}{N} \sum_{i=1}^N \mathcal{H}_t(z_i)$
    $\mathcal{S}_t \leftarrow \emptyset$
    for each $z_i \in \mathcal{D}$ **do**
      if $\mathcal{H}_t(z_i) < \bar{\mathcal{H}}_t$ **then**
        With probability r: set $c_i \leftarrow c_i + 1$ and skip z_i
        Otherwise: add z_i to $\mathcal{S}_t$ with loss weight $(1-r)^{-1}$
      else
        Add z_i to $\mathcal{S}_t$ with loss weight 1
      end if
    end for
    for minibatch $\mathcal{B} \subset \mathcal{S}_t$ **do**
      Update θ using gradient of weighted loss on $\mathcal{B}$
      Update $\mathcal{H}_{t+1}(z_i) \leftarrow \ell(z_i; \theta)$ for $z_i \in \mathcal{B}$
    end for
  else
    if $t = T_s + 1$ **then**
      $\rho_i \leftarrow c_i / T_s$ for all i
      $w_i \leftarrow \dfrac{\max(\rho_i, \rho_{\min})}{\frac{1}{N} \sum_{j=1}^N \max(\rho_j, \rho_{\min})}$ for all i
    end if
    Annealing phase: full data, fixed per-sample weights
    for minibatch $\mathcal{B} \subset \mathcal{D}$ **do**
      Update θ using gradient of $\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} w_i \, \ell(z_i; \theta)$
    end for
  end if
end for
return θ

F. WLIB algorithm

Algorithm 1 provides the full pseudocode for WLIB. The pruning phase ($t \leq T_s$) is identical to InfoBatch: easy samples (loss below the epoch-wise mean $\bar{\mathcal{H}}_t$) are pruned with probability r, and the loss of retained easy samples is rescaled by $(1-r)^{-1}$ to preserve the expected gradient. Throughout this phase, we maintain a per-sample prune count c_i, which determines the cumulative pruning rate $\rho_i = c_i / T_s$ at the start of annealing. At epoch $T_s + 1$, the pruning rule is disabled, the rescaling is removed, and the per-sample weights w_i are computed once from ρ_i as in Eq. 2 and held fixed for the remainder of training. The construction adds no new hyperparameters beyond $\rho_{\min}$ and no extra computation, since ρ_i is derived from quantities InfoBatch already tracks.