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

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

[1]Saarland University, Germany [2]CISPA Helmholtz Center for Information Security, Germany

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. Code is available at `https://github.com/chinmayj23/prune_to_protect`.

1 Introduction

Deep neural networks achieve state-of-the-art performance by training on massive datasets with heavily overparameterized architectures [Zhang et al., 2017]. A direct consequence of this scaling is *memorization*: for some training examples, the model's prediction becomes dependent on whether the example was present during training [Feldman and Zhang, 2020]. This dependence creates a direct privacy vulnerability. Membership Inference Attacks (MIAs) [Shokri et al., 2017, Zarifzadeh et al., 2024, Carlini et al., 2022a] exploit the learned model to infer whether a given example belonged to the training set. Thus, highly memorized examples are especially susceptible to MIAs, as their inclusion induces a stronger example-specific effect on the trained model. Consistent with this view, recent work [Choi et al., 2023] shows that per-example memorization strongly predicts per-example MIA vulnerability, implying that highly memorized samples carry disproportionate privacy risk.

Memorization, however, is not uniform across the training set. It is concentrated on a small subset of atypical, hard-to-learn examples that lie in the tail of the data distribution. Both empirical and theoretical work suggest that learning, and in some cases memorizing, such long-tail examples is necessary for strong generalization [Feldman and Zhang, 2020, Feldman, 2020, Zhou et al., 2026, Brown et al., 2021]. This creates a fundamental tension: the same examples whose memorization supports generalization are also those that incur the highest privacy risk.

This observation motivates the central question of this work: Can we design a *training-time procedure* of these hard examples using dynamic data-pruning to reduce tail memorization, without sacrificing the generalization performance it enables?

Dynamic data pruning offers a principled way to control the influence of individual samples during training. InfoBatch [Qin et al., 2024] is especially relevant in this context. At each training epoch, InfoBatch uses per-example loss to distinguish easy and hard examples. Low-loss examples are stochastically pruned, whereas high-loss examples are retained. During the final annealing phase,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

InfoBatch turns off dynamic pruning and trains on the full dataset. While this strategy was introduced as a means of improving training efficiency, it also changes the optimization dynamics at the level of individual samples: hard examples are exposed to the model more frequently and thus contribute more gradient updates than easy examples. This exposure asymmetry has direct implications for memorization. However, whether this mechanism can be used to reduce privacy-relevant memorization, rather than merely accelerate training, remains unexplored.

Our analysis of InfoBatch yields three findings:

1. InfoBatch reduces memorization on hard examples. Under standard training, these examples form much of the high-memorization tail. Under InfoBatch, they rarely reach the extreme memorization values observed under standard training; instead, the tail mass shifts toward lower and intermediate scores. A general intuition is that reducing the exposure of hard, highly memorized examples should reduce memorization [Feldman and Zhang, 2020]. Our findings are surprising from this perspective. InfoBatch retains hard examples more often than easy examples, thereby increasing their relative exposure, yet decreasing their memorization. Thus, memorization is not governed by exposure frequency alone, but by how sample exposure interacts with the dynamics of optimization.
2. In InfoBatch, the early pruning phase alters the sample exposure and thereby delays the onset of memorization. As the pruning schedule is annealed and the full dataset is exposed with uniform weights across the samples, memorization rises sharply. This rise is driven mainly by hard examples, whose memorization grows during the annealing phase after staying low relative to standard training in earlier stages of optimization.
3. InfoBatch's reduction of hard-sample memorization translates to improved robustness against MIAs. Less-memorized hard samples exhibit higher predictive entropy, which weakens the member–non-member signal exploited by the attacker, particularly in the low-false-positive-rate (FPR) regime.

These findings show that InfoBatch can improve training efficiency while reducing privacy-relevant memorization. They also expose a late-stage vulnerability: as pruning is annealed, memorization rises sharply. We therefore propose WLIB—**W**eighted **L**oss **I**nfo**B**atch—that targets the annealing phase. At the start of annealing, WLIB computes each example's cumulative pruning rate, normalizes these rates to a unit mean, and fixes them as a multiplicative weight on that example's loss. This weighting uses the pruning rate to rebalance sample influence: hard examples, which are retained frequently and have low pruning rates, receive smaller weights and contribute less to the late memorization increase; easy examples, which are pruned more often, receive larger weights and preserve the accuracy gains from annealing. Thus, WLIB maintains the accuracy and efficiency offered by InfoBatch, while targeting the memorization-critical phase of InfoBatch and reducing its late-stage memorization.

We summarize the main contributions of the paper as follows.

- We show, for the first time, that dynamic data pruning with InfoBatch not only improves training efficiency, but also has privacy-relevant benefits due to its sample-level training strategy.
- We identify a late-stage privacy failure mode of InfoBatch, resulting in a sharp increase in memorization. To mitigate this effect, we introduce WLIB, resulting in improved MIA robustness over standard training and InfoBatch while retaining InfoBatch-level efficiency and model utility.
- We experimentally demonstrate that WLIB reduces hard-example memorization and improves robustness to membership inference attacks across multiple datasets compared to InfoBatch and standard training, while preserving the accuracy and training efficiency offered by InfoBatch.

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 [2020] 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., 2024, 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., 2019, Jagielski et al., 2023]. 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. Direct evidence of privacy leakage is provided by extraction and reconstruction results in both generative and classification settings [Carlini et al., 2021, Haim et al., 2022, Balle et al., 2021]. Privacy-driven training methods, including DP-SGD [Abadi et al., 2016] and recent membership and reconstruction audits [Boenisch et al., 2023a, Kaissis et al., 2023] target this risk through optimizer-level noise or post-hoc auditing rather than through how data are exposed during training. 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., 2019], early gradient or error signals [Paul et al., 2021], generalization influence [Yang et al., 2023], leave-one-out influence [Tan et al., 2023], diversity–difficulty balancing [Maharana et al., 2024], decision-boundary reconstruction [Yang et al., 2024],information maximization [Tan et al., 2025], coreset objectives [Mirzasoleiman et al., 2020, Sorscher et al., 2022], 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., 2024], 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. This literature treats pruning as an efficiency mechanism; we use the same mechanism as a sample-level intervention on memorization while inheriting its efficiency. By changing how often each sample is presented during training, dynamic pruning also acts as a curriculum [Bengio et al., 2009], and curricula are known to affect membership-inference risk [Chen et al., 2025]. We elaborate on this connection in Section 7.

Closest to our perspective, Ye et al. [2026] shows that statically pruning a language-model corpus before training reshapes what is memorized by altering the effective frequency of retained data. Our setting is orthogonal in two respects: the pruning is dynamic and per-epoch rather than a single pre-training selection, and we address the privacy concerns of memorization through data pruning. To our knowledge, no prior work studies the joint effect of dynamic data pruning on memorization and membership-inference robustness, and our paper closes this gap.

3 Background

InfoBatch. Qin et al. [2024] proposed InfoBatch, a dynamic data pruning method that accelerates training by reducing the participation of examples that are currently well learned. 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 falls 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

$$P_t(z) = r \text{ if } H_t(z) < \bar{H}_t \text{ (\textit{easy} examples)}, \qquad P_t(z) = 0 \text{ if } H_t(z) \geq \bar{H}_t \text{ (\textit{hard} examples)}. \quad (1)$$

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 refresh their score to the current loss.

InfoBatch trains in two phases. For a run of T epochs, pruning is applied during the first $T_s = \lfloor \delta \cdot T \rfloor$ epochs, with $\delta \in (0, 1)$. In the remaining $T - T_s$ epochs, the *annealing phase*, pruning and rescaling

are disabled, and all samples contribute equally to each update. We use $\delta = 0.875$ throughout (Appendix A). This stage is crucial for recovering any performance lost during pruning, since it reintroduces the previously pruned easy samples at full weight. However, as we observe in our experiments, it is also the stage in which memorization rises sharply.

Pruning rate. InfoBatch makes training non-uniform across examples, with examples estimated to be easy contributing less frequently than harder examples. To summarize this non-uniformity, we record how often each example is pruned during training. Let $p_i(t) \in \{0, 1\}$ indicate whether example i is pruned at epoch t. We define its cumulative pruning rate up to epoch T, $\rho_i(T)$, as

$$\rho_i(T) = \frac{1}{T} \sum_{t=1}^{T} p_i(t). \tag{2}$$

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*.

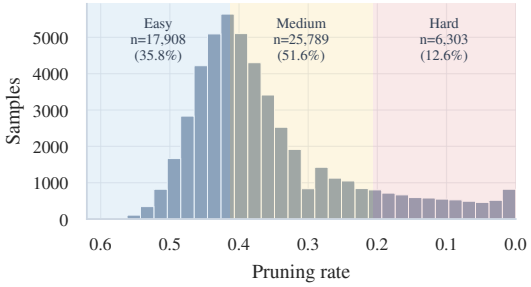


Figure 1: Distribution of ρ on CIFAR-10 under InfoBatch, partitioned into three equal-width bins defining easy (n=17,908), medium (n=25,789), and hard (n=6,303) groups. Lower ρ indicates samples persistently retained during training.

We use the pruning rate as an aggregate, loss-based measure of sample hardness. Prior work commonly uses the gradient or training loss in the early epochs to score or stratify examples in data pruning and curriculum methods [Paul et al., 2021, Sanyal et al., 2025]. Pruning rate follows the same principle, since each pruning decision depends on the loss-based score; we show the correlation between ρ and $\mathcal{L}$ in Fig. 7 in the appendix. However, instead of measuring at a single checkpoint, ρ aggregates these relative loss-based decisions over the training trajectory. Additionally, the signal remains meaningful throughout training rather than trivially small once the loss has converged, and it requires no additional computation beyond what InfoBatch maintains.

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

Membership Inference Attack. A membership inference attack tests whether a sample z was part of the training set of a target model θ. Recent attacks frame this as a hypothesis test. LiRA [Carlini et al., 2022a] uses shadow models trained on data from the same distribution as the target's training data. In its online variant, LiRA trains shadow models with z (IN) and without z (OUT). Each model assigns a probability to the ground-truth label (y). LiRA applies a logit transformation to these probabilities and fits separate Gaussian distributions to the IN and OUT values and evaluates both

Gaussian densities at the target model's transformed probability. Their ratio provides the membership score. However, training the shadow models can impose substantial computational cost.

RMIA [Zarifzadeh et al., 2024] improves on LiRA by additionally comparing z against population samples. For reference points $z' \sim \pi$ drawn from the population, RMIA computes the membership score

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

where $\Pr(z \mid \theta)$ is the likelihood of z under the target model, $\Pr(z)$ is its average likelihood over shadow models, and $\gamma \geq 1$ is the pairwise threshold. Normalizing by $\Pr(z)$ calibrates the test to sample difficulty, and the comparison with z' calibrates it to the population. This gives RMIA a higher TPR at low FPR than LiRA, in particular when few shadow models are available [Zarifzadeh et al., 2024], as in our ImageNet setting with 2 shadow models. We therefore use RMIA as our primary attack, and additionally report LiRA on CIFAR-10 in Appendix I. 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].

4 Analysis: effect of InfoBatch on memorization

Memorization measures how much a trained model depends on individual training examples [Feldman and Zhang, 2020]. Dynamic data pruning reshapes this dependence by changing how often each example contributes to the gradient, and we use this lens to ask whether the same intervention can simultaneously alter the memorization dynamics. We study these effects on CIFAR-10 with a ResNet-20 model. Detailed setup is deferred to Appendix A.

Pruning reduces hard-sample memorization despite increased exposure. By construction, as seen in Figure 1, hard examples are the ones InfoBatch retains most often: low pruning rate identifies samples whose loss persistently sits above the epoch-wise mean and are therefore rarely skipped. The hard samples, therefore, have a relatively higher exposure than easy samples. Intuitively, we would expect this higher exposure to increase the model's dependence on them and hence increase their memorization.

To measure this, we compute mean memorization using Equation 4 within each pruning-rate group at multiple checkpoints throughout training on CIFAR-10. In Figure 2, the hard-group trajectory under InfoBatch sits below standard training across the run and converges to a lower final value, 0.4693 versus 0.4993. The easy and medium groups remain near zero or saturate around 0.05 under both methods, with curves overlapping (as shown in Appendix D). The distinguishing effect of dynamic pruning is captured in the hard-sample group.

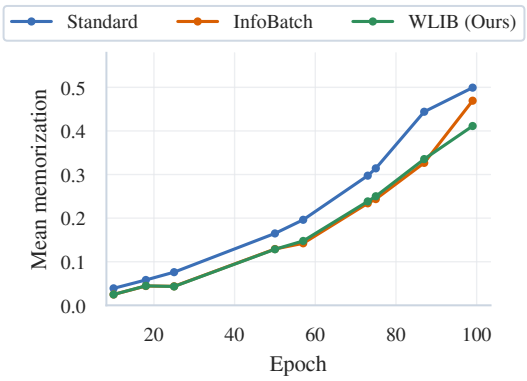


Figure 2: Mean memorization trajectory on CIFAR-10 within the hard group ($\rho \in [0.00, 0.21]$). InfoBatch and WLIB suppress memorization through most of training.

Counterintuitively, hard samples, in spite of having a relatively higher exposure through InfoBatch, attain lower memorization. This rules out a simple monotone relationship in which increased exposure necessarily increases memorization. One possible explanation for this empirical reduction in memorization relates to how the loss distribution changes. In standard training, the model quickly learns easy samples and 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.

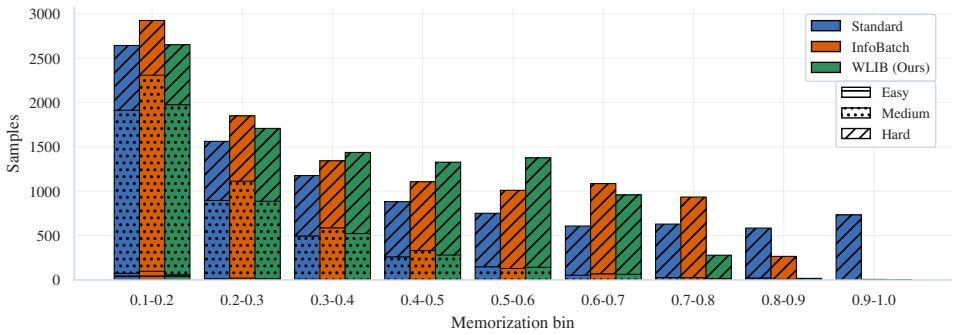


Figure 3: Distribution of final per-sample memorization scores on CIFAR-10 (scores ≥ 0.1), stacked by difficulty group via hatching. InfoBatch and WLIB progressively depopulate the extreme bins occupied by standard training and shift hard-sample mass toward intermediate values.

The reduction in hard-group mean is modest in magnitude, but the effect on the distribution is structural. In Figure 3, we examine the per-example memorization scores at the end of training. The high-memorization tail is composed almost entirely of hard examples under all methods, with contributions from medium examples vanishing around the 0.5 bin and easy being negligible beyond 0.2. Under standard training, hard examples populate a long tail extending to $[0.9, 1.0]$. Under InfoBatch, the $[0.9, 1.0]$ bin is empty and the $[0.8, 0.9]$ bin shrinks, with the displaced mass moving into intermediate bins below 0.8. As we will see in Appendix G, these high memorization bins are also the most vulnerable to membership inference attacks. This privacy-relevant region of the memorization distribution, the extreme tail, is therefore depopulated under InfoBatch even though the hard examples that previously occupied it were the ones being retained most aggressively.

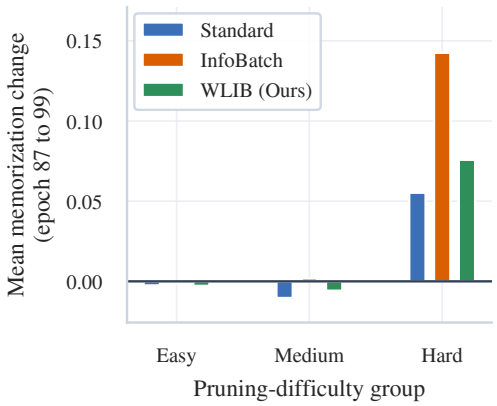


Figure 4: Mean memorization change between the start of the annealing phase (epoch 87) and the last epoch (epoch 99) on CIFAR-10, by difficulty group. Easy and medium groups change negligibly; the late-phase rise is entirely a hard-sample phenomenon, and is largest under InfoBatch.

Memorization surge in the late phase. Having established the benefits of pruning, we next isolate the effect of the annealing phase. On CIFAR-10, InfoBatch begins annealing at epoch 87: dynamic pruning is disabled, and training resumes on the full dataset with uniform loss weights (Section 3). This transition separates the run into two regimes, allowing us to measure how memorization changes once full-dataset exposure is restored.

Figure 4 shows the change in mean memorization between the beginning of the annealing phase (epoch 87) and the end of training (epoch 99). Under InfoBatch, memorization rises by $+0.143$ during this period. For comparison, we measure the change in standard training over the same epochs, even though standard training has no annealing phase. InfoBatch's rise is about $2.6\times$ larger.

This pattern shows that InfoBatch delays hard-sample memorization. As shown in Figure 2, standard training reaches higher memorization earlier, whereas InfoBatch keeps hard examples at lower memorization during pruning and increases mainly near the end. Because InfoBatch enters the annealing phase from a lower memorization level, this late increase does not entirely erase the earlier reduction. Hence, at the final epoch, hard-sample memorization under InfoBatch remains below that of standard training. Thus, InfoBatch has a phase-dependent effect: pruning delays hard-sample memorization, while annealing partially offsets this delay by restoring full exposure late in training.

The late-phase surge concentrates at low pruning rates. The pattern we have analyzed until now is defined at the level of difficulty groups: hard examples receive higher relative exposure under InfoBatch, yet their memorization decreases. We next examine if the hard group, which is of particular interest to us, has finer structure within. We observe that in the hard group, the cumulative pruning rate ρ remains an informative difficulty signal. Examples with lower ρ are pruned less often because their losses stayed high relative to the epoch-wise mean; they are therefore the hardest examples within the hard group. Note that this memorization surge occurs in the annealing phase when the model is trained with the full dataset, suggesting that when training with the full dataset, following Arpit et al. [2017], the harder samples are memorized.

We measure how these examples behave during annealing by comparing two checkpoints: the onset of annealing at epoch 87 and the final epoch. For each hard example, we compute the change in memorization over this interval and relate it to its cumulative pruning rate. Figure 5 shows a clear negative correlation: hard examples with lower pruning rates experience the largest late-stage increase in memorization. The increase is $+0.27$ for the least-pruned examples, $+0.14$ in the middle of the range, and $+0.05$ for the most-pruned examples within the hard group. Thus, the annealing-phase memorization surge is not uniform across hard examples. It is concentrated on the hardest, most frequently retained samples.

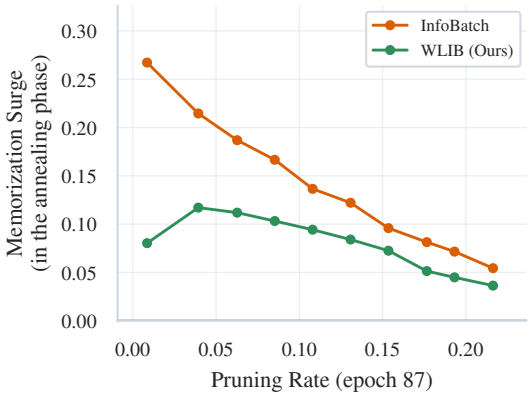


Figure 5: Mean memorization surge during the annealing phase within the CIFAR-10 hard examples, plotted against the cumulative pruning rate ρ anchored at epoch 87 (beginning of the annealing phase). The late-phase rise is largest at low ρ and is suppressed most strongly there by WLIB. Full range of pruning rate vs memorization surge during the annealing phase in Figure 10 in Appendix F

The purpose of the annealing phase is to reinstate the contribution from the samples that were aggressively pruned in the pruning phase, and close the performance gap compared to standard training. A more targeted intervention in this phase is necessary to achieve this goal while preventing the surge in memorization of the hard samples. WLIB addresses this shortcoming using the pruning rate for the intended intervention of reinstating samples that received low exposure while holding the memorization state of samples that have benefited from the pruning phase.

5 WLIB: Weighted Loss InfoBatch

By the end of the pruning phase, hard examples are in a state that is favorable for privacy: their memorization has been kept below the levels reached under standard training (Section 4), and the privacy-relevant tail of the distribution is correspondingly redistributed. The annealing phase then shifts the memorization state toward the standard-training regime on these examples (Section 4), with the largest movement concentrated in the lowest-ρ samples (Section 4). WLIB (Weighted-Loss InfoBatch) is designed to preserve the favorable state of hard examples while still allowing easy examples to recover their contribution during the annealing phase and thus continue to maintain comparable generalization performance to InfoBatch.

The pruning phase is unchanged: examples are pruned exactly as in InfoBatch. At the start of the annealing phase, each example's cumulative pruning rate ρ_i is computed once and fixed for the remainder of training. We then convert these fixed pruning rates into unit-mean example weights for per-example loss in the annealing phase.

$$\mathcal{L}_{\text{WLIB}} = \frac{1}{B}\sum_{i\in\mathcal{B}} w_i \ell_i, \qquad w_i = \frac{\max(\rho_i, \rho_{\min})}{\frac{1}{N}\sum_{j=1}^{N}\max(\rho_j, \rho_{\min})}, \qquad \rho_i = \frac{c_i}{T_s}. \tag{6}$$

where $\mathcal{B}$ is a minibatch of size B, ℓ_i is the loss of example i, c_i is the number of times example i was pruned before the annealing phase, T_s is the epoch at which annealing begins, N is the number of training examples, and $\rho_{\min}$ is a small value used to prevent any sample from being fully ignored.

Hard examples, which have low ρ, receive small weights and contribute correspondingly little to the gradient updates, while easy examples, which were relatively less exposed earlier through pruning, are up-weighted and recover their contribution as the annealing phase intends. Easy samples receive a larger weight in the annealing phase, which results in a more consistent gradient (similar to that of standard training). This helps recover the accuracy, as we observe empirically in Figure 6. Simultaneously, the down-weighting of hard samples protects them from the memorization surge. This construction introduces no new hyperparameters beyond $\rho_{\min}$ and adds no additional computation, since InfoBatch already maintains ρ. Full pseudocode is provided in Appendix H.

The effect of this design is visible in the same figures used in the analysis. In Figure 2, the hard-group trajectory under WLIB stays close to its pre-annealing value at epoch 87 through the annealing phase and ends below both InfoBatch and standard training at convergence, consistent with the favorable pruning-phase state being preserved. In Figure 4, the hard-group rise drops from $+0.143$ under InfoBatch to $+0.076$ under WLIB, while the easy and medium groups remain unchanged, indicating that the intervention acts on the group whose memorization the late phase would otherwise drive up without disturbing the rest. In Figure 5, the effect of WLIB can be seen at a finer level. The group of samples that were worsening the most with InfoBatch is protected with WLIB.

6 Results

Setup. We design our empirical setup to study how the reshaping of memorization distribution improves robustness to membership inference attacks. For privacy, we implement RMIA (Section 3) on CIFAR-10, CIFAR-100 [Krizhevsky, 2009], and ImageNet [Deng et al., 2009], covering small- to large-scale image classification with both convolutional and transformer backbones [He et al., 2016, Zagoruyko and Komodakis, 2016, Touvron et al., 2021]. We compare standard training, InfoBatch, and WLIB under a fixed architecture, optimizer, schedule, and attack pipeline. We use TPR at 0.01% FPR as the primary MIA metric and AUC as a secondary metric, following standard low-FPR auditing practice [Carlini et al., 2022a]. We primarily report results under a defense-unaware attack, where shadow models use standard training. In order to further establish robustness to an adaptive attack, we evaluate a defense-aware attack on CIFAR-10 with ResNet-20, where shadow models are trained with the target's method.

Results. Table 1 shows that dynamic pruning consistently reduces RMIA success. Across all three datasets, TPR at 0.01% FPR decreases from standard training to InfoBatch and further to WLIB. The same ordering holds for AUC and for TPR at 1% FPR. On DeiT-Small, InfoBatch performs similarly to standard training, whereas WLIB achieves the lowest AUC and TPR at 1% FPR. Thus, WLIB reduces membership-inference leakage across datasets with the clearest gains in the low-FPR regime. Additionally, under LiRA on CIFAR-10, WLIB is also the least vulnerable method (Appendix I).

This privacy gain does not come with degraded utility. As shown in Figure 6, WLIB closely follows the accuracy trajectory of InfoBatch throughout training. At convergence, it remains within 0.1 percentage points of InfoBatch on CIFAR-10 and ImageNet, and slightly exceeds standard training on CIFAR-100 and improves over InfoBatch on DeiT-Small (Table 1). It also retains the training-cost reduction of InfoBatch. On CIFAR-10 and CIFAR-100, it processes 35% and 32% fewer samples than standard training and trains $1.47\times$ and $1.46\times$ faster (Table 4 in § H). Thus, WLIB improves membership-inference robustness while preserving the accuracy and efficiency benefits of InfoBatch.

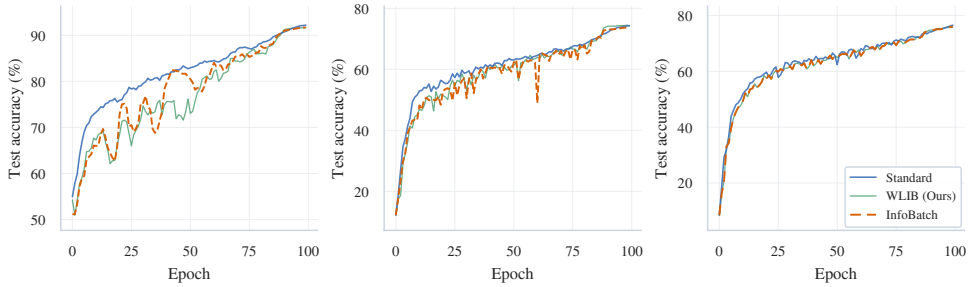
To understand the privacy gain in WLIB, we trace the per-sample structure of memorization. Figure 11 in Appendix G plots memorization against predictive entropy on CIFAR-10. At 0.01% FPR, RMIA detections concentrate in a small high-risk region: samples with high memorization and low entropy. These samples depend strongly on training inclusion and receive confident predictions, making them especially informative to the attacker. Under standard training, this region contains only 6.0% of samples but accounts for 77.2% of detected members. InfoBatch reduces the detection rate within this region, and WLIB reduces it further, from 40.4% under standard training to 15.1% under InfoBatch and 4.8% under WLIB. Moreover, of total detected members, the fraction coming from this region also decreases from 77.2% to 59.5% and then to 43.8%.

These results explain the aggregate privacy gains against RMIA through the observed memorization dynamics. WLIB lowers memorization and increases predictive entropy on hard, highly exposed samples, weakening the attack signal where RMIA detections concentrate. It also prevents the

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	$\mathbf{92.21}$
	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)	$\mathbf{88.06 \pm 0.27}$	$\mathbf{1.22 \pm 0.70}$	$\mathbf{2.39 \pm 0.70}$	$\mathbf{21.59 \pm 1.06}$	$\mathbf{74.34}$
ImageNet (ResNet-50)	Standard	73.7 ± 0.15	0.0	1.50 ± 1.7	17.4 ± 0.11	$\mathbf{76.53}$
	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	0.00	19.4 ± 0.60	$\mathbf{72.9}$
	InfoBatch	79.30 ± 0.09	0.0	0.00	19.5 ± 0.09	72.2
	WLIB (Ours)	$\mathbf{77.12 \pm 0.08}$	0.0	0.00	$\mathbf{17.3 \pm 0.14}$	72.5

remaining leakage from concentrating in a small, identifiable high-risk subset. This, together with the preserved accuracy trajectories in Figure 6, shows that WLIB improves the privacy–utility tradeoff through dynamic pruning.



(a) CIFAR-10 test accuracy (b) CIFAR-100 test accuracy (c) ImageNet test accuracy

Figure 6: Test accuracy across training epochs on CIFAR-10, CIFAR-100, and ImageNet. WLIB tracks InfoBatch throughout training and remains close to standard training at the end.

Ablations. We ablate the annealing phase and the pruning-rate weights of WLIB on CIFAR-10 (Table 2, details in Appendix A.4). *No annealing*, which keeps pruning active until the end of training, tests whether annealing is needed at all, or whether skipping it maintains the accuracy while achieving more privacy. *Early stopping*, which ends training at the start of annealing, tests whether the pruning phase alone is sufficient. Both reduce leakage relative to InfoBatch, but lower accuracy to 91.10% and 90.42%, compared to 91.64% for WLIB. Thus, annealing is necessary to recover accuracy.

Loss weights, the per-sample loss at the start of annealing, indicate whether loss at a single checkpoint suffices as a hardness proxy instead of the cumulative pruning rate. *Shuffled ρ weights*, a random permutation of ρ across samples, checks if down-weighting random samples is sufficient, or whether specific samples must be down-weighted. Both leak almost as much as InfoBatch, so the gain requires down-weighting the specific samples identified by the cumulative pruning rate. *Constant hard weights*, the same down-weight for the hardest third of samples, test whether identifying hard samples and down-weighting them together suffices. They match WLIB under the defense-unaware attack, but leak more under the defense-aware attack and lose 0.5 points of accuracy. Thus, individual weights are needed for robustness to the defense-aware attack and for performance. Under the defense-aware attack, WLIB has the lowest leakage of all variants (see also Appendix I).

Table 2: Ablations of WLIB on CIFAR-10 (ResNet-20) (%). The first block changes the annealing phase. The second block replaces the pruning-rate weights. For standard training, both attacks coincide.

Method	Test Acc. ↑	Defense-unaware				Defense-aware			
		AUC ↓	TPR@0% FPR ↓	TPR@0.01% FPR ↓	TPR@1% FPR ↓	AUC ↓	TPR@0% FPR ↓	TPR@0.01% FPR ↓	TPR@1% FPR ↓
Standard	92.21	71.82	1.87	3.15	13.76	71.82	1.87	3.15	13.76
InfoBatch	91.74	66.69	0.74	1.25	8.47	69.23	1.01	1.90	10.78
No annealing	91.10	65.44	0.47	0.98	7.54	68.20	0.92	1.59	10.03
Early stopping	90.42	63.19	0.33	0.62	5.91	66.69	0.61	1.25	8.74
Loss weights	91.26	65.95	0.58	1.15	8.00	68.27	1.12	1.68	10.10
Shuffled ρ weights	91.26	66.20	0.58	1.22	8.04	68.97	0.93	1.75	10.47
Constant hard weights	91.17	60.84	0.19	0.34	4.25	66.12	0.65	1.13	8.11
WLIB (Ours)	91.64	61.43	0.20	0.40	4.58	65.76	0.56	1.02	7.88

7 Discussion

In this work, we show that the per-example exposure induced by dynamic data pruning reshapes the memorization distribution, and that this reshaping has consequences for membership-inference risk. We provide principled insights into why data-pruning with InfoBatch, an intervention designed for efficiency, also acts on the privacy-relevant tail of memorization, and give a concrete account of how that effect arises and how it can be controlled. Producing formal privacy guarantees is outside the scope of this work, and methods such as DP-SGD [Abadi et al., 2016, Dwork and Roth, 2014, Boenisch et al., 2023b] remain the appropriate tool when such guarantees are required.

Two implications follow from our analysis. The first concerns curriculum. The order and frequency with which examples are presented to the model have long been studied as a lever for optimization and generalization [Bengio et al., 2009, Jiang et al., 2019, Mindermann et al., 2022], and our results show that the same lever shapes memorization in ways that matter for privacy. Treating examples as equally important throughout training hides this degree of freedom. Allowing the curriculum to vary across samples opens a surface for privacy-aware training that operates on the data-exposure dimension of the training pipeline, and is potentially compatible with techniques that hold privacy guarantees. The second concerns which examples to prune. The intuitive strategy for protecting privacy-vulnerable samples is to identify and remove them, but the onion effect of Carlini et al. [2022b] shows that removing the most memorized layer exposes a new layer beneath it, so vulnerability persists under removal. Our results point to an inversion of this strategy. Pruning the easy majority, the samples that contribute least to the privacy-relevant tail, redistributes memorization without requiring vulnerable examples to be identified in advance.

Counterintuitively, pruning what is well learned and exposing the hard-to-learn, atypical examples more, is an effective strategy for reducing privacy-relevant memorization than pruning examples that appear vulnerable. The interaction of this mechanism with other axes of the training pipeline and with data distributions of varying shape is a rich space for future work.

8 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 undoes the benefits. WLIB 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. The training-time treatment of hard examples can therefore be designed to reduce tail memorization. This asymmetric per-example exposure can redistribute the privacy-relevant tail of memorization, which allows hard examples to be protected without giving up the generalization their memorization supports. Efficiency is preserved as a direct consequence of the pruning schedule, which WLIB inherits from InfoBatch. These findings encourage further study of data exposure as a mechanism for privacy-aware training.

Impact statement

This work studies how dynamic data pruning and loss reweighting affect memorization and vulnerability to membership inference. In our experiments, WLIB reduces the success of the evaluated attacks while retaining the training efficiency and comparable overall predictive accuracy of InfoBatch. Reductions in attack success are also observed in our class- and subgroup-level analyses, including for the minority subgroup studied on CelebA. These findings may inform training methods that reduce computational costs and privacy risks for individuals whose data are used for model training. The reported privacy improvements are empirical and do not constitute formal privacy guarantees. Their applicability to other datasets, populations, and attacks requires further evaluation.

Disclosure of funding

The authors gratefully acknowledge funding from the European Research Council (ERC) under the Horizon Europe Framework Programme (HORIZON) for proposal number 101116395 SPARSE-ML.

References

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

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*, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/1e14bfe2714193e7af5abc64ecbd6b46-Abstract.html.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (SP)*, 2017. 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*, 2024. URL https://proceedings.mlr.press/v235/zarifzadeh24a.html.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *IEEE Symposium on Security and Privacy (SP)*, 2022a. URL https://ieeexplore.ieee.org/document/9833649.

Jihye Choi, Shruti Tople, Varun Chandrasekaran, and Somesh Jha. Why train more? effective and efficient membership inference via memorization. *arXiv preprint arXiv:2310.08015*, 2023. URL https://arxiv.org/abs/2310.08015.

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*, 2020. URL https://doi.org/10.1145/3357713.3384290.

Mo Zhou, Haoyang Ma, and Rong Ge. Memorizing long-tail data can help generalization through composition. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=UBoCMU5iYV.

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*, 2021. 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. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=C61sk5LsK6.

Devansh Arpit, Stanisław Jastrzębski, 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*, 2017. URL https://proceedings.mlr.press/v70/arpit17a.html.

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*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/7bc4f74e35bcfe8cfe43b0a860786d6a-Abstract-Conference.html.

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*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/59404fb89d6194641c69ae99ecdf8f6d-Abstract-Conference.html.

Michal Lukasik, Vaishnavh Nagarajan, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. What do larger image classifiers memorise? *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=Ew73inSyhG.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TatRHT_1cK.

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*, 2023. URL https://proceedings.mlr.press/v202/maini23a.html.

Niklas Stoehr, Mitchell A. Gordon, Chiyuan Zhang, and Owen Lewis. Localizing paragraph memorization in language models. *arXiv preprint arXiv:2403.19851*, 2024. URL https://arxiv.org/abs/2403.19851.

Mariya Toneva, Alessandro Sordoni, Remi 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://openreview.net/forum?id=BJlxm30cKm.

Matthew Jagielski, Om Thakkar, Florian Tramer, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Guha Thakurta, Nicolas Papernot, and Chiyuan Zhang. Measuring forgetting of memorized training examples. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=7bJizxLKrR.

Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. The privacy onion effect: Memorization is relative. In *Advances in Neural Information Processing Systems*, 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/564b5f8289ba846ebc498417e834c253-Paper-Conference.pdf.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

Niv Haim, Gal Vardi, Gilad Yehudai, Ohad Shamir, and Michal Irani. Reconstructing training data from trained neural networks. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/906927370cbeb537781100623cca6fa6-Paper-Conference.pdf.

Borja Balle, Giovanni Cherubin, and Jamie Hayes. Reconstructing training data with informed adversaries. In *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021. URL https://openreview.net/forum?id=Yi2DZTbnBl4.

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016. URL https://doi.org/10.1145/2976749.2978318.

Franziska Boenisch, Adam Dziedzic, Roei Schuster, Ali Shahin Shamsabadi, Ilia Shumailov, and Nicolas Papernot. Reconstructing individual data points in federated learning hardened with differential privacy and secure aggregation. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, 2023a. URL https://arxiv.org/abs/2301.04017.

Georgios Kaissis, Jamie Hayes, Alexander Ziller, and Daniel Rueckert. Bounding data reconstruction attacks with the hypothesis testing interpretation of differential privacy. *arXiv preprint arXiv:2307.03928*, 2023. URL https://arxiv.org/abs/2307.03928.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, 2018. URL https://doi.ieeecomputersociety.org/10.1109/CSF.2018.00027.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *Proceedings of the 36th International Conference on Machine Learning*, 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*, 2022. 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. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=Uj7pF-D-YvT.

Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=4wZiAXD29TQ.

Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. Data pruning via moving-one-sample-out. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/3abe23bf7e295b44369c24465d68987a-Abstract-Conference.html.

Adyasha Maharana, Prateek Yadav, and Mohit Bansal. $\mathbb{D}^2$ pruning: Message passing for balancing diversity & difficulty in data pruning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=thbtoAkCe9.

Shuo Yang, Zhe Cao, Sheng Guo, Ruiheng Zhang, Ping Luo, Shengping Zhang, and Liqiang Nie. Mind the boundary: Coreset selection via reconstructing the decision boundary. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/yang24b.html.

Haoru Tan, Sitong Wu, Wei Huang, Shizhen Zhao, and XIAOJUAN QI. Data pruning by information maximization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=93XT0lKOct.

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*, 2020. URL https://proceedings.mlr.press/v119/mirzasoleiman20a.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 *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=UmvSlP-PyV.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021a. URL https://ojs.aaai.org/index.php/AAAI/article/view/16988.

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*, 2021b. URL https://proceedings.mlr.press/v139/killamsetty21a.html.

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. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJg2b0VYDr.

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.

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.

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.

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 *Proceedings of the 39th International Conference on Machine Learning*, 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. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=6gzPSMUAz2.

Feng Hong, Yueming Lyu, Jiangchao Yao, Ya Zhang, Ivor W. Tsang, and Yanfeng Wang. Diversified batch selection for training acceleration. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/hong24c.html.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009. URL https://doi.org/10.1145/1553374.1553380.

Joann Qiongna Chen, Xinlei He, Zheng Li, Yang Zhang, and Zhou Li. A comprehensive study of privacy risks in curriculum learning. *Proceedings on Privacy Enhancing Technologies*, 2025. URL https://doi.org/10.56553/popets-2025-0033.

Jiayuan Ye, Vitaly Feldman, and Kunal Talwar. Cram less to fit more: Training data pruning improves memorization of facts. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=YL9lAEeoXw.

Sunny Sanyal, Hayden Prairie, Rudrajit Das, Ali Kavis, and Sujay Sanghavi. Upweighting easy samples in fine-tuning mitigates forgetting. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=13HPTmZKbM.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009. doi: 10.1109/CVPR.2009.5206848.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL https://doi.org/10.1109/CVPR.2016.90.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016. URL https://bmva-archive.org.uk/bmvc/2016/papers/paper087/index.html.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. URL https://proceedings.mlr.press/v139/touvron21a.html.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 2014. URL https://doi.org/10.1561/0400000042.

Franziska Boenisch, Christopher Mühl, Roy Rinberg, Jannis Ihrig, and Adam Dziedzic. Individualized pate: Differentially private machine learning with individual privacy guarantees. *Proceedings on Privacy Enhancing Technologies*, 2023b. URL https://doi.org/10.56553/popets-2023-0010.

Sree Harsha Nelaturu, Advait Gadhikar, and Rebekka Burkholz. Turboprune: High-speed distributed lottery ticket training. GitHub repository, 2024. URL https://github.com/nelaturuharsha/TurboPrune.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, 2015. URL https://doi.org/10.1109/ICCV.2015.425.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

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 3. All runs use a triangular learning-rate schedule, and our training pipeline builds on the TurboPrune framework [Nelaturu et al., 2024].

Table 3: 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 estimate per-sample memorization using the subset-sampling estimator of Feldman and Zhang [2020], which approximates Eq. 3 by training many models on overlapping random subsets of the training set. Each subset is drawn independently and contains a uniformly random 70% of the training data, so every example appears in roughly 70% of the runs.

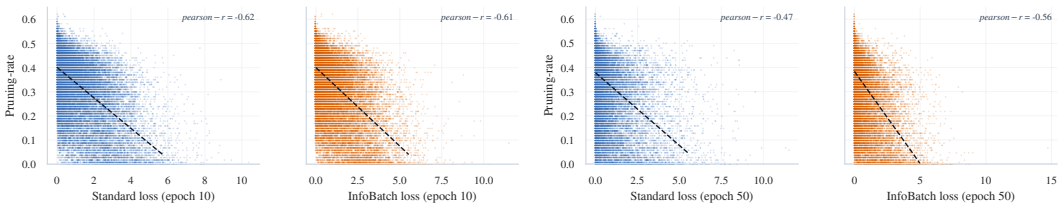
We use $M = 400$ subsets on CIFAR-10, with each model trained under the configuration in Table 3. Scores are estimated separately for standard training, InfoBatch, and WLIB, with all 400 runs of each method using that method's training procedure.

A.3 Membership inference attack

We 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]; we use $\gamma = 2.0$ throughout. All target and shadow models are trained under the configuration in Table 3.

Throughout the paper, we report results under a *defense-unaware attack*: 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. Additionally, we evaluate a defense-aware attack, whose shadow models are trained with the target's method, in the ablations (Table 2) and in Appendix I.

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



(a) Epoch 10 (b) Epoch 50

Figure 7: 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.

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.

A.4 Ablations

All ablations in Table 2 use the CIFAR-10 configuration in Table 3, with $r = 0.5$, $\delta = 0.875$, and $\rho_{\min} = 0.05$ as for WLIB. Privacy is evaluated with the RMIA protocol of Appendix A.3 under both the defense-unaware and the defense-aware attack.

No annealing applies the InfoBatch pruning rule for all T epochs, so no annealing phase takes place. Since this variant is noisier than the others, its results are averaged over 5 seeds.

Early stopping ends training at epoch $T_s = 87$, the epoch at which annealing would begin. We adapt the learning-rate schedule to end at this epoch, since keeping the original schedule further reduces accuracy.

The remaining variants keep the WLIB pipeline and replace only the weights in the annealing phase. *Loss weights* use the per-sample loss at epoch 87 instead of ρ_i. The loss is min-max normalized and inverted, so that low-loss samples receive high weights, as with ρ_i. It is then clamped at $\rho_{\min}$ and normalized to unit mean as in Eq. 6. *Shuffled ρ weights* apply a random permutation to the ρ_i across samples before computing w_i. The weights therefore have the same distribution as in WLIB, but are no longer assigned to the samples whose pruning history produced them. *Constant hard weights* apply a constant down-weight to the hardest third of samples by ρ_i.

B Pruning rate as a hardness proxy

We use the cumulative pruning rate ρ_i (Eq. 2) 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.

Per-sample loss is the established hardness proxy in data pruning and curriculum learning [Paul et al., 2021, 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 7 shows the relationship at epoch 10 and epoch 50, for both standard training and InfoBatch.

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 Theoretical motivation for loss-based pruning and pruning-rate weighting

We can directly motivate our approach by locally linearizing the neural network around θ_0:

$$f_\theta(x_i) \approx f_{\theta_0}(x_i) + J_i(\theta - \theta_0), \tag{7}$$

so squared-loss training locally reduces to a least-squares problem with NTK matrix $K = JJ^\top$ and hat matrix

$$H = K(K + \lambda I)^{-1}. \tag{8}$$

For linear smoothers, the leave-one-out identity (obtained via the Sherman–Morrison–Woodbury formula) gives

$$r_{i,(-i)} = \frac{r_i}{1 - h_{ii}}, \tag{9}$$

where $r_i = y_i - \hat{f}_i$ and $h_{ii} = H_{ii}$. Moreover, the classical Cook's-distance influence theorem gives, up to normalization,

$$I_i \propto r_i^2 \frac{h_{ii}}{(1 - h_{ii})^2} = 2\ell_i \frac{h_{ii}}{(1 - h_{ii})^2}. \tag{10}$$

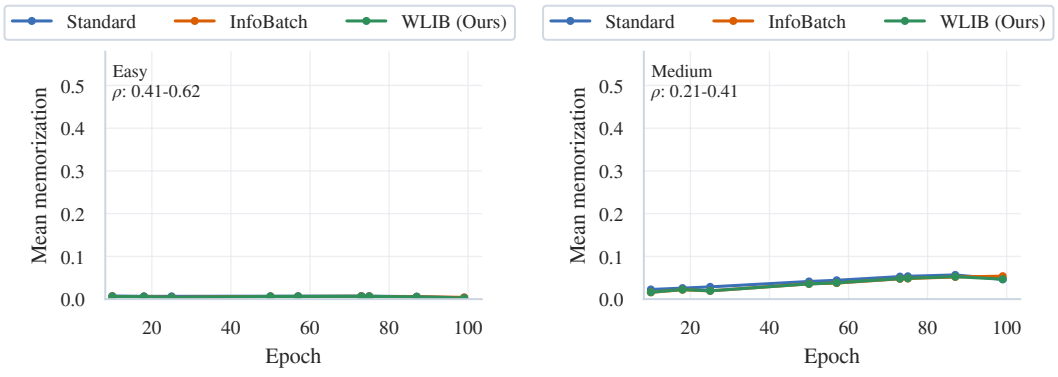
Hence, for comparable NTK leverage, higher-loss samples have greater marginal influence on the fitted network. Loss is therefore a practical proxy for unresolved sample influence, rather than memorization itself. This provides a theoretical motivation for InfoBatch: loss-based pruning preferentially allocates exposure to samples with greater unresolved influence. WLIB extends this principle to annealing: cumulative pruning history identifies samples that have persistently received high exposure, and down-weighting them later can reduce unnecessary continued sample-specific fitting.

Because each pruning decision is loss-based, the pruning rate aggregates the sample's relative-loss trajectory over the entire pruning phase; Figure 7 confirms the correlation between pruning rate and loss. This cumulative signal is more suitable than the loss at a single annealing checkpoint, which is noisy and does not capture persistent exposure. Indeed, replacing the pruning rate with loss at epoch 87 worsens AUC from 0.614/0.658 to 0.660/0.683 under defense-unaware/aware attacks and reduces accuracy from 91.64% to 91.26%.

Thus, prioritizing high-influence samples early while limiting their repeated late-stage exposure provides a principled mechanism for improving efficiency and reducing memorization.

D 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 8 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 8: 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.

E Memorization distribution at the onset of annealing

To isolate what WLIB's two phases each contribute, we measure the memorization distribution at the start of the annealing phase. Up to this point, WLIB and InfoBatch share an identical training procedure, so this snapshot captures the state both methods inherit from the pruning phase before WLIB's reweighting takes effect. Comparing it to the final-epoch distribution (Fig. 3) separates the effect of the pruning phase, which WLIB inherits, from the effect of the reweighted annealing phase, which is WLIB's contribution.

Figure 9 shows that the pruning phase has already done substantial work: the bins above 0.7 are essentially empty under both InfoBatch and WLIB, while standard training still occupies the tail up to 1.0, and hard-sample mass is concentrated below 0.6. By the final epoch (Fig. 3), InfoBatch's tail has shifted back toward standard training's while WLIB's has not, showing that the pruning phase alone is necessary but not sufficient: WLIB's reweighted annealing is what holds this state in place to the end of training.

Note. The figure is measured at epoch $T_s = 87$. WLIB's reweighting takes effect from epoch $T_s + 1$ onwards, so the small differences between the InfoBatch and WLIB bars here reflect a single epoch of annealing rather than different pruning behavior.

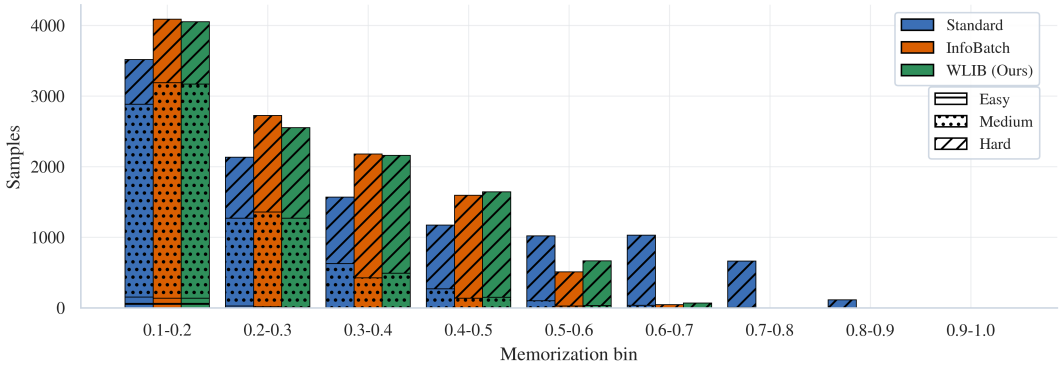


Figure 9: Distribution of per-sample memorization scores on CIFAR-10 at the onset of the annealing phase ($T_s = 87$), stacked by difficulty group via hatching. The bins above 0.7 are already empty under both InfoBatch and WLIB, while standard training still populates the tail up to 1.0.

F Memorization surge across the full pruning-rate range

The main paper plots the late-stage memorization surge against ρ within the hard group, where the effect is strongest. Figure 10 extends this view to the full range of pruning rates on CIFAR-10.

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.

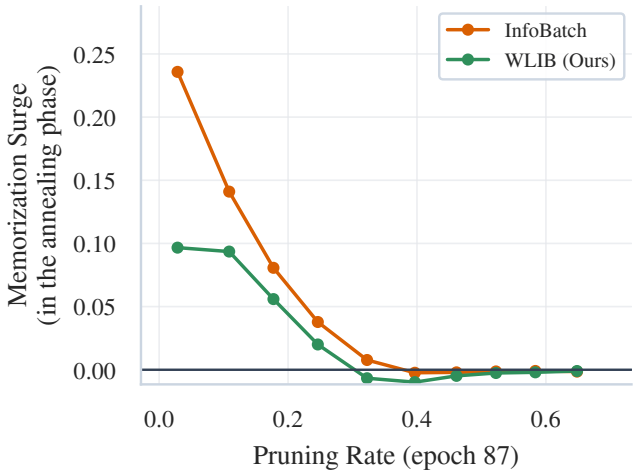


Figure 10: Mean memorization surge between epoch 87 and epoch 99 on CIFAR-10, plotted against ρ at epoch 87 across the full range.

G 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.

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.

H 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

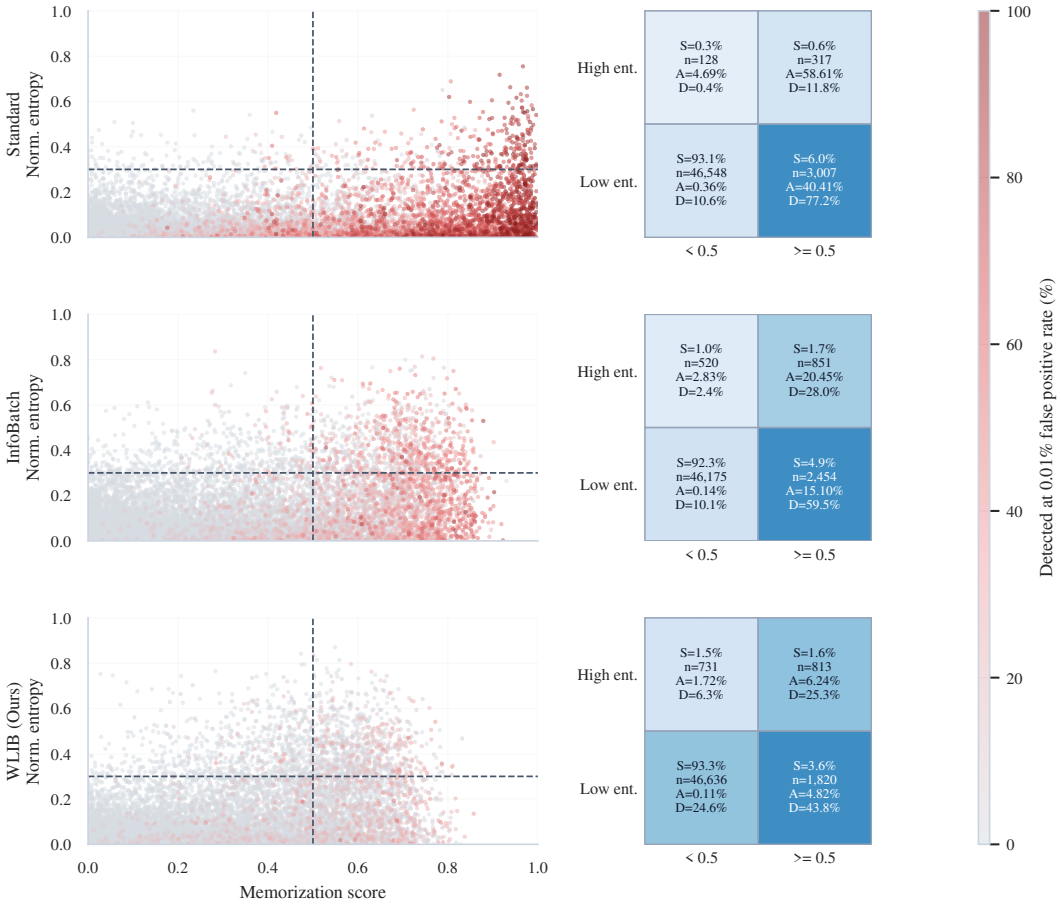


Figure 11: CIFAR-10 RMIA detection at 0.01% FPR in the memorization–entropy plane, partitioned into quadrants at memorization 0.5 and normalized entropy 0.3. Per-quadrant annotations: sample share S, count n, mean detection rate A, share D of total detected mass.

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. 6 and held fixed for the remainder of training. The construction adds no new hyperparameters beyond $\rho_{\min}$.

Computation. The pruning phase of WLIB is identical to InfoBatch. At the start of annealing, the weights w_i are computed once from the prune counts c_i, and during annealing each per-sample loss is multiplied by its fixed weight. WLIB therefore adds no forward or backward passes. Table 4 reports the resulting training cost. Since InfoBatch prunes only samples with below-mean loss, the pruning probability r does not directly determine the savings. We therefore report the fraction of data actually retained per epoch and the total number of processed samples, together with compute and wall-clock time. On both datasets, WLIB processes the same number of samples and FLOPs as InfoBatch, and its wall-clock time differs by at most 1 s. Measured throughput is 67.3 vs. 67.5 batches/s on CIFAR-10 and 47.4 vs. 47.4 batches/s on CIFAR-100 for WLIB and InfoBatch, respectively.

Memory. In addition to the per-sample scores maintained by InfoBatch, WLIB stores two fp32 scalars per sample, c_i and w_i, i.e., 8 bytes per sample. This amounts to 0.4 MB for CIFAR-10 and CIFAR-100 ($N = 50{,}000$) and 10.2 MB for ImageNet ($N \approx 1.28$ M). The measured peak-memory overhead over InfoBatch is 0.6 MiB on CIFAR-10 and 1 MiB on CIFAR-100.

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)
    Compute mean score $\bar{\mathcal{H}}_t \leftarrow \frac{1}{N} \sum_{i=1}^N \mathcal{H}_t(z_i)$
    $\mathcal{S}_t \leftarrow \emptyset$
    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 all $z_i \in \mathcal{B}$
    end for
  else
    if $t = T_s + 1$ **then**
      Compute unit-mean weights once at start of annealing
      $\rho_i \leftarrow c_i / T_s$ 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

Distributed training. In data-parallel training, each worker keeps c_i and w_i for its own partition of the data. These values are not part of the gradient all-reduce, so per-step communication is unchanged. The only additional communication is a single *all-gather* of the pruning rates at the start of annealing, which is needed to compute the normalization in Eq. 6.

I Defense-aware attack and LiRA

In addition to the defense-unaware attack used throughout the paper (Appendix A.3), we evaluate a defense-aware attack, whose shadow models are trained with the same method and hyperparameters as the target. We report RMIA and LiRA [Carlini et al., 2022a] under both attacks, on CIFAR-10 with ResNet-20.

LiRA. LiRA treats membership inference as a per-sample likelihood-ratio test. For a target sample $z = (x, y)$ and model θ, it uses the logit-scaled confidence on the true label, $\phi(z; \theta) = \log \frac{p}{1-p}$ with $p = f_\theta(x)_y$. Over the shadow models trained with z (IN) and without z (OUT), it fits Gaussians

Table 4: Training cost across datasets and methods over 100 epochs. Retained is the average fraction of the training set used per epoch, including the annealing phase; Samples is the total number of processed samples. Time is wall-clock training time on a single NVIDIA A6000.

Dataset	Method	Retained (%) ↓	Samples (M) ↓	Compute (TFLOPs) ↓	Time (s) ↓	Speedup ↑
CIFAR-10 (ResNet-20)	Standard	100	5.00	1224	144	1.00×
	InfoBatch	65	3.23	791	97	1.48×
	WLIB (Ours)	65	3.23	791	98	1.47×
CIFAR-100 (WRN-28-2)	Standard	100	5.00	6431	824	1.00×
	InfoBatch	68.5	3.42	4402	564	1.46×
	WLIB (Ours)	68.5	3.42	4402	563	1.46×

$\mathcal{N}(\mu_{\text{in}}, \sigma^2_{\text{in}})$ and $\mathcal{N}(\mu_{\text{out}}, \sigma^2_{\text{out}})$ to this statistic, and scores membership by

$$\text{Score}_{\text{LiRA}}(z;\theta) = \frac{\mathcal{N}\big(\phi(z;\theta);\, \mu_{\text{in}}, \sigma^2_{\text{in}}\big)}{\mathcal{N}\big(\phi(z;\theta);\, \mu_{\text{out}}, \sigma^2_{\text{out}}\big)}. \tag{11}$$

Unlike RMIA (Eq. 5), LiRA calibrates only against the IN and OUT distributions of the same sample and does not compare against population samples. We use online LiRA with the same target and shadow models as online RMIA (64 shadow models per target).

Table 5: RMIA and LiRA results on CIFAR-10 (ResNet-20) under the defense-unaware and defense-aware attacks (%). For standard training, both attacks coincide.

Attack	Method	Setting	AUC ↓	TPR@0% FPR ↓	TPR@0.01% FPR ↓	TPR@1% FPR ↓
RMIA	Standard	–	71.82	1.87	3.15	13.76
	InfoBatch	Unaware	66.69	0.74	1.25	8.47
	InfoBatch	Aware	69.23	1.01	1.90	10.78
	WLIB	Unaware	61.43	0.20	0.40	4.58
	WLIB	Aware	65.76	0.56	1.02	7.88
LiRA	Standard	–	71.82	2.04	3.43	14.10
	InfoBatch	Unaware	66.44	0.84	1.39	8.28
	InfoBatch	Aware	67.86	1.09	1.66	10.45
	WLIB	Unaware	60.97	0.22	0.45	4.39
	WLIB	Aware	64.95	0.40	0.80	7.47

The defense-aware attack is stronger than the defense-unaware attack against both InfoBatch and WLIB. Under both attacks, for both RMIA and LiRA, WLIB is the least vulnerable method on all metrics. WLIB under the defense-aware attack is also less vulnerable than InfoBatch under the defense-unaware attack on all metrics.

J Per-class and per-subgroup results

WLIB down-weights hard samples during annealing. Since hard samples may include rare classes or minority subgroups, we report accuracy and RMIA leakage per class on CIFAR-10 and per subgroup on CelebA [Liu et al., 2015] to confirm whether the gains in privacy does not come at the cost of rare-class or minority subgroups' accuracy.

CIFAR-10. Table 6 reports the per-class difference between WLIB and InfoBatch. WLIB has lower AUC on every class. Accuracy increases on 6 of the 10 classes, and the largest decrease is limited to 0.4 percentage points (cat). Thus, the gain in privacy does not cost accuracy for any specific class.

CelebA. We train ResNet-10t on CelebA and evaluate on the four subgroups defined by hair colour (blonde, non-blonde) and sex. Blonde/male is a minority subgroup, accounting for only 0.85% of the training data. We use online RMIA with the protocol of Appendix A.3. We report raw test accuracy and group-adjusted accuracy [Sagawa et al., 2020]. Results are given in Table 7.

Overall, WLIB has the lowest AUC and TPR at all reported FPRs, and the highest raw and group-adjusted accuracy. On the minority subgroup (blonde/male), accuracy under the three methods differs

Table 6: Per-class difference between WLIB and InfoBatch on CIFAR-10 (ResNet-20), in percentage points. Negative Δ AUC indicates lower leakage under WLIB.

Class	Δ Test Acc.	Δ AUC
Airplane	$+0.40$	-4.41
Automobile	$+0.10$	-2.42
Bird	-0.10	-6.66
Cat	-0.40	-8.88
Deer	$+0.70$	-5.03
Dog	-0.10	-8.06
Frog	$+0.40$	-4.03
Horse	$+0.30$	-4.57
Ship	0.00	-2.34
Truck	$+0.40$	-2.97

Table 7: RMIA results and test accuracy on CelebA (ResNet-10t), overall and per subgroup (%). Adj. Test Acc. is the group-adjusted accuracy [Sagawa et al., 2020], defined only overall.

Metric	Method	Overall	Non-blonde female	Non-blonde male	Blonde female	Blonde male
AUC ↓	Standard	53.3	53.0	50.4	59.8	75.6
	InfoBatch	54.5	54.2	50.9	62.3	80.6
	WLIB	52.9	52.8	50.6	57.4	68.5
TPR@0% FPR ↓	Standard	0.00	0.01	0.02	0.00	0.51
	InfoBatch	0.00	0.01	0.02	0.01	0.34
	WLIB	0.00	0.00	0.02	0.00	0.06
TPR@0.01% FPR ↓	Standard	0.02	0.04	0.07	0.01	0.51
	InfoBatch	0.02	0.03	0.07	0.02	0.34
	WLIB	0.00	0.01	0.05	0.00	0.06
TPR@1% FPR ↓	Standard	2.9	2.8	1.4	4.6	4.2
	InfoBatch	3.5	3.4	1.6	5.8	4.7
	WLIB	2.2	2.3	1.5	2.6	0.37
Test Acc. ↑	Standard	95.63	96.0	99.6	86.8	36.7
	InfoBatch	95.56	95.7	99.6	87.0	39.3
	WLIB	95.71	95.9	99.7	87.2	37.0
Adj. Test Acc. ↑	Standard	95.69	–	–	–	–
	InfoBatch	95.61	–	–	–	–
	WLIB	95.76	–	–	–	–

by at most 2.6 percentage points, while WLIB reduces AUC from 75.6 to 68.5 and TPR at 1% FPR from 4.2% to 0.37% relative to standard training. WLIB also has the lowest AUC on the two blonde subgroups and on non-blonde/female; on non-blonde/male, all three methods are close to random (50.4–50.9). Thus, the gain in privacy does not come at the cost of the minority subgroup's accuracy.

K Comparison with static data pruning

We compare against InfoMax [Tan et al., 2025], a static pruning method that selects a subset before training by maximizing its information content. We report results on CIFAR-10 (ResNet-20) under the defense-unaware RMIA protocol.

InfoMax reduces RMIA success only marginally relative to standard training, and has lower test accuracy than both InfoBatch and WLIB.

Table 8: RMIA results and test accuracy on CIFAR-10 (ResNet-20) for static pruning with InfoMax (%).

Method	AUC ↓	TPR@0% FPR ↓	TPR@0.01% FPR ↓	TPR@1% FPR ↓	Test Acc. ↑
Standard	71.82	1.87	3.15	13.76	92.21
InfoMax	71.47	1.78	3.03	13.72	91.55
InfoBatch	66.69	0.74	1.25	8.47	91.74
WLIB	61.43	0.20	0.40	4.58	91.64

L Hard-sample accuracy during training

Section 4 infers from memorization trajectories that InfoBatch alters *when* the hard samples are learned. However, the trajectories alone do not show that hard samples are learned earlier. We therefore measure this directly. Table 9 reports the training accuracy of the hard group on CIFAR-10 (ResNet-20) at the same number of optimization steps for all methods. Comparing at matched steps compares how far each method has fitted the hard samples for the same number of updates. InfoBatch and WLIB complete training after about 6300 steps. Hence, we report the hard sample accuracy up to that point of training.

Table 9: Training accuracy of the CIFAR-10 hard group (%) at matched optimization steps.

Step	Standard	InfoBatch	WLIB
1000	54.5	50.6	50.6
2000	47.7	60.6	60.6
3000	64.2	67.4	67.4
4000	65.9	77.8	77.8
5000	64.2	90.3	90.3
6300	75.5	98.7	96.6

Standard training is slightly ahead at step 1000. From step 2000 onwards, InfoBatch fits the hard group faster, and the gap grows to 23.2 percentage points at step 6300. WLIB matches InfoBatch until annealing and ends slightly lower, since it down-weights hard samples in this phase. Together with the lower accumulation of hard-group memorization in Figure 2, this shows that InfoBatch fits hard samples earlier than standard training.

M Code availability

Our code is available at `https://github.com/chinmayj23/prune_to_protect` under the MIT license. It includes implementations of standard training, InfoBatch, WLIB and the ablations in Table 2, together with the RMIA and LiRA attacks and the memorization estimator. The repository provides the training configurations of Table 3 and maps the tables and figures in this paper to the commands that produce them.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction are closely aligned with the research presented in the main text.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The discussion section highlights that this paper sheds light on a mechanism and does not have privacy guarantees.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This work does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The experiments in the main text and the experiment setup in the Appendix clearly put forth the necessary experimental setup for the training methods, memorization estimation, and robust membership inference attacks. These contain all the information required to reproduce the claims made in the text.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the repository: `https://github.com/chinmayj23/prune_to_protect`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Table 3 in Appendix A specifies all the training details necessary to understand the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All results are reported as mean $\pm$ standard deviation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Details on the compute resources are provided in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read and adhere to the code of ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See the Impact statement.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper is an empirical evaluation of a mechanism and poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the references to the original assets used.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented, and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new assets are introduced in this work.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLM was not used for the important aspects of this research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.