RCAP: Robust, Class-Aware, Probabilistic Dynamic Dataset Pruning

Atif Hassan[1] **Swanand Khare**[2] **Jiaul H. Paik**[1]

[1]Department of Artificial Intelligence, IIT Kharagpur, Kharagpur, West Bengal, India
[2]Department of Mathematics, IIT Kharagpur, Kharagpur, West Bengal, India

Abstract

Dynamic data pruning techniques aim to reduce computational cost while minimizing information loss by periodically selecting representative subsets of input data during model training. However, existing methods often struggle to maintain strong worst-group accuracy, particularly at high pruning rates, across balanced and imbalanced datasets. To address this challenge, we propose RCAP, a Robust, Class-Aware, Probabilistic dynamic dataset pruning algorithm for classification tasks. RCAP applies a closed-form solution to estimate the fraction of samples to be included in the training subset for each individual class. This fraction is adaptively adjusted in every epoch using class-wise aggregated loss. Thereafter, it employs an adaptive sampling strategy that prioritizes samples having high loss for populating the class-wise subsets. We evaluate RCAP on six diverse datasets ranging from class-balanced to highly imbalanced using five distinct models across three training paradigms: training from scratch, transfer learning, and fine-tuning. Our approach consistently outperforms state-of-the-art dataset pruning methods, achieving superior worst-group accuracy at all pruning rates. Remarkably, with only 10% data, RCAP delivers $> 1\%$ improvement in performance on class-imbalanced datasets compared to full data training while providing an average $8.69\times$ speedup. The code can be accessed at `https://github.com/atif-hassan/RCAP-dynamic-dataset-pruning`

1 INTRODUCTION

The remarkable success of deep learning across domains such as computer vision [He et al., 2016, Dosovitskiy et al., 2021], natural language processing [Brown et al., 2020, Radford et al., 2019, OpenAI, 2023], and speech [Radford et al., 2023, Baevski et al., 2020] is largely fueled by training massive networks on datasets with millions or even billions of samples. However, this scale of training demands exorbitant computational resources over prolonged periods, incurring unsustainable monetary costs [Mindermann et al., 2022]. These expenses not only limit accessibility for resource-constrained researchers but also discourage investment in model refinement activities like hyper-parameter tuning and architecture search. Consequently, reducing training costs has emerged as a critical research challenge in deep learning.

One promising approach to mitigate these costs is to reduce the number of training updates which can be achieved by shrinking the dataset size. Approaches such as dataset distillation [Zhao and Bilen, 2023, Cazenavette et al., 2022], coreset selection [Xia et al., 2024, Yang et al., 2024, Zheng et al., 2023] and data pruning [Zhang et al., 2024, Yang et al., 2024, Okanovic et al., 2024a, Qin et al., 2024] have garnered attention with data pruning striking the best balance between performance and training cost by removing the least informative examples [Paul et al., 2021].

Pruning methods typically use scoring mechanisms to identify the most informative samples for training. Static pruning techniques [Paul et al., 2021, Yang et al., 2023, Zhang et al., 2024, Yang et al., 2024] select a fixed subset prior to training, discarding the remaining data to reduce storage and computation. However, their scoring mechanism relies on training a model for multiple epochs before determining sample importance. This process is not only expensive but also model-dependent, thus restricting its applicability to diverse downstream architectures. Dynamic dataset pruning, in contrast, recomputes subsets during training, leveraging accessible metrics like per-sample loss to adaptively select data for each epoch [Raju et al., 2021, Qin et al., 2024, Okanovic et al., 2024a]. This dynamic approach ensures that the sampled subset evolves with model training, offering near loss-less average performance even at high pruning rates while reducing overall training time.

1.1 MOTIVATION

Developing robust models is a crucial aspect of real-world AI applications as it mitigates bias against under-represented/minority groups. However, existing state-of-the-art dynamic pruning algorithms, such as RS2 [Okanovic et al., 2024a] and InfoBatch [Qin et al., 2024], overlook a critical metric: worst-group accuracy, essential for evaluating model robustness, especially in class-imbalanced datasets. Moreover, even in class-balanced datasets these methods often neglect class-specific hardness, achieving strong average performance but underperforming on harder or minority groups. For instance in CIFAR10, certain classes, such as cats and dogs, accumulate higher loss in comparison to other groups, leading to non-robust models with poor worst-class performance [Vysogorets et al., 2024]. Thus, we aim to answer the following question,

> "*Does incorporating class hardness, while performing data pruning, enhance model robustness across both balanced and imbalanced data settings?*"

1.2 OUR CONTRIBUTION

We propose RCAP, a novel, Robust, Class-Aware, Probabilistic dynamic dataset pruning algorithm for classification tasks. RCAP automatically determines the appropriate subset size for individual classes through a parameter which is updated in every epoch based on the aggregated class-wise loss of the previous epoch. Thereafter, RCAP prioritizes samples with higher loss for each subset by sampling from a distribution over per-sample losses.

We evaluate RCAP across a diverse set of datasets, spanning various scales and class imbalance levels. These include class-balanced datasets of medium scale (CIFAR10 and CIFAR100), a class-balanced large-scale dataset (ImageNet), a moderately imbalanced small-scale dataset (Waterbirds), a relatively high imbalance medium-scale dataset (CelebA), and an extremely imbalanced large-scale dataset (iNaturalist). Our experiments employ five distinct network architectures, ResNet18, ResNet50, EfficientNetV2, Dinov2 and EfficientFormerV2 across three training paradigms: training from scratch, transfer learning, and fine-tuning.

We compare against seven state-of-the-art baselines, including both dynamic and static data pruning techniques. To the best of our knowledge, this is the first comprehensive evaluation of dynamic dataset pruning algorithms in both class-balanced and imbalanced data settings in terms of worst-group performance. The results demonstrate that RCAP consistently surpasses all methods, achieving significantly superior worst-group accuracy, especially at high pruning rates across all architectures, datasets and training paradigms.

2 PRELIMINARIES

2.1 NOTATIONS

We denote $\mathcal{S} = \{(X_i, y_i)\}_{i=1}^n$ as a labelled set of input and target pairs. Here, $X_i \in \mathcal{X}$ and $y_i \in \mathbb{N}_c$ where $\mathcal{X}$ is the input space while $\mathbb{N}_c = \{1, \cdots, c\}$ with c being the number of classes and n the total number of samples. Here, $(\mathbf{X}, Y) \sim \mathcal{P}_\mathcal{D}$ where $\mathcal{P}_\mathcal{D}$ is the underlying distribution. Given a label $j \in \mathbb{N}_c$, define $\mathcal{S}_j = \{(X_k, y_k)\}_{k=1}^{n_j}$ where $\forall k, y_k = j$. Then clearly, $\mathcal{S} = \bigcup_{j=1}^c \mathcal{S}_j$ and $n = \sum_j n_j$. Let $r \in (0,1)$ be the pruning rate supplied by the user such that the total number of samples to be selected is $(1-r)n$. We define the retain set, $\mathcal{S}^t \subset \mathcal{S}$ as the subset of samples selected for training at epoch t where $|\mathcal{S}^t| = (1-r)n$. Here, $t = \{1, 2, \cdots, T\}$ where T is the total number of epochs. The retain set comprises class-wise subsets, $\mathcal{S}_j^t = \mathcal{S}^t \cap \mathcal{S}_j,\ \forall j \in \mathbb{N}_c$ where $|\mathcal{S}_j^t| = \alpha_j^t n_j$ such that, $\sum_{j=1}^c \alpha_j^t n_j = (1-r)n$. Here, α_j^t is the fraction of samples to be selected to form the subset for class j at epoch t. The set of unused samples, $\mathcal{S} \setminus \mathcal{S}^t$, at epoch t form the pruned set. Let $f_\theta(\cdot)$ be any arbitrary model parameterized by $\theta \in \mathbb{R}^m$. Let $\widetilde{f}_{\theta^t}(X_i) = \sigma\left(f_{\theta^t}(X_i)\right) \in \mathbb{R}^c$ at epoch t such that $\forall t, \forall i,\ \left\|\widetilde{f}_{\theta^t}(X_i)\right\|_1 = 1$. Here, $\sigma(\cdot)$ is the Softmax function. The loss function is denoted as, $L : \mathbb{R}^c \times \mathbb{N}_c \to \mathbb{R}$ with its value at epoch t for some input X_i being represented as $L\left(\widetilde{f}_{\theta^t}\left(X_i\right), y_i\right)$. For brevity, we represent the loss at epoch t for some input X_i as $L\left(\widetilde{f}_{\theta^t}(X_i)\right)$. The derivative of $L\left(\widetilde{f}_{\theta^t}\left(X_i\right)\right)$ at epoch t for any input X_i is denoted as $\nabla_{\theta^t} L\left(\widetilde{f}_{\theta^t}(X_i)\right)$. Let $\mathcal{B}^p$ denote a batch of examples at iteration p with $|\mathcal{B}^p| = b$ being the batch size. Then the total number of iterations at epoch t over the entire dataset and a subset, $\mathcal{S}^t \subset \mathcal{S}$, are $\lceil |\mathcal{S}|/b \rceil$ and $\lceil |\mathcal{S}^t|/b \rceil$, respectively. We use η to denote the learning rate.

3 RCAP

An effective data pruning algorithm should account for class-wise performance when selecting samples for the retain set. This is because the performance of individual groups/classes vary for a given classification task, thus requiring non-uniform representation in the selected subset. Some methods such as Data Diet [Paul et al., 2021], implicitly address this by inducting high-error samples into the retain set. However, at high pruning rates, such strategies risk discarding classes with consistently low-error samples. Other techniques such as MetriQ [Vysogorets et al., 2024] incorporate class-wise performance but rely on ad-hoc rules to determine sample allocation. To overcome these limitations, we propose the following two fundamental problems that any effective data pruning algorithm should solve:

- Determining the appropriate subset size for each class in the retain set.
- Selecting the most informative samples within each subset.

We address the first problem by adaptively adjusting the class-wise subset size in each epoch, as formalized in Theorem 3.1 (Section 3.1). The second problem is tackled through a novel epoch-wise adaptive sampling strategy, detailed in Section 3.2.

3.1 ADAPTIVE PER-CLASS SUBSET SIZE

Allocating more training samples to classes that a model perceives as difficult can lead to performance improvements on underrepresented or challenging groups [Vysogorets et al., 2024]. Theorem 3.1 formalizes this intuition, demonstrating that classes with higher loss values should have a proportionally larger representation in the training subset.

Theorem 3.1. *Let, the total empirical error be given by,*

$$E^{t+1} = \sum_j \frac{p_j}{\alpha_j^{t+1} n_j} \widetilde{E}_j^{t+1}$$

$$\textit{where} \quad \widetilde{E}_j^{t+1} = \sum_{X_i \in \mathcal{S}_j^{t+1}} L\left(\widetilde{f}_{\theta^t}\left(X_i\right)\right) \quad \textit{and} \quad p_j = \frac{n_j}{n}$$

Then, under the assumption of full batch gradient descent, the optimal solution to the minimization problem

$$\min_{\alpha_j^{t+1}} E^{t+1}$$

$$\textit{subject to} \quad \sum_{j=1}^{c} \alpha_j^{t+1} n_j = (1-r)n$$

$$\textit{is given by} \quad \widehat{\alpha}_j^{t+1} = \frac{\sqrt{p_j \widetilde{E}_j^{t+1}}}{\sum_j \sqrt{p_j \widetilde{E}_j^{t+1}}}(1-r)\frac{n}{n_j}$$

Proof. Introducing the Lagrange multiplier λ, the optimization problem becomes,

$$G = E^{t+1} + \lambda\left(\sum_{j=1}^{c} \alpha_j^{t+1} n_j - (1-r)n\right)$$

If $\left(\widehat{\alpha}_j^{t+1}, \widehat{\lambda}\right)$ is an optimal pair, then the optimality conditions imply:

$$\begin{aligned} \left.\frac{\partial G}{\partial \alpha_j^{t+1}}\right|_{\left(\widehat{\alpha}_j^{t+1}, \widehat{\lambda}\right)} &= -\frac{p_j \widetilde{E}_j^{t+1}}{\left(\widehat{\alpha}_j^{t+1}\right)^2 n_j} + \widehat{\lambda} n_j = 0 \\ \implies \widehat{\alpha}_j^{t+1} &= \frac{\sqrt{p_j \widetilde{E}_j^{t+1}}}{\sqrt{\widehat{\lambda}} n_j} \end{aligned} \tag{1}$$

Substituting the value of $\widehat{\alpha}_j^{t+1}$ in the constraint gives us,

$$\begin{aligned} \frac{1}{\sqrt{\widehat{\lambda}}} \sum_j \sqrt{p_j \widetilde{E}_j^{t+1}} &= (1-r)n \\ \implies \frac{1}{\sqrt{\widehat{\lambda}}} &= \frac{(1-r)n}{\sum_j \sqrt{p_j \widetilde{E}_j^{t+1}}} \end{aligned} \tag{2}$$

Replacing the value of $\sqrt{\widehat{\lambda}}$ from Eqn. 2 in Eqn. 1, we get,

$$\widehat{\alpha}_j^{t+1} = \frac{\sqrt{p_j \widetilde{E}_j^{t+1}}}{\sum_j \sqrt{p_j \widetilde{E}_j^{t+1}}}(1-r)\frac{n}{n_j} \tag{3}$$

□

Remark 1. Eqn. 3 provides a closed-form solution for determining the appropriate class-wise fraction, which suggests allocating more samples to classes with larger error. By prioritizing high-error groups, the total training error can be reduced. However, Eqn. 3 cannot be directly implemented since the optimal value for the class-wise fraction of samples in the retained set for epoch $t+1$ requires loss values that are yet to be observed. Instead, approximating $\widetilde{E}_j^{t+1}$ with $\widetilde{E}_j^t$ in Eqn. 3 can resolve this issue. However, doing so incurs some approximation error which, as shown in Eqn. 4, is bounded.

$$\begin{aligned} \left|\widetilde{E}_j^t - \widetilde{E}_j^{t+1}\right| &\leq \frac{\eta K_1}{(1-r)n} \left\| \sum_{X_i \in \mathcal{S}^t} \nabla_{\theta^{t-1}} L\left(\widetilde{f}_{\theta^{t-1}}(X_i)\right) \right\|_2 \\ &+ \left|\mathcal{S}_j^t\right| K_2 \left\|\widetilde{f}_{\theta^t}\left(X\right) - \widetilde{f}_{\theta^t}\left(X'\right)\right\|_2 + \sum_{i=\left|\mathcal{S}_j^t\right|+1}^{\left|\mathcal{S}_j^{t+1}\right|} L\left(\widetilde{f}_{\theta^t}\left(X_i'\right)\right) \end{aligned} \tag{4}$$

Here K_1 and K_2 are the Lipschitz constants for L with respect to the change in parameters and input, respectively, while $X \in \mathcal{S}_j^t$ and $X' \in \mathcal{S}_j^{t+1}$. The full derivation is provided in Section B of the Appendix. The gradient norm reduces exponentially during training [Boyd, 2004] while $\eta \ll 1$ and $(1-r)n \gg 1$ ensure that the first term in the R.H.S. quickly converges early on in training. The second term converges quickly early on in training [Paul et al., 2021] as $\left\|\widetilde{f}_{\theta^t}\left(X\right) - \widetilde{f}_{\theta^t}\left(X'\right)\right\|_2$ is the norm of the difference between the confidence scores across c classes for two samples from the same class with $\|X\|_1 = \|X'\|_1 = 1$. Under the assumption that the fraction of samples allocated to each class does not change significantly between consecutive epochs, the third term in the R.H.S. of the inequality also decreases as training progresses. Thus, the approximation error reduces as training progresses. Therefore, rewriting Eqn. 3:

$$\widehat{\alpha}_j^{t+1} = \frac{\sqrt{p_j \widetilde{E}_j^t}}{\sum_j \sqrt{p_j \widetilde{E}_j^t}}(1-r)\frac{n}{n_j} \tag{5}$$

We find that this approximation to the optimal value of α_j^{t+1}, obtained in Eqn. 5, works well in practice as demonstrated in Tables 1 and 2.

Implementation Detail: Note that $\widetilde{E}_j^0$ is the class-wise aggregated loss at model initialization which determines $\widehat{\alpha}_j^1$. Furthermore, closely inspecting Eqn. 5 reveals that the condition $\widehat{\alpha}_j^{t+1} > 1$ is plausible as the fraction is unconstrained. To mitigate this issue, we perform the following operation,

$$\widehat{\alpha}_j^{t+1} = \begin{cases} 1 & \text{if } \widehat{\alpha}_j^{t+1} > 1 \\ \frac{\sqrt{p_j \widetilde{E}_j^t}}{\sum_j \left(\sqrt{p_j \widetilde{E}_j^t}\right) m_j} (1-r) \frac{n-k}{n_j} & \text{otherwise} \end{cases}$$

where,

$$m_j = \begin{cases} 1 & \text{if } \widehat{\alpha}_j^{t+1} < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$k = \sum_j (1-m_j) n_j \tag{6}$$

In doing so, we guarantee that $\widehat{\alpha}_j^{t+1} \leq 1$ with excess values being re-distributed among the remaining classes.

3.2 ADAPTIVE PER-CLASS SAMPLE SELECTION

The goal of any dynamic dataset pruning algorithm is to train a model on a carefully selected subset of data at each epoch such that the model's performance is indistinguishable from a model trained on the full dataset. Formally, this goal can be expressed as,

$$\mathop{\mathbb{E}}_{(X_i,y_i)\sim\mathcal{P}_\mathcal{D}} \left[\left| L\left(\widetilde{f}_{\widetilde{\theta}^T}(X_i)\right) - L\left(\widetilde{f}_{\theta^T}(X_i)\right) \right| \right] \leq \epsilon \tag{7}$$

where,

$$\theta^T = \theta^1 - \eta \sum_{t=1}^{T} \sum_{p=1}^{\left\lceil \frac{|\mathcal{S}|}{b} \right\rceil} \frac{1}{b} \sum_{(X_i,y_i)\in\mathcal{B}^p} \nabla_{\theta^{t,p}} L\left(\widetilde{f}_{\theta^{t,p}}(X_i)\right) \tag{8}$$

$$\widetilde{\theta}^T = \theta^1 - \eta \sum_{t=1}^{T} \sum_{p=1}^{\left\lceil \frac{|\mathcal{S}^t|}{b} \right\rceil} \frac{1}{b} \sum_{(\widetilde{X}_i,\widetilde{y}_i)\in\widetilde{\mathcal{B}}^p} \nabla_{\widetilde{\theta}^{t,p}} L\left(\widetilde{f}_{\widetilde{\theta}^{t,p}}\left(\widetilde{X}_i\right)\right) \tag{9}$$

Here, $\mathcal{B}^p$ and $\widetilde{\mathcal{B}}^p$ are batches sampled from $\mathcal{S}$ and $\mathcal{S}^t$, respectively, at iteration p. Here, θ^T and $\widetilde{\theta}^T$ are the parameters obtained after training on $\mathcal{S}$ and its subset, respectively. Similarly, $\theta^{t,p}$ and $\widetilde{\theta}^{t,p}$ are the parameters obtained at epoch t and iteration p after training on $\mathcal{S}$ and $\mathcal{S}^t$, respectively. We now look at the condition to achieve Eqn. 7. Let $\frac{1}{b}\sum_{(X_i,y_i)\in\mathcal{B}^p} \nabla_{\theta^{t,p}} L\left(\widetilde{f}_{\theta^{t,p}}(X_i)\right) = g^{t,p}$ and $\frac{1}{b}\sum_{(X_i,y_i)\in\widetilde{\mathcal{B}}^p} \nabla_{\widetilde{\theta}^{t,p}} L\left(\widetilde{f}_{\widetilde{\theta}^{t,p}}(X_i)\right) = \widetilde{g}^{t,p}$. Assuming that L is Lipschitz continuous having Lipschitz constant K_1 with respect to the change in parameters, we get:

$$\left| L\left(\widetilde{f}_{\widetilde{\theta}^T}(X_i)\right) - L\left(\widetilde{f}_{\theta^T}(X_i)\right) \right| \leq K_1 \left\| \widetilde{\theta}^T - \theta^T \right\|_2 \tag{10}$$

Replacing $\widetilde{\theta}^T$ and θ^T from Eqns. 8 and 9 in Eqn. 10, and taking expectation on both sides, we get:

$$\mathop{\mathbb{E}}_{(X_i,y_i)\sim\mathcal{P}_\mathcal{D}} \left[\left| L\left(\widetilde{f}_{\widetilde{\theta}^T}(X_i)\right) - L\left(\widetilde{f}_{\theta^T}(X_i)\right) \right| \right] \leq$$

$$K_1 \eta \mathop{\mathbb{E}}_{(X_i,y_i)\sim\mathcal{P}_\mathcal{D}} \left[\left\| \sum_{t=1}^{T} \left(\sum_{p=1}^{\left\lceil \frac{|\mathcal{S}|}{b} \right\rceil} g^{t,p} - \sum_{p=1}^{\left\lceil \frac{|\mathcal{S}^t|}{b} \right\rceil} \widetilde{g}^{t,p} \right) \right\|_2 \right] \tag{11}$$

Hence, to achieve Eqn. 7, the right-hand-side in Eqn. 11 needs to be minimized. One can observe that in each epoch, the term $\sum_{p=1}^{\lceil (|\mathcal{S}|/b) \rceil} g^{t,p}$ is dominated by the samples with the largest gradient norm. We empirically find that the cross-entropy loss and the magnitude of the gradient exhibit a monotonic relation (see Section A in the Supplementary Materials). This empirical relation is further reinforced by Paul et al. [2021], as they observe that "examples that are learned faster and maintain small error over training have a smaller GraNd score on average," where the GraNd score is the gradient norm of a sample. Thus, we choose to form $\mathcal{S}^t$ with high-loss samples to approximately minimize the right-hand side in Eqn. 11. A naïve approach involves sorting samples in $\mathcal{S}$ by their loss values which is computationally expensive ($O(\log n)$ per sample, e.g. [Paul et al., 2021]). Instead, RCAP samples from every $\mathcal{S}_j$ by defining $\mathcal{S}_j^t \subseteq \mathcal{S}_j$ as the set of examples sampled at epoch t for class j in the following manner.

$$\mathcal{S}_j^{t+1} = \{(X_i, y_i)\}_{i=1}^{\widehat{\alpha}_j^{t+1} n_j} \sim \mathcal{P}_j^{t+1}(X_i) \tag{12a}$$

$$\mathcal{P}_j^{t+1}(X_i) = \frac{e^{\left(\phi_j^t(X_i)/\beta\right)}}{\sum_{X_q\in\mathcal{S}_j} e^{\left(\phi_j^t(X_q)/\beta\right)}} \tag{12b}$$

$$\phi_j^t(X_i) = \begin{cases} \gamma\left(L\left(\widetilde{f}_{\theta^t}(X_i)\right), j\right) & \text{if } X_i \in \mathcal{S}_j^t \\ \phi_j^{t-1}(X_i) & \text{otherwise} \end{cases} \tag{12c}$$

$$\gamma(x, j) = \min\left(x, \max\left(\phi_j^0(X_i)\right)\right) \quad \forall X_i \in \mathcal{S}_j \tag{12d}$$

$$\phi_j^0(X_i) = L\left(\widetilde{f}_{\theta^0}(X_i)\right) \tag{12e}$$

Before training ensues, all examples are forward passed through a randomly initialized network and the corresponding loss values are stored in ϕ_j^0 as shown in Eqn. 12e. These loss values correspond to completely random predictions. Next, ϕ_j^0 is used to compute the aggregate class-wise losses, $\widetilde{E}_j^0$, that determine the fraction of samples to be allocated per class, $\widehat{\alpha}_j^1$, as shown in Eqn. 5. Next, a class-wise probability distribution is generated over the collected loss values, ϕ_j^0, using a Softmax function with β as the temperature hyper-parameter as shown in Eqn. 12b. The training subset is then generated by sampling over this distribution as per Eqn. 12a. The model is then trained using these samples. Following an epoch of training, ϕ_j^0 is updated with the new loss values corresponding to the selected samples, forming ϕ_j^1 as per Eqn. 12c which in turn determines $\widehat{\alpha}_j^2$. The distribution is updated using Eqn. 12b and sampling re-occurs

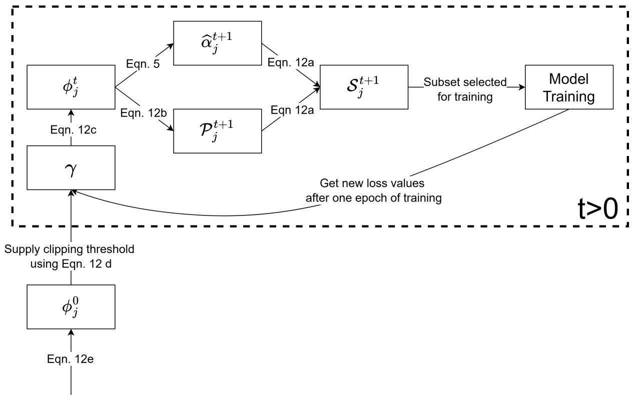


Figure 1: An overview of the sequence of steps involved in RCAP

as per Eqn. 12a. This iterative process continues until the end of training. Fig. 1 gives a graphical overview of the sequence of steps involved in each epoch. It is important to note that the softmax-based sampling distribution built from the loss values can become highly skewed due to the presence of a few large values, leading to unstable or biased subset selection. Hence, we define a clipping function in Eqn. 12d with the clipping threshold as the maximum loss observed in epoch 0, before training begins. If a sample's loss exceeds this baseline during training, we assume the model is making a deliberate or persistent error, possibly due to label noise or input corruption. By capping the per-sample loss before computing the sampling distribution, we reduce the likelihood of repeatedly selecting such samples, thereby keeping the pruning process fairly robust.

Crucially, RCAP's per-class sample size determination and per-class sample selection modules incur no additional computational overhead, as they are determined entirely based on loss values that are computed during the forward pass. Such a strategy allows our proposed approach to achieve a per-sample time complexity of $O(1)$. Algorithm 1 provides the implementation details, where $I[j] = \{i \mid \forall\, y_i \in Y, y_i = j\}$ at $t = 1$.

Algorithm 1: The proposed RCAP Algorithm

Input : Dataset $\mathcal{S} = (\mathbf{X}, Y)$, Number of classes c, Pruning rate $r \in (0, 1]$, Number of training epochs, T, Softmax temperature β and Set of indices selected I

Output : Trained Model.

```
α, m = [ ], [ ]
n = length(Y)
φ ← L(X_i)  ∀X_i ∈ S
for t = 1 to T do
    for j = 1 to c do
        idx ← {i | ∀ y_i ∈ Y, y_i = j}
        m[j] = length(idx)
        α[j] = ⌊ (sqrt(n c[j] / n) φ[I[j]]) / (Σ sqrt(n c[j] / n) φ[I[j]]) × (1 − r) × n / m[j] ⌋
        Use Eqn. 6 to fix violating α
        P = e^(φ[idx]/β) / Σ e^(φ[idx]/β)
        I[j] = {i | ∀ (X_i, y_i) ∈ 𝒳 ∼ P(X[idx], Y[idx])}
          and |𝒳| = (α[j] × m[j])
    Update model parameters using I
    Update φ
```

$$\alpha[j] = \left\lfloor \frac{\sqrt{\frac{nc[j]}{n}}\,\phi[I[j]]}{\sum \sqrt{\frac{nc[j]}{n}}\,\phi[I[j]]} \times (1-r) \times \frac{n}{m[j]} \right\rfloor$$

$$\mathcal{P} = \frac{e^{(\phi[\text{idx}]/\beta)}}{\sum e^{(\phi[\text{idx}]/\beta)}}$$

$$\text{I[j]} = \{i \mid \forall\, (X_i, y_i) \in \mathcal{X} \sim \mathcal{P}\left(\mathbf{X}[idx], Y[idx]\right)\} \text{ and } |\mathcal{X}| = (\alpha[j] \times m[j])$$

4 EXPERIMENTS

4.1 BASELINES

We evaluate our proposed approach against seven representative baselines: two static and four dynamic data pruning methods as well as a coreset selection technique. **CCS** [Zheng et al., 2023] is a state-of-the-art coreset selection technique that maximizes data distribution coverage. **TDDS** [Zhang et al., 2024] is the current leading static data pruning method. It incorporates training dynamics to determine sample importance. **MetriQ** [Vysogorets et al., 2024] is a class-ratio-aware static data pruning method designed to reduce classification bias. **UCB** Raju et al. [2021] is one of the earliest dynamic data pruning approaches utilizing sample uncertainty and Reinforcement Learning inspired exploration to prune unimportant samples. **InfoBatch** [Qin et al., 2024] is a state-of-the-art dynamic data pruning method that selects samples based on their loss and adaptively determines the pruning ratio via a hyperparameter. **RS2** [Okanovic et al., 2024b] is another state-of-the-art dynamic data pruning approach that performs pruning by random selection with and without replacement. **Note:** For brevity's sake, we omit comparisons with older methods (e.g., GraNd, CRAIG, GradMatch, Glister, and CREST) as all considered baselines have demonstrated superior performance in prior studies.

4.2 DATASET AND MODEL DETAILS

We benchmark RCAP on six diverse datasets in terms of scale and class imbalance using five distinct networks. **CIFAR10** [Krizhevsky et al., 2009] is a medium-scale, class-balanced dataset comprising 10 classes, each containing 5000 samples over which we trained the ResNet18 model [He et al., 2016] from scratch. **CIFAR100** [Krizhevsky et al., 2009] is a medium-scale, class-balanced dataset comprising 100 classes, each containing 500 samples, over which we trained the ResNet18 model from scratch as well. **ImageNet** [Deng et al., 2009] is a large-scale, relatively class-balanced dataset of over 1.2 million images comprising 1000 classes, each containing approximately 1300 samples with slight variations. We trained a two layer MLP on top of the Dinov2-b model [Radosavovic et al., 2020] on this dataset. **Waterbirds** [Sagawa et al., 2019] is a moderately class-imbalanced, small scale dataset with 4795 images

Table 1: Worst Group Accuracy (Top-1) averaged over three separate runs. The best scores are shown in bold while the second best are underlined. The time, in minutes, required by RCAP in comparison to full data training is also reported.

Dataset	Prune Rate	CCS(%)	MetriQ(%)	TDDS(%)	UCB(%)	InfoBatch(%)	RS2 w/r(%)	RS2 w/o(%)	RCAP(%)	Time
CIFAR10	00%	91.13±0.29	91.13±0.29	91.13±0.29	91.13±0.29	91.13±0.29	91.13±0.29	91.13±0.29	91.13±0.29	23.3
	50%	88.87±0.29	<u>90.53±0.41</u>	90.27±0.33	90.00±0.45	89.97±0.48	89.83±0.24	90.10±0.59	**90.60±0.14**	13.3
	70%	83.50±1.31	86.63±0.29	84.57±1.11	87.97±0.45	88.50±0.16	88.43±0.49	<u>88.60±0.22</u>	**89.73±0.38**	6.7
	80%	77.53±0.87	82.50±0.62	80.17±1.11	84.53±0.34	86.53±0.97	87.43±0.48	<u>88.10±0.50</u>	**88.70**±0.37	5.3
	90%	67.20±0.36	71.30±1.08	68.63±0.95	73.17±0.38	<u>83.53±0.45</u>	79.63±0.45	80.47±0.33	**85.07±0.34**	3.3
CIFAR100	00%	55.00±1.41	55.00±1.41	55.00±1.41	55.00±1.41	55.00±1.41	55.00±1.41	55.00±1.41	55.00±1.41	23.3
	50%	43.67±0.47	<u>54.00±1.41</u>	43.67±0.47	49.33±0.47	52.33±0.47	53.67±1.25	<u>54.00±0.00</u>	**55.00±1.63**	13.3
	70%	34.67±0.47	43.67±0.94	23.33±1.70	42.57±1.89	51.00±1.63	50.33±1.70	<u>52.00±0.82</u>	**52.67±0.94**	6.7
	80%	21.00±0.82	30.33±0.47	15.67±0.47	33.33±1.89	<u>50.33±0.47</u>	<u>50.33±1.70</u>	49.00±0.00	**50.67±0.47**	5.3
	90%	07.00±0.82	11.00±1.63	05.33±0.47	14.33±1.25	<u>46.67±0.47</u>	35.67±0.94	35.33±0.94	**48.33±1.89**	3.3
ImageNet	00%	20.67±2.49	20.67±2.49	20.67±2.49	20.67±2.49	20.67±2.49	20.67±2.49	20.67±2.49	20.67±2.49	249.0
	50%	00.00±0.00	00.00±0.00	00.00±0.00	00.00±0.00	18.67±2.49	<u>22.67±0.94</u>	19.33±0.94	**24.00±0.00**	130.5
	70%	00.00±0.00	00.00±0.00	00.00±0.00	00.00±0.00	18.00±2.83	<u>20.00±1.64</u>	<u>20.00±4.32</u>	**24.00±2.83**	82.7
	80%	00.00±0.00	00.00±0.00	00.00±0.00	00.00±0.00	14.00±0.00	20.67±0.94	<u>23.33±0.94</u>	**24.00±1.63**	58.0
	90%	00.00±0.00	00.00±0.00	00.00±0.00	00.00±0.00	20.00±2.83	19.33±1.89	<u>23.33±2.49</u>	**26.00±0.00**	30.5
Waterbirds	00%	90.27±0.67	90.27±0.67	90.27±0.67	90.27±0.67	90.27±0.67	90.27±0.67	90.27±0.67	90.27±0.67	70.0
	50%	89.97±0.47	89.22±0.27	90.10±0.35	50.00±0.00	<u>90.48±0.71</u>	<u>90.48±0.17</u>	89.72±1.42	**91.34±0.01**	35.0
	70%	91.02±0.06	82.35±0.87	90.11±0.18	50.00±0.00	<u>90.60±0.61</u>	90.10±0.18	90.35±0.35	**92.09±0.09**	20.0
	80%	90.40±0.21	81.73±0.26	<u>90.71±0.52</u>	50.00±0.00	89.83±0.25	89.61±1.70	89.60±0.94	**91.60±0.18**	15.0
	90%	90.27±0.04	79.05±0.37	<u>90.48±0.18</u>	50.00±0.00	89.06±0.57	89.78±0.35	88.97±0.47	**91.21±0.38**	10.0
CelebA	00%	90.14±0.35	90.14±0.35	90.14±0.35	90.14±0.35	90.14±0.35	90.14±0.35	90.14±0.35	90.14±0.35	21.7
	50%	86.43±0.80	<u>91.72±1.40</u>	87.44±1.80	50.00±0.00	86.47±2.62	88.99±0.91	87.97±2.46	**92.30±0.16**	12.1
	70%	86.28±1.27	<u>91.45±0.30</u>	88.29±2.12	50.00±0.00	84.46±5.06	85.76±2.21	88.27±2.01	**92.19±0.22**	5.8
	80%	89.58±0.31	<u>90.70±0.26</u>	86.16±0.59	50.00±0.00	82.17±0.34	88.21±1.58	88.96±1.54	**91.64±0.57**	4.0
	90%	84.75±1.99	<u>89.39±1.48</u>	80.57±3.02	50.00±0.00	79.35±0.26	81.49±1.72	84.25±2.05	**91.24±0.41**	2.0
iNaturalist	00%	69.66±1.17	69.66±1.17	69.66±1.17	69.66±1.17	69.66±1.17	69.66±1.17	69.66±1.17	69.66±1.17	58.5
	50%	65.62±0.13	<u>65.97±0.48</u>	49.32±1.79	0.00±0.00	62.76±0.26	61.73±1.46	63.01±1.37	**66.44±2.06**	34.1
	70%	61.12±0.12	<u>65.94±0.82</u>	48.61±1.39	00.00±0.00	40.42±8.91	51.37±2.05	54.79±0.00	**69.18±2.06**	19.6
	80%	61.53±2.09	<u>65.70±0.70</u>	26.05±1.74	00.00±0.00	36.31±6.17	40.42±4.8	37.68±0.69	**69.18±0.69**	10.2
	90%	56.64±0.89	<u>65.14±4.31</u>	00.00±0.00	00.00±0.00	05.48±4.11	03.41±0.71	00.69±0.69	**68.49±2.74**	6.7

split into land birds and water birds ($76.8\% vs. 23.2\%$). We fine-tuned the EfficientNet-b3 model [Tan and Le, 2019] pre-trained on ImageNet. **CelebA** [Liu et al., 2015] is a relatively high class-imbalanced, medium-scale dataset containing over $160K$ images. We chose the blonde 85.1% vs not blonde 14.9%, binary classification task. We train an EfficientFormerV2 [Li et al., 2023] from scratch, for this dataset. **iNaturalist** [Van Horn et al., 2018] is a large-scale, extremely imbalanced dataset with over $600K$ images across 13 superclasses. The largest group contains $196,613$ images, while the smallest has 381. We fine-tune an ImageNet pre-trained ResNet50 [He et al., 2016].

4.3 TRAINING DETAILS

To ensure fair evaluation, we re-implement all baselines and verify that the Top-1 average accuracy of each method matches its corresponding reported value. For robustness analysis, we report the worst group accuracy and corresponding average group accuracy (Top-1). In doing so, the average group accuracy of each method as reported in their corresponding manuscripts changes considerably, especially at high pruning rates. All static methods utilize the same network for subset selection and training which is the best-case scenario for such techniques. For a fair comparison between InfoBatch and other baselines, we adjust the number of training iterations as recommended by Qin et al. [2024]. To understand RCAP's training efficiency, we report its training time across all pruning rates as well as the total time for full dataset training, in minutes. Further training specifics are detailed in Section C of the Supplementary Material.

4.4 RESULTS

Table 1 presents the Top-1 worst-group accuracy across six datasets at four pruning rates. RCAP consistently outperforms all baselines in every experiment. Notably, on ImageNet, Waterbirds and CelebA, it surpasses full-data train-

Table 2: Average Group Accuracy (Top-1) averaged over three separate runs. The best scores are shown in bold while the second best are underlined. The time, in minutes, required by RCAP in comparison to full data training is also reported.

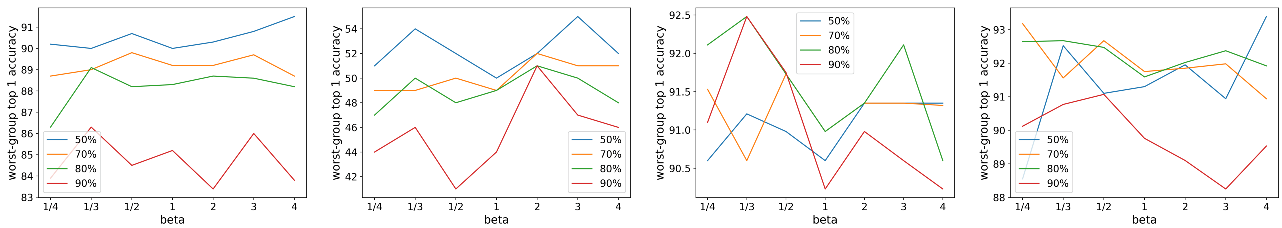
Dataset	Prune Rate	CCS(%)	MetriQ(%)	TDDS(%)	UCB(%)	InfoBatch(%)	RS2 w/r(%)	RS2 w/o(%)	RCAP(%)	Time
CIFAR10	00%	95.41±0.10	95.41±0.10	95.41±0.10	95.41±0.10	95.41±0.10	95.41±0.10	95.41±0.10	95.41±0.10	23.3
	50%	93.85±0.16	92.81±0.19	**94.88±0.06**	94.78±0.04	94.84±0.04	94.85±0.14	94.64±0.17	94.81±0.24	13.3
	70%	90.10±0.59	89.89±0.40	91.79±0.61	93.80±0.17	94.14±0.21	94.19±0.16	94.09±0.39	**94.40±0.14**	6.7
	80%	86.50±0.38	86.31±0.54	89.59±0.72	91.24±0.28	93.44±0.36	93.22±0.29	**93.50±0.32**	93.04±0.16	5.3
	90%	81.20±0.17	76.26±1.00	83.60±1.03	84.29±0.83	**91.83±0.85**	88.82±0.24	89.11±0.81	91.45±0.27	3.3
CIFAR100	00%	77.85±0.56	77.85±0.56	77.85±0.56	77.85±0.56	77.85±0.56	77.85±0.56	77.85±0.56	77.85±0.56	23.3
	50%	69.16±0.51	69.50±0.44	71.87±0.54	74.95±0.32	76.03±0.47	76.35±0.36	76.76±0.38	**76.90±0.30**	13.3
	70%	64.85±0.20	60.98±0.36	64.94±0.45	69.95±0.98	75.53±0.30	74.86±0.59	**75.76±0.23**	75.63±0.13	6.7
	80%	53.15±1.30	51.03±1.35	57.13±1.33	62.13±0.48	**74.68±0.11**	73.77±0.71	73.68±0.40	74.62±0.13	5.3
	90%	35.42±0.72	29.99±1.04	40.98±1.69	40.03±1.65	**71.93±0.38**	66.90±0.29	66.59±0.76	70.92±0.22	3.3
ImageNet	00%	84.47±0.05	84.47±0.05	84.47±0.05	84.47±0.05	84.47±0.05	84.47±0.05	84.47±0.05	84.47±0.05	249
	50%	74.78±0.19	71.78±0.16	78.78±0.27	80.23±0.39	**84.40±0.03**	84.17±0.09	84.31±0.07	84.37±0.04	130.5
	70%	73.95±0.21	70.95±0.39	75.82±0.72	77.29±0.15	84.09±0.18	83.88±0.11	**84.14±0.06**	83.89±0.13	82.7
	80%	69.25±0.19	70.98±0.21	71.09±0.44	72.43±0.59	83.78±0.05	83.77±0.03	**83.94±0.01**	83.46±0.06	58.0
	90%	62.23±0.41	70.16±0.49	69.13±0.63	71.24±0.96	83.46±0.06	83.19±0.02	83.49±0.03	**83.54±0.02**	30.5
Waterbirds	00%	90.87±0.30	90.87±0.30	90.87±0.30	90.87±0.30	90.87±0.30	90.87±0.30	90.87±0.30	90.87±0.30	70.0
	50%	90.84±0.40	89.42±0.24	90.71±0.21	50.00±0.00	91.65±0.61	91.15±0.74	90.54±1.08	**91.78±0.42**	35.0
	70%	91.40±0.06	84.41±1.62	90.98±0.58	50.00±0.00	91.42±0.64	90.30±0.16	90.82±0.36	**92.26±0.56**	20.0
	80%	90.61±0.14	83.49±0.96	90.93±0.59	50.00±0.00	90.09±0.18	90.01±1.30	90.53±0.29	**91.98±0.40**	15.0
	90%	90.33±0.34	81.84±0.34	90.77±0.08	50.00±0.00	89.42±0.72	89.99±0.46	89.36±0.35	**91.47±0.33**	10.0
CelebA	00%	92.04±0.35	92.04±0.35	92.04±0.35	92.04±0.35	92.04±0.35	92.04±0.35	92.04±0.35	92.04±0.35	21.7
	50%	91.15±0.09	**93.14±0.36**	91.45±0.57	50.00±0.00	91.28±0.59	91.94±0.26	90.87±0.35	92.84±0.36	12.1
	70%	91.28±0.47	92.42±0.05	91.10±0.27	50.00±0.00	90.19±1.75	90.93±0.67	91.03±1.05	**93.00±0.07**	5.8
	80%	91.48±0.09	91.68±0.19	90.53±0.38	50.00±0.00	89.44±0.09	90.20±0.79	90.19±0.54	**92.39±0.26**	4.0
	90%	87.08±0.44	91.14±0.62	87.46±0.64	50.00±0.00	87.79±0.19	88.50±0.48	89.38±0.55	**91.47±0.31**	2.0
iNaturalist	00%	83.62±0.06	83.62±0.06	83.62±0.06	83.62±0.06	83.62±0.06	83.62±0.06	83.62±0.06	83.62±0.06	58.5
	50%	84.03±0.06	84.19±0.16	80.32±0.26	07.69±0.00	81.65±0.05	81.45±0.18	81.87±0.04	**84.26±0.09**	34.1
	70%	81.86±1.14	82.50±0.17	76.12±0.48	07.69±0.00	75.40±0.74	78.82±0.33	79.85±0.15	**84.12±0.36**	19.6
	80%	80.25±0.38	82.81±0.36	70.20±0.30	07.69±0.00	76.73±0.03	76.31±0.16	76.43±0.86	**83.93±0.24**	10.2
	90%	79.74±0.16	78.32±4.50	63.19±0.14	07.69±0.00	70.67±0.85	70.40±0.32	70.34±0.76	**82.97±0.57**	6.7

ing, even at a 90% pruning rate. The improvement stems from the class imbalance in these datasets, ranging from mild to high. By pruning aggressively, RCAP naturally retains fewer samples from the majority class while preserving most or all minority-class samples. This results in a more balanced classification task, which enhances worst-group accuracy. Other pruning methods like MetriQ and CCS also benefit from the same effect. We find that on large-scale imbalanced datasets with few classes, particularly CelebA and iNaturalist, static pruning methods such as CCS and MetriQ perform significantly better than dynamic pruning techniques. However, such methods fail entirely, in terms of worst group accuracy, on ImageNet where the number of classes is large. Note that all static pruning techniques are evaluated in their best-case scenario, i.e., the architecture used for data pruning and consequent training on the retained data are the same. We find that UCB, which was originally only tested on CIFAR10 and CIFAR100, performs the worst among all baselines with the resultant models producing random output for four out of six datasets. Table 2 reports Top-1 average-group accuracy. RCAP performs comparably to existing methods on class-balanced datasets. On class-imbalanced datasets, it outperforms all baselines and even full-data training, primarily due to its gains in worst-group accuracy.

Beyond accuracy, RCAP is highly efficient. It delivers up to $8.69\times$ speed-up in comparison to full-data training with less than 1% drop in performance on ImageNet, Waterbirds, CelebA and iNaturalist datasets, on average, as demonstrated in Table 1 (or Table 2). This combination of efficiency and robustness establishes RCAP as the new state-of-the-art in robust dynamic dataset pruning.

4.5 PERFORMANCE WITH VARYING β

The Softmax temperature hyper-parameter, β, is the sole hyper-parameter in RCAP and plays a critical role in deter-



(a) CIFAR10 (b) CIFAR100 (c) Waterbirds (d) CelebA

Figure 2: Variation of the Softmax temperature hyper-parameter, β, across different pruning rates over four datasets: CIFAR10, CIFAR100, Waterbirds, and CelebA.

mining the sampling probabilities, thereby influencing the overall performance of our algorithm. Specifically, β controls the sharpness of the sampling distribution with values > 1 promoting a more uniform sampling strategy, while values < 1 prioritizing samples having high loss by assigning them higher sampling probabilities. To evaluate the impact of β on RCAP's performance, we conduct an ablation study by varying β within the range $[\frac{1}{4}, 4]$ across four datasets, CIFAR10, CIFAR100, Waterbirds, and CelebA over four different pruning rates, 50%, 70%, 80%, and 90%. The results, presented in Fig. 2, reveal that the choice of β is crucial, especially at higher pruning rates where fewer samples are retained. Interestingly, at relatively moderate pruning rates (50%), $\beta > 1$ performs better with the performance being more stable across a wider range of β, allowing for more flexibility in hyper-parameter selection. On the other hand, at high pruning rates (90%), $\beta < 1$ achieves better results with the choice of β becoming increasingly critical as fewer samples are retained. Suboptimal values lead to sharp performance drops, particularly in Waterbirds and CIFAR100. These findings suggest that β must be carefully tuned based on dataset characteristics and pruning rates. We recommend that starting with $\beta = \left\{\frac{1}{3}, \frac{1}{2}, 2, 3\right\}$ provides a strong baseline across most scenarios.

5 RELATED WORK

Dataset pruning algorithms aim to identify and remove less "informative" examples, minimizing the performance gap between models trained on subsets and full datasets. This is achieved through various sample importance estimation metrics that measure the amount of "information" imparted by an example during model training.

Geometry based methods reduce redundancy by leveraging spatial similarity in feature space. Popular techniques like Herding [Welling, 2009] minimize the distance between coreset and dataset centers, while K-Center Greedy [Sener and Savarese, 2018] minimizes the maximum distance to the nearest coreset sample.
Uncertainty based methods prioritize low-confidence samples using metrics like least confidence, entropy, and margin [Coleman et al., 2020]. For example, Chang et al. [2017] use predictive distribution variance for selection.
Loss based methods focus on samples contributing higher loss or gradient values. Forgetting events [Toneva et al., 2019], GraNd, and EL2N scores [Paul et al., 2021] are prominent techniques. CCS [Zheng et al., 2023], a state-of-the-art one-shot coreset selection technique maximizes data distribution coverage while utilizing stratified sampling to form the retain set. InfoBatch [Qin et al., 2024], a dynamic pruning approach, combines loss thresholds with score based sampling and uniform sampling along with gradient scaling for bias reduction.
Decision boundary based methods prioritize samples near decision boundaries. Adversarial DeepFool [Ducoffe and Precioso, 2018] measures perturbations needed to alter predictions, while CAL [Margatina et al., 2021] emphasizes predictive divergence among neighbors.
Gradient matching based methods optimize coresets to approximate full-dataset gradients. CRAIG [Mirzasoleiman et al., 2020] and GradMatch [Killamsetty et al., 2021a] minimize gradient error, with GradMatch introducing penalties to prevent over-reliance on few samples.
Bilevel optimization based methods frame sample selection as an optimization problem. Retrieve [Killamsetty et al., 2021c] applies this to semi-supervised learning, while Glister [Killamsetty et al., 2021b] introduces robustness by adding a validation set on the outer optimization and the log-likelihood in the bilevel optimization.
Training dynamics incorporating methods track importance over epochs. Dyn-Unc [He et al., 2024] averages prediction uncertainty across epochs, and TDDS [Zhang et al., 2024] aligns gradients over the full training run.
Submodularity based methods maximize diversity and informativeness through submodular functions like Facility Location and Log Determinant [Iyer et al., 2021]. Prism [Kaushal et al., 2021] targets labeling efficiency in large datasets.
Proxy based methods train a proxy model (a smaller, shallower version of the original model) on the entire training dataset to determine the importance of each sample [Coleman et al., 2020, Sachdeva et al., 2021].
Random sampling based methods perform uniform sam-

pling and are tough-to-beat baselines [Ayed and Hayou, 2023]. RS2 [Okanovic et al., 2024b] employs dynamic uniform sampling (with and without replacement), while MetriQ [Vysogorets et al., 2024] adjusts sample fractions by class.

6 CONCLUSION

We present RCAP, a novel, Robust, Class-Aware, Probabilistic dynamic dataset pruning algorithm tailored for classification tasks. In every epoch, RCAP applies a closed-form solution to estimate the fraction of samples that need to be included in the training subset for each individual class. Thereafter, RCAP employs a novel, adaptive sampling strategy that prioritizes samples having a higher loss for populating the class-wise subset. Our method incurs no computational overhead, achieving an impressive $8.69\times$ speed-up on average across multiple datasets while maintaining $< 1\%$ drop in performance with respect to full data training. Extensive evaluation on six datasets, ranging from class-balanced to highly imbalanced, across four pruning rates and three distinct training paradigms, shows that RCAP significantly improves worst-group accuracy while maintaining competitive average-group accuracy compared to seven state-of-the-art pruning methods.

Limitations and Future Scope: RCAP requires a few training epochs to accurately approximate the optimal class-wise fractions, which could hinder its utility in scenarios requiring immediate effectiveness. For instance, LLMs are few-shot learners [Brown et al., 2020] but are computationally expensive to train due to their reliance on massive datasets. Therefore, we aim to refine RCAP to reduce approximation error early in training, enhancing its applicability to LLMs and other large-scale models. Another important limitation is the β hyper-parameter, which currently requires manual selection. A promising direction of future work is to make β adaptive during training by utilizing the loss values or annealing schedules.

References

Fadhel Ayed and Soufiane Hayou. Data pruning and neural scaling laws: fundamental limitations of score-based algorithms. *Trans. Mach. Learn. Res.*, 2023, 2023.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Stephen Boyd. Convex optimization. *Cambridge UP*, 2004.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-20, 2022*, pages 4749–4758. IEEE, 2022. doi: 10.1109/CVPRW56347.2022.00521.

Haw-Shiuan Chang, Erik G. Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1002–1012, 2017.

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 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

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*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Melanie Ducoffe and Frédéric Precioso. Adversarial active learning for deep networks: a margin based approach. *CoRR*, abs/1802.09841, 2018.

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, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90.

Muyang He, Shuo Yang, Tiejun Huang, and Bo Zhao. Large-scale dataset pruning with dynamic uncertainty. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024 - Workshops, Seattle, WA, USA, June 17-18, 2024*, pages 7713–7722. IEEE, 2024. doi: 10.1109/CVPRW63382.2024.00767.

Rishabh K. Iyer, Ninad Khargoankar, Jeff A. Bilmes, and Himanshu Asanani. Submodular combinatorial information measures with applications in machine learning. In *Algorithmic Learning Theory, 16-19 March 2021, Virtual Conference, Worldwide*, volume 132 of *Proceedings of Machine Learning Research*, pages 722–754. PMLR, 2021.

Vishal Kaushal, Suraj Kothawade, Ganesh Ramakrishnan, Jeff A. Bilmes, and Rishabh K. Iyer. PRISM: A unified framework of parameterized submodular information measures for targeted data subset selection and summarization. *CoRR*, abs/2103.00128, 2021.

KrishnaTeja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, Abir De, and Rishabh K. Iyer. GRAD-MATCH: gradient matching based data subset selection for efficient deep model training. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5464–5474. PMLR, 2021a.

KrishnaTeja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh K. Iyer. GLISTER: generalization based data subset selection for efficient and robust learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 8110–8118. AAAI Press, 2021b. doi: 10.1609/AAAI.V35I9.16988.

KrishnaTeja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh K. Iyer. RETRIEVE: coreset selection for efficient and robust semi-supervised learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 14488–14501, 2021c.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yanyu Li, Ju Hu, Yang Wen, Georgios Evangelidis, Kamyar Salahi, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Rethinking vision transformers for mobilenet size and speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16889–16900, 2023.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 3730–3738. IEEE Computer Society, 2015. doi: 10.1109/ICCV.2015.425.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2022.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 650–663. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.51.

Sören Mindermann, Jan Markus Brauner, Muhammed 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 *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 15630–15649. PMLR, 2022.

Baharan Mirzasoleiman, Jeff A. Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6950–6960. PMLR, 2020.

Samuel G. Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 754–762. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00081.

Patrik Okanovic, Roger Waleffe, Vasilis Mageirakos, Konstantinos Nikolakakis, Amin Karbasi, Dionysios Kalogerias, Nezihe Merve Gürel, and Theodoros Rekatsinas. Repeated random sampling for minimizing the time-to-accuracy of learning. In *The Twelfth International Conference on Learning Representations*, 2024a.

Patrik Okanovic, Roger Waleffe, Vasilis Mageirakos, Konstantinos E. Nikolakakis, Amin Karbasi, Dionysios S. Kalogerias, Nezihe Merve Gürel, and Theodoros Rekatsinas. Repeated random sampling for minimizing the time-to-accuracy of learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024b.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in neural information processing systems*, 34:20596–20607, 2021.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR, 2023.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross B. Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10425–10433. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01044.

Ravi S Raju, Kyle Daruwalla, and Mikko Lipasti. Accelerating deep learning with dynamic data pruning. *arXiv preprint arXiv:2111.12621*, 2021.

Noveen Sachdeva, Carole-Jean Wu, and Julian McAuley. Svp-cf: Selection via proxy for collaborative filtering data. *arXiv preprint arXiv:2107.04984*, 2021.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *CoRR*, abs/1911.08731, 2019.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019.

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 *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Artem Vysogorets, Kartik Ahuja, and Julia Kempe. Robust data pruning: Uncovering and overcoming implicit bias. *CoRR*, abs/2404.05579, 2024. doi: 10.48550/ARXIV.2404.05579.

Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 1121–1128. ACM, 2009. doi: 10.1145/1553374.1553517.

Xiaobo Xia, Jiale Liu, Shaokun Zhang, Qingyun Wu, Hongxin Wei, and Tongliang Liu. Refined coreset selection: Towards minimal coreset size under model performance constraints. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

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 *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

Xin Zhang, Jiawei Du, Yunsong Li, Weiying Xie, and Joey Tianyi Zhou. Spanning training progress: Temporal dual-depth scoring (TDDS) for enhanced dataset pruning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 26213–26222. IEEE, 2024. doi: 10.1109/CVPR52733.2024.02477.

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*, pages 6503–6512. IEEE, 2023. doi: 10.1109/WACV56688.2023.00645.

Haizhong Zheng, Rui Liu, Fan Lai, and Atul Prakash. Coverage-centric coreset selection for high pruning rates. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

RCAP: Robust, Class-Aware, Probabilistic Dynamic Dataset Pruning (Supplementary Material)

Atif Hassan[1] **Swanand Khare**[2] **Jiaul H. Paik**[1]

[1]Department of Artificial Intelligence, IIT Kharagpur, Kharagpur, West Bengal, India
[2]Department of Mathematics, IIT Kharagpur, Kharagpur, West Bengal, India

A ADDITIONAL SIMULATION RESULTS

To further support our argument, we train a small feed-forward neural network using the Adam optimizer on a toy, three-class classification dataset consisting of 90 examples and provide a plot of the loss and gradient norm of 10 randomly selected examples over a 100 epoch training run. The figure demonstrates that there indeed is a monotonic relation.

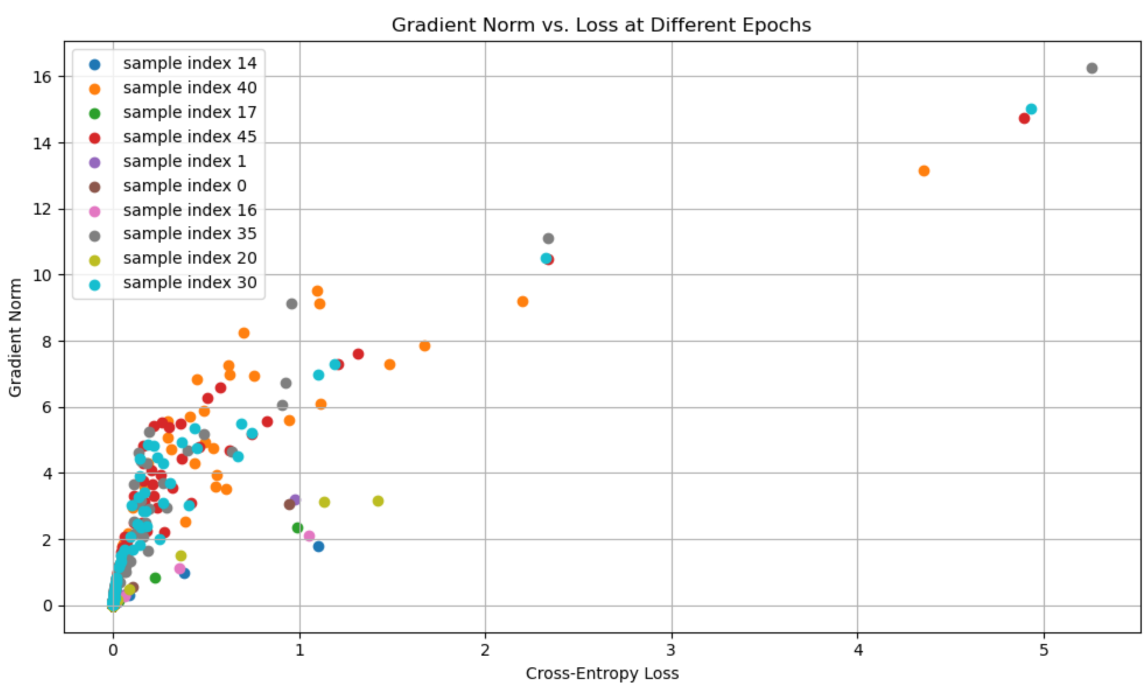


Figure 3: Visualizing the relationship between cross-entropy loss against gradient norm.

B APPROXIMATION ERROR

Approximating $\widehat{\alpha}_j^{t+1}$ with $\widetilde{E}_j^t$ incurs some approximation error. In this section, we derive an upper bound on this error. Specifically,

$$\begin{aligned}
\left|\widetilde{E}_j^t - \widetilde{E}_j^{t+1}\right| &= \left|\sum_{X_i \in \mathcal{S}_j^t} L\left(\widetilde{f}_{\theta^{t-1}}(X_i)\right) - \sum_{X_i' \in \mathcal{S}_j^{t+1}} L\left(\widetilde{f}_{\theta^t}(X_i')\right)\right| \\
&= \left|\sum_{X_i \in \mathcal{S}_j^t} L\left(\widetilde{f}_{\theta^{t-1}}(X_i)\right) - \sum_{X_i \in \mathcal{S}_j^t} L\left(\widetilde{f}_{\theta^t}(X_i)\right) + \sum_{X_i \in \mathcal{S}_j^t} L\left(\widetilde{f}_{\theta^t}(X_i)\right) - \sum_{X_i' \in \mathcal{S}_j^{t+1}} L\left(\widetilde{f}_{\theta^t}(X_i')\right)\right| \qquad (13) \\
&\le \left|\sum_{X_i \in \mathcal{S}_j^t} L\left(\widetilde{f}_{\theta^{t-1}}(X_i)\right) - \sum_{X_i \in \mathcal{S}_j^t} L\left(\widetilde{f}_{\theta^t}(X_i)\right)\right| + \left|\sum_{X_i \in \mathcal{S}_j^t} L\left(\widetilde{f}_{\theta^t}(X_i)\right) - \sum_{X_i' \in \mathcal{S}_j^{t+1}} L\left(\widetilde{f}_{\theta^t}(X_i')\right)\right|
\end{aligned}$$

We now derive separate upper bounds for both terms in Eqn. 13. Assuming that L is Lipschitz continuous having a Lipschitz constant K_1 with respect to the change in parameters θ, we get:

$$\left|\sum_{X_i \in \mathcal{S}_j^t} L\left(\widetilde{f}_{\theta^{t-1}}(X_i)\right) - \sum_{X_i \in \mathcal{S}_j^t} L\left(\widetilde{f}_{\theta^t}(X_i)\right)\right| \le K_1 \left\|\theta^{t-1} - \theta^t\right\|_2$$

Using the gradient-descent update rule we get:

$$\left|\sum_{X_i \in \mathcal{S}_j^t} L\left(\widetilde{f}_{\theta^{t-1}}(X_i)\right) - \sum_{X_i \in \mathcal{S}_j^t} L\left(\widetilde{f}_{\theta^t}(X_i)\right)\right| \le \frac{\eta K_1}{(1-r)n} \left\|\sum_{X_i \in \mathcal{S}^t} \nabla_{\theta^{t-1}} L\left(\widetilde{f}_{\theta^{t-1}}(X_i)\right)\right\|_2 \qquad (14)$$

We now derive the upper bound for the second term in Eqn. 13. Without loss of generality, we assume that $\left|\mathcal{S}_j^{t+1}\right| > \left|\mathcal{S}_j^t\right|$

$$\begin{aligned}
\left|\sum_{X_i \in \mathcal{S}_j^t} L\left(\widetilde{f}_{\theta^t}(X_i)\right) - \sum_{X_i' \in \mathcal{S}_j^{t+1}} L\left(\widetilde{f}_{\theta^t}(X_i')\right)\right| &\le \left|\sum_{i=1}^{\left|\mathcal{S}_j^t\right|} L\left(\widetilde{f}_{\theta^t}(X_i)\right) - L\left(\widetilde{f}_{\theta^t}(X_i')\right)\right| + \sum_{i=\left|\mathcal{S}_j^t\right|+1}^{\left|\mathcal{S}_j^{t+1}\right|} L\left(\widetilde{f}_{\theta^t}(X_i')\right) \\
&\le \left|\mathcal{S}_j^t\right| \left|L\left(\widetilde{f}_{\theta^t}(X)\right) - L\left(\widetilde{f}_{\theta^t}(X')\right)\right| + \sum_{i=\left|\mathcal{S}_j^t\right|+1}^{\left|\mathcal{S}_j^{t+1}\right|} L\left(\widetilde{f}_{\theta^t}(X_i')\right)
\end{aligned}$$

where inputs X and X' are selected such that, $\forall i \in \{1, 2, \cdots \left|\mathcal{S}_j^t\right|\}, L\left(\widetilde{f}_{\theta^t}(X_i)\right) - L\left(\widetilde{f}_{\theta^t}(X_i')\right) \le L\left(\widetilde{f}_{\theta^t}(X)\right) - L\left(\widetilde{f}_{\theta^t}(X')\right)$. Again, assuming that L is Lipschitz continuous having a Lipschitz constant K_2 with respect to the change in input, we get:

$$\left|\sum_{X_i \in \mathcal{S}_j^t} L\left(\widetilde{f}_{\theta^t}(X_i)\right) - \sum_{X_i' \in \mathcal{S}_j^{t+1}} L\left(\widetilde{f}_{\theta^t}(X_i')\right)\right| \le \left|\mathcal{S}_j^t\right| K_2 \left\|\widetilde{f}_{\theta^t}(X) - \widetilde{f}_{\theta^t}(X')\right\|_2 + \sum_{i=\left|\mathcal{S}_j^t\right|+1}^{\left|\mathcal{S}_j^{t+1}\right|} L\left(\widetilde{f}_{\theta^t}(X_i')\right) \qquad (15)$$

Finally, applying Eqns. 14 and 15 in Eqn. 13, we get:

$$\left|\widetilde{E}_j^t - \widetilde{E}_j^{t+1}\right| \le \frac{\eta K_1}{(1-r)n} \left\|\sum_{X_i \in \mathcal{S}^t} \nabla_{\theta^{t-1}} L(\widetilde{f}_{\theta^{t-1}}(X_i))\right\|_2 + \left|\mathcal{S}_j^t\right| K_2 \left\|X - X'\right\|_2 + \sum_{i=\left|\mathcal{S}_j^t\right|+1}^{\left|\mathcal{S}_j^{t+1}\right|} L\left(\widetilde{f}_{\theta^t}(X_i')\right) \qquad (16)$$

C TRAINING DETAILS

We run all our tasks on a single NVIDIA A100 GPU in combination with an Intel Xeon processor. We use the Pytorch Lightning library to implement all methods. Each reported result is averaged over three different runs usings seeds, $0, 27, 100$. Apart from standard image augmentations, we also employ TrivialAugmentWide Müller and Hutter [2021]. In all our experiments, we use the CosineAnnealing Scheduler Loshchilov and Hutter [2022].

Table 3: All the training details required to reproduce our results.

Dataset	Model	Augmentations	Optimizer	LR	Weight Decay	Batch Size	Epochs
CIFAR10	ResNet18	RandomCrop RandomHorizontalFLip	SGD momentum= 0.9	0.1	$5e^{-4}$	128	200
CIFAR100	ResNet18	RandomCrop RandomHorizontalFLip	SGD momentum= 0.9	0.1	$5e^{-4}$	128	200
ImageNet	Frozen dinov2_vitb14_reg with two linear layers $2304 \rightarrow 512 \rightarrow 1000$	Resize CenterCrop TrivialAugmentWide	AdamW	0.001	–	256	10
Waterbirds	pretrained efficientnet_b3	Resize RandomCrop RandomHorizontalFlip TrivialAugmentWide	AdamW	0.00004	$5e^{-4}$	32	300
CelebA	EfficientFormerV2	CenterCrop RandomHorizontalFlip TrivialAugmentWide	AdamW	0.001	$5e^{-4}$	256	5
iNaturalist	pretrained ResNet50	Resize CenterCrop RandomHorizontalFlip TrivialAugmentWide	AdamW	0.001	$5e^{-4}$	256	5

Table 4: β values used across all datasets and pruning rates.

Dataset	50%	70%	80%	90%
CIFAR10	3	1	2	1
CIFAR100	3	2	2	2
ImageNet	$\frac{1}{3}$	4	2	2
Waterbirds	3	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
CelebA	2	2	3	1
iNaturalist	2	3	3	$\frac{1}{3}$