ORDERDP: A THEORETICALLY GUARANTEED LOSSLESS DYNAMIC DATA PRUNING FRAMEWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Data pruning (DP), as an oft-stated strategy to alleviate heavy training burdens, reduces the volume of training samples according to a well-defined pruning method while striving for near-lossless performance. However, existing approaches, which commonly select highly informative samples, can lead to biased gradient estimation compared to full-dataset training. Furthermore, the analysis of this bias and its impact on final performance remains ambiguous. To address these challenges, we propose **`OrderDP`**, a plug-and-play framework that aims to obtain stable, unbiased, and near-lossless training acceleration with theoretical guarantees. Specifically, **`OrderDP`** first randomly selects a subset and then chooses the top-q samples, where unbiasedness is established with respect to a surrogate loss. This ensures that **`OrderDP`** conducts unbiased training in terms of the surrogate objective. We further establish convergence and generalization analyses, elucidating how **`OrderDP`** affects optimal performance and enables well-controlled acceleration while ensuring guaranteed final performance. Empirically, we evaluate **`OrderDP`** against comprehensive baselines on CIFAR-10, CIFAR-100, and ImageNet-1K, demonstrating competitive accuracy, stable convergence, and exact control—all with a simpler design and faster runtime, while reducing training cost by over 40%. Delivering both strong performance and computational efficiency, our method serves as a robust and easily adaptable tool for data-efficient learning.

1 INTRODUCTION

Neural scaling laws have revealed a consistent empirical pattern across a wide range of domains (Amari et al., 1992; Hestness et al., 2017; Kaplan et al., 2020): model performance tends to improve predictably, often as a power law (Hernandez et al., 2021; Cherti et al., 2023; Chen et al., 2023), with increased model size and the data volume. This observation has fueled a surge in computational demands and financial costs, as larger models and datasets are leveraged to push the boundary of model capabilities. In this context, data pruning (DP) has emerged as a promising strategy to alleviate training costs by selectively removing less informative samples (Killamsetty et al., 2021b; Mirzasoleiman et al., 2020; Qin et al., 2024; Raju et al., 2021), offering a pathway to enhance training efficiency without compromising model performance.

Depending on when sample selection is performed, data pruning strategies can be broadly classified into ***static pruning*** and ***dynamic pruning***. ① Static pruning assigns an informativeness score to each training sample *before* training, typically using data influence functions (Borsos et al., 2020; Koh & Liang, 2017; Yang et al., 2022) or coreset selection strategies (Huggins et al., 2016; Campbell & Broderick, 2019; Kim et al., 2023). ② Dynamic pruning, on the other hand, performs sample selection *during* training, updating scores on-the-fly based on evolving model states or gradients (Raju et al., 2021; Qin et al., 2024; Chen et al., 2024). By continuously adapting to the training dynamics, it can better identify and retain the most influential samples at each stage, potentially yielding higher performance under constrained training budgets. A more comprehensive survey of static and dynamic pruning methods is included in Appendix B.

Within supervised learning, data pruning aims to reduce data volume without sacrificing performance, thereby achieving near-lossless[1] pruning. However, the discarded data can cause the distribution

[1]Here, near-lossless means matching full-data accuracy up to normal stochastic fluctuations (typically within 0.1%) while achieving a noticeable training speedup.

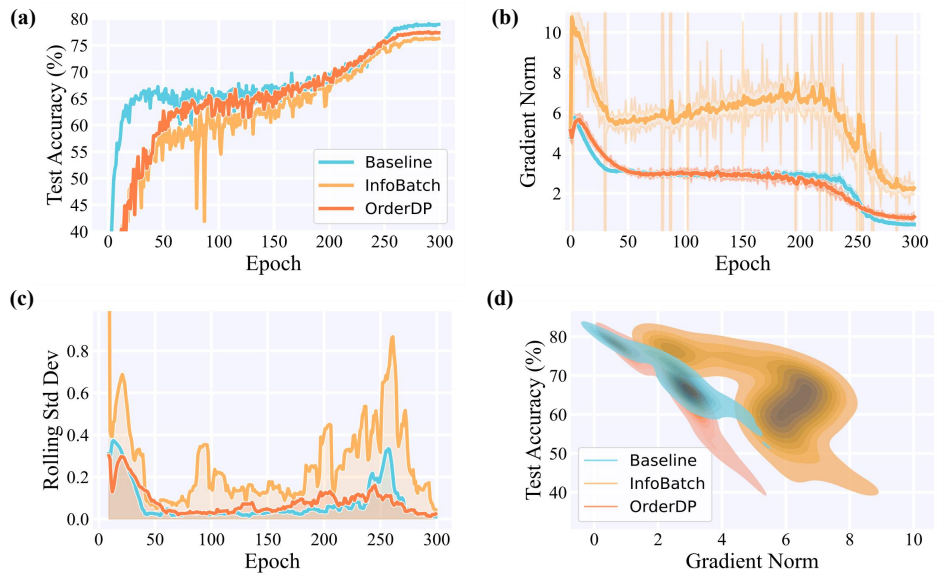


Figure 1: **Training dynamics of ResNet-18 on the CIFAR-100 under a 70% data pruning ratio.** Method comparison: full-dataset training (Baseline), representative dynamic pruning strategy (InfoBatch), and our proposed method (OrderDP). **(a-c)** Test accuracy, gradient norm (shadow area denotes standard deviation), and temporal stability of gradient norm over training epochs. **(d)** Joint distribution of test accuracy and gradient norm throughout training.

shift and bias gradient. Although selecting a portion of the discarded data randomly via calibration protocols (Ayed & Hayou, 2023) can theoretically ensure unbiasedness, finding the optimal proportion can be difficult in practice. Inspired by this method, recent dynamic methods such as InfoBatch (Qin et al., 2024) achieve this goal by rescaling the bias gradient toward the expected loss. However, when training on a specific dataset, gradient bias in both scale and direction may still arise from the discrepancy between empirical and expected loss. This bias is further amplified under extreme pruning, where large scaling factors are applied and stabilization techniques such as annealing are often required. These challenges reveal an incomplete understanding of the principles underlying "near-lossless" pruning. A critical questions arise as to *what ensures this property? how the bias should be analyzed, and whether pruning can be pushed further toward more extreme regimes?* To investigate these questions, we conduct comparative studies and report a representative result on CIFAR-100, comparing full-dataset training with InfoBatch (Qin et al., 2024) under a 70% pruning ratio to test its limits. Further experiments are presented in Appendix D, E, and we highlight several key observations as follows:

❶ *Gradient norm serves as a reliable proxy for model performance*: Under full-dataset training, test accuracy exhibits a strong linear correlation with gradient norm (Pearson's $R = -0.93$), as shown in Figure 1 (d), which suggest the magnitude of gradients is a stable and informative indicator of both training progress and generalization, which echoes prior observations in related studies (Zhao et al., 2022; Zhang et al., 2023).

❷ *Dynamic data pruning suffers from training instability*: Compared to full-dataset training, dynamic one displays pronounced fluctuations in test accuracy and volatile gradient norms. Moreover, the rolling standard deviation reveals irregular and noisy optimization dynamics, indicating reduced training stability, as shown in Figure 1 (a-c).

❸ *Gradient estimation under dynamic pruning is still biased*: Figure 1 (b,d) demonstrate that the dynamic method induces a noticeable shift in the overall scale of gradient norms relative to the baseline. This shift is more significant for InfoBatch, as a large scaling factor is imposed. The previously observed linear relationship between accuracy and gradient norm weakens, suggesting that it still distorts gradient estimates and introduces bias during training.

Together, these findings highlight that training instability and biased gradient estimation are two key limitations of existing dynamic DP strategies. We provide an extended empirical analysis of

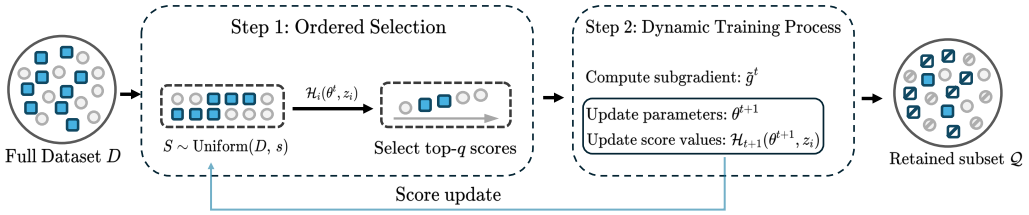


Figure 2: Illustration of the proposed **OrderDP** framework: at each iteration, a candidate batch is sampled uniformly, the top-q examples are selected by score to compute a subgradient and update model parameters, and scores are refreshed only for the retained samples.

gradient bias in Appendix E. To this end, we take a step towards designing a DP method to mitigate both issues, thus achieving **stable, unbiased, and near-lossless** data pruning. Inspired by the recent stochastic optimization based on ordering statistics (Kawaguchi & Lu, 2020; Mehta et al., 2023), we propose a simple yet effective DP framework, **OrderDP**, which aims to obtain near-lossless pruning with improved efficiency, even at large pruning ratios. At the beginning of each epoch, we randomly sample a batch of data points from the full dataset to form a pruning candidate pool. These candidates are then ranked in descending order based on their loss values, and the Top-q samples are selected as the most informative ones for model training. We formulate **OrderDP** as an optimization algorithm that minimizes a proposed surrogate loss. We theoretically establish convergence analyses using an unbiased gradient estimation of the surrogate loss. Furthermore, generalization analysis is provided in terms of the surrogate loss and expected loss, demonstrating the effectiveness of **OrderDP**.

Our approach has several desirable properties. First, **OrderDP** ensures unbiased gradient estimation and works with standard training pipelines without architectural changes or auxiliary approximations. It maintains an exactly controlled pruning ratio with rigorous theoretical guarantees on convergence and generalization. The surrogate loss fully captures the bias and enables principled, loss-aware pruning while sustaining strong stability. Empirically, we validate **OrderDP** on CIFAR-10 (Krizhevsky et al., a), CIFAR-100 (Krizhevsky et al., b), and ImageNet-1K (Deng et al., 2009). Across all benchmarks, **OrderDP** achieves near-lossless performance at moderate pruning ratios and surpasses state-of-the-art methods. On ImageNet-1K, it retains full accuracy at 40% pruning with the lowest total computation, leading to faster runtime. These results show that **OrderDP** not only sustains strong performance but also delivers superior efficiency, robustness, and a simple plug-and-play design, making it a practical solution for scalable deep learning.

2 METHOD

Inspired by the iterative update process of stochastic gradient descent (SGD) (Kawaguchi & Lu, 2020; Amari, 1993), we propose Ordered Data Pruning (**OrderDP**), a dynamic strategy that integrates adaptive sample selection into the SGD pipeline to reduce training cost without sacrificing accuracy. The overall framework is illustrated in Figure 2. **OrderDP** leverages a score-value mechanism to rank and retain the most informative samples at each iteration.

2.1 PRELIMINARIES

We begin with the standard empirical risk minimization formulation over a dataset $\mathcal{D} = \{z_i\}_{i=1}^n$:

$$\mathcal{L}(\theta) := \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_i(\theta, z_i) \tag{1}$$

where $\theta \in \mathbb{R}^d$ denotes the model parameters and each per-sample loss $\mathcal{L}_i(\theta, z_i)$: $\mathbb{R}^d \to \mathbb{R}_{\geq 0}$ measures the discrepancy on example z_i. Solving this ERM via (mini-batch) stochastic gradient descent is at the core of most modern machine learning tasks.

Score Value Function. To drive dynamic pruning, we associate each sample z_i with a nonnegative *score value* $\mathcal{H}_i(\theta) = \mathcal{H}_i(\theta, z_i)$, which quantifies its importance. In general, $\mathcal{H}_i$ can be any function of model state and data (e.g., gradient norm or influence measure), but we adopt the instantaneous

Algorithm 1: Dynamic Training Process with **OrderDP**

Input: Initial parameters θ^1, initial scores $\mathcal{H}_1(\theta^1, z_i)$ for all $i \in [n]$, learning rates $\{\eta^t\} > 0$, exploration size s, exploitation size q.
Output: Final parameters θ^T.

for $t = 1, 2, \ldots, T$ **do**
- // Ordered Data Pruning (**OrderDP**)
- Sample candidate batch $S^t \subseteq D$ uniformly at random, with $|S^t| = s$.
- Select subset $\mathcal{Q}^t \subseteq S^t$ of top-q scores:
  $$\mathcal{Q}^t \in \arg\max_{\substack{Q \subseteq S^t \\ |Q|=q}} \sum_{i \in Q} \mathcal{H}_t(\theta^t, z_i).$$
- // Compute a subgradient
- $\tilde{g}^t \in \partial L_{\mathcal{Q}^t}(\theta^t)$, where $L_{\mathcal{Q}^t}(\theta^t) = \frac{1}{q} \sum_{i \in \mathcal{Q}^t} \mathcal{L}_i(\theta^t, z_i)$.
- // Update model parameters
- $\theta^{t+1} \leftarrow \theta^t - \eta^t \tilde{g}^t$.
- // Update score values
- $\mathcal{H}_{t+1}(\theta^{t+1}, z_i) = \begin{cases} \mathcal{L}_i(\theta^t, z_i), & i \in \mathcal{Q}^t, \\ \mathcal{H}_t(\theta^t, z_i), & \text{otherwise.} \end{cases}$

loss $\mathcal{H}_i(\theta) = \mathcal{L}_i(\theta, z_i)$ as a simple, adaptive proxy: higher loss indicates greater need for retention. By updating scores only for samples that remain active, we avoid full-dataset recomputation each step. Concretely, if $\mathcal{Q}_t \subseteq \mathcal{D}$ denotes the set of retained examples at epoch t, then

$$\mathcal{H}_{t+1}(\theta^{t+1}, z_i) = \begin{cases} \mathcal{L}_i(\theta^{t+1}, z_i), & i \in \mathcal{Q}_t, \\ \mathcal{H}_t(\theta^t, z_i), & i \notin \mathcal{Q}_t. \end{cases} \tag{2}$$

This rule ensures that only the losses of selected samples are refreshed, while others retain their previous scores. Together, the ERM objective and the score value function enable dynamic data pruning: by ranking samples via $\mathcal{H}_i(\theta)$, we focus computation on the most informative subset each iteration, reducing training cost without hurting accuracy.

2.2 The OrderDP framework

Building on the ERM and score-value preliminaries, Ordered Data Pruning (**OrderDP**) integrates dynamic sample selection into the SGD loop. At each iteration, a candidate batch is uniformly sampled, the top-q samples are selected by score values, and a subgradient computed on this subset is used for parameter update. Scores are refreshed only for the retained samples, while others remain unchanged. The complete procedure is given in Algorithm 1.

OrderDP combines uniform sampling with score-based ranking to preserve diversity while focusing on informative samples, and enforces an exact prune ratio of $1 - (q/s) \cdot (s/|\mathcal{D}|)$ for predictable speed-ups. Uniform sampling ensures every sample has a non-zero chance of being selected, improving robustness (see Part 4.2, (Shah et al., 2020)) and reducing dependence on sorting. The sorting step can be reduced to $O(\log q)$ time per sample (even $O(1)$ when $q = 1$), with constant memory overhead, unlike other dynamic methods such as UCB (Raju et al., 2021) and InfoBatch (Qin et al., 2024), which require either $O(\log n)$ time or $O(n)$ storage.

3 Theoretical Analysis

In this section, we show that **OrderDP** provides unbiased gradient estimates for a surrogate loss and achieves standard convex–Lipschitz convergence rates, then establish its generalization error bound via a spectral-risk analysis. Full proofs are provided in Appendix A.

3.1 Bias and Convergence Analysis

In this subsection, we analyze the convergence of **OrderDP** by first capturing the bias introduced by selective pruning. Specifically, we define a surrogate loss that yields unbiased gradient updates:

$$\mathcal{L}_q(\theta) := \frac{1}{q}\sum_{j=1}^{n} \gamma_j \, \mathcal{L}_{(j)}(\theta), \quad \text{and where } \gamma_j = \sum_{l=\max\{1,s-n+j\}}^{\min\{q,j\}} \frac{\binom{j-1}{l-1}\binom{n-j}{s-l}}{\binom{n}{s}}, \tag{3}$$

where $\mathcal{L}_{(j)}(\cdot)$ is the j-th rank of per-sample loss and each weight γ_j depends only on (n, s, q). This construction guarantees that the gradient estimator $\tilde{g}^t$ produced by **OrderDP** is unbiased with respect to $\mathcal{L}_q$.

Theorem 1. *Under the definitions above, the update $\tilde{g}^t$ in Algorithm 1 satisfies*

$$\mathbb{E}[\tilde{g}^t] \notin \partial\mathcal{L}(\theta^t) \quad \textit{but} \quad \mathbb{E}[\tilde{g}^t] \in \partial\mathcal{L}_q(\theta^t), \tag{4}$$

i.e., it is an unbiased estimator of a (sub-)gradient of $\mathcal{L}_q$.

Theorem 1 shows that the biased gradient estimation of **OrderDP** w.r.t. empirical loss $\mathcal{L}$ can be interpreted as an unbiased method for minimizing the surrogate objective $\mathcal{L}_q$. $\mathcal{L}_q$ is well-defined for any θ, and **OrderDP** enjoys three key advantages: ① By choosing s and q, the prune ratio $1 - \frac{q}{n}$ is easily adjusted; ② Computing γ_j adds no per-epoch cost, unlike other dynamic methods requiring $O(n)$ time and memory for weight tables; ③ **OrderDP** preserves unbiased, lower-variance gradient estimates for $\mathcal{L}_q$—eliminating the need for annealing. See Appendix A.1 for the complete proof.

Another view of Theorem 1 is that the parameters (n, s, q) shape the surrogate loss $\mathcal{L}_q(\theta)$ via the weights $\{\gamma_j\}$, which inherently represent the selective pruning. Inspired by the asymptotic approximation in (Kawaguchi & Lu, 2020), we obtain:

Proposition 2. *Denote $z = j/n$ and $\gamma(z) = \sum_{l=1}^{q} z^{\,l-1}(1-z)^{\,s-l}\,\frac{s!}{(l-1)!\,(s-l)!}$. Then, as $j, n \to \infty$ it holds that*

$$\lim_{j,n\to\infty,\; j/n=z} n\,\gamma_j \;=\; \gamma(z). \tag{5}$$

Furthermore, $1 - \frac{n}{s}\,\gamma(z)$ is the cumulative distribution of $\mathrm{Beta}(z; s-q)$.

The weight sequence $\{\gamma_j/q\}_{j=1}^{n}$ generated by **OrderDP** forms a non-uniform probability distribution (Kawaguchi & Lu, 2020; Mehta et al., 2023; Shah et al., 2020), which can be easily verified through a numerical simulation showing that $\sum_{j=1}^{n} \gamma_j/q = 1$ for given (n, s, q). A non-trivial proof is also provided. For the structure of γ_j itself, Fig 5 shows that γ_j monotonically decays. If we fix (s, q), the cliff becomes smoother and closer to $r(z)$ as j, n increase. Similar observations are also found in (Kawaguchi & Lu, 2020), but we make a more general formulation of γ_j. More discussions, proof and empirical validation are deferred to Appendix A.2 and Appendix D.2.

Building on the unbiased gradient estimates of **OrderDP**, we leverage the classic mini-batch SGD analysis to obtain the following guarantee.

Theorem 3. *Let $(\theta^t)_{t=0}^{T}$ be the sequence generated by Algorithm 1. Suppose there exists a finite $\theta^* \in \arg\min_\theta \mathcal{L}_q(\theta)$, $\mathcal{L}_q(\theta^*) < \infty$. If each $\mathcal{L}_i(\cdot)$ is convex and G-Lipschitz, then*

$$\min_{0\le t\le T} \mathbb{E}\big[\mathcal{L}_q(\theta^t) - \mathcal{L}_q(\theta^*)\big] \le \frac{\eta_{\max}\big(\|\theta^1 - \theta^*\|_2^2 + G^2\sum_{t=1}^{T}(\eta^t)^2\big)}{2\,\eta_{\min}\sum_{t=1}^{T}\eta^t}. \tag{6}$$

This matches the standard $O(1/\sqrt{T})$ convergence rate of mini-batch SGD under the same convexity and Lipschitz assumptions, demonstrating that **OrderDP** attains identical theoretical guarantees despite pruning. In particular, choosing $\eta^t = \|\theta^1 - \theta^*\|_2/\big(G\sqrt{T}\big)$ yields the error bound $(G\|\theta^1 - \theta^*\|_2)/\sqrt{T}$. By using the averaged iterate $\hat{\theta}^T = (1/\sum_{t=1}^{T})\sum_{t=1}^{T}\eta^t\theta^t$, the dependence on $\eta_{\max}$ and $\eta_{\min}$ can be removed, that is $\mathbb{E}[L_q(\bar{\theta}^T) - L_q(\theta^*)] \le (\|\theta^1 - \theta^*\|_2^2 + G^2\sum_{t=1}^{T}(\eta^t)^2)/(2\sum_{t=1}^{T}\eta^t)$. The full proof is provided in Appendix A.3. Empirical evidence supporting the convergence assumptions is provided in Appendix D.3.

Thus, despite pruning a large fraction of data each epoch, **OrderDP** does not slow optimization in expectation, ensuring computational savings without loss in convergence speed.

3.2 Generalization Analysis

Having established convergence for the surrogate loss $\mathcal{L}_q(\theta)$, we now quantify its approximation to the expected risk $\mathcal{L}(\theta^*) = \mathbb{E}_{z\sim\mathcal{D}}[\mathcal{L}(\theta^*, z)]$. Pruning creates a non-uniform sampling bias. Motivated by the 1-Wasserstein distance (Mehta et al., 2023), we rewrite $\mathcal{L}_q(\theta) = \sum_{j=1}^{n} \frac{\gamma_j}{q} \mathcal{L}_{(j)}(\theta)$ and $\mathcal{L}(\theta) = \sum_{j=1}^{n} \frac{1}{n} \mathcal{L}_{(j)}(\theta)$, thereby revealing the bias from the gap between $\{\gamma_j/q\}$ and uniform weights $\{1/n\}$. Noting that $\mathbb{E}[\mathcal{L}_q(\theta, D)] = \mathbb{E}[\sum_{j=1}^{s}(\hat{\gamma}_j/q)\,\mathcal{L}_{i_{(j)}}(\theta)]$ with $\hat{\gamma}_j = (n/s)\gamma_j$, we decompose the generalization gap $\mathbb{E}[\mathcal{L}_q(\theta, D)] - \mathcal{L}(\theta^*)$ into a bias term $\mathbb{E}[\mathcal{L}_q(\theta, D)] - \mathbb{E}[\mathcal{L}(\theta, D)]$ and a sampling error $\mathbb{E}[\mathcal{L}(\theta, D)] - \mathcal{L}(\theta^*)$, where the expectation is over the random minibatch $\{i_1, \ldots, i_s\}$.

Theorem 4. *(Generalization error bound). Under the same assumption of Theorem 3, the following satisfies for any θ^t in the sequence $\Theta = \{\theta\}_{t=1}^{T}$ generated by* **OrderDP**:

$$\mathcal{L}(\theta^*) - \mathbb{E}[\mathcal{L}_q(\theta^t, D)] \le \underbrace{\sqrt{2}C_s B\sqrt{\frac{n-s}{s(n-1)}} - \mathcal{Q}_n(\theta^t; s, q)}_{\text{bias term}} + \underbrace{\frac{\eta_{max}(\|\theta^1 - \theta^*\|_2^2 + G^2\sum_{t=1}^{T}(\eta^t)^2)}{2\eta_{min}\sum_{t=1}^{T}\eta^t}}_{\text{unbiased term}},$$

where $C_s = \sup_{t\in(0,1)} |s(t) - u(t)|$[2], $B = \inf_{\theta\in\Theta}\max_{i\in[1,n]}|\mathcal{L}_i(\theta, D_i)| < \infty$, *and* $\mathcal{Q}_n(\theta; s, q) := \inf_{\theta\in\Theta}\sum_{i=1}^{n}(\frac{r_i(\theta,D)}{q} - \frac{1}{n})\mathcal{L}_i(\theta, D_i)$. *The expectation is over the random batch sampling.*

The bias term bounds the bias from selective pruning of **OrderDP**; the unbiased term is the standard optimization error, which vanishes as $T \to \infty$ with suitable η^t. The dependence on $\eta_{\max}$ and $\eta_{\min}$ can be removed by using the averaged iterate; see proof in Appendix A.4. In contrast, the value C_s and $\mathcal{Q}_n(\theta; s, q)$ remain finite and quantify the deviation of $\{\gamma_j\}$ from uniformity (Mehta et al., 2023), as confirmed by simulations in Figure 6. As $q \to s$, $\big(r_i(\theta, D)/q - 1/n\big) \to 0$, so $\mathcal{Q}_n(\theta; s, q) \to 0$; and as $s \to n$, $\sqrt{2}C_s B\sqrt{\frac{n-s}{s(n-1)}} \to 0$, implying $\mathcal{L}(\theta^*) \le \mathbb{E}[\mathcal{L}_q(\theta^t, D)]$. Thus, by minimizing $\mathcal{L}_q(\theta^t, D)$, **OrderDP** also minimizes expected generalization error. In the special case $s = q$, it reduces to standard mini-batch SGD ($r_i(\theta, D)/q = 1/n$, $C_s = 0$) and the bias vanishes. The approximation behavior characterized in Theorem 4 is further illustrated empirically in Appendix D.2.

Theorem 4 shows that **OrderDP** 's modifies the distribution shift as the gap between the surrogate loss $\mathcal{L}_q$ and the original loss $\mathcal{L}$, and the gap is fully captured by the values C_s and $\mathcal{Q}_n$ of the biased term. For a high pruning ratio, i.e., small exploitation size q or a small exploration size s since $q \le s$, the distribution of $\frac{\gamma_j}{q}, j \in \{1, \ldots, n\}$ exhibates a large range. This leads to a significant bias compared to the uniform distribution (which has a range of 0), a substantial discrepancy between C_s and $Q_n(\theta; s, q)$, and consequently, a poor approximation. As the pruning ratio decreases (i.e., q approaches s), the range of the γ_j distribution narrows, and its shift from the uniform distribution diminishes and both C_s and $Q_n(\theta; s, q)$ decrease, thereby improving the approximation. Specifically, when $q = s$, OrderDP reduces to standard SGD. In this case, the bias term vanishes, yielding $\mathcal{L}_q = \mathcal{L}$. We visualize the distribution shift in Figure 7.

In summary, Theorem 4 demonstrates that **OrderDP** 's generalization error comprises a vanishing optimization term and a bounded pruning bias, maintaining SGD-rate convergence while controlling dynamic pruning bias, which is consistent with the observation in Figure 1 (a-c).

4 Experimental Settings

4.1 Datasets and Tasks

To comprehensively validate the effectiveness of our proposed **OrderDP**, we conduct experiments on a range of image classification benchmarks: CIFAR-10 and CIFAR-100 (Krizhevsky et al., a;b), ImageNet-1K (Deng et al., 2009).

CIFAR datasets comprise 32×32 color images across 10 and 100 categories, respectively. Each split includes 50,000 training and 10,000 test samples, providing balanced classification evaluation. ImageNet-1K, as a 1,000-class subset of ImageNet-21k, contains 1,281,167 training images and 50,000 validation images, spanning a variety of object categories.

[2] $s(t)$ and $u(t)$ refer to the probability density of the spectrum γ_j distribution and uniform distribution on $(0, 1)$. Details can be found in (Mehta et al., 2023).

Table 1: Static pruning results (accuracy, %) on `CIFAR10` and `CIFAR100` with ResNet-18. Accuracy (%, ↑). Best in **bold**. Performance gaps to full-data are in blue / orange.

Dataset	CIFAR10			CIFAR100		
Prune Ratio %	30	50	70	30	50	70
Static Random	94.6$_{\downarrow 1.0}$	93.3$_{\downarrow 2.3}$	90.2$_{\downarrow 5.4}$	73.8$_{\downarrow 4.4}$	72.1$_{\downarrow 6.1}$	69.7$_{\downarrow 8.5}$
CD (Agarwal et al., 2020)	95.0$_{\downarrow 0.6}$	94.3$_{\downarrow 1.3}$	90.8$_{\downarrow 4.8}$	74.2$_{\downarrow 4.0}$	72.3$_{\downarrow 5.9}$	70.3$_{\downarrow 7.9}$
Herding (Welling, 2009)	92.2$_{\downarrow 3.4}$	88.0$_{\downarrow 7.6}$	80.1$_{\downarrow 15.5}$	73.1$_{\downarrow 5.1}$	71.8$_{\downarrow 6.4}$	69.6$_{\downarrow 8.0}$
K-Center (Sener & Savarese, 2018)	94.7$_{\downarrow 0.9}$	93.9$_{\downarrow 1.7}$	90.9$_{\downarrow 4.7}$	74.1$_{\downarrow 4.1}$	72.2$_{\downarrow 6.0}$	70.2$_{\downarrow 8.0}$
Least Confidence (Coleman et al., 2019)	95.0$_{\downarrow 0.6}$	94.5$_{\downarrow 1.1}$	90.3$_{\downarrow 5.3}$	74.2$_{\downarrow 4.0}$	72.3$_{\downarrow 5.9}$	69.8$_{\downarrow 8.4}$
Margin (Coleman et al., 2019)	94.9$_{\downarrow 0.7}$	94.3$_{\downarrow 1.3}$	90.9$_{\downarrow 4.7}$	74.0$_{\downarrow 4.2}$	72.2$_{\downarrow 6.0}$	70.2$_{\downarrow 8.0}$
Forgetting (Toneva et al., 2018)	94.7$_{\downarrow 0.9}$	94.1$_{\downarrow 1.5}$	91.7$_{\downarrow 3.9}$	75.3$_{\downarrow 2.9}$	73.1$_{\downarrow 5.1}$	69.9$_{\downarrow 8.3}$
GraNd-4 (Paul et al., 2021)	95.3$_{\downarrow 0.3}$	94.6$_{\downarrow 1.0}$	91.2$_{\downarrow 4.4}$	74.6$_{\downarrow 3.6}$	71.4$_{\downarrow 6.8}$	68.8$_{\downarrow 9.4}$
DeepFool (Ducoffe & Precioso, 2018)	95.1$_{\downarrow 0.5}$	94.1$_{\downarrow 1.5}$	90.0$_{\downarrow 5.6}$	74.2$_{\downarrow 4.0}$	73.2$_{\downarrow 5.0}$	69.8$_{\downarrow 6.4}$
Craig (Mirzasoleiman et al., 2020)	94.8$_{\downarrow 0.8}$	93.3$_{\downarrow 3.3}$	88.4$_{\downarrow 7.2}$	74.4$_{\downarrow 3.8}$	71.9$_{\downarrow 6.3}$	69.7$_{\downarrow 8.5}$
Glister (Killamsetty et al., 2021b)	95.2$_{\downarrow 0.4}$	94.0$_{\downarrow 1.6}$	90.9$_{\downarrow 4.7}$	74.6$_{\downarrow 3.6}$	73.2$_{\downarrow 5.0}$	70.4$_{\downarrow 7.8}$
Influence (Koh & Liang, 2017)	93.1$_{\downarrow 2.5}$	91.3$_{\downarrow 4.3}$	88.3$_{\downarrow 7.3}$	74.4$_{\downarrow 3.8}$	72.0$_{\downarrow 6.2}$	68.9$_{\downarrow 9.5}$
EL2N-2 (Toneva et al., 2018)	94.4$_{\downarrow 1.2}$	93.2$_{\downarrow 2.4}$	89.8$_{\downarrow 5.8}$	74.1$_{\downarrow 4.1}$	71.0$_{\downarrow 7.2}$	68.5$_{\downarrow 9.7}$
EL2N-20 (Toneva et al., 2018)	95.3$_{\downarrow 0.3}$	95.1$_{\downarrow 0.5}$	91.9$_{\downarrow 3.7}$	77.2$_{\downarrow 1.0}$	72.1$_{\downarrow 6.1}$	-
DP (Yang et al., 2023)	94.9$_{\downarrow 0.7}$	93.8$_{\downarrow 1.8}$	90.8$_{\downarrow 4.8}$	77.2$_{\downarrow 1.0}$	73.1$_{\downarrow 5.1}$	-
OrderDP	**95.6**$_{\uparrow 0.0}$	**95.3**$_{\downarrow 0.2}$	**95.0**$_{\downarrow 0.6}$	**78.2**$_{\uparrow 0.0}$	**77.9**$_{\downarrow 0.3}$	**76.7**$_{\downarrow 1.5}$
Whole Dataset	95.6$_{\pm 0.1}$			78.2$_{\pm 0.1}$		

4.2 Implementation Details

In this section, we provide a succinct overview of the implementation details for our experiments, including backbone models and training details.

Backbone models. For classification, we train ResNet-18 and ResNet-50 (He et al., 2016) on CIFAR-10/100 and ImageNet-1K.

Training Details. For **`OrderDP`**, an exploration ratio of 0.5 related to s (i.e., $s/|\mathcal{D}| = 0.5$) and an exploitation ratio of 0.6 related to q (i.e., $q/s = 0.6$) are used by default when no other values are specified. All models are trained with the OneCycle scheduler, which employs cosine annealing, using SGD with a momentum of 0.9 and a weight decay of 5×10^{-4}. Images are augmented through normalization, random cropping, and horizontal flipping unless stated otherwise. The implementation is based on PyTorch (Paszke, 2019). All other details are deferred to the Appendix C and D.1.

5 Empirical Studies

5.1 Empirical analysis on CIFAR

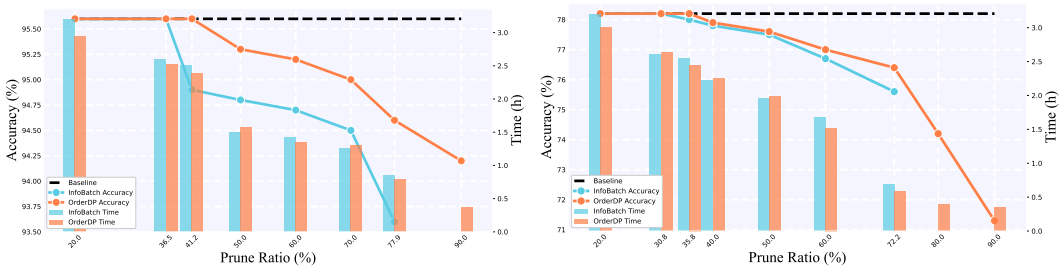
For a comprehensive comparison on CIFAR-10/100, we consider two categories of DP methods as baselines: static DP and dynamic DP. From static DP, we include 15 representative methods: static random pruning, CD (Agarwal et al., 2020), Herding (Welling, 2009), K-means (Sorscher et al., 2022), Least Confidence and Entropy (Coleman et al., 2019), Forgetting (Toneva et al., 2018), GraNd and EL2N (Paul et al., 2021), DeepFool (Ducoffe & Precioso, 2018), Craig (Mirzasoleiman et al., 2020), Glister (Killamsetty et al., 2021b), Influence (Koh & Liang, 2017), and DP (Yang et al., 2022). From dynamic DP, we adopt four methods: dynamic random pruning, ϵ-greedy (Raju et al., 2021), UCB (Raju et al., 2021), and InfoBatch[3] (Qin et al., 2024), along with our proposed method **`OrderDP`**, which also belongs here.

Performance comparison. From Tables 1 and 2, our systematic study suggests the following trends: ① Dynamic random pruning outperforms static random by preserving higher sample diversity, and both ϵ-greedy and UCB adaptively explore sample importance, but **`OrderDP`** consistently surpasses other baselines in accuracy and robustness across all prune ratios. ② At 30% pruning, only **`OrderDP`** matches full-data accuracy. ③ Under 50% and 70%, **`OrderDP`** has the smallest accuracy drop,

[3]In the original experiments of InfoBatch (Qin et al., 2024), an annealing algorithm was incorporated. To ensure fair comparison, we have removed this component from all implementations.

Table 2: Dynamic pruning results (accuracy, %) on `CIFAR10` and `CIFAR100` with ResNet-18 and ResNet-50. Accuracy (%, ↑). Best in **bold**. Performance gaps to full-data are in blue / orange.

Dataset	CIFAR10						CIFAR100					
Backbone	ResNet-18			ResNet-50			ResNet-18			ResNet-50		
Prune Ratio %	30	50	70	30	50	70	30	50	70	30	50	70
Dynamic Random	94.8$_{\downarrow 0.8}$	94.5$_{\downarrow 1.1}$	93.0$_{\downarrow 2.6}$	95.1$_{\downarrow 0.5}$	94.9$_{\downarrow 0.7}$	93.6$_{\downarrow 2.0}$	77.3$_{\downarrow 0.9}$	75.3$_{\downarrow 2.9}$	72.8$_{\downarrow 5.4}$	77.9$_{\downarrow 2.7}$	76.1$_{\downarrow 4.5}$	73.9$_{\downarrow 6.7}$
ϵ-greedy	95.2$_{\downarrow 0.4}$	94.9$_{\downarrow 0.7}$	94.1$_{\downarrow 1.5}$	95.4$_{\downarrow 0.2}$	95.1$_{\downarrow 0.5}$	94.3$_{\downarrow 1.3}$	76.4$_{\downarrow 1.8}$	74.8$_{\downarrow 3.4}$	72.9$_{\downarrow 5.3}$	77.2$_{\downarrow 3.4}$	76.3$_{\downarrow 4.3}$	74.1$_{\downarrow 6.5}$
UCB	95.3$_{\downarrow 0.3}$	94.7$_{\downarrow 0.9}$	93.9$_{\downarrow 1.7}$	95.5$_{\downarrow 0.1}$	95.0$_{\downarrow 0.6}$	94.2$_{\downarrow 1.4}$	77.3$_{\downarrow 0.9}$	75.3$_{\downarrow 2.9}$	73.2$_{\downarrow 5.0}$	78.0$_{\downarrow 2.6}$	76.5$_{\downarrow 4.1}$	74.3$_{\downarrow 6.3}$
InfoBatch	95.6$_{\downarrow 0.0}$	95.0$_{\downarrow 0.6}$	94.5$_{\downarrow 1.1}$	95.6$_{\downarrow 0.0}$	95.3$_{\downarrow 0.3}$	94.7$_{\downarrow 0.9}$	78.1$_{\downarrow 0.1}$	77.7$_{\downarrow 0.5}$	75.9$_{\downarrow 2.3}$	80.4$_{\downarrow 0.2}$	78.6$_{\downarrow 2.0}$	76.4$_{\downarrow 4.2}$
OrderDP	**95.6**$_{\uparrow 0.0}$	**95.3**$_{\downarrow 0.2}$	**95.0**$_{\downarrow 0.6}$	**95.6**$_{\uparrow 0.0}$	**95.4**$_{\downarrow 0.2}$	**95.0**$_{\downarrow 0.6}$	**78.2**$_{\uparrow 0.0}$	**77.9**$_{\downarrow 0.3}$	**76.7**$_{\downarrow 1.5}$	**80.6**$_{\uparrow 0.0}$	**79.8**$_{\downarrow 0.8}$	**77.9**$_{\downarrow 2.7}$
Whole Dataset	95.6$_{\pm 0.1}$			95.6$_{\pm 0.1}$			78.2$_{\pm 0.1}$			80.6$_{\pm 0.1}$		



(a) Accuracy vs. Training Time on CIFAR-10 (b) Accuracy vs. Training Time on CIFAR-100

Figure 3: More accuracy and time results for different prune ratios on CIFAR-10/100 for **`OrderDP`** and InfoBatch, using ResNet-18. The lossless pruning ratios are marked in the figure.

outperforming both static and existing dynamic methods. ④ Compared to InfoBatch, **`OrderDP`** consistently yields higher accuracy as pruning becomes more aggressive.

Efficiency comparison. Table 3 reports end-to-end training time and GPU-hours under identical settings. **`OrderDP`** achieves the fastest training and lowest GPU-hours, improving upon InfoBatch without loss in accuracy. Additional CIFAR results and extended comparisons are in Appendix D.

Extended comparison of varying pruning ratios. To further evaluate the performance of **`OrderDP`**, we compare it with InfoBatch, a state-of-the-art data pruning algorithm, across different prune ratios on the CIFAR-10 and CIFAR-100 datasets. The results are demonstrated in Figure 3. It can be observed that **`OrderDP`** not only achieves higher accuracy at every pruning ratio, but also remains comparable to other algorithms and reduces total training time in most cases. Moreover, InfoBatch cannot prune to an extreme ratio (limited by 77.9% on CIFAR-10 or 72.2% on CIFAR-100 in our setting) due to its fixed retention mechanism. ***`OrderDP`*** *supports arbitrary pruning ratios because data retention is fully specified by the the exploration size s and exploitation size q (see Section 5.3).* These results confirm that **`OrderDP`** 's sample-selection strategy delivers near-optimal efficiency and robustness, making it particularly well-suited for resource-constrained scenarios where preserving accuracy is paramount.

5.2 Empirical analysis on ImageNet-1K

We evaluate Dynamic Random, UCB (Raju et al., 2021), InfoBatch (Qin et al., 2024) and our **`OrderDP`** on ImageNet-1K with ResNet-50 at 40% pruning (Table 3). **`OrderDP`** matches/exceeds all baselines in accuracy while not significantly increasing the total GPU runtime; it retains full-data performance at 40% pruning and incurs only a 0.4% drop at 60% (Table 4). ① *Efficiency:* **`OrderDP`** completes training faster and uses fewer GPU-hours than all competing methods. ② *Robustness:* It shows no loss at moderate prune ratios and only minimal degradation under aggressive pruning. Together, these findings confirm that **`OrderDP`** achieves near-lossless accuracy with a substantial reduction in compute, making it ideal for large-scale training under tight resource budgets.

5.3 Ablation experiment

We study how two-stage pruning decomposition, parameterized by the exploration size s and exploitation size q, affects **`OrderDP`** 's performance on CIFAR-10/100 with ResNet-18 (Figure 4).

Table 3: Comparison of performance and time cost on ImageNet-1K. Results are reported with ResNet-50 under 40% prune ratio for 90 epochs on a 2-L40-GPU server. "Total (n*h)" is the total node hour.

	Random	ϵ-greedy	UCB	InfoBatch	Ours	Full Data
Acc (%)	$73.4_{\pm 0.3}$	$75.2_{\pm 0.3}$	$75.4_{\pm 0.3}$	$75.6_{\pm 0.2}$	**76.4**$_{\pm 0.2}$	$76.4_{\pm 0.2}$
Time (h)	21.1	21.1	21.1	21.6	21.5	35.2
Total (n*h)	42.2	42.2	42.2	43.2	43.0	70.4

Table 4: Experiments on ImageNet-1K. The models here are all implemented based on ResNet-50$_{\text{PyTorch}}$.

Prune Ratio %	30	40	60
InfoBatch	$76.4_{\downarrow 0.0}$	$75.6_{\downarrow 0.8}$	$74.9_{\downarrow 1.5}$
OrderDP	**76.4**$_{\uparrow 0.0}$	$76.4_{\uparrow 0.0}$	$76.0_{\downarrow 0.4}$
Whole Dataset	$76.4_{\pm 0.1}$		

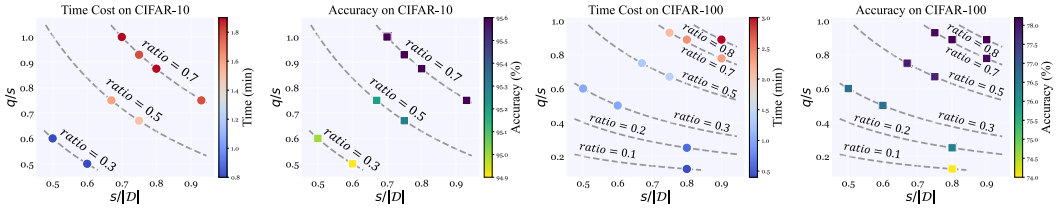


Figure 4: Performance with different ratio parameters. Here $(q/s) \cdot (s/|\mathcal{D}|)$ represents the retained data ratio, and thus the prune ratio is calculated as $1 - (q/s) \cdot (s/|\mathcal{D}|)$. Results are reported with ResNet-18.

Fixed prune ratio Under a fixed effective prune ratio, we vary the decomposition of the retained data portion by adjusting the exploration ratio $s/|D|$ and the exploitation ratio q/s, while keeping their product $(q/s) \cdot (s/|D|)$ unchanged. As shown in Figure 4, **OrderDP** achieves identical accuracy across all decompositions on both datasets, demonstrating its precise control over the prune ratio. The training time remains stable across different decompositions, indicating consistent computational cost when the overall prune ratio is fixed.

Varying prune ratios. As the prune ratio grows, training time drops sharply while accuracy degrades more slowly—up to about 70% pruning, where we see over 95% on CIFAR-10 and over 76% on CIFAR-100 with half the compute. Beyond that, further pruning gives diminishing accuracy but continues to cut runtime. This shows **OrderDP** 's ability to trace a smooth efficiency–performance frontier and lets practitioners pick the "sweet spot" matching their compute budget.

Our ablation shows that decoupling exploration (s) and exploitation (q) achieves exact pruning ratios without efficiency loss and yields a smooth accuracy–cost frontier, enabling straightforward budget selection. We further provide stability results under multiple runs in Appendix D.6.

5.4 Sensitivity Analysis

Cross-architecture robustness evaluation. Table 6 reports the maximum lossless prune ratios of InfoBatch and **OrderDP** on ResNet-18/50 across CIFAR-10, CIFAR-100, and ImageNet-1K. InfoBatch usually caps in the mid-30% range, while **OrderDP** extends this by 4–6 points, especially on harder datasets, showing its ability to prune more aggressively without accuracy loss.

We adopt the Timm (Wightman et al., 2021) ImageNet training stack, which combines mixed-precision training with strong augmentation and regularization methods such as MixUp, and CutMix (Zhong et al., 2017; Zhang et al., 2018; Yun et al., 2019), and observe that **OrderDP** continues to yield lossless speedups under this stronger recipe, indicating that it is compatible with existing acceleration and augmentation pipelines. Beyond CNN-based architectures, **OrderDP** also maintains lossless accuracy at 20%–30% pruning on Swin-Tiny (Liu et al., 2021) and ViT-Base (MAE) (He et al., 2021) (Table 5), showing that the loss-based ordering remains stable on Vision Transformers, and that **OrderDP** naturally transfers across heterogeneous architectures as a plug-and-play module.

Table 5: Cross-architecture robustness evaluation on ImageNet-1K. ViT-Base (MAE) is pretrained with **OrderDP** for 300 epochs and fine-tuned for 100 epochs. Swin-Tiny is trained from scratch with **OrderDP**.

Model	Prune Ratio	Original	OrderDP
R-50$_{\text{Timm}}$	29.8%	78.4	$78.3_{\downarrow 0.1}$
Swin-T	22.1%	81.5	$81.4_{\downarrow 0.1}$
ViT-B (MAE)	30.8%	82.8	$82.8_{\uparrow 0.0}$

Table 6: Cross-architecture robustness results of OrderDP. 'Full Dataset' denotes training on the original dataset without pruning.

	CIFAR-10		CIFAR-100		ImageNet-1K	
	R-18	R-50	R-18	R-50	R-18	R-50
Full Dataset	95.6	95.6	78.2	80.6	70.5	76.4
InfoBatch	95.5	95.6	78.2	80.6	70.4	76.4
Saved (%)	36.5	37.1	30.8	36.3	21.8	34.3
OrderDP	95.5	95.6	78.2	80.6	70.5	76.5
Saved (%)	**41.2**	**42.6**	**35.8**	**41.1**	**27.3**	**39.8**

Table 7: Comparison of accuracy (%) and saved cost (%) on CIFAR-10 when trained with R-50 using different optimizers.

	SGD	AdamW	LARS	LAMB
Full Dataset	95.6	94.3	95.5	95.0
InfoBatch	95.6	94.3	95.5	95.0
Saved (%)	37.1	37.0	37.1	37.1
OrderDP	95.6	94.4	95.5	95.0
Saved (%)	**42.6**	**42.4**	**42.5**	**42.6**

Note: All the results are obtained from an 2-L40-GPU server.

Cross-optimizer robustness evaluation. We test widely used optimizers—SGD (Bottou et al., 1991), AdamW (Loshchilov & Hutter, 2019), LARS (You et al., 2017), and LAMB (You et al., 2019)—on ResNet-50/CIFAR-10 (Table 7). InfoBatch saves 37.1% of training cost across all optimizers, while **`OrderDP`** raises savings to about 42.5%. This consistent gain shows that **`OrderDP`** 's dynamic sample selection is optimizer-agnostic: by focusing on high-loss examples, it reduces redundant computation and delivers plug-and-play efficiency without accuracy loss.

6 Conclusion

In this paper, we introduced **`OrderDP`**, a novel dynamic data pruning framework that provides rigorous theoretical guarantees while achieving substantial training acceleration. Unlike prior approaches, **`OrderDP`** ensures unbiased gradient estimation and offers exact control of the pruning ratio, leading to more stable and efficient data pruning. Our theoretical analysis establishes both convergence guarantees and generalization bounds, demonstrating its robustness across datasets and pruning levels. Empirically, **`OrderDP`** consistently attains equal or better accuracy than state-of-the-art baselines, while reducing runtime and overall computational cost by 40–45%. Moreover, its simpler plug-and-play design makes it easy to integrate into existing pipelines. These findings highlight the potential of our method as a scalable solution that balances efficiency, accuracy, and theoretical rigor.

Ethics Statement

Our work focuses on improving training efficiency in deep learning through dynamic data pruning. All experiments are conducted on widely used public benchmark datasets (CIFAR-10, CIFAR-100, and ImageNet-1K), which do not involve any personally identifiable information, sensitive attributes, or human subjects. The study does not pose foreseeable risks related to privacy, fairness, or security. Moreover, no external sponsorship or conflicts of interest have influenced the design, analysis, or reporting of this work. As such, we believe our research complies with the ICLR Code of Ethics and raises no ethical concerns.

Reproducibility Statement

We are committed to ensuring the reproducibility of our results. To this end, we provide detailed descriptions of datasets (CIFAR-10, CIFAR-100, ImageNet-1K), model architectures (ResNet-18, ResNet-50), hyperparameters, and training protocols in the main text and Appendix. For reproducibility, our implementation is based on PyTorch, with standard data augmentation (normalization, random cropping, horizontal flipping), SGD optimizer with momentum, weight decay, and OneCycle learning rate scheduling. We will submit the full source code and configuration files in the supplementary material to enable independent verification of our experiments. In addition, ablation studies and sensitivity analyses provide transparency into the robustness of our method across pruning ratios, optimizers, and architectures.

REFERENCES

Carlo Acerbi and Dirk Tasche. On the coherence of expected shortfall. *Journal of banking & finance*, 26(7):1487–1503, 2002. 18

Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *ECCV*, pp. 137–153. Springer, 2020. 7, 20

Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5): 185–196, 1993. 3

Shun-ichi Amari, Naotake Fujita, and Shigeru Shinomoto. Four types of learning curves. *Neural Computation*, 4(4):605–618, 1992. 1

Fadhel Ayed and Soufiane Hayou. Data pruning and neural scaling laws: fundamental limitations of score-based algorithms. *arXiv preprint arXiv:2302.06960*, 2023. 2, 21

Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in neural information processing systems*, 33:14879–14890, 2020. 1

Léon Bottou et al. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91 (8):12, 1991. 10

Trevor Campbell and Tamara Broderick. Automated scalable bayesian inference via hilbert coresets. *Journal of Machine Learning Research*, 20(15):1–38, 2019. 1

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. Instruction mining: Instruction data selection for tuning large language models. *arXiv preprint arXiv:2307.06290*, 2023. 21

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. *Advances in Neural Information Processing Systems*, 30, 2017. 20

Dingshuo Chen, Yanqiao Zhu, Jieyu Zhang, Yuanqi Du, Zhixun Li, Qiang Liu, Shu Wu, and Liang Wang. Uncovering neural scaling laws in molecular representation learning. *Advances in Neural Information Processing Systems*, 36:1452–1475, 2023. 1

Dingshuo Chen, Zhixun Li, Yuyan Ni, Guibin Zhang, Ding Wang, Qiang Liu, Shu Wu, Jeffrey Yu, and Liang Wang. Beyond efficiency: Molecular data pruning for enhanced generalization. *Advances in Neural Information Processing Systems*, 37:18036–18061, 2024. 1

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2818–2829, 2023. 1

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 *ICLR*, 2019. 7, 20

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*, pp. 248–255. Ieee, 2009. 3, 6

Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018. 7, 20

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 7

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. 9

Muyang He, Shuo Yang, Tiejun Huang, and Bo Zhao. Large-scale dataset pruning with dynamic uncertainty. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7713–7722, 2024. 20

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021. 1

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017. 1

Jonathan Huggins, Trevor Campbell, and Tamara Broderick. Coresets for scalable bayesian logistic regression. *Advances in neural information processing systems*, 29, 2016. 1

Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. 20

Rishabh Iyer, Ninad Khargoankar, Jeff Bilmes, and Himanshu Asanani. Submodular combinatorial information measures with applications in machine learning. In *Algorithmic Learning Theory*, pp. 722–754. PMLR, 2021. 20

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1

Kenji Kawaguchi and Haihao Lu. Ordered sgd: A new stochastic optimization framework for empirical risk minimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 669–679. PMLR, 2020. 3, 5

Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pp. 5464–5474. PMLR, 2021a. 20

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021b. 1, 7, 20

Sungnyun Kim, Sangmin Bae, and Se-Young Yun. Coreset sampling from open-set for fine-grained self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7537–7547, 2023. 1

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1885–1894. JMLR. org, 2017. 1, 7, 20

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). a. URL http://www.cs.toronto.edu/~kriz/cifar.html. 3, 6

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research). b. URL http://www.cs.toronto.edu/~kriz/cifar.html. 3, 6

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 9

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 10

Zhiyun Lu, Yongqiang Wang, Yu Zhang, Wei Han, Zhehuai Chen, and Parisa Haghani. Unsupervised data selection via discrete speech representation for asr. *arXiv preprint arXiv:2204.01981*, 2022. 21

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764*, 2021. 20

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*, 2023. 21

Ronak Mehta, Vincent Roulet, Krishna Pillutla, Lang Liu, and Zaid Harchaoui. Stochastic optimization for spectral risk measures. In *International Conference on Artificial Intelligence and Statistics*, pp. 10112–10159. PMLR, 2023. 3, 5, 6, 18, 19

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *ICML*. PMLR, 2020. 1, 7

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*, 2024. URL https://openreview.net/forum?id=JnRStoIuTe. 20

A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019. 7

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. 7, 20

Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, xu Zhao Pan, 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. 1, 2, 4, 7, 8, 20

Ravi S Raju, Kyle Daruwalla, and Mikko Lipasti. Accelerating deep learning with dynamic data pruning, 2021. 1, 4, 7, 8, 20

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018. 7, 20

Vatsal Shah, Xiaoxia Wu, and Sujay Sanghavi. Choosing the sample with lowest loss makes sgd robust. In *International Conference on Artificial Intelligence and Statistics*, pp. 2120–2130. PMLR, 2020. 4, 5

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022. 7, 20

Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. Data pruning via moving-one-sample-out. *Advances in neural information processing systems*, 36: 18251–18262, 2023. 20

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. *arXiv preprint arXiv:1812.05159*, 2018. 7, 20

Max Welling. Herding dynamical weights to learn. In *ICMLg*, pp. 1121–1128, 2009. 7, 20

Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm, 2021. 9

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818, 2023. 21

Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. *arXiv preprint arXiv:2205.09329*, 2022. 1, 7, 20

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. 7

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 10

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019. 10

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019. 9

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018. 9

Xingxuan Zhang, Renzhe Xu, Han Yu, Hao Zou, and Peng Cui. Gradient norm aware minimization seeks first-order flatness and improves generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20247–20257, 2023. 2

Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. In *International conference on machine learning*, pp. 26982–26992. PMLR, 2022. 2

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation, 2017. 9

A Proof of Theoretical Analysis

A.1 Proof of Theorem 1

Proof. We just need to show that $\tilde{g}$ is an unbiased estimator of a sub-gradient of $L_q(\theta)$ at θ^t, namely $E\tilde{g} \in \partial L_q(\theta^t)$. At first, it holds that

$$\begin{aligned} E\tilde{g}^t &= \frac{1}{q} E \sum_{i \in Q} g_i^t = \frac{1}{q} \sum_{i=1}^{n} P(i \in Q) g_i^t \\ &= \frac{1}{q} \sum_{j=1}^{n} P((j) \in Q) g_{(j)}^t \,, \end{aligned}$$

where $g_i^t \in \partial L_i(\theta^t)$ is a sub-gradient of L_i at θ^t. In the above equality chain, the third equality is simply the definition of expectation, and the last equality is because $((1), (2), \ldots, (n))$ is a permutation of $(1, 2, \ldots, n)$.

For any given index j, $P((j) \in Q) \neq \frac{q}{n}$ thus $E\tilde{g}^t \notin \partial L(\theta^t)$. To analyze $P((j) \in Q)$, define $A_j = ((1), (2), \ldots, (j-1))$, and $A_j^c = ((j+1), (j+2), \ldots, (n))$ then

$$\begin{aligned} &P((j) \in Q) = P\left((j) \in q\text{-argmax}_{i \in S} \mathcal{H}_i(\theta)\right) \\ &= P\left((j) \in S \text{ and } S \text{ contains at most } q-1 \text{ items in } A_j\right) \\ &= P\left((j) \in S\right) P\left(S \text{ contains at most } q-1 \text{ items in } A_j \mid (j) \in S\right) \\ &= P\left((j) \in S\right) \textstyle\sum_{l=l_1}^{l_2} P\left((j) \text{ appears at } l \text{ position in } S \mid (j) \in S\right), \end{aligned} \tag{7}$$

where $0 \leq l_1 \leq l_2 \leq s$ measures the possible positions of $(j) \in S$ These two variables vary depending on the choice of (j). For examples, if $(j) = (1)$, (j) should be included in Q since (1) would be at the top-1 position of S.

Notice that S is randomly chosen from sample index set $(1, 2, \ldots, n)$ without replacement. There are in total $\binom{n}{s}$ different sets S such that $|S| = s$. Among them, there are $\binom{n-1}{s-1}$ different sets S which contains the index (j), thus

$$P\left((j) \in S\right) = \frac{\binom{n-1}{s-1}}{\binom{n}{s}} \,. \tag{8}$$

Given the condition $(j) \in S$, (j) appears at l position means S contains $l-1$ items in A_j and $s-l$ items in A_j^c, thus we have the constraints:

$$s - l \leq n - j \quad \text{and} \quad 1 \leq l - 1 \leq j - 1.$$

Thus we conclude $s - n + j \leq l \leq j$, i.e., $l_1 = \max\{1, s-n+j\}$ and $l_2 = \min\{1, j\}$. There are $\binom{n-j}{s-l}$ such possible set S for $(j) \in S$, whereby it holds that

$$\begin{aligned} &P\left(S \text{ contains at most } q-1 \text{ items in } A_j \mid (j) \in S\right) \\ &= \sum_{l=l_1}^{l_2} P\left((j) \text{ appears at } l \text{ position in } S \mid (j) \in S\right) \\ &= \sum_{l=\max\{1,s-n+j\}}^{\min\{q,j\}} \frac{\binom{j-1}{l-1}\binom{n-j}{s-l}}{\binom{n-1}{s-1}} \end{aligned} \tag{9}$$

Substituting Equations (7) and (8) into Equation (6), we arrive at

$$P((j) \in Q) = \frac{\binom{n-1}{s-1}}{\binom{n}{s}} \sum_{l=\max\{1,s-n+j\}}^{\min\{q,j\}} \frac{\binom{j-1}{l-1}\binom{n-j}{s-l}}{\binom{n-1}{s-1}} = \sum_{l=\max\{1,s-n+j\}}^{\min\{q,j\}} \frac{\binom{j-1}{l-1}\binom{n-j}{s-l}}{\binom{n}{s}} = \gamma_j. \tag{10}$$

Therefore,

$$E\tilde{g}^t = \frac{1}{q} \sum_{j=1}^{n} P((j) \in Q) g_{(j)}^t = \frac{1}{q} \sum_{j=1}^{n} \gamma_j g_{(j)}^t \in \partial L_q(\theta^t) \,, \tag{11}$$

where the last inequality is due to the additivity of sub-gradient (for both convex and weakly convex function) □. □

A.2 Proof of Proposition 2

Proof. We will show that

$$\lim_{j,n\to\infty, j/n=z} \gamma_j = \frac{1}{n}\frac{s!}{(l-1)!(s-l)!}\sum_{l=1}^{q}\left(\frac{j}{n}\right)^{l-1}\left(1-\frac{j}{n}\right)^{s-l}. \tag{12}$$

We begin the proof by changing the variable $z = \frac{j}{n}$.

At first, the Stirling's approximation yields that when n and j are both sufficiently large, it holds that

$$\binom{n}{j} \sim \sqrt{\frac{n}{2\pi j(n-j)}}\frac{n^n}{j^j(n-j)^{n-j}}. \tag{13}$$

Thus,

$$\lim_{j,n\to\infty, j/n=z} \frac{\binom{n-s}{j-l}}{\binom{n-1}{j-1}} = \frac{\frac{n^{n-s}}{j^{j-l}(n-j)^{n-j-s+l}}}{\frac{n^{n-1}}{j^{j-1}(n-j)^{n-j}}} = \frac{j^{l-1}(n-j)^{s-l}}{n^{s-1}} = \left(\frac{j}{n}\right)^{l-1}\left(\frac{n-j}{n}\right)^{s-l} \tag{14}$$

where the first equality utilizes Equation (10) and the fact that $s, l, 1$ are negligible in the limit case (except the exponent terms).

On the other hand, it holds by rearranging the factorial numbers that

$$\frac{1}{n}\frac{\binom{n-s}{j-l}}{\binom{n-1}{j-1}}\frac{s!}{(l-1)!(s-l)!} = \frac{\binom{j-1}{l-1}\binom{n-j}{s-l}}{\binom{n}{s}}. \tag{12}$$

Recall $\gamma_j = \sum_{l=\max\{1,s-n+j\}}^{\min\{q,j\}} \frac{\binom{j-1}{l-1}\binom{n-j}{s-l}}{\binom{n}{s}}$. Let

$$\gamma_j = \sum_{l=\max\{1,s-n+j\}}^{\min\{q,j\}} \frac{\binom{j-1}{l-1}\binom{n-j}{s-l}}{\binom{n}{s}} = \sum_{l=1}^{q}\frac{\binom{j-1}{l-1}\binom{n-j}{s-l}}{\binom{n}{s}}, \tag{15}$$

where we set the value to 0 for $l \in [1, s-n+j]$ and $[j, q]$ if $s-n+j > 1$ and $j < q$. Therefore, we conclude the following by noticing $s > q$,

$$\begin{aligned}
\frac{d}{dz}\gamma(z) &= \sum_{l=2}^{q}(l-1)z^{l-2}(1-z)^{s-l}\frac{s!}{(l-1)!(s-l)!} - \sum_{l=1}^{q}(s-l)z^{l-1}(1-z)^{s-l-1}\frac{s!}{(l-1)!(s-l)!} \\
&= \sum_{l=2}^{q} z^{l-2}(1-z)^{s-l}\frac{s!}{(l-2)!(s-l)!} - \sum_{l=1}^{q} z^{l-1}(1-z)^{s-l-1}\frac{s!}{(l-1)!(s-l-1)!} \\
&= \sum_{l=1}^{q-1} z^{l-1}(1-z)^{s-l-1}\frac{s!}{(l-1)!(s-l-1)!} - \sum_{l=1}^{q} z^{l-1}(1-z)^{s-l-1}\frac{s!}{(l-1)!(s-l-1)!} \\
&= -z^{q-1}(1-z)^{s-q-1}\frac{s!}{(q-1)!(s-q-1)!} \\
&= -z^{q-1}(1-z)^{s-q-1}\frac{(s-1)!}{(q-1)!(s-q-1)!}s.
\end{aligned} \tag{16}$$

In other words, $1 - \frac{1}{s}\gamma(z)$ is the cumulative distribution function of Beta$(q, s-q)$ when $n \to \infty$. □

A.3 Proof of Theorem 3

Full version of Theorem 3: Let $(\theta^t)_{t=0}^T$ be the sequence generated by Algorithm 1. Suppose there exists a finite $\theta^* \in \arg\min_\theta \mathcal{L}_q(\theta)$, $\mathcal{L}_q(\theta^*) < \infty$. If each $\mathcal{L}_i(\cdot)$ is convex and G-Lipschitz, then

$$\min_{0\le t\le T} \mathbb{E}\left[\mathcal{L}_q(\theta^t) - \mathcal{L}_q(\theta^*)\right] \le \frac{\eta_{\max}\left(\|\theta^1 - \theta^*\|_2^2 + G^2\sum_{t=1}^{T}(\eta^t)^2\right)}{2\,\eta_{\min}\sum_{t=1}^{T}\eta^t}. \tag{17}$$

Moreover, if we define the weighted average $\bar{\theta}^T = \frac{1}{\sum_{t=1}^T \eta^t} \sum_{t=1}^T \eta^t \theta^t$, then

$$\mathbb{E}\big[\mathcal{L}_q(\bar{\theta}^T) - \mathcal{L}_q(\theta^*)\big] \leq \frac{\|\theta^1 - \theta^*\|_2^2 + G^2 \sum_{t=1}^T (\eta^t)^2}{2 \sum_{t=1}^T \eta^t}. \tag{18}$$

Proof. Consider the one update at epoch t, we have

$$\|\theta^{t+1} - \theta^*\|_2^2 = \|\theta^t - \theta^*\|_2^2 - 2\eta^t \langle \tilde{g}^t, \theta^t - \theta^* \rangle + (\eta^t)^2 \|\tilde{g}^t\|_2^2. \tag{19}$$

Taking the conditional expectation of v^t given θ of equation 19 yields

$$\mathbb{E}[\|\theta^{t+1} - \theta^*\|_2^2] \leq \|\theta^t - \theta^*\|_2^2 - 2\eta^t \langle \mathbb{E}[\tilde{g}^t], \theta^t - \theta^* \rangle + (\eta^t)^2 G^2, \tag{20}$$

where we use $\|\tilde{g}^t\|_2 \leq G$. Because we maintain an unbiased gradient estimator $\mathbb{E}[\tilde{g}^t] \in \partial \mathcal{L}_q(\theta^t)$, we have that with convexity of $\mathcal{L}_q$, we have

$$-\langle \mathbb{E}[\tilde{g}^t], \theta^t - \theta^* \rangle \leq -(\mathcal{L}_q(\theta^t) - \mathcal{L}_q(\theta^*)), \tag{21}$$

where the $\theta^* \overset{\text{def}}{:=} \arg\min_\theta \mathcal{L}_q(\theta)$. Substituted equation 21 into equation 20 gives

$$\begin{aligned} &\mathbb{E}\left[\|\theta^{t+1} - \theta^*\|_2^2\right] \leq \|\theta^t - \theta^*\|_2^2 - 2\eta^t \left(\mathcal{L}_q(\theta^t) - \mathcal{L}_q(\theta^*)\right) + (\eta^t)^2 G^2. \\ \implies\ &2\eta^t (\mathcal{L}_q(\theta^t) - \mathcal{L}_q(\theta^*)) \leq \left(\|\theta^t - \theta^*\|_2^2 - \mathbb{E}[\|\theta^{t+1} - \theta^*\|_2^2]\right) + (\eta^t)^2 G^2. \end{aligned} \tag{22}$$

Take the expectation over the entire sequence $\theta^1, \ldots, \theta^{t+1}$, sum over $t = 1, \ldots, T$, we have

$$2 \sum_{t=1}^T \eta^t \mathbb{E}[\mathcal{L}_q(\theta^t) - \mathcal{L}_q(\theta^*)] \leq \left(\|\theta^1 - \theta^*\|_2^2 - \mathbb{E}[\|\theta^{T+1} - \theta^*\|_2^2]\right) + G^2 \sum_{t=1}^T (\eta^t)^2. \tag{23}$$

It shows that

$$\frac{1}{\sum_{t=1}^T \eta^t} \sum_{t=1}^T \eta^t \mathbb{E}[\mathcal{L}_q(\theta^t) - \mathcal{L}_q(\theta^*)] \leq \frac{\|\theta^1 - \theta^*\|_2^2 + G^2 \sum_{t=1}^T (\eta^t)^2}{2 \sum_{t=1}^T \eta^t}. \tag{24}$$

With the observation that

$$\begin{aligned} &\frac{1}{\sum_{t=1}^T \eta^t} \sum_{t=1}^T \eta^t \mathbb{E}[\mathcal{L}_q(\theta^t) - \mathcal{L}_q(\theta^*)] \\ =&\frac{1}{\frac{1}{T}\sum_{t=1}^T \eta^t} \frac{1}{T} \sum_{t=1}^T \eta^t \mathbb{E}[\mathcal{L}_q(\theta^t) - \mathcal{L}_q(\theta^*)] \\ \geq& \frac{\min_{1 \leq t \leq T} \eta^t \mathbb{E}[\mathcal{L}_q(\theta^t) - \mathcal{L}_q(\theta^*)]}{\max_{1 \leq t \leq T} \eta^t} \\ \geq& \frac{\eta_{min}}{\eta_{max}} \min_{1 \leq t \leq T} \mathbb{E}[\mathcal{L}_q(\theta^t) - \mathcal{L}_q(\theta^*)], \end{aligned} \tag{25}$$

where $\eta_{min} = \min_{1 \leq t \leq T} \eta^t$ and $\eta_{max} = \max_{1 \leq t \leq T} \eta^t$. Therefore, it holds that

$$\begin{aligned} \frac{\eta_{min}}{\eta_{max}} \min_{1 \leq t \leq T} \mathbb{E}[\mathcal{L}_q(\theta^t) - \mathcal{L}_q(\theta^*)] \leq& \frac{1}{\sum_{t=1}^T \eta^t} \sum_{t=1}^T \eta^t \mathbb{E}[\mathcal{L}_q(\theta^t) - \mathcal{L}_q(\theta^*)] \\ \leq& \frac{\|\theta^1 - \theta^*\|_2^2 + G^2 \sum_{t=1}^T (\eta^t)^2}{\sum_{t=1}^T \eta^t}. \end{aligned} \tag{26}$$

Then we can derive that

$$\min_{1 \leq t \leq T} \mathbb{E}[\mathcal{L}_q(\theta^t) - \mathcal{L}_q(\theta^*)] \leq \frac{\eta_{max}(\|\theta^1 - \theta^*\|_2^2 + G^2 \sum_{t=1}^T (\eta^t)^2)}{2\eta_{min} \sum_{t=1}^T \eta^t}. \tag{27}$$

□

A.4 Proof of Theorem 4

Proof. We begin this proof by leveraging the concepts of spectral risk measure (Acerbi & Tasche, 2002; Mehta et al., 2023). the surrogate loss $\mathcal{L}_q(\theta, D) = \sum_{i=1}^n \frac{\gamma_i}{q} L_{(i)}(\theta) = \sum_{i=1}^n \sigma_j Z_{(i)}$ is called an L-risk with a spectrum $\sigma_i = \frac{\gamma_i}{q}$ and $Z_{(i)} = L_{(i)}(\theta)$ for $i \in [n]$, which can be regarded as a functional of the CDF known as a *spectral risk measure*. $\{Z_i\}_{i=1}^n$ are arbitrary real-valued i.i.d. random variables drawn from CDF F. For our case, these refer to data instance D_i of n samples drawn from distribution $\mathcal{D}$ under parameter vector θ.

Let $F_n(z) := \frac{1}{n}\sum_{i=1}^n 1_{(-\infty,z]}(Z_i)$ denote the (random) empirical CDF of the sample and define the empirical *quantile function* (or inverse CDF) as

$$F_n^{-1}(t) := \inf\{z : F_n(z) \geq t\} \quad \text{for } t \in (0,1). \tag{28}$$

The population quantile function is defined similarly as

$$F^{-1}(t) := \inf\{z : F(z) \geq t\}. \tag{29}$$

The empirical quantile function can be written in terms of the order statistics as $F_n^{-1}(t) = Z_{(\lceil nt \rceil)}$. Notice in particular that when $t \in \left(\frac{i-1}{n}, \frac{i}{n}\right)$, we have that $F_n^{-1}(t) = Z_{(i)}$, where end-points are chosen to make F_n^{-1} left continuous.

The spectrum σ of an L-risk is typically defined as a discretization of a probability density s on $(0,1)$, such that

$$\sigma_i = \int_{(i-1)/n}^{i/n} s(t)\,\mathrm{d}t, \tag{30}$$

so that it need not be redefined for every n. Given both the construction of s and F_n^{-1}, we can rewrite the L-risk as

$$\begin{aligned}
\mathcal{L}_q(\theta, D) &= \sum_{i=1}^n \sigma_i Z_{(i)} = \sum_{i=1}^n \left(\int_{(i-1)/n}^{i/n} s(t)\,\mathrm{d}t \right) Z_{(i)} \\
&= \sum_{i=1}^n \left(\int_{(i-1)/n}^{i/n} s(t) \cdot Z_{(\lceil nt \rceil)}\,\mathrm{d}t \right) \\
&= \int_0^1 s(t) \cdot F_n^{-1}(t)\,\mathrm{d}t =: \mathbb{L}_s\left[F_n\right],
\end{aligned} \tag{31}$$

where $\mathbb{L}_s[G] := \int_0^1 s(t) G^{-1}(t)\,\mathrm{d}t$ is called a spectral risk measure with spectrum s applied to CDF G.

It stands to reason that $\mathbb{L}_s[F_n]$ converges to $\mathbb{L}_s[F]$ in an appropriate sense. This convergence is governed by the Wasserstein distance between the empirical and population distribution, which we briefly recall here. In this section, we control the bias term appearing in the convergence analysis. The following lemmas consider a set of real numbers, representing losses at a single $\theta \in \mathbb{R}^d$. Let $x_1, \ldots, x_n \in \mathbb{R}$ be call the *full batch*, and let $X_1, \ldots X_m$ be a random sample selected uniformly *without* replacement from $\{x_1, \ldots, x_n\}$, called the *minibatch*. Let

$$F_n(x) := \frac{1}{n}\sum_{i=1}^n 1_{(-\infty,x]} \text{ and } F_{n,m}(x) := \frac{1}{m}\sum_{j=1}^m 1_{(-\infty,x)} \tag{32}$$

be the empirical CDFs, and let

$$F_n^{-1}(t) := \inf\{x : F_n(x) \geq t\} \text{ and } F_{n,m}(t) := \inf\{x : F_{n,m}(x) \geq t\}. \tag{33}$$

be the empirical quantile functions of the full batch and minibatch, respectively. Similarly, let

$$\mu_n := \sum_{i=1}^n \delta_{x_i} \text{ and } \mu_{n,m} = \sum_{j=1}^m \delta_{X_j} \tag{34}$$

be the empirical measures of the full batch and minibatch, respectively, with δ_x indicating a Dirac point mass at x. Let $u(t) := 1_{(0,1)}t$ be the uniform spectrum.

Recall the expressions of the $\mathcal{L}$-risk. We denote $\mathcal{L}(\theta^*) = \mathbb{E}_{D\sim\mathcal{D}}[\mathcal{L}(\theta^*, D)]$ as the optimal value. For the sampled distribution, we have

$$\mathbb{E}[\mathcal{L}_q(\theta^t, D)] = \mathbb{E}\left[\sum_{j=1}^{s} \frac{\hat{\gamma}_j}{q} \mathcal{L}_{i_{(j)}}(\theta^t)\right] = \mathbb{E}[\mathbb{L}_s[F_{n,s}(\cdot;\theta^t)]], \tag{35}$$

and for the uniform distribution, we define

$$\mathbb{E}[\mathcal{L}_u(\theta^t, D)] = \mathbb{E}\left[\sum_{j=1}^{s} \frac{1}{s} \mathcal{L}_{i_{(j)}}(\theta^t)\right] = \mathbb{E}[\mathbb{L}_u[F_{n,s}(\cdot;\theta^t)]]. \tag{36}$$

Moreover, the full-batch loss satisfies

$$\mathcal{L}(\theta^t, D) = \mathcal{L}_u(\theta^t, D) = \mathbb{L}_u[F_n(\cdot;\theta^t)]. \tag{37}$$

Here, the distributions s and u correspond to the sampling distribution of γ in $\mathcal{L}_q(\theta^t, D)$ at step t and the uniform distribution, respectively. The expectation is taken over the minibatch $\{i_1, \ldots, i_s\}$. Therefore, we establish the generalization error over the set $\Theta := \{\theta^i\}_{i=1}^T$.

$$\begin{aligned}
&\mathcal{L}(\theta^*) - \mathbb{E}[\mathcal{L}_q(\theta^t, D)] \\
\leq &\sup_{\theta\in\Theta} \mathcal{L}(\theta^*) - \mathbb{E}[\mathcal{L}_q(\theta^t, D)] \\
= &\sup_{\theta\in\Theta} \mathcal{L}(\theta^*) - \mathbb{E}[\mathbb{L}_s[F_{n,s}]] - \mathbb{L}_s[F_n] + \mathbb{L}_s[F_n] - \mathbb{E}[\mathbb{L}_u[F_{n,s}]] + \mathbb{E}[\mathbb{L}_u[F_{n,s}]] - \mathbb{L}_u[F_n] + \mathbb{L}_u[F_n] \\
= &\sup_{\theta\in\Theta} \mathcal{L}(\theta^*) - \mathbb{E}[\mathbb{L}_u[F_{n,s}]] + \mathbb{L}_u[F_n] - \mathbb{L}_s[F_n] \\
&- \Big(\mathbb{E}[\mathbb{L}_s[F_{n,s}]] - \mathbb{L}_s[F_n] - \big(\mathbb{E}[\mathbb{L}_u[F_{n,s}]] - \mathbb{L}_u[F_n]\big)\Big) \\
\leq &\sup_{\theta\in\Theta} \mathcal{L}(\theta^*) - \mathbb{E}[\mathbb{L}_u[F_{n,s}]] + \sup_{\theta\in\Theta} \big(\mathbb{L}_u[F_n] - \mathbb{L}_s[F_n]\big) \\
&+ \sup_{\theta\in\Theta} \Big(- \mathbb{E}[\mathbb{L}_s[F_{n,s}]] + \mathbb{L}_s[F_n] + \mathbb{E}[\mathbb{L}_u[F_{n,s}]] - \mathbb{L}_u[F_n]\Big) \\
\leq &\inf_{\theta\in\Theta} \big|\mathbb{E}[\mathbb{L}_u[F_{n,s}]] - \mathcal{L}(\theta^*)\big| + \sup_{\theta\in\Theta} \big(\mathbb{L}_u[F_n] - \mathbb{L}_s[F_n]\big) \\
&+ \sup_{\theta\in\Theta} \Big|\mathbb{E}[\mathbb{L}_s[F_{n,s}]] - \mathbb{L}_s[F_n] - \big(\mathbb{E}[\mathbb{L}_u[F_{n,s}]] - \mathbb{L}_u[F_n]\big)\Big| \\
\leq &\frac{\eta_{\max}\Big(\|\theta^1 - \theta^*\|_2^2 + G^2 \sum_{t=1}^T (\eta^t)^2\Big)}{2\eta_{\min} \sum_{t=1}^T \eta^t} \\
&+ \sup_{\theta\in\Theta} \big(\mathbb{L}_u[F_n] - \mathbb{L}_s[F_n]\big) + \sup_{\theta\in\Theta} \|s - u\|_\infty \, \mathbb{E}[\|F_{n,m}^{-1} - F_n^{-1}\|_1] \\
\leq &\underbrace{\frac{\eta_{max}(\|\theta^1 - \theta^*\|_2^2 + G^2 \sum_{t=1}^T (\eta^t)^2)}{2\eta_{min} \sum_{t=1}^T \eta^t}}_{\text{unbiased part}} \underbrace{-\mathcal{Q}_n(\theta^t; s, q) + \sqrt{2} C_s B \sqrt{\frac{n-s}{s(n-1)}}}_{\text{biased part}},
\end{aligned} \tag{38}$$

where the third inequality follows the Theorem 3 and the fourth inequality follows Lemma 14 in (Mehta et al., 2023). We denote $C_s = \sup_{t\in(0,1)} |s(t) - u(t)|$, $B = \inf_{\theta\in[1,T]} \max_{i\in[1,n]} |\mathcal{L}_i(\theta, z_i)| < \infty$, and $\mathcal{Q}_n(\theta; s, q) := \inf_{\theta\in\Theta} \sum_{i=1}^n (\frac{r_i(\theta,D)}{q} - \frac{1}{n}) \mathcal{L}_i(\theta, z)$.

Moreover, if we use the weighted average $\bar{\theta}^T = \frac{1}{\sum_{t=1}^T \eta^t} \sum_{t=1}^T \eta^t \theta^t$ as output of **`OrderDP`**, the dependence on $\eta_{\max}$ and $\eta_{\min}$ can be removed and it holds that:

$$
\begin{aligned}
&\mathcal{L}(\theta^*) - \mathbb{E}[\mathcal{L}_q(\bar{\theta}^T, D)] \\
=&\mathcal{L}(\theta^*) - \mathbb{E}[\mathbb{L}_s[F_{n,s}]] - \mathbb{L}_s[F_n] + \mathbb{L}_s[F_n] - \mathbb{E}[\mathbb{L}_u[F_{n,s}]] + \mathbb{E}[\mathbb{L}_u[F_{n,s}]] - \mathbb{L}_u[F_n] + \mathbb{L}_u[F_n] \\
=&\mathcal{L}(\theta^*) - \mathbb{E}[\mathbb{L}_u[F_{n,s}]] + \mathbb{L}_u[F_n] - \mathbb{L}_s[F_n] - [\mathbb{E}[\mathbb{L}_s[F_{n,s}]] - \mathbb{L}_s[F_n] - (\mathbb{E}[\mathbb{L}_u[F_{n,s}]] - \mathbb{L}_u[F_n])] \\
\leq& \left|\mathbb{E}[\mathbb{L}_u[F_{n,s}]] - \mathcal{L}(\theta^*)\right| + \left(\mathbb{L}_u[F_n] - \mathbb{L}_s[F_n]\right) \\
&+ \left|\mathbb{E}[\mathbb{L}_s[F_{n,s}]] - \mathbb{L}_s[F_n] - \left(\mathbb{E}[\mathbb{L}_u[F_{n,s}]] - \mathbb{L}_u[F_n]\right)\right| \\
\leq& \left|\mathbb{E}[\mathbb{L}_u[F_{n,s}]] - \mathcal{L}(\theta^*)\right| + \sup_{\theta\in\Theta}\left(\mathbb{L}_u[F_n] - \mathbb{L}_s[F_n]\right) \\
&+ \left|\mathbb{E}[\mathbb{L}_s[F_{n,s}]] - \mathbb{L}_s[F_n] - \left(\mathbb{E}[\mathbb{L}_u[F_{n,s}]] - \mathbb{L}_u[F_n]\right)\right| \\
\leq& \underbrace{\frac{(\|\theta^1 - \theta^*\|_2^2 + G^2\sum_{t=1}^T (\eta^t)^2)}{2\sum_{t=1}^T \eta^t}}_{\text{unbiased part}} \underbrace{-\mathcal{Q}_n(\theta^t; s, q) + \sqrt{2}C_s B\sqrt{\frac{n-s}{s(n-1)}}}_{\text{biased part}}.
\end{aligned}
\tag{39}
$$

where the last inequality follows from (18) and other terms remain unchanged.

□

B Related Works

Static Data Pruning. Static pruning techniques aim to pre-select a compact subset of the training data that can approximate the utility of the full dataset. A wide range of criteria have been proposed for this purpose. Diversity-based methods such as Contextual Diversity (CD) (Agarwal et al., 2020), Herding (Welling, 2009), and k-Center (Sener & Savarese, 2018) remove redundant samples by ensuring broad feature-space coverage. Difficulty-based strategies including Cal (Margatina et al., 2021) and Deepfool (Ducoffe & Precioso, 2018) prioritize hard-to-learn examples near decision boundaries. Error- and gradient-driven approaches such as GraNd and EL2N (Paul et al., 2021) and MOSO (Tan et al., 2023) instead exploit training dynamics or loss sensitivity. In parallel, uncertainty-based sampling (Coleman et al., 2019), influence-function analysis (Koh & Liang, 2017), and gradient matching approaches like GradMatch (Killamsetty et al., 2021b;a) provide alternative means of quantifying informativeness. More principled frameworks include bilevel optimization (Killamsetty et al., 2021b) and submodular subset selection (Iyer et al., 2021), where algorithms such as FL and Graph Cut (GC) (Iyer et al., 2021) explicitly balance coverage and information gain. Early computer vision work such as (Huh et al., 2016) also emphasized the importance of dataset diversity for transferable representations. While effective in specific cases, static approaches often require costly pre-computation, and their heuristics may not generalize well across architectures or datasets, particularly at ImageNet scale.

Dynamic Data Pruning. Dynamic methods instead make pruning decisions adaptively during training by leveraging information from the evolving model state. Early efforts such as ActiveBias (Chang et al., 2017) adjusted sampling probabilities based on prediction confidence, while forgetting-based measures (Toneva et al., 2018) revealed that unstable or frequently forgotten examples often provide valuable signal. Raju et al. (Raju et al., 2021) introduced exploration-based policies such as ϵ-greedy and UCB, where uncertainty estimates guide the retention of high-value samples. Recent work has also examined improving random sampling policies themselves: Okanovic et al. (Okanovic et al., 2024) showed that repeated random sampling can significantly reduce time-to-accuracy, offering a complementary perspective to loss-based dynamic pruning approaches such as InfoBatch and **OrderDP**. More recently, InfoBatch (Qin et al., 2024) proposed an unbiased gradient estimator, showing that loss-based pruning can accelerate training without compromising accuracy on benchmarks like CIFAR-10/100 and ImageNet-1K. Building on this line of research, Yang et al. (Yang et al., 2022) and Sorscher et al. (Sorscher et al., 2022) extended dynamic pruning principles to large-scale pretraining, while He et al. (He et al., 2024) incorporated dynamically updated uncertainty estimates. In particular, Sorscher et al. (Sorscher et al., 2022) demonstrated that the optimal choice between hard and easy samples can depend on dataset scale, an observation that is complementary to the top-q strategy analyzed in this work. Despite these advances, dynamic methods still face challenges: the

achievable "lossless" pruning ratio on new datasets is unpredictable, sorting operations can become expensive at scale, and empirical instability often emerges under aggressive pruning ratios. Ayed and Hayou (Ayed & Hayou, 2023) further analyze the fundamental bias of score-based pruning and show that reweighting can recover unbiasedness with respect to the original loss. Their perspective is complementary to ours: while they study limitations under L, we provide guarantees for a ranking-induced surrogate L_q whose gap to L is explicitly controlled.

Cross-domain Data Selection and Pruning. Beyond computer vision, pruning and selection strategies have been expanded to other domains such as NLP and speech. In speech recognition, unsupervised data selection has been explored through discrete speech units (Lu et al., 2022). For large-scale NLP pretraining, several studies investigate pruning and mixture optimization to accelerate convergence. (Marion et al., 2023) explored pruning strategies for pretraining corpora, while (Xie et al., 2023) introduced DoReMi, a framework that dynamically optimizes data mixtures for faster language model pretraining. Instruction tuning has further motivated task-specific pruning, exemplified by (Cao et al., 2023), who proposed instruction mining to select relevant subsets for downstream tasks. These works highlight that pruning is not limited to vision but constitutes a broader principle of efficient data utilization across modalities.

C Experimental Infrastructures

Software infrastructures. All experiments are implemented in Python 3.12.4 using PyTorch 2.3.1 with CUDA 11.8 support. Key libraries include NumPy 1.26.4, pandas 2.2.3, torchvision 0.18.1, matplotlib 3.10.1, and scikit-learn 1.6.1 for data processing and analysis. We also employ `accelerate` 1.6.0 for multi-GPU training and `tqdm` 4.67.1 for progress visualization.

Hardware infrastructures. We conduct all experiments on a computer server with 2 NVIDIA L40 GPUs (with 48GB memory each), a single Intel Xeon Gold 6448Y CPU (32 physical cores), and 944 GiB of system RAM.

D Additional Empirical Results

D.1 Experimental Setup Details

We provide software/hardware infrastructures in Appendix C; here we detail dataset-specific training setups throughout this paper.

CIFAR-10/100: The CIFAR-10/100 experiment with ResNet-18 can be reproduced with SGD using a maximum learning rate of 0.2 for the OneCycle scheduler under a batch size of 128. For experiments with ResNet-50, SGD is used with a maximum learning rate of 0.03 and batch size of 128 for baseline, InfoBatch, and **`OrderDP`**.

ImageNet-1K: The tests are implemented based on `Pytorch/examples`. The LARS optimizer and a maximum learning rate of 6.4 / 1.98 are used for batch size 1024 on ImageNet-1K experiments with ResNet-50/18.

D.2 Validation of Theoretical Properties

Both Figure 5 and Figure 6 illustrate theoretical properties derived in Appendix A. Figure 5 empirically validates Proposition 2 by fixing $(s, q) = (100, 30)$ and increasing n, showing how $n\,\gamma_j$ converges to $\gamma(z)$ as n, s, and q increase. Figure 6 illustrates Theorem 4 by comparing γ_j to uniform sampling for different q (fix $(n, s) = (200, 100)$ and increase $q \to 100$), highlighting that the gap between the two distributions vanishes (i.e., $\mathcal{Q}_n(\theta; s, q)$ and C_s approach 0) as q increases.

In addition to the above validations, we further examine the normalization behavior of the weights $\{\gamma_j\}$ derived in Eq. (10). Although providing a fully symbolic proof for the combinatorial form of γ_j is algebraically involved, the construction of Algorithm 1 implies that exactly q samples are selected at each iteration, suggesting that $\sum_{j=1}^{n} \gamma_j$ should be close to q, and therefore $\sum_{j=1}^{n} \gamma_j / q \approx 1$.

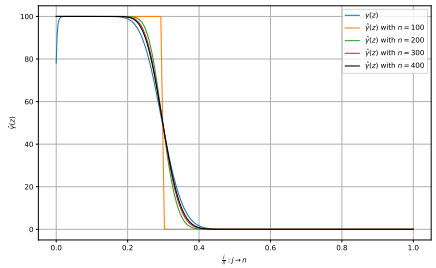


Figure 5: Empirical weight decay curves $n\,\gamma_j$ versus normalized index j/n, demonstrating convergence to the limiting density $\gamma(z)$ and the smoothing of the "cliff" as n increases.

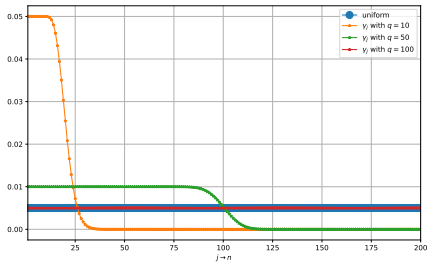


Figure 6: Comparison of the sampling weights γ_j against uniform sampling for different exploitation sizes q, illustrating the deviation captured by the bias term C_s in Theorem 4.

Figure 7 plots the normalized distributions γ_j/q for different values of q (with $(n, s) = (400, 100)$). As q increases, the curves gradually flatten and approach the uniform distribution, consistent with Theorem 4.

Figure 8 further shows the empirical values of $\sum_{j=1}^{n} \gamma_j/q$ across $q \in \{10, 20, \ldots, 100\}$, all of which lie extremely close to 1 (within floating-point error). This provides strong numerical evidence that the weights induced by Algorithm 1 are properly normalized in practice.

Proof. We also provide proof of the claim $\sum_{j=1}^{n} \gamma_j/q = 1$ via Mathematical Induction.

For any n, s, when $q = 1$, we can show exactly that

$$\sum_j \frac{\gamma_j}{q} = \sum_j \frac{\binom{n-1}{s-1}}{\binom{n}{s}} = 1.$$

Suppose $\sum_j \gamma_j/q = 1$ for any $q = m$ where $m \in \mathbb{N}$ and $1 \le m \le s-1$, we show $\sum_j \gamma_j/q = 1$ for $q = m+1$. The case $q = m$ can be rewritten as

$$\sum_j \frac{\gamma_j}{q} = \frac{1}{m} \sum_j \sum_{l=\max\{1,s-n+j\}}^{\min\{m,j\}} \frac{\binom{j-1}{l-1}\binom{n-j}{s-l}}{\binom{n}{s}} = 1.$$

Thus for the case $q = m+1$, we have

$$\begin{aligned}
\sum_j \frac{\gamma_j}{q} &= \frac{1}{m+1} \sum_j \sum_{l=\max\{1,s-n+j\}}^{\min\{m+1,j\}} \frac{\binom{j-1}{l-1}\binom{n-j}{s-l}}{\binom{n}{s}} \\
&= \frac{1}{m+1} \sum_j \left(\sum_{l=\max\{1,s-n+j\}}^{\min\{m,j\}} \frac{\binom{j-1}{l-1}\binom{n-j}{s-l}}{\binom{n}{s}} + \frac{\binom{j-1}{m}\binom{n-j}{s-m-1}}{\binom{n}{s}} \right) \\
&= \frac{1}{m+1} \left(m + \sum_{j=m+1}^{n-s+m+1} \frac{\binom{j-1}{m}\binom{n-j}{s-m-1}}{\binom{n}{s}} \right)
\end{aligned}$$

where the last equality follows that $\binom{j-1}{m}\binom{n-j}{s-m-1} > 0$ for $m+1 \le j \le n-s+m+1$ else 0. The remain proof is to show $\sum_{j=m+1}^{n-s+m+1} \frac{\binom{j-1}{m}\binom{n-j}{s-m-1}}{\binom{n}{s}} = 1$.

Consider selecting a subset of size s from the set $\{1, 2, \ldots, n\}$. Arrange the elements of the subset in increasing order: $a_1 \ge a_2 \ge \cdots \ge a_s$ Then a_{m+1} is the $(m+1)$-th largest element. Let $j = m+1$, i.e., the $(m+1)$-th largest element is at position j.

To construct such a subset, the following conditions must be satisfied:

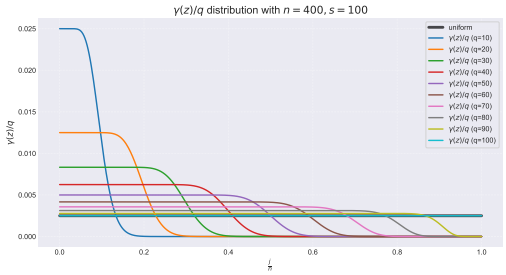


Figure 7: Normalized distributions γ_j/q for different q under $(n, s) = (400, 100)$. As q increases, the curves flatten and approach the uniform distribution.

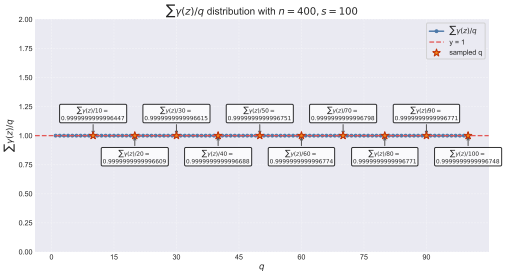


Figure 8: Empirical normalization of $\sum_{j=1}^{n} \gamma_j/q$. Across all $q \in \{10, \ldots, 100\}$, the values remain extremely close to 1.

- Choose m elements from the first $j-1$ elements (i.e., $a_1, \ldots, a_m$), which can be done in $\binom{j-1}{m}$ choices.
- Choose $s-m-1$ elements from the remaining $n-j$ elements (i.e., $a_{m+2}, \ldots, a_s$), which can be done in $\binom{n-j}{s-m-1}$ choices.

Thus, the number of subsets where the $m+1$-th largest element is exactly at position j is: $\binom{j-1}{m}\binom{n-j}{s-m-1}$

Summing over j from $m+1$ to $n-s+m+1$ (since j must be at least $m+1$ and at most $n-s+m+1$ to ensure enough elements remain), we obtain the total number of subsets of size s: $\sum_{j=m+1}^{n-s+m+1} \binom{j-1}{m}\binom{n-j}{s-m-1} = \binom{n}{s}$,

Therefore, the original claim holds for all q: $\sum_{j=1}^{n} \gamma_j/q = 1$. □

D.3 Validation of Convergence Assumptions

To further support the validity of Theorem 3, we analyze whether the selected coreset stabilizes during training. Although the selection depends on sampling scores H, which may vary across epochs, our analysis shows that the coreset indeed becomes stable in later stages of training.

Coreset Dynamics. Our analysis does not assume a fixed coreset. Instead, OrderDP naturally determines the coreset through the pruning strategy (captured by γ_j in Proposition 2), where the sample $z_j \in$ coreset follows an approximate Beta-distribution.

Theoretical Parallel to SGD. The applicability of Theorem 3 is analogous to SGD's convergence guarantees: (i) SGD converges by deterministic batches per epoch (the sample $z_j \in$ selected follows uniform sampling); (ii) OrderDP achieves convergence after the coreset stabilizes (via Beta-distributed sampling).

To empirically verify this stabilization, we measure the *Jaccard Similarity* between the coreset at the current epoch and that from the immediately preceding epoch, defined as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{40}$$

A higher similarity indicates that the selected set of samples remains consistent across epochs. Table 8 shows that OrderDP consistently achieves higher coreset stability than InfoBatch, confirming the practical applicability of Theorem 3.

D.4 Sample Coverage under Different Pruning Ratios

To further examine the exploration behavior of **OrderDP**, we track for each training sample the number of times it is selected into the update set throughout the entire training process. Figures 9, 10,

Table 8: Jaccard Similarity between consecutive checkpoints on CIFAR-100 with ResNet-18. InfoBatch and OrderDP are trained for 200 epochs (checkpoints at 20%, 40%, 60%, 80%, and final 100%), with learning rate = 0.03 and batch size = 128. Each setting is repeated 5 times, and mean ± std are reported. Higher values indicate greater stability of the coreset.

Prune Ratio	Method	0–20%	20–40%	40–60%	60–80%	80–100%	100%
40%	InfoBatch	**0.583±0.050**	0.496±0.010	0.479±0.003	0.481±0.006	0.512±0.010	0.522±0.007
	OrderDP	0.645±0.057	**0.692±0.023**	**0.713±0.008**	**0.743±0.023**	**0.757±0.034**	**0.767±0.008**
70%	InfoBatch	**0.593±0.012**	0.532±0.024	0.459±0.019	0.403±0.018	0.455±0.036	0.512±0.012
	OrderDP	0.552±0.049	**0.592±0.016**	**0.647±0.020**	**0.661±0.018**	**0.678±0.023**	**0.704±0.090**

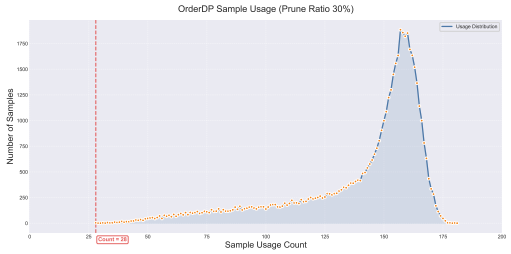


Figure 9: Sample usage count distribution (30% prune ratio).

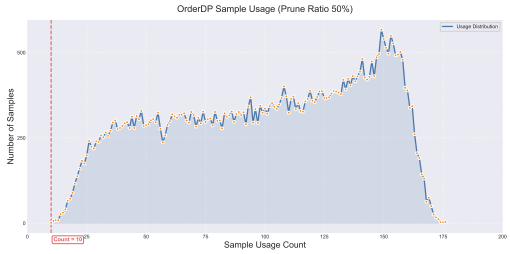


Figure 10: Sample usage count distribution (50% prune ratio).

and 11 show the empirical distributions of sample usage counts on CIFAR-10 under prune ratios of 30%, 50%, and 99%, respectively.

Across all pruning ratios, we observe that *no sample has zero usage count*: every example is selected at least once during training. Under practical pruning settings (e.g., 30%–50%), most samples fall within a reasonably concentrated range of usage counts, indicating that **OrderDP** does not permanently discard any data point but instead explores the entire dataset with a frequency controlled by (s, q).

These empirical findings are fully consistent with our theoretical analysis of coverage and directly support our response to reviewer questions regarding whether OrderDP eventually sees the entire dataset.

D.5 Time-to-Accuracy Curves

To complement the wall-clock results in Figure 3 and to directly address the reviewer's suggestion on evaluating time-to-accuracy, we report curves showing the relationship between training time and accuracy. These curves provide a practical view of how fast different methods reach comparable accuracy levels in real training scenarios.

Figures 12 and 13 present the Time-to-Accuracy curves on CIFAR-10 using ResNet-18 under prune ratios of 40% and 70%. Consistent with our findings throughout the paper, **OrderDP** achieves faster accuracy improvement and maintains stable convergence compared with both InfoBatch and Random, especially under higher pruning levels.

D.6 Stability Analysis

We further include a stability study of dynamic pruning methods under multiple independent runs. In this analysis, we evaluate CIFAR-100 with ResNet-18 at a 70% real pruning ratio across 10 runs. InfoBatch often exhibits large mid-training gradient oscillations and occasional convergence failures, while OrderDP consistently converges smoothly in every trial. Moreover, InfoBatch relies on late-stage full-data "annealing" to stabilize training, whereas OrderDP maintains exact pruning control via (s, q) without requiring such rescue. This highlights the robustness and practicality of our approach. Under aggressive pruning, InfoBatch's rescaling further causes severe fluctuations in both gradient and accuracy.

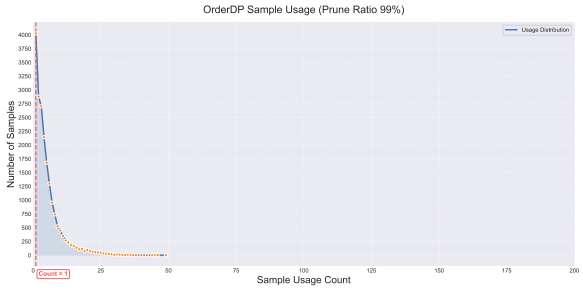


Figure 11: Sample usage count distribution under a 99% prune ratio. Even under extreme pruning, every sample is selected at least once.

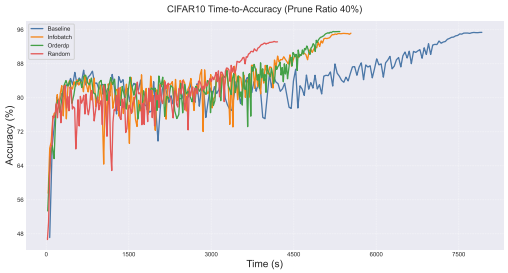


Figure 12: Time-to-Accuracy on CIFAR-10 at 40% prune ratio.

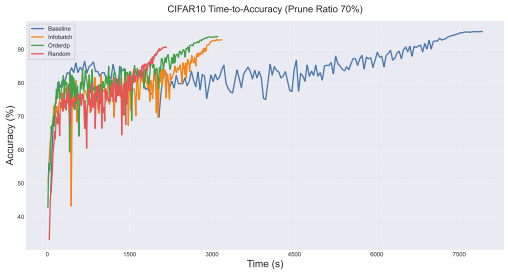


Figure 13: Time-to-Accuracy on CIFAR-10 at 70% prune ratio.

Detailed results are summarized in Table 9 (real prune ratio ≈ 0.61, InfoBatch nominal prune ratio ≈ 0.99; 200 epochs; learning rate = 0.03; batch size = 128; averaged over 10 runs).

E Extended Analysis of Gradient Bias

E.1 Mechanisms and Toy Example

This subsection provides additional clarification on how OrderDP eliminates biased gradient estimation. The method addresses the issue through three main mechanisms. **Uniform exploration.** Before pruning, s points are randomly sampled so that every sample, regardless of gradient magnitude, has equal probability of entering the coreset. This ensures the gradient distribution is much closer to that of full-data SGD compared with InfoBatch's biased rescaling strategy. **Unbiased surrogate loss.** Instead of pruning the original loss directly, a new loss $\mathcal{L}_q$ is constructed with closed-form weights γ_j derived from a two-stage sampler. Theorem 1 guarantees each update is an unbiased estimator of $\nabla\mathcal{L}_q(\theta)$, while Theorems 2, 3, and 4 provide convergence and generalization guarantees, avoiding the need for late-stage full-data annealing. **Exact pruning control.** By explicitly setting (s, q), any target prune ratio (e.g., 70%, 80%, 90%) can be realized precisely, unlike InfoBatch's mean-threshold scheme, which fluctuates around $\sim 77\%$.

To make the difference clear, we present a toy comparison under an 80% prune ratio with 5 samples of gradients $g = \{1, 2, 3, 4, 5\}$. In **InfoBatch (mean-threshold + rescale)**, samples 3, 4, 5 are always kept, while 1 and 2 are included with probability 0.2, and their gradients rescaled by a factor $1/(1 - 0.8) = 5$. This leads to four possible outcomes summarized in Table 10.

From Table 10, the expected gradient estimate is 4.336 with prune ratio < 0.8, indicating bias.

In contrast, for **OrderDP (5 samples, 80% prune)**, we randomly sample a batch with size $s \in \{1, \ldots, 5\}$ and select the top-1 element. The detailed probability calculation for each index is as follows:

Table 9: Stability comparison on CIFAR-100 with ResNet-50 across 10 runs under varying **training progress** (percentage of epochs). Reported are **Accuracy (% ± Std)** and **Gradient Std**. OrderDP shows smoother convergence and eliminates instability observed in InfoBatch.

Method	Metric	0–30%	30–50%	50–70%	70–100%	Final
ϵ-Greedy	Acc ± Std	48.01 ± 4.60	49.77 ± 2.54	52.61 ± 1.42	66.24 ± 1.23	74.77 ± 0.30
	Grad ± Std	3.08 ± 1.19	3.49 ± 0.62	2.78 ± 0.57	2.05 ± 0.45	1.53 ± 0.37
UCB	Acc ± Std	49.70 ± 4.80	50.66 ± 2.31	54.34 ± 1.82	67.97 ± 1.12	75.41 ± 0.40
	Grad ± Std	4.08 ± 1.69	3.69 ± 1.02	2.99 ± 0.27	2.35 ± 0.38	1.33 ± 0.14
InfoBatch	Acc ± Std	45.74 ± 3.56	52.08 ± 2.55	47.72 ± 3.63	68.92 ± 3.01	76.72 ± 0.70
	Grad ± Std	7.35 ± 1.78	5.88 ± 1.24	4.35 ± 9.48	3.56 ± 4.55	2.89 ± 1.67
OrderDP	Acc ± Std	48.00 ± 3.23	56.00 ± 2.01	61.00 ± 1.34	72.00 ± 0.56	78.32 ± 0.20
	Grad ± Std	4.08 ± 1.19	3.49 ± 0.62	2.78 ± 0.47	2.05 ± 0.55	1.03 ± 0.33
Whole Dataset	Acc ± Std	56.58 ± 1.56	62.36 ± 0.76	66.84 ± 0.52	72.24 ± 0.40	80.60 ± 0.20
	Grad ± Std	3.88 ± 0.79	3.09 ± 0.41	2.45 ± 0.42	1.93 ± 0.55	0.88 ± 0.20

Table 10: Toy example of InfoBatch under an 80% prune ratio with 5 gradients. The table shows the kept set, probability of selection, rescaled gradients, average gradient, and prune rate.

Kept set	Probability	Gradients	Avg. grad.	Prune rate
$\{3,4,5\}$	0.64	3, 4, 5	4.0	0.60
$\{1,3,4,5\}$	0.16	$5\cdot 1,3,4,5 = 5,3,4,5$	4.25	0.20
$\{2,3,4,5\}$	0.16	$5\cdot 2,3,4,5 = 10,3,4,5$	5.5	0.20
$\{1,2,3,4,5\}$	0.04	$5\cdot 1, 5\cdot 2,3,4,5 = 5,10,3,4,5$	5.4	0.00

$$
\begin{aligned}
\{1\}:&\quad \tfrac{1}{5}\cdot\tfrac{1}{5}\,(s=1) = \tfrac{1}{25},\\
\{2\}:&\quad \tfrac{1}{5}\cdot\tfrac{1}{5}\,(s=1) + \tfrac{1}{10}\cdot\tfrac{1}{5}\,(s=2) = \tfrac{3}{50},\\
\{3\}:&\quad \tfrac{1}{5}\cdot\tfrac{1}{5}\,(s=1) + \tfrac{3}{10}\cdot\tfrac{1}{5}\,(s=2) + \tfrac{1}{10}\cdot\tfrac{1}{5}\,(s=3) = \tfrac{3}{25},\\
\{4\}:&\quad \tfrac{1}{5}\cdot\tfrac{1}{5}\,(s=1) + \tfrac{3}{10}\cdot\tfrac{1}{5}\,(s=2) + \tfrac{3}{10}\cdot\tfrac{1}{5}\,(s=3) + \tfrac{1}{10}\cdot\tfrac{1}{5}\,(s=4) = \tfrac{9}{50},\\
\{5\}:&\quad \tfrac{1}{5}\cdot\tfrac{1}{5}\,(s=1) + \tfrac{2}{5}\cdot\tfrac{1}{5}\,(s=2) + \tfrac{3}{5}\cdot\tfrac{1}{5}\,(s=3) + \tfrac{4}{5}\cdot\tfrac{1}{5}\,(s=4) + \tfrac{1}{5}\,(s=5) = \tfrac{3}{5}.
\end{aligned}
$$

The expected gradient estimate under this distribution is 4.06 with prune ratio exactly 0.8. Therefore, OrderDP not only maintains the target pruning ratio precisely but also achieves gradient estimates closer to the true full gradient ($= 3$), effectively eliminating the bias observed in InfoBatch.

E.2 Gradient Direction Analysis

In addition to gradient magnitude analysis, we also examine gradient directions by measuring the *cosine similarity* between the gradients computed with each pruning method and the full-data gradient at matched checkpoints (same model weights). We train on CIFAR-100 with ResNet-18 for 200 epochs, evaluate at 20%, 40%, 60%, 80%, and 100% of training progress, using a learning rate of 0.03 and batch size of 128. Each setting is repeated 5 times, and we report mean ± std. Results under pruning ratios 40% and 70% are summarized in Table 11.

These results demonstrate that **OrderDP's gradients align more closely with full-data gradients**, particularly at high pruning ratios, thereby reducing directional bias compared with InfoBatch.

F Limitations and Future Work

While **`OrderDP`** excels on moderate-scale vision benchmarks, its performance on very large architectures, streaming inference scenarios, and heterogeneous hardware platforms, as well as in self-supervised or multi-modal settings, remains to be explored. In future work, we will extend **`OrderDP`** to adaptive pruning schedules, investigate its integration with transformer and graph models, and study its behavior under distribution shift and noisy labels.

Table 11: Cosine similarity between pruned and full-data gradients on CIFAR-100 (ResNet-18) under pruning ratios 40% and 70%. Each experiment is repeated 5 times, and mean ± std are reported. Higher values indicate stronger alignment with full-data gradients.

Prune Ratio	Method	0–20%	20–40%	40–60%	60–80%	80–100%	100%
40%	InfoBatch	0.915±0.044	0.940±0.008	0.916±0.007	0.904±0.011	0.897±0.008	0.895±0.009
	OrderDP	**0.943±0.035**	**0.951±0.007**	**0.934±0.008**	**0.921±0.009**	**0.908±0.014**	**0.906±0.011**
70%	InfoBatch	0.825±0.037	0.807±0.027	0.763±0.021	0.758±0.023	0.743±0.026	0.716±0.029
	OrderDP	**0.853±0.048**	**0.896±0.018**	**0.901±0.015**	**0.893±0.014**	**0.868±0.021**	**0.844±0.018**

Another promising direction is to develop noise-robust variants of **`OrderDP`**. Although the current work focuses on the top-q strategy, the surrogate-loss framework introduced in this paper is more general and can naturally incorporate min-q selection or mixed hard/easy sampling schemes by modifying the weight structure $\{\gamma_j\}$. Such extensions may help suppress extreme outliers, improve stability under label noise, and adapt the pruning strategy across different stages of training (e.g., hard-sample emphasis in early stages and easy-sample regularization in later stages). We plan to further explore these variants and evaluate their performance in noisy or adversarial settings.

G The Use of Large Language Models (LLMs)

Large language models (LLMs) were used solely for linguistic refinement and editing of the manuscript. All scientific ideas, methodological contributions, and experimental results are entirely conceived, implemented, and validated by the authors.