EVOLVED SAMPLING: EFFICIENT TRAINING THROUGH LOSS-DYNAMICS-AWARE DATA SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Data selection aims to accelerate training while preserving model performance. To achieve this, a common approach is to identify informative data samples with significant contributions to the training. In this work, we propose **Evolved Sampling** (**ES**), a simple yet effective framework for *dynamic* sampling along the training process. This method conducts batch-level data selection based on *predicted loss dynamics*, involving not only the current state of training dynamics but also accounting for dynamical evolutions in the form of loss variations along the training. Since ES leverages lightweight *re-weighting computations only regarding losses*, it significantly reduces the back propagation time with maintained model performance. Due to its conciseness, ES is also readily extensible to incorporate set-level data selection (to form ES with pruning, **ESWP**) for further accelerations. As a plug-and-play framework, ES(WP) is mathematically proved to converge, and consistently achieves lossless training accelerations across multiple pre-training and post-training tasks with various scales, saving up to 43.7% wall-clock time. Our results motivate further investigations on the data efficiency aspect of modern large-scale machine learning and LLMs.

1 INTRODUCTION

Foundation models have advanced a wide range of applications in language (Radford et al., 2019; Xu et al., 2026; Team et al., 2026) and vision (Rombach et al., 2022; Bai et al., 2025). Training these models requires web-scale datasets and substantial computation and memory resources, motivating efficient approaches to large-scale machine learning. *Data-efficient* learning handles this challenge by assessing the importance of training samples and allocating computation. By prioritizing informative samples, data selection aims to reduce training costs while preserving model performance.

Data selection methods can be broadly divided into offline and online approaches. Offline methods select a subset before training, often using pre-computed representations or quality scores (Xie et al., 2023b; Sachdeva et al., 2024). Such selection can incur substantial pre-processing costs, and a fixed subset cannot adapt to changes in sample importance during training. Online methods instead update sample importance as training proceeds. They mainly operate at two complementary levels: *set-level* selection prunes the training set at the beginning of each epoch (Qin et al., 2024; Thao Nguyen et al., 2023), whereas *batch-level* selection chooses a subset of each batch for back propagation (Mindermann et al., 2022). A shared weighting scheme for both levels would allow these sources of computational savings to be combined.

Beyond the level at which selection is applied, its effectiveness depends on the metric used to assign sampling weights/scores. Current losses provide inexpensive scores, but do not distinguish samples with similar losses and different learning trajectories (Kumar et al., 2023). Gradient-based criteria can capture how individual samples contribute to the overall update but incur additional computation and memory costs (Gu et al., 2025). This motivates a sampling rule that properly evaluates sample importance along training dynamics from readily available loss values.

In this work, we propose **Evolved Sampling** (**ES**), a dynamic data selection framework that computes sampling weights from **loss predictions** of individual samples. This re-weighting strategy not only considers the current state of training dynamics (current losses), but also accounts for *dynamical evolutions* (loss variations) along the training. Our analysis suggests that ES further encodes correlations between individual gradients and the overall update direction, despite that there are only lightweight

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

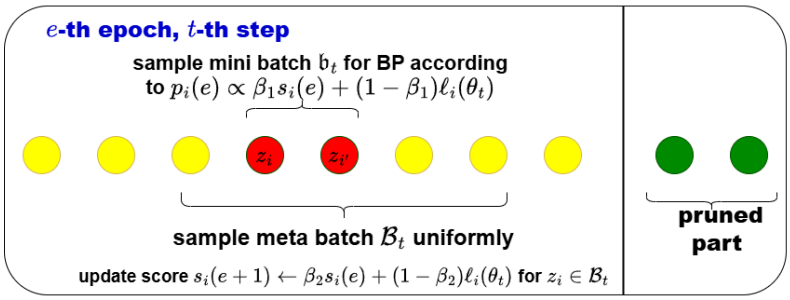


Figure 1: Overview of ES and ESWP. ES uniformly samples a meta-batch $\mathcal{B}_t$ and then samples a mini-batch $\mathfrak{b}_t$ (red) for back propagation (BP), with sampling weights computed from current losses and historical scores. Historical scores are updated for all samples in $\mathcal{B}_t$ using an exponential moving average. ESWP additionally uses the same weighting scheme to retain a subset of the dataset at the beginning of each epoch; green samples are pruned for that epoch.

loss values used for weight calculations of ES, and consequently reduces backward computations by selecting subsets of batches during back propagations. Reusing the same weights at the beginning of each epoch for dataset pruning yields **Evolved Sampling with Pruning** (**ESWP**).

Our contributions are summarized as follows:

- We introduce ES and its pruning extension ESWP, a unified framework for online data selection at the batch and set levels using only lightweight loss-based weight updates. We show that the re-weighting rule incorporates dynamical evolutions and implicitly captures per-sample gradient correlations along the training process.
- We establish a convergence bound for ES-weighted SGD with a constant learning rate and characterize when data selection tightens this bound. Under additional boundedness assumptions, we also derive a generalization bound relative to the weighted empirical loss.
- We evaluate ES(WP) across vision and language tasks with various scales. For vision tasks, we study ResNet and MAE pre-training (He et al., 2016; 2022), and ViT-Large fine-tuning (Dosovitskiy et al., 2021). For language tasks, we investigate GPT-2-Small/XL pre-training (Radford et al., 2019) and fine-tuning of Qwen2.5-Math-1.5B (Yang et al., 2024) and Qwen3.5-9B-Base (Qwen Team, 2026). Motivated by recent interests in looped Transformers (Zhu et al., 2025; Wang et al., 2026), we also evaluate fine-tuning of ALBERT (Lan et al., 2020). The results demonstrate comparable or improved performance of ES(WP) across a wide range of training budgets, with a wall-clock time saving up to 43.7%.

2 Related Work

Loss-based selection. Most online selection methods weight samples by their losses. Ordered SGD (Kawaguchi & Lu, 2020) back-propagates the top-q samples of each batch; InfoBatch (Qin et al., 2024) prunes samples with below-average losses and rescales the remaining gradients; KAKURENBO (Thao Nguyen et al., 2023) also uses prediction accuracy and confidence to hide samples before moving them back; Kumar et al. (2023) and Balaban et al. (2023) derive loss-based weights from robust optimization; while Attendu & Corbeil (2023) and Raju et al. (2021) use an exponential moving average of historical losses. In contrast, ES captures both current states and dynamical variations in losses.

Model- and gradient-based selection. Other methods rely on metrics from additional models or per-sample gradients. Mindermann et al. (2022); Deng et al. (2023); Xie et al. (2023a) score samples with reference models that must be trained or stored. Gradient-based methods, such as local approximation-based (Wang et al., 2024a) and counterfactual-based (Wang et al., 2025) selection, reduce weighting costs via ghost inner products and generalized Gauss–Newton approximations. ES requires neither extra models nor gradient computations for weight calculations.

3 METHODS AND ANALYSIS

In this section, we first formulate the dynamic data selection problem and motivate loss-based selection methods. Then, we introduce the evolved sampling (ES) method. Finally, we provide a theoretical convergence analysis of ES.

3.1 PRELIMINARIES

The classic setting of general machine learning tasks is as follows. Given a dataset $\mathcal{D} := \{\boldsymbol{z}_i\}_{i=1}^n$ with $\boldsymbol{z}_i := (\boldsymbol{x}_i, y_i)$ (labeled) or $\boldsymbol{z}_i := \boldsymbol{x}_i$ (unlabeled) of size n, the goal is to solve the empirical risk minimization (ERM) problem: $\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \hat{L}_n(\boldsymbol{\theta}) := \frac{1}{n}\sum_{i=1}^n \ell_i(\boldsymbol{\theta})$, where $\ell_i(\boldsymbol{\theta}) := \ell(\boldsymbol{z}_i;\boldsymbol{\theta})$ denotes the loss function, and $\hat{L}_n(\boldsymbol{\theta})$ represents the empirical mean loss over n data samples. When n is large, a common routine is to compute stochastic gradient on a random batch instead of the whole training set. For instance, starting from an initialization $\boldsymbol{\theta}(0) = \boldsymbol{\theta}_0$, the SGD optimizer updates model parameters by $\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta_t \nabla_{\boldsymbol{\theta}} \hat{L}_n(\boldsymbol{\theta}(t)) \approx \boldsymbol{\theta}(t) - \frac{\eta_t}{B}\sum_{j\in\mathcal{B}_t} \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}(t))$, where $\{\eta_t\}_{t\in\mathbb{N}}$ denotes learning rates, and $\mathcal{B}_t \subset [n]$ denotes the batch with the size $|\mathcal{B}_t| = B$. The dynamic data selection/re-weighting problem is then formulated as $\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta_t \sum_{j\in\mathcal{B}_t} p_j(t) \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}(t))$, where $\{p_j(t)\}$ are time-varying sampling probabilities over the training dataset, and typically as certain normalized weights: $p_j(t) \propto w_j(t)$ (e.g. $w_j(t) \equiv 1/B$ for the standard batched sampling). We are supposed to find appropriate schemes of these weights/probabilities to speed up training.

3.2 LOSS-WEIGHTED SAMPLING

Standard gradient descent weights all training samples equally. This can be *inefficient* since different samples may have *varied importance to distinct learning tasks at different training stages*: as training proceeds, there are inevitably samples fitted more accurately compared to others, and these well-fitted samples should receive lower priority thereafter. Hence, it is reasonable to assign *adaptive* weights to data samples during training.

An intuitive approach is to measure sample importance by loss magnitude during training, assigning larger weights to samples with higher losses. Experiments in Katharopoulos & Fleuret (2017) and Kawaguchi & Lu (2020) show that this approach can accelerate training, while Sow et al. (2025) further demonstrates the efficiency of loss-based sample re-weighting in large-scale LLM pre-training.

For loss-weighted sampling, one calculates the sampling probability as

$$p_i(t) \propto w_i(t) = \ell_i(\boldsymbol{\theta}(t)), \tag{3.1}$$

i.e., the weight is set as the current (non-negative) loss value. On top of that, there are some variants of loss-weighted sampling strategies: For instance, Kumar et al. (2023) sets $w_i(t) = g(\ell_i(\boldsymbol{\theta}(t)))$, where the function $g(\cdot)$ is pre-defined based on the robust optimization theory; Kawaguchi & Lu (2020) directly selects top-q samples in terms of current losses per training step, which can be regarded as another realization of Kumar et al. (2023); Sow et al. (2025) performs elementary transformations on individual sample losses to improve the stability of weights' updates. Nevertheless, there are still *limited* quantitative justifications on why loss-based sample re-weighting can *accelerate in theory*, even for simple idealized objectives. We step towards this direction in the following.

Warmup: A motivating quadratic example. In this part, we provide *the first* theoretical analysis on the *provable acceleration* by loss-based re-weighting for idealized quadratic objectives.

Theorem 3.1 (Acceleration via loss-based re-weighting)**.** *Consider the sample-wise quadratic loss $\ell_i(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^\star)^\top \boldsymbol{H}_i(\boldsymbol{\theta}-\boldsymbol{\theta}^\star)$, $\boldsymbol{H}_i \succeq 0$, $i = 1, \cdots, n$, and the empirical average loss $\hat{L}_n(\boldsymbol{e}) = \frac{1}{n}\sum_{i=1}^n \ell_i(\boldsymbol{e}) = \frac{1}{2}\boldsymbol{e}^\top \boldsymbol{H}\boldsymbol{e}$, $\boldsymbol{H} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{H}_i$ (with $\boldsymbol{e} := \boldsymbol{\theta}-\boldsymbol{\theta}^\star$). Assume that $\boldsymbol{H} \succ 0$, and let $\mu = \lambda_{\min}(\boldsymbol{H}) \ll 1$ (typically). Comparing the standard gradient flow $\dot{\boldsymbol{e}}_u = -\nabla \hat{L}_n(\boldsymbol{e}_u)$ with the loss-weighted gradient flow $\dot{\boldsymbol{e}}_w = -\sum_{i=1}^n p_i(\boldsymbol{e}_w)\nabla \ell_i(\boldsymbol{e}_w)$, $p_i(\boldsymbol{e}) := \frac{\ell_i(\boldsymbol{e})}{\sum_{j=1}^n \ell_j(\boldsymbol{e})}$, we have*

1. *$\|\boldsymbol{e}_u(t)\| \sim \exp\{-\mu t\}$ ($t \to +\infty$) if $\boldsymbol{e}_u(0)^\top \boldsymbol{v}_1 \neq 0$;*
2. Non-deceleration*: Unconditionally, we have $\|\boldsymbol{e}_w(t)\| \le \|\boldsymbol{e}_w(0)\| \exp\{-\mu_w t\}$ with $\mu_w \ge \mu$;*

3. ***Strict acceleration**: If* $\mathrm{Var}_i\big(\boldsymbol{v}_{min}^\top \boldsymbol{H}_i \boldsymbol{v}_{min}\big) > 0$ *for any eigenvector* $\boldsymbol{v}_{min}$ *associated with the minimum eigenvalues of* $\boldsymbol{H}$,[1] *we further have* $\mu_w > \mu$.

The proof is deferred to Appendix B.1. Theorem 3.1 suggests that, around in-common minimizers, re-weighting based on local quadratic curvatures can strictly accelerate the standard training (given only mild non-degeneration conditions that are almost met unconditionally).

3.3 Evolved Sampling

Besides theoretically provable acceleration justifications for idealized objectives, loss-based sampling requires only lightweight calculations compared to many other more complex re-weighting schemes (e.g. gradient-weighted sampling). However, during the complex neural network training in practice, individual sample losses can appear intricate (oscillation) patterns, incurring potential sensitivity issues of re-weighting by only current (individual) losses, affecting the overall training efficiency.

To tackle these issues without performing expensive gradient calculations, we propose to re-weight by simply individual **loss predictions** $w_i(t) = \ell_i(\boldsymbol{\theta}(t+\Delta t))$, i.e. to stress samples that are still difficult (i.e. with higher losses) in future updates. This strategy not only considers the current state of training dynamics, but also accounts for *dynamical evolutions*: By Taylor expansion, $\ell_i(\boldsymbol{\theta}(t+\Delta t)) \approx \ell_i(\boldsymbol{\theta}(t)) + \frac{d}{dt}\ell_i(\boldsymbol{\theta}(t))\Delta t$ ($\Delta t > 0$ sufficiently small). In fact, $\big\{\frac{d}{dt}\ell_i(\boldsymbol{\theta}(t))\big\}_{i=1}^n$ denote the per-sample loss variations during training, which can also oscillate and be negative. To alleviate the instability issue and ensure the positivity of weights, we consider a smoothed version

$$p_i(t) \propto w_i(t) = \mathrm{EMA}_t(\ell_i(\boldsymbol{\theta}(t))) + c \cdot \frac{d}{dt}\mathrm{EMA}_t(\ell_i(\boldsymbol{\theta}(t))), \quad \text{for some } c > 0. \tag{3.2}$$

Let $s_i(t) := \mathrm{EMA}_t(\ell_i(\boldsymbol{\theta}(t)))$, with the continuous-time formulation $\frac{d}{dt}s_i(t) = \beta(\ell_i(\boldsymbol{\theta}(t)) - s_i(t))$, $s_i(0) = 1/n$ for some $\beta > 0$. Then

$$w_i(t) = s_i(t) + c\frac{d}{dt}s_i(t) = (1 - c\beta)s_i(t) + c\beta\ell_i(\boldsymbol{\theta}(t)). \tag{3.3}$$

To ensure positivity, we require $c\beta \in (0, 1)$; to stress the differential term reflecting dynamical variations, we set c as large as possible, i.e., $c\beta = 1 - \beta_1$ for some $\beta_1 > 0$ sufficiently small, which yields $w_i(t) = \beta_1 s_i(t) + (1 - \beta_1)\ell_i(\boldsymbol{\theta}(t))$. Taking the Euler discretization of s-dynamics gives $s_i(t_k) = s_i(t_{k-1}) + \beta\Delta t(\ell_i(\boldsymbol{\theta}(t_k)) - s_i(t_k)) \approx (1-\beta\Delta t)s_i(t_{k-1}) + \beta\Delta t\ell_i(\boldsymbol{\theta}(t_k))$, $\Delta t = t_{k+1} - t_k$. Note $\beta_2 := 1 - \beta\Delta t \in (0, 1)$ (close to 1), one can rewrite it as $s_i(t_k) = \beta_2 s_i(t_{k-1}) + (1 - \beta_2)\ell_i(\boldsymbol{\theta}(t_k))$, and hence obtain the following evolved sampling (ES) scheme.

For any $i \in [n]$ and $t \in \mathbb{N}$, define the evolved sampling (ES) probability as

$$\begin{aligned} p_i(t) \propto w_i(t) &= \beta_1 s_i(t-1) + (1 - \beta_1)\ell_i(\boldsymbol{\theta}(t)), \\ s_i(t) &= \beta_2 s_i(t-1) + (1 - \beta_2)\ell_i(\boldsymbol{\theta}(t)) \end{aligned} \tag{3.4}$$

with $s_i(0) = 1/n$, and $\beta_1, \beta_2 \in (0, 1)$ as two hyper-parameters (typically $(\beta_1, \beta_2) \approx (0, 1)$).

Proposition 3.2. *For the sampling scheme Equation* (3.4)*, we have*

$$w_i(t) = (1 - \beta_2)\sum_{k=1}^{t} \beta_2^{t-k}\ell_i(\boldsymbol{\theta}(k)) + O(\beta_2^t) + (\beta_2 - \beta_1)\sum_{k=1}^{t-1} \beta_2^{t-1-k} \underbrace{\big(\ell_i(\boldsymbol{\theta}(k+1)) - \ell_i(\boldsymbol{\theta}(k))\big)}_{\text{dynamical variation in loss}}. \tag{3.5}$$

Furthermore, consider the SGD optimizer with the sampling scheme Equation (3.4)*:* $\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta_t \sum_{j\in\mathcal{B}_t} p_j(t)\nabla_{\boldsymbol{\theta}}\ell_j(\boldsymbol{\theta}(t))$*, for* $t \gg 1$*, we have*

$$w_i(t) \approx (1 - \beta_2)\sum_{k=1}^{t} \beta_2^{t-k}\ell_i(\boldsymbol{\theta}(k)) - (\beta_2 - \beta_1)\sum_{k=1}^{t-1} \beta_2^{t-1-k}\eta_k c_i(k), \tag{3.6}$$

where $c_i(k) := \big\langle \nabla_{\boldsymbol{\theta}}\ell_i(\boldsymbol{\theta}(k)), \sum_{j\in\mathcal{B}_k} p_j(k)\nabla_{\boldsymbol{\theta}}\ell_j(\boldsymbol{\theta}(k))\big\rangle$ *denotes the inner product between the* i*-th sample's gradient and full gradient at the* k*-th iteration.*

[1]The eigenvector $\boldsymbol{v}_{\min}$ can be unique given appropriate conditions, e.g., $\lambda_{\min}(\boldsymbol{H})$ has the single multiplicity, which is often the case in practice.

The proof is deferred to Appendix B.2. Intuitively, $c_i(k)$ represents the individual-to-whole gradient "alignment" or correlation along training trajectories. As shown in Equation (3.6), despite that there are only calculations regarding values of losses in the sampling scheme Equation (3.4), Equation (3.4) *implicitly* leverages additional *correlations between gradients* to determine sample weights: When the individual gradient positively correlates with the whole gradient (i.e. the better-learned sample in step with the overall learning direction), we have $c_i(k) > 0$, and its sample weight is decreased as the second term of $w_i(t)$ is negative; conversely, when the individual gradient negatively correlates with the whole gradient (i.e. the worse-learned sample not in step with the overall learning direction), we have $c_i(k) < 0$, and its sample weight is increased as the second term of $w_i(t)$ is positive.

Remark 1. *We further explain why augmented loss variations should work intuitively. Given any data sample $\boldsymbol{z}_i$, if $\ell_i(\cdot)$ always increases, the augmented "variation" term in Equation* (3.5) *is positive and hence its sampling weight is increased, which is reasonable since the model continually underfits $\boldsymbol{z}_i$ and should then value $\boldsymbol{z}_i$ more. Conversely, if $\ell_i(\cdot)$ always decreases, the augmented "variation" term in Equation* (3.5) *is negative and hence its weight is decreased, which is also reasonable since the model continually fits $\boldsymbol{z}_i$ well and should then value $\boldsymbol{z}_i$ less. That said, the augmented "variation" term in Equation* (3.5) *plays a role of "damping".*

Values of betas. We discuss some benefits of the configured typical values ($(\beta_1, \beta_2) \approx (0, 1)$).

1. Equation (3.4) reduces to Equation (3.1) when setting $\beta_1 = \beta_2 = 0$, to no re-weighting when setting $\beta_1 = \beta_2 = 1$, and to the common moving average (MA) of historical losses by setting $\beta_1 = \beta_2$. Our configuration of $(\beta_1, \beta_2) \approx (0, 1)$ *avoids* these degeneration cases.
2. Based on Equation (3.5), the strengths of losses and their dynamical variations can be flexibly balanced via (β_1, β_2). By setting $(\beta_1, \beta_2) \approx (0, 1)$, we are able to exploit the *long-term dynamical history* during training (via β_2, s-dynamics), while simultaneously responding to *current dynamical states* of losses (via β_1, w-dynamics).

ES(WP) framework. We refer to the optimization with the scheme Equation (3.4) as Evolved Sampling (ES), which conducts dynamic data selection on a batch-level. To further incorporate set-level data selection, we extend ES to prune samples at each epoch with the same sampling scheme Equation (3.4), leading to **E**volved **S**ampling **W**ith **Pr**uning (ESWP), as illustrated in Figure 1. We optionally adopt the annealing technique (Qin et al. (2024)) to enhance learning performance. As a plug-and-play framework, ES(WP) can be integrated into any optimizers. In practice, the simple and elegant design of the sampling scheme Equation (3.4) is surprisingly effective, as shown in Section 4.

3.4 Theoretical Analysis

We establish the following estimate on the convergence rate and generalization gap of SGD weighted by the sampling scheme Equation (3.4). The proof is deferred to Appendix B.3.

Theorem 3.3. *Assume that $\ell_i(\cdot)$ is convex and L_i-smooth for each $i \in [n]$, and there exists $\boldsymbol{\theta}^*$ such that $\hat{L}_n(\boldsymbol{\theta}^*) = 0$. Then, for the SGD optimizer with the sampling scheme Equation* (3.4)*, i.e. $\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta_t \sum_{j \in \mathcal{B}_t} p_j(t) \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}(t))$, with the constant learning rate $\eta_t \equiv \eta = 1/(2L)$ ($L := \max_{i \in [n]} L_i$), there exists (β_1, β_2) such that*

$$\hat{L}_n\left(\frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{\theta}(t)\right) - \hat{L}_n(\boldsymbol{\theta}^*) \leq \frac{2L\mathbb{E}\left\|\boldsymbol{\theta}(0) - \boldsymbol{\theta}^*\right\|_2^2}{T} - \mathcal{R}_1(T), \tag{3.7}$$

where $\mathcal{R}_1(\cdot) : \mathbb{N} \mapsto \mathbb{R}$ is a non-negative function. Furthermore, if both data and model parameters are bounded, for any $\delta > 0$, with probability at least $1 - \delta$, we have

$$\mathbb{E}_{\boldsymbol{z}}\left[\ell(\boldsymbol{z}; \boldsymbol{\theta}(t))\right] - \sum_{i=1}^{n} p_i(t)\ell(\boldsymbol{z}_i; \boldsymbol{\theta}(t)) \leq \left(2\mathrm{Rad}_{\mathcal{D}}(\boldsymbol{\Theta}) + 4\sup_{\boldsymbol{\theta} \in \boldsymbol{\Theta}, \boldsymbol{z}} \ell(\boldsymbol{z}; \boldsymbol{\theta}) \cdot \sqrt{\frac{2\ln(4/\delta)}{n}}\right) - \mathcal{R}_2(t), \tag{3.8}$$

where $\mathrm{Rad}_{\mathcal{D}}(\boldsymbol{\Theta}) \overset{def}{=} \frac{1}{n}\mathbb{E}_{\boldsymbol{\sigma} \sim \{\pm 1\}^n}\left[\sup_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \sum_{i=1}^{n} \sigma_i \ell(\boldsymbol{z}_i; \boldsymbol{\theta})\right]$ denotes the (empirical) Rademacher complexity, and $\mathcal{R}_2(\cdot) : \mathbb{N} \mapsto \mathbb{R}$ is also a non-negative function.

Theoretical comparisons. As shown in Theorem 3.3, the first error term in Equation (3.7) agrees with the vanilla case (standard training without data selection), which stands for the sub-linear convergence rate. However, the additional remainder term $-\mathcal{R}_1(\cdot)$ is non-positive, leading to a sharper convergence rate of this ES-weighted training dynamics. Similarly, the first error term in Equation (3.8) also agrees with the vanilla case, and the additional non-positive remainder term $-\mathcal{R}_2(\cdot)$ implies a reduced generalization gap of the ES-weighted dynamics.

3.5 Complexity Analysis

Computation efficiency. The primary source of savings comes from the substantial reduction in the effective batch size during back propagations (BPs), compared with standard training. Although ES(WP) introduces an extra forward pass (FP) on the selected mini-batch the overhead is modest since FP requires much fewer FLOPs than BP. Consequently, the reduction in BP dominates the overall time complexity, leading to a significant acceleration effect, as observed empirically in Section 4.2.

Memory efficiency. From Equation (3.4), the only additional memory cost of ES(WP) is to store the current score/weight value of each sample, which is negligible compared to the memory cost in high-dimensional data itself. Moreover, because ES(WP) reduces the effective sample size in BP, it further decreases memory usage, also numerically verified in Section 4.2.

More significant benefits under resource constraints. The advantages of ES(WP) become even more significant in low-resource scenarios where GPU memory is limited and gradient accumulation is required, a typical scenario in fine-tuning LLMs. Suppose the micro-batch size on each GPU is b_{micro}. Then under standard sampling, the number of BP per update step is $\lceil B/b_{\text{micro}} \rceil$. In contrast, ES(WP) requires only $\lceil b/b_{\text{micro}} \rceil$ BP passes. When $b_{\text{micro}} \leq b$, the time spent on BP under standard sampling can be up to B/b times greater than that of ES(WP).

3.6 Hyper-Parameters Tuning

The primary hyper-parameters in the sampling scheme Equation (3.4) are betas (β_1, β_2), which balance dynamical losses and their variations during training. In experiments, we take the default values of (β_1, β_2), which are obtained by a fine-grained grid search in small-scale simulations (Section 4.4). These defaults are consistently validated to be (locally) optimal in small-scale experiments (Section 4.4), and their superior effectiveness robustly remains on large-scale real-world tasks (Section 4.2, (ii) & (iii), Section 4.3). The other hyper-parameters, including mini-batch sizes, the pruning ratio and annealing epochs, trade off learning performance and training efficiency. For all data selection methods, larger pruning ratios often accelerate convergence but degrade final performance. Therefore, these hyper-parameters are user-defined based on users' practical demands and computation resources, as in previous methods such as Qin et al. (2024); Thao Nguyen et al. (2023); Raju et al. (2021).

4 Experiments

4.1 Experimental Setup

We evaluate ES(WP) on vision and language tasks. For vision, we train ResNet-18/50 (He et al., 2016) on CIFAR-10/100, pre-train ViT-Large with MAE (He et al., 2022) on ImageNet-1K, and fully fine-tune an ImageNet-21K pre-trained ViT-Large (Dosovitskiy et al., 2021) on ImageNet-1K. For language, we pre-train GPT-2-Small on OpenWebText and GPT-2-XL on FineWeb (Radford et al., 2019), and fine-tune Qwen2.5-Math-1.5B (Yang et al., 2024) and Qwen3.5-9B-Base (Qwen Team, 2026) (with LoRA) on NuminaMath-CoT. Due to recent interest in looped Transformers (Wang et al., 2026), we also fine-tune ALBERT (Lan et al., 2020). Experiment details are provided in Appendix D.

We compare against standard batched sampling without data selection (Baseline) and two groups of data selection methods. *Set-level* methods: random pruning (Random), UCB (Raju et al., 2021), KAKURENBO (KA; Thao Nguyen et al., 2023), and InfoBatch (Qin et al., 2024); *batch-level* methods: loss-based sampling (Loss; Katharopoulos & Fleuret, 2017) and Ordered SGD (Order; Kawaguchi & Lu, 2020). ES selects at the batch level, and ESWP at both levels. We report task performance and the wall-clock training time saved relative to Baseline. By default, ES uses $(\beta_1, \beta_2) = (0.2, 0.9)$ and ESWP uses $(0.2, 0.8)$, with a mini-batch to meta-batch size ratio of $b/B = 25\%$ and, where applicable, an annealing ratio of 5%. The default pruning ratio for ESWP

Table 1: Accuracy (%) and saved training time (%). (a) ResNet training on CIFAR (test accuracy). (b) Full fine-tuning of ViT-Large on ImageNet-1K (validation accuracy).

(a) CIFAR

	CIFAR-10 (R-18)		CIFAR-100 (R-18)		CIFAR-100 (R-50)	
Baseline	95.4		78.8		81.1	
Random	$95.3_{\downarrow 0.1}$	18%	$78.4_{\downarrow 0.4}$	20%	$80.8_{\downarrow 0.3}$	19%
UCB (Raju et al. (2021))	$95.2_{\downarrow 0.2}$	18%	$77.6_{\downarrow 1.2}$	18%	$80.5_{\downarrow 0.6}$	24%
KA (Thao Nguyen et al. (2023))	$95.3_{\downarrow 0.1}$	21%	$78.1_{\downarrow 0.7}$	21%	$80.2_{\downarrow 0.9}$	24%
InfoBatch (Qin et al. (2024))	$95.3_{\downarrow 0.1}$	21%	$78.4_{\downarrow 0.4}$	24%	$80.4_{\downarrow 0.7}$	28%
Loss (Katharopoulos & Fleuret (2017))	$95.3_{\downarrow 0.1}$	11%	$78.4_{\downarrow 0.4}$	10%	$80.5_{\downarrow 0.6}$	12%
Order (Kawaguchi & Lu (2020))	$\mathbf{95.4}_{\uparrow 0.0}$	11%	$78.5_{\downarrow 0.3}$	10%	$80.9_{\downarrow 0.2}$	12%
ES	$\mathbf{95.4}_{\uparrow 0.0}$	10%	$\mathbf{78.8}_{\uparrow 0.0}$	10%	$\mathbf{81.1}_{\uparrow 0.0}$	11%
ESWP	$95.3_{\downarrow 0.1}$	**24%**	$78.6_{\downarrow 0.2}$	**24%**	$80.8_{\downarrow 0.3}$	**31%**

(b) ImageNet-1K

	Time ↓	Acc. (%)
Baseline	–	84.4
Random	24.5%	$84.5_{\uparrow 0.1}$
UCB	23.6%	$84.2_{\downarrow 0.2}$
KA	25.3%	$84.3_{\downarrow 0.1}$
InfoBatch	23.5%	$84.7_{\uparrow 0.3}$
Loss	36.4%	$84.3_{\downarrow 0.1}$
Order	38.2%	$84.2_{\downarrow 0.2}$
ES	26.0%	$84.7_{\uparrow 0.3}$
ESWP	**40.7%**	$\mathbf{85.0}_{\uparrow 0.6}$

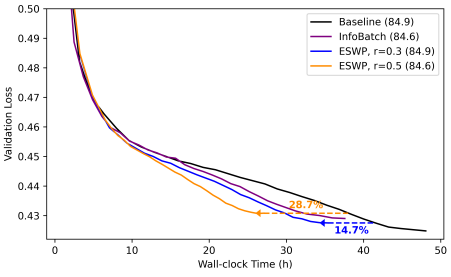
and Random is 20%. Loss and Order use the same b and B as ES(WP). For UCB, KA, and InfoBatch, we use the default hyperparameters from their original papers, listed in Appendix D.11. For more types of data selection baselines (e.g. gradient-based, static/early-training-based), see Appendix D.4.

4.2 Vision Tasks

(i) ResNet training on CIFAR. Table 1(a) compares ResNet-18/50 training on CIFAR-10/100. It is shown that the batch-level data selection methods (Loss, Order, ES) typically exhibit limited accelerations on these small-scale tasks, since they often require additional forward propagation overheads that are not negligible compared to BPs. Nevertheless, ES is the only algorithm that achieves lossless accelerations across all sampling methods. Notably, ESWP saves the most computation time while maintaining the best performance among set-level data selection methods (UCB, KA, InfoBatch).

(ii) ViT-Large fine-tuning on ImageNet-1K. We fully fine-tune ViT-Large, pre-trained on ImageNet-21K, on ImageNet-1K. As shown in Table 1(b), ES consistently achieves the best performance among batch-level data selection methods and the second-highest accuracy across all sampling methods. Notably, ESWP attains the highest accuracy and the most significant wall-clock time reduction, suggesting that ESWP inherits the advantages of *both* set- and batch-level data selection methods. In addition, it is observed that the training speed-up of batch-level methods gets far more significant given these large-scale tasks, conversely surpassing the set-level methods compared to (i). This is due to the dominance of the saved computation in BPs.

Memory usage is also lower: ES and ESWP use 49.7 GB and 49.1 GB, respectively, compared with 52.4 GB for Baseline. Thus, both methods reduce memory usage and training time in this setting.



	Time (h)	Time ↓	Acc. (%)
Baseline	48.1	–	84.9
InfoBatch	37.6	21.8%	$84.6_{\downarrow 0.3}$
ESWP ($r = 0.3$)	35.1	27.0%	$\mathbf{84.9}_{\uparrow 0.0}$
ESWP ($r = 0.5$)	**27.1**	**43.7%**	$84.6_{\downarrow 0.3}$

Figure 2 (left) and Table 2 (right): MAE pre-training of ViT-Large on ImageNet-1K. Left: reconstruction loss, with validation accuracy (%) after fine-tuning in parentheses in the legend. Right: pre-training time and fine-tuning accuracy. r denotes the pruning ratio.

(iii) MAE pre-training on ImageNet-1K. We pre-train ViT-Large using MAE and then fully fine-tune it on ImageNet-1K without data selection. We plot the reconstruction loss curves in Figure 2 and report final accuracy after fine-tuning in Table 2. It is shown that ESWP achieves lossless acceleration over Baseline and consistently outperforms the previous SOTA method InfoBatch.

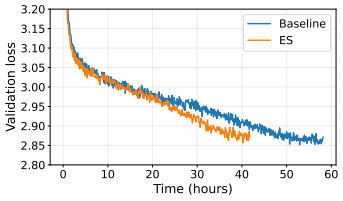


Figure 3: Validation loss v.s. pre-training time for GPT-2-Small on OpenWebText (300B tokens).

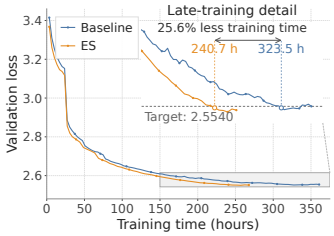


Figure 4: Validation loss v.s. pre-training time for GPT-2-XL on FineWeb (300B tokens).

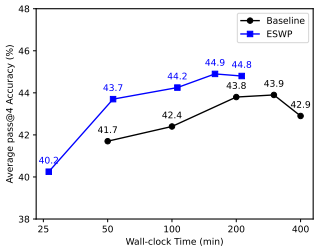


Figure 5: Average pass@4 accuracy v.s. fune-tuning time for Qwen2.5-Math-1.5B on 30K NuminaMath-CoT samples.

4.3 Language Tasks

GPT-2 pre-training. We pre-train GPT-2-Small on OpenWebText using AdamW and GPT-2-XL on FineWeb using Muon, both for about 300B tokens. Figures 3 and 4 compare validation loss against training time for GPT-2-Small and GPT-2-XL, respectively. Full details are given in Appendix D.9.

To assess downstream performance, we also evaluate the models on GLUE. Table 3 reports results for both model sizes. For GPT-2-Small, ES increases the mean score from 78.6 to 79.0 while saving 28.2% of pre-training time. For GPT-2-XL, ES improves seven of eight task scores and increases the mean from 83.7 to 85.0, with a reported pre-training time saving of 25.6%. We additionally evaluate GPT-2-XL on nine zero-shot benchmarks (Table 4). ES improves four of nine task scores (even with reduced training time) and increases the mean accuracy from 44.51% to 44.68%.

Table 3: GLUE scores of GPT-2-Small and GPT-2-XL. Time↓ denotes pre-training time saved relative to the Baseline of the same model size.

Model	Method	CoLA	SST-2	QNLI	QQP	MNLI-m	MRPC	RTE	STS-B	Avg.	Time↓
GPT-2-Small	Baseline	**46.0**	93.5	88.9	**90.1**	**83.0**	79.2	60.6	87.8	78.6	–
	ES	44.2	**93.6**	**89.1**	**90.1**	**83.0**	**79.7**	**63.9**	**88.7**	**79.0**	**28.2%**
GPT-2-XL	Baseline	59.0	95.2	91.8	90.2	**87.6**	79.2	76.2	**90.1**	83.7	–
	ES	**60.7**	**95.6**	**92.1**	**90.6**	87.3	**86.0**	**77.6**	**90.1**	**85.0**	**25.6%**

Table 4: Zero-shot accuracy (%) of GPT-2-XL on nine benchmarks. W.G. denotes WinoGrande and C.QA denotes CommonsenseQA.

	MMLU	ANLI	HellaSwag	PIQA	SIQA	W.G.	ARC-E	C.QA	WSC	Avg.	Time↓
Baseline	**29.61**	31.88	**55.56**	**74.16**	40.17	**55.33**	41.98	**33.42**	38.46	44.51	–
ES	29.31	**33.17**	54.66	72.80	**41.15**	55.25	**43.56**	32.84	**39.42**	**44.68**	**25.6%**

ALBERT fine-tuning on GLUE. To evaluate an architecture with cross-layer parameter sharing, we fully fine-tune ALBERT (Lan et al., 2020) on GLUE. This setting is motivated by recent interest in looped Transformers (Wang et al., 2026). As shown in Table 5, ES achieves the highest mean score, 83.9, compared with 83.4 for Baseline, while saving 20.2% of training time. ESWP achieves a mean score of 83.6 and the largest time saving, 33.1%, among the compared methods.

Qwen2.5-Math-1.5B fine-tuning under memory constraints. We fine-tune Qwen2.5-Math-1.5B on 30K samples from NuminaMath-CoT using a single NVIDIA A100 (40GB) GPU. Figure 5 plots pass@4 accuracy averaged over MATH-500 (Hendrycks et al., 2021) and OlympiadBench (He et al., 2024) against training time. ESWP reaches comparable or higher accuracy with less training time in this setting. Selecting fewer samples for backpropagation reduces the gradient accumulation required when memory limits the micro-batch size.

Table 5: The validation metric (%) and saved time of fully fine-tuning ALBERT-Base on GLUE.

	CoLA	SST2	QNLI	QQP	MNLI-m	MRPC	RTE	STSB	Avg.	Time↓
Baseline	56.7	92.2	91.1	**90.3**	**84.7**	88.5	74.0	89.6	83.4	-
InfoBatch	57.9	92.1	91.2	**90.3**	84.5	89.2	73.8	**89.7**	83.6$_{\uparrow 0.2}$	28.3%
Loss	55.1	92.3	91.4	90.2	84.4	88.6	69.6	89.5	82.6$_{\downarrow 0.8}$	20.8%
Order	55.4	92.6	91.3	90.1	80.9	84.6	63.2	89.4	80.9$_{\downarrow 2.5}$	20.8%
ES	**58.4**	92.4	91.4	90.2	84.5	88.7	**75.8**	89.6	**83.9**$_{\uparrow 0.5}$	20.2%
ESWP	57.5	**93.1**	**91.7**	90.0	**84.7**	**89.8**	72.8	89.4	83.6$_{\uparrow 0.2}$	**33.1%**

Qwen3.5-9B LoRA fine-tuning. We fine-tune Qwen3.5-9B-Base with LoRA on 100K samples from NuminaMath-CoT. Table 6 compares ES and Baseline on GSM8K, MATH-500, and OlympiadBench at a matched training time of 250 minutes. ES improves accuracy by 0.53, 1.40, and 2.52 percentage points, respectively. These results show higher accuracy at the same training-time budget across all three benchmarks.

Table 6: Accuracy (%) after LoRA fine-tuning of Qwen3.5-9B-Base on 100K NuminaMath-CoT samples at 250 minutes. ES outperforms Baseline on all benchmarks.

Method	GSM8K	MATH-500	OlympiadBench
Baseline	91.05	68.00	28.04
ES	**91.58**	**69.40**	**30.56**

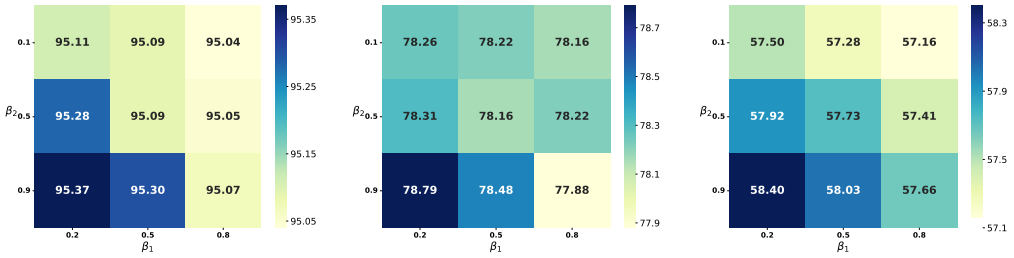
4.4 Ablation Studies

Loss variations. We evaluate the effects of dynamical loss variations and annealing in ES(WP). Table 7 compares combinations of "Loss" ($\beta_1 = \beta_2 = 0$), "NonDif" ($\beta_1 = \beta_2$, without dynamical loss variations), "Dif" ($\beta_1 \neq \beta_2$, with dynamical loss variations), and "A" (annealing). Annealing improves loss-based sampling on both tasks. With annealing enabled, incorporating dynamical loss variations improves performance over the "NonDif" variant on both tasks. The full ES variant achieves the best performance in both settings.

Table 7: Ablations on the effect of augmented dynamical variations of losses and annealing.

	ResNet-50 CIFAR-100	ALBERT-Base CoLA
Loss	80.5	55.1
Loss + A	80.8	55.8
Loss + NonDif	80.5	57.6
Loss + Dif	**81.1**	57.5
Loss + A + NonDif	80.4	57.6
ES = Loss + A + Dif	**81.1**	**58.4**

Choices of (β_1, β_2). We evaluate the sensitivity of ES to (β_1, β_2) by training ResNet-18 on CIFAR-10/100 and fine-tuning ALBERT-Base on CoLA. As shown in Figure 6, the default values $(\beta_1, \beta_2) = (0.2, 0.9)$ achieve the best performance among the tested configurations in all three settings.



(a) CIFAR-10 (ResNet-18) (b) CIFAR-100 (ResNet-18) (c) CoLA (ALBERT-Base)

Figure 6: The effect of (β_1, β_2).

5 Conclusion

In this work, we propose a simple yet effective dynamic data selection framework, Evolved Sampling (ES). By further approximating and flexibly tuning loss predictions or dynamical evolutions for sample weights, Evolved Sampling is lightweight and can achieve improved or lossless performance with significant training accelerations. The effectiveness and efficiency of ES are consistently validated by numerous experiments across vision and language tasks with various scales.

AI USE STATEMENT

In this work, we used generative AI tools for implement methods. We have not used generative AI tools for help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, and interpret results. Generate synthetic data sets, assist with translation, clean and reformat dataset, and support qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for create or modify scientific figures or images, create or edit software code, summarize or analyse existing literature, sourcing/searching for information, edit a research paper to improve readability, and identify relevant literature. We have reviewed all AI-assisted work. For code, we checked the implementations against the algorithms described in the paper and confirmed that they reproduce the reported results. For figures, we verified that all plotted values match the underlying experimental logs. For literature, we read each identified paper, confirmed that every citation refers to an existing publication, and checked that our descriptions of prior work are faithful to the original sources. For writing, we ensured that all edits preserved our intended meaning and introduced no new claims. We take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We have made the following efforts to ensure reproducibility. **Algorithm.** ES(WP) is fully specified by the sampling scheme Equation (3.4) and the pseudocode in Algorithm 1, with further details, including the annealing strategy, in Appendix C. **Theory.** All assumptions are stated in Proposition 3.2 and Theorem 3.3, and complete proofs are provided in Appendices B.2 and B.3. **Experiments.** The datasets, models, compared methods, and default hyperparameters of ES(WP) are described in Section 4. Per-task training settings, including optimizers, learning-rate schedules, batch sizes, numbers of epochs or iterations, and hardware, are given in Appendix D: CIFAR in Appendix D.5, ViT-Large fine-tuning in Appendix D.6, MAE and GPT-2 pre-training in Appendix D.9, Qwen2.5-Math fine-tuning in Appendix D.7, and Qwen3.5 LoRA fine-tuning in Appendix D.8. The default hyperparameters of all compared methods are listed in Appendix D.11. All datasets and pre-trained models are publicly available, and our implementations build on public frameworks (PyTorch, Timm, nanoGPT, verl, and ms-swift).

REFERENCES

Amro Kamal Mohamed Abbas, Kushal Tirumala, Daniel Simig, Surya Ganguli, and Ari S. Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023. URL https://openreview.net/forum?id=4vlGm9gv6c.

Guillaume Alain, Alex Lamb, Chinnadhurai Sankar, Aaron Courville, and Yoshua Bengio. Variance reduction in SGD by distributed importance sampling. *arXiv preprint arXiv:1511.06481*, 2015.

Jean-Michel Attendu and Jean-Philippe Corbeil. NLU on data diets: Dynamic data subset selection for NLP classification tasks. In Nafise Sadat Moosavi, Iryna Gurevych, Yufang Hou, Gyuwan Kim, Young Jin Kim, Tal Schuster, and Ameeta Agrawal (eds.), *Proceedings of the Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pp. 129–146, Toronto, Canada (Hybrid), July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.sustainlp-1.9. URL https://aclanthology.org/2023.sustainlp-1.9.

Olivier Bachem, Mario Lucic, and Andreas Krause. Coresets for nonparametric estimation - the case of DP-Means. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 209–217, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/bachem15.html.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Valeriu Balaban, Jayson Sia, and Paul Bogdan. Robust learning under label noise by optimizing the tails of the loss distribution. In *International Conference on Machine Learning and Applications (ICMLA)*, pp. 520–527, 2023. doi: 10.1109/ICMLA58977.2023.00078.

Mayee F. Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. Skill-it! A data-driven skills framework for understanding and training language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 36000–36040. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/70b8505ac79e3e131756f793cd80eb8d-Paper-Conference.pdf.

Hoyong Choi, Nohyun Ki, and Hye Won Chung. BWS: Best window selection based on sample scores for data pruning across broad ranges. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8672–8701. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/choi24c.html.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJg2b0VYDr.

Rudrajit Das, Xi Chen, Bertram Ieong, Parikshit Bansal, and Sujay Sanghavi. Understanding the training speedup from sampling with approximate losses. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10127–10147. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/das24b.html.

Sanjoy Dasgupta, Daniel Hsu, Stefanos Poulis, and Xiaojin Zhu. Teaching a black-box learner. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1547–1555. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/dasgupta19a.html.

Zhijie Deng, Peng Cui, and Jun Zhu. Towards accelerated model training via Bayesian data selection. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8513–8527. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1af3e0bf5905e33789979f666c31192d-Paper-Conference.pdf.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

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

Guillaume Garrigos and Robert M. Gower. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.

Yuxian Gu, Li Dong, Hongning Wang, Yaru Hao, Qingxiu Dong, Furu Wei, and Minlie Huang. Data selection via optimal control for language models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=dhAL5fy8wS.

Ayoub El Hanchi, David A. Stephens, and Chris J. Maddison. Stochastic reweighted gradient descent. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8359–8374. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/hanchi22a.html.

Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pp. 291–300, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138520. doi: 10.1145/1007352.1007400. URL https://doi.org/10.1145/1007352.1007400.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

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.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Lingxiao Huang, Shaofeng H.-C. Jiang, Jianing Lou, and Xuan Wu. Near-optimal coresets for robust clustering. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Nc1ZkRW8Vde.

Nishant Jain, Arun S. Suggala, and Pradeep Shenoy. Improving generalization via meta-learning on hard samples. *arXiv preprint arXiv:2403.12236*, 2024.

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

Angelos Katharopoulos and François Fleuret. Biased importance sampling for deep neural network training. *arXiv preprint arXiv:1706.00043*, 2017.

Angelos Katharopoulos and Francois Fleuret. Not all samples are created equal: Deep learning with importance sampling. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2525–2534. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/katharopoulos18a.html.

Kenji Kawaguchi and Haihao Lu. Ordered SGD: A new stochastic optimization framework for empirical risk minimization. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 669–679. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/kawaguchi20a.html.

Ahmed Khaled and Peter Richtárik. Better theory for SGD in the nonconvex world. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=AU4qHN2VkS. Survey Certification.

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.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. GRAD-MATCH: Gradient matching based data subset selection for efficient deep model training. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5464–5474. PMLR, 18–24 Jul 2021b. URL https://proceedings.mlr.press/v139/killamsetty21a.html.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. GLISTER: Generalization based data subset selection for efficient and robust learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):8110–8118, May 2021c. doi: 10.1609/aaai.v35i9.16988. URL https://ojs.aaai.org/index.php/AAAI/article/view/16988.

Krishnateja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh Iyer. RETRIEVE: Coreset selection for efficient and robust semi-supervised learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 14488–14501. Curran Associates, Inc., 2021d. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/793bc52a941b3951dfdb85fb04f9fd06-Paper.pdf.

Germain Kolossov, Andrea Montanari, and Pulkit Tandon. Towards a statistical theory of data selection under weak supervision. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=HhfcNgQn6p.

Ramnath Kumar, Kushal Majmundar, Dheeraj Nagaraj, and Arun Sai Suggala. Stochastic re-weighted gradient descent via distributionally robust optimization. *arXiv preprint arXiv:2306.09222*, 2023.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1eA7AEtvS.

Michael Langberg and Leonard J. Schulman. Universal ϵ-approximators for integrals. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pp. 598–607, USA, 2010. Society for Industrial and Applied Mathematics. ISBN 9780898716986.

Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6781–6792. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/liu21f.html.

Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. In *ICLR 2016 Workshop Track*, 2016.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 650–663, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.51. URL https://aclanthology.org/2021.emnlp-main.51.

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

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6950–6960. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/mirzasoleiman20a.html.

Alexander Munteanu, Chris Schwiegelshohn, Christian Sohler, and David Woodruff. On coresets for logistic regression. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi,

and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31, pp. 6561–6570. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/63bfd6e8f26d1d3537f4c5038264ef36-Paper.pdf.

Deanna Needell, Nathan Srebro, and Rachel Ward. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/b3310bba2be31e673a7ded3386994599-Paper.pdf.

Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 5186–5198. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/299a23a2291e2126b91d54f3601ec162-Paper.pdf.

Yonatan Oren, Shiori Sagawa, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust language modeling. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4227–4237, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1432. URL https://aclanthology.org/D19-1432.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 20596–20607. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/ac56f8fe9eea3e4a365f29f0f1957c55-Paper.pdf.

Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Zhaopan Xu, Daquan Zhou, Lei Shang, Baigui Sun, Xuansong Xie, and Yang You. InfoBatch: Lossless training speed up by unbiased dynamic data pruning. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=C61sk5LsK6.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

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

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Noveen Sachdeva, Benjamin Coleman, Wang-Cheng Kang, Jianmo Ni, Lichan Hong, Ed H. Chi, James Caverlee, Julian McAuley, and Derek Zhiyuan Cheng. How to train data-efficient LLMs. *arXiv preprint arXiv:2402.09668*, 2024.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Shai Shalev-Shwartz and Yonatan Wexler. Minimizing the maximal loss: How and why. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 793–801, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/shalev-shwartzb16.html.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, pp. 369–386. SPIE, 2019.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 19523–19536. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/7b75da9b61eda40fa35453ee5d077df6-Paper-Conference.pdf.

Daouda Sow, Herbert Woisetschläger, Saikiran Bulusu, Shiqiang Wang, Hans-Arno Jacobsen, and Yingbin Liang. Dynamic loss-based sample reweighting for improved large language model pretraining. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=gU4ZgQNsOC.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Truong Thao Nguyen, Balazs Gerofi, Edgar Josafat Martinez-Noriega, François Trahay, and Mohamed Wahib. KAKURENBO: Adaptively hiding samples in deep neural network training. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 37900–37922. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/7712b1075f5e0eae297702845714098f-Paper-Conference.pdf.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJlxm30cKm.

Jiachen T. Wang, Tong Wu, Dawn Song, Prateek Mittal, and Ruoxi Jia. GREATS: Online selection of high-quality data for LLM training in every iteration. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 131197–131223. Curran Associates, Inc., 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ed165f2ff227cf36c7e3ef88957dadd9-Paper-Conference.pdf.

Jiachen T. Wang, Dawn Song, James Zou, Prateek Mittal, and Ruoxi Jia. Capturing the temporal dependence of training data influence. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=uHLgDEgiS5.

Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. CAFE: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12196–12205, June 2022.

Shaowen Wang, Ge Zhang, Kairong Luo, Yuhao Wu, Shaofan Liu, Jiaheng Liu, Wenhao Huang, Shen Yan, and Jian Li. Smelt: Scaling laws for compute-matched moe looped transformers. *arXiv preprint arXiv:2609.01343*, 2026.

Ziteng Wang, Jianfei Chen, and Jun Zhu. Efficient backpropagation with variance controlled adaptive sampling. In *International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=gEwKAZZmSw.

Ross Wightman et al. Pytorch image models, 2019.

Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=7D5EECbOaf9.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 69798–69818. Curran Associates, Inc., 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/dcba6be91359358c2355cd920da3fcbd-Paper-Conference.pdf.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 34201–34227. Curran Associates, Inc., 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6b9aa8f418bde2840d5f4ab7a02f663b-Paper-Conference.pdf.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

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

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. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Syx4wnEtvH.

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

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6514–6523, January 2023.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1–9, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/zhaoa15.html.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, Ziniu Li, Haoran Que, Boyi Wei, Zixin Wen, Fan Yin, He Xing, et al. Scaling latent reasoning via looped language models. *arXiv preprint arXiv:2510.25741*, 2025.

A Related Work: More Details

Static sampling. Methods to sampling statically can be based on geometry, uncertainty, error, meta optimization, dataset distillation, etc. With numerous studies on theoretical guarantees (Har-Peled & Mazumdar (2004); Huang et al. (2023); Bachem et al. (2015)), the coreset selection is designed to approximate original datasets with smaller (weighted) subsets, typically achieved by clustering in representation spaces (Xia et al. (2023); Abbas et al. (2023); Sorscher et al. (2022)). Uncertainty-based methods use probability metrics such as the confidence, entropy (Coleman et al. (2020)) and distances to decision boundaries (Ducoffe & Precioso (2018); Margatina et al. (2021); Dasgupta et al. (2019); Liu et al. (2021)). Sampling methods based on errors assume that training samples with more contributions to errors are more important. Errors are evaluated with metrics such as forgetting events (Toneva et al. (2019)), GraNd & EL2N score (Paul et al. (2021)), and sensitivity (Langberg & Schulman (2010); Munteanu et al. (2018)). The meta optimization methods apply a bilevel framework to learn the re-weighting. In general, existing studies such as Retrieve (Killamsetty et al. (2021d)), Glister (Killamsetty et al. (2021c)), MOLERE (Jain et al. (2024)), CAFE (Wang et al. (2022)) and so on, consider the data selection as the outer objective (over selection weights), and the optimization of model parameters on selected subsets as the inner objective. Dataset distillation aims to synthesize an informative but smaller subset from the original (large) dataset. Although there are multiple implementations to reduce the overall loads, such as distributed kernel computation (Nguyen et al. (2021)), category decoupling (Wang et al. (2022)), random modeling (Zhao & Bilen (2023)) and so on, the dataset distillation still requires to optimize over both the model and data, and is hence expensive. A recent work Sachdeva et al. (2024) also leverages the zero-shot reasoning capability of instruction-tuned large language models (LLMs) to directly assess the quality of data examples. These static sampling methods require extra training, leading to considerable costs in both computation and memory.

Dynamic sampling. Methods to sampling dynamically typically leverage metrics based on losses and gradients along the training process. Loss-adaptive sampling re-weights data points during the training according to current losses (Jiang et al. (2019); Loshchilov & Hutter (2016); Schaul et al. (2016); Kawaguchi & Lu (2020); Qin et al. (2024); Thao Nguyen et al. (2023); Kumar et al. (2023); Balaban et al. (2023); Katharopoulos & Fleuret (2017); Shrivastava et al. (2016); Das et al. (2024)) and historical losses (Attendu & Corbeil (2023); Raju et al. (2021); Oren et al. (2019); Sagawa et al. (2020)). To name a few, Ordered SGD (Kawaguchi & Lu (2020)) selects top-q samples in terms of the loss ranking per training step. InfoBatch (Qin et al. (2024)) randomly prunes a portion of less informative samples with losses below the average and then re-scales the gradients. KAKURENBO (Thao Nguyen et al. (2023)) combines current losses with the prediction accuracy and confidence to design a sampling framework with moving-back. Kumar et al. (2023) and Balaban et al. (2023) assign weights as functions of current losses based on the robust optimization theory. Attendu & Corbeil (2023) and Raju et al. (2021) use the exponential moving average over past losses for sampling. There are also studies adopting additional reference models, including Mindermann et al. (2022); Deng et al. (2023); Xie et al. (2023a) and so on. These methods either use the information of losses inadequately, or require to train or exploit extra architectures. Gradient-based sampling methods involve (i) gradient matching, such as CRAIG (Mirzasoleiman et al. (2020)) and Grad-Match (Killamsetty et al. (2021b)), which approximate the "full" gradients computed on original datasets via the gradients computed on subsets; (ii) gradient adaption, where the sampling probability is basically determined by current scales of gradients (Hanchi et al. (2022); Katharopoulos & Fleuret (2018)). Obviously, gradient-based sampling methods lead to much more computation and memory overheads than loss-based methods. A recent work Wang et al. (2024b) uses a intricate layer-wise sampling scheme with complex variance control, which develops former literature Zhao & Zhang (2015); Alain et al. (2015); Needell et al. (2014) applying importance sampling methods to accelerate the convergence by reducing variances. A very recent work Gu et al. (2025) also leverages the optimal control theory (i.e. Pontryagin's maximum principle, PMP) to formulate and decide sampling weights, where both the gradient and Hessian are computed and all historical checkpoints are stored. Obviously, these methods usually suffer from significant computation and memory loads, since extra complexities of at least model dimensions are introduced at every training step. Although there are other gradient-based data selection methods (e.g. Wang et al. (2024a): local approximation-based selection; Wang et al. (2025): counterfactual-based selection) developing computation reduction techniques such as the ghost inner-product (of gradients) and generalized Gauss-Newton approximation (of Hessians), these methods are not directly extendable to other popular optimizers like Adam.

Set-level versus batch-level. Dynamic sampling methods can be divided into two categories based on the level where data selection is performed: (i) *set*-level selection, to prune the whole dataset at the beginning of each epoch (Qin et al. (2024); Raju et al. (2021); Thao Nguyen et al. (2023); Attendu & Corbeil (2023)); (ii) *batch*-level selection, to sample subsets from the original batches for back propagations (Kawaguchi & Lu (2020); Katharopoulos & Fleuret (2017; 2018); Mindermann et al. (2022)). These two types of methods, facilitating training accelerations from different perspectives, are not mutually exclusive. However, to the best of our knowledge, we are not aware of any algorithms combining both of them.

B Proofs and Supplemental Theory

B.1 Proof of Theorem 3.1

Proof. Consider the sample-wise quadratic loss

$$\ell_i(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^\star)^\top \boldsymbol{H}_i(\boldsymbol{\theta} - \boldsymbol{\theta}^\star), \qquad \boldsymbol{H}_i \succeq 0,\ i = 1, \cdots, n,$$

where all samples share the same zero minimizer $\boldsymbol{\theta}^\star$. Define $\boldsymbol{e} := \boldsymbol{\theta} - \boldsymbol{\theta}^\star$, then $\ell_i(\boldsymbol{e}) = \frac{1}{2}\boldsymbol{e}^\top \boldsymbol{H}_i \boldsymbol{e}$. The empirical average loss is

$$\hat{L}_n(\boldsymbol{e}) = \frac{1}{n}\sum_{i=1}^n \ell_i(\boldsymbol{e}) = \frac{1}{2}\boldsymbol{e}^\top \boldsymbol{H}\boldsymbol{e}, \quad \boldsymbol{H} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{H}_i.$$

Assume that $\boldsymbol{H} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{H}_i \succ 0$, and let $\mu = \lambda_{\min}(\boldsymbol{H}) > 0$ (typically, $\mu \ll 1$).

The standard gradient flow is

$$\dot{\boldsymbol{e}}_{\mathrm{u}} = -\nabla \hat{L}_n(\boldsymbol{e}_{\mathrm{u}}) = -\boldsymbol{H}\boldsymbol{e}_{\mathrm{u}}, \tag{B.1}$$

while the loss-weighted gradient flow is defined as

$$\dot{\boldsymbol{e}}_w = -\sum_{i=1}^n p_i(\boldsymbol{e}_w)\nabla \ell_i(\boldsymbol{e}_w), \quad p_i(\boldsymbol{e}) := \frac{\ell_i(\boldsymbol{e})}{\sum_{j=1}^n \ell_j(\boldsymbol{e})}.$$

Since $\nabla \ell_i(e) = \boldsymbol{H}_i \boldsymbol{e}$, we have

$$\dot{\boldsymbol{e}}_{\mathrm{w}} = -\frac{\sum_{i=1}^n \ell_i(\boldsymbol{e}_{\mathrm{w}})\boldsymbol{H}_i}{\sum_{j=1}^n \ell_j(\boldsymbol{e}_{\mathrm{w}})}\boldsymbol{e}_{\mathrm{w}}. \tag{B.2}$$

The aim is to quantitatively compare the standard gradient flow (B.1) and loss-weighted gradient flow (B.2), particularly in the convergence rate.

(i) For the standard gradient flow, (B.1) is a linear ODE, which can be solved explicitly. In fact, let $\boldsymbol{H} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$ be the spectral decomposition, we have

$$\boldsymbol{e}_{\mathrm{u}}(t) = \sum_{j=1}^d (\boldsymbol{e}_{\mathrm{u}}(0)^\top \boldsymbol{v}_j)\exp\{-\lambda_j t\}\boldsymbol{v}_j,$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \cdots, \lambda_d)$ ($\mu = \lambda_1 \le \cdots \le \lambda_d$) and $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_d]$ collect all eigenvalues and eigenvectors of $\boldsymbol{H}$. Therefore, if $\boldsymbol{e}_{\mathrm{u}}(0)^\top \boldsymbol{v}_1 \neq 0$, we obtain

$$\|\boldsymbol{e}_{\mathrm{u}}(t)\| \sim \exp\{-\mu t\}, \quad t \to +\infty. \tag{B.3}$$

(ii) For the loss-weighted gradient flow (B.2), we analyze this dynamics using a radial-directional decomposition. Write

$$\boldsymbol{e} = r\boldsymbol{u}, \quad r := \|\boldsymbol{e}\|,\ \boldsymbol{u} = \boldsymbol{e}/\|\boldsymbol{e}\|.$$

For each sample $i \in [n]$, define the directional curvature

$$a_i(\boldsymbol{u}) = \boldsymbol{u}^\top \boldsymbol{H}_i \boldsymbol{u},$$

we have $\ell_i(r\boldsymbol{u}) = \frac{1}{2}r^2 a_i(\boldsymbol{u})$, and $p_i(r\boldsymbol{u}) = a_i(\boldsymbol{u})/(\sum_{j=1}^n a_j(\boldsymbol{u}))$. That is, the weights depend only on the direction $\boldsymbol{u}$ and are completely independent of the radius r. Next, since $\dot{\boldsymbol{e}} = \dot{r}\,\boldsymbol{u} + r\dot{\boldsymbol{u}}$, and $\boldsymbol{u}^\top \boldsymbol{u} = 1$ leads to $\boldsymbol{u}^\top \dot{\boldsymbol{u}} = 0$, we get $\dot{r} = \boldsymbol{u}^\top \dot{\boldsymbol{e}}$. Then, since (B.2) can be rewritten as $\dot{\boldsymbol{e}}_{\mathrm{w}} = -\frac{\sum_{i=1}^n a_i(\boldsymbol{u}_{\mathrm{w}})\boldsymbol{H}_i}{\sum_{j=1}^n a_j(\boldsymbol{u}_{\mathrm{w}})} r_{\mathrm{w}}\boldsymbol{u}_{\mathrm{w}}$, we get

$$\dot{r}_{\mathrm{w}} = -\frac{\sum_{i=1}^n a_i^2(\boldsymbol{u}_{\mathrm{w}})}{\sum_{j=1}^n a_j(\boldsymbol{u}_{\mathrm{w}})} r_{\mathrm{w}} =: -\rho(\boldsymbol{u}_{\mathrm{w}}) r_{\mathrm{w}}.$$

Let $\bar{a}(\boldsymbol{u}) = \frac{1}{n}\sum_{i=1}^n a_i(\boldsymbol{u}) = \boldsymbol{u}^\top \boldsymbol{H}\boldsymbol{u}$, and define

$$\mathrm{Var}_i(a_i(\boldsymbol{u})) = \frac{1}{n}\sum_{i=1}^n \left(a_i(\boldsymbol{u}) - \bar{a}(\boldsymbol{u})\right)^2.$$

Using $\frac{1}{n}\sum_{i=1}^n a_i^2(\boldsymbol{u}) = \bar{a}(\boldsymbol{u})^2 + \mathrm{Var}_i(a_i(\boldsymbol{u}))$, we obtain

$$\rho(\boldsymbol{u}) = \frac{\bar{a}(\boldsymbol{u})^2 + \mathrm{Var}_i(a_i(\boldsymbol{u}))}{\bar{a}(\boldsymbol{u})} = \boldsymbol{u}^\top \boldsymbol{H}\boldsymbol{u} + \frac{\mathrm{Var}_i\big(\boldsymbol{u}^\top \boldsymbol{H}_i \boldsymbol{u}\big)}{\boldsymbol{u}^\top \boldsymbol{H}\boldsymbol{u}},$$

and consequently,

$$\frac{\dot{r}_{\mathrm{w}}}{r_{\mathrm{w}}} = -\boldsymbol{u}_{\mathrm{w}}^\top \boldsymbol{H}\boldsymbol{u}_{\mathrm{w}} - \frac{\mathrm{Var}_i\big(\boldsymbol{u}_{\mathrm{w}}^\top \boldsymbol{H}_i \boldsymbol{u}_{\mathrm{w}}\big)}{\boldsymbol{u}_{\mathrm{w}}^\top \boldsymbol{H}\boldsymbol{u}_{\mathrm{w}}}.$$

Let $\mu_{\mathrm{w}} = \min_{\|\boldsymbol{u}\|=1} \rho(\boldsymbol{u}) = \rho(\boldsymbol{u}_{\min})$, we have $\frac{\dot{r}_{\mathrm{w}}}{r_{\mathrm{w}}} \le -\mu_{\mathrm{w}}$, and integrating both sides gives

$$\|\boldsymbol{e}_{\mathrm{w}}(t)\| = r_{\mathrm{w}}(t) \le \|\boldsymbol{e}_{\mathrm{w}}(0)\| \exp\{-\mu_{\mathrm{w}} t\}. \tag{B.4}$$

The remaining is to compare μ_{w} and μ (Equation (B.3) v.s. Equation (B.4)), which completely characterize the (asymptotic) convergence rate. First, for any $\boldsymbol{u} : \|\boldsymbol{u}\| = 1$, we have $\rho(\boldsymbol{u}) \ge \boldsymbol{u}^\top \boldsymbol{H}\boldsymbol{u} \ge \lambda_{\min}(\boldsymbol{H}) \Rightarrow \mu_{\mathrm{w}} \ge \mu$, which implies the *non-deceleration* of the loss-weighted gradient flow. Additionally, if we further assume that $\mathrm{Var}_i\big(\boldsymbol{v}_{\min}^\top \boldsymbol{H}_i \boldsymbol{v}_{\min}\big) > 0$ for any eigenvector $\boldsymbol{v}_{\min}$ associated with the minimum eigenvalues of $\boldsymbol{H}$ (e.g. $\mu = \lambda_1 < \lambda_2$, $v_{\min} = v_1$), we can derive $\mu_{\mathrm{w}} > \mu$,[2] which implies the **strict acceleration** of the loss-weighted gradient flow. □

B.2 Proof of Proposition 3.2

Proof. Define $\boldsymbol{w}(t) := [w_i(t)]_{i\in[n]}$, $\boldsymbol{s}(t) := [s_i(t)]_{i\in[n]}$, and $\boldsymbol{l}(t) := [\ell_i(\boldsymbol{\theta}(t))]_{i\in[n]}$ for any $t \in \mathbb{N}$. The sampling scheme Equation (3.4) can be rewritten as

$$\begin{aligned} \boldsymbol{w}(t) &= \beta_1 \boldsymbol{s}(t-1) + (1-\beta_1)\boldsymbol{l}(t), \\ \boldsymbol{s}(t) &= \beta_2 \boldsymbol{s}(t-1) + (1-\beta_2)\boldsymbol{l}(t), \quad \boldsymbol{s}(0) = \mathbf{1}/n. \end{aligned} \tag{B.5}$$

In Equation (B.5), let the first equation minus the second, we get

$$\boldsymbol{w}(t) - \boldsymbol{s}(t) = (\beta_2 - \beta_1)(\boldsymbol{l}(t) - \boldsymbol{s}(t-1)). \tag{B.6}$$

The second equation gives

$$\boldsymbol{s}(t) - \boldsymbol{s}(t-1) = (1-\beta_2)(\boldsymbol{l}(t) - \boldsymbol{s}(t-1)). \tag{B.7}$$

Combining Equation (B.6) with Equation (B.7), we have

$$\boldsymbol{w}(t) = \boldsymbol{s}(t) + \frac{\beta_2 - \beta_1}{1-\beta_2}(\boldsymbol{s}(t) - \boldsymbol{s}(t-1)), \tag{B.8}$$

which proves the first equality.

Expanding the second equation, by induction we get

$$\boldsymbol{s}(t) = \beta_2^t \boldsymbol{s}(0) + (1-\beta_2)\sum_{k=1}^t \beta_2^{t-k}\boldsymbol{l}(k), \tag{B.9}$$

[2] In fact, if $\mu_{\mathrm{w}} = \mu$, i.e. $\rho(u_{\min}) = \lambda_{\min}(H)$, then we need both $\rho(u_{\min}) = u_{\min}^\top H u_{\min}$ and $u_{\min}^\top H u_{\min} = \lambda_{\min}(H)$. The first equality implies $\mathrm{Var}_i\big(u_{\min}^\top H_i u_{\min}\big) = 0$, while the second equality implies that $u_{\min}$ is an eigenvector associated with the minimum eigenvalues of H. This is a contradiction.

hence

$$\begin{aligned}\boldsymbol{s}(t)-\boldsymbol{s}(t-1) &= \beta_2^{t-1}(\beta_2-1)\boldsymbol{s}(0)+(1-\beta_2)\left[\sum_{k=1}^{t}\beta_2^{t-k}\boldsymbol{l}(k)-\sum_{k=1}^{t-1}\beta_2^{t-1-k}\boldsymbol{l}(k)\right]\\ &= -(1-\beta_2)\beta_2^{t-1}\boldsymbol{s}(0)+(1-\beta_2)\left[\beta_2^{t-1}\boldsymbol{l}(1)+\sum_{k=2}^{t}\beta_2^{t-k}\boldsymbol{l}(k)-\sum_{k=1}^{t-1}\beta_2^{t-1-k}\boldsymbol{l}(k)\right]\\ &= -(1-\beta_2)\beta_2^{t-1}\boldsymbol{s}(0)+(1-\beta_2)\left[\beta_2^{t-1}\boldsymbol{l}(1)+\sum_{k=1}^{t-1}\beta_2^{t-1-k}(\boldsymbol{l}(k+1)-\boldsymbol{l}(k))\right]\\ &\approx (1-\beta_2)\sum_{k=1}^{t-1}\beta_2^{t-1-k}(\boldsymbol{l}(k+1)-\boldsymbol{l}(k)) \qquad \text{(B.10)}\end{aligned}$$

for relatively large t, and the approximation error is exponentially small (due to $\lim_{t\to+\infty}\beta_2^t=0$ for any $\beta_2\in(0,1)$). Combining Equation (B.8), Equation (B.9) and Equation (B.10) yields Equation (3.5).

Given the stochastic gradient descent (SGD) training dynamics, the model parameters are updated by

$$\boldsymbol{\theta}(t+1)=\boldsymbol{\theta}(t)-\eta_t\sum_{j\in\mathcal{B}_t}p_j(t)\nabla_{\boldsymbol{\theta}}\ell_j(\boldsymbol{\theta}(t)), \qquad \text{(B.11)}$$

where $\{\eta_t\}_{t\in\mathbb{N}}$ denotes learning rates, and $\mathcal{B}_t\subset[n]$ denotes the subset of $\{1,2,\cdots,n\}$, with the (batch) size $|\mathcal{B}_t|=B$. Then, by Taylor expansion, we have

$$\begin{aligned}\ell_i(\boldsymbol{\theta}(k+1))-\ell_i(\boldsymbol{\theta}(k)) &= \left\langle\nabla_{\boldsymbol{\theta}}\ell_i(\boldsymbol{\theta}(k)),\boldsymbol{\theta}(k+1)-\boldsymbol{\theta}(k)\right\rangle+O(\|\boldsymbol{\theta}(k+1)-\boldsymbol{\theta}(k)\|_2^2) \qquad \text{(B.12)}\\ &= -\eta_k c_i(k)+O(\|\boldsymbol{\theta}(k+1)-\boldsymbol{\theta}(k)\|_2^2), \qquad \text{(B.13)}\end{aligned}$$

where $c_i(k):=\left\langle\nabla_{\boldsymbol{\theta}}\ell_i(\boldsymbol{\theta}(k)),\sum_{j\in\mathcal{B}_k}p_j(k)\nabla_{\boldsymbol{\theta}}\ell_j(\boldsymbol{\theta}(k))\right\rangle$ denotes the inner product between the i-th sample's gradient and full gradient at the k-th iteration, representing the individual-to-whole gradient "alignment" along training trajectories. This yields

$$\begin{aligned}&(\beta_2-\beta_1)\sum_{k=1}^{t-1}\beta_2^{t-1-k}(\ell_i(\boldsymbol{\theta}(k+1))-\ell_i(\boldsymbol{\theta}(k)))\\ =&-(\beta_2-\beta_1)\sum_{k=1}^{t-1}\beta_2^{t-1-k}[\eta_k c_i(k)+O(\|\boldsymbol{\theta}(k+1)-\boldsymbol{\theta}(k)\|_2^2)]\\ \overset{(i)}{\approx}&-(\beta_2-\beta_1)\sum_{k=1}^{t-1}\beta_2^{t-1-k}\eta_k c_i(k)-(\beta_2-\beta_1)\sum_{k=t-O(1)}\beta_2^{t-1-k}\cdot O(\|\boldsymbol{\theta}(k+1)-\boldsymbol{\theta}(k)\|_2^2)]\\ \overset{(ii)}{\approx}&-(\beta_2-\beta_1)\sum_{k=1}^{t-1}\beta_2^{t-1-k}\eta_k c_i(k), \qquad \text{(B.14)}\end{aligned}$$

where (i) is due to the fact that β_2^s ($\beta_2\in(0,1)$) is exponentially small for relatively large $s>0$ and $t\gg1$, and (ii) is a consequence of convergence. The proof is completed. □

B.3 Proof of Theorem 3.3

We begin by proving some lemmas. The first two lemmas and their proofs are standard, and can be found in e.g. Garrigos & Gower (2023).

Lemma B.1. *If $f:\mathbb{R}^d\mapsto\mathbb{R}$ is L-smooth, then for any $\boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^d$, we have*

$$f(\boldsymbol{y})\le f(\boldsymbol{x})+\langle\nabla f(\boldsymbol{x}),\boldsymbol{y}-\boldsymbol{x}\rangle+\frac{L}{2}\|\boldsymbol{y}-\boldsymbol{x}\|_2^2. \qquad \text{(B.15)}$$

Proof. For any fixed $\boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^d$, let $\phi(t):=f(\boldsymbol{x}+t(\boldsymbol{y}-\boldsymbol{x}))$. Then we have

$$f(\boldsymbol{y})-f(\boldsymbol{x})=\phi(1)-\phi(0)=\int_0^1\phi'(t)\mathrm{d}t=\int_0^1\langle\nabla f(\boldsymbol{x}+t(\boldsymbol{y}-\boldsymbol{x})),\boldsymbol{y}-\boldsymbol{x}\rangle\,\mathrm{d}t$$

$$
\begin{aligned}
&= \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \int_0^1 \langle \nabla f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle \, \mathrm{d}t \\
&\leq \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \int_0^1 \|\nabla f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x})\|_2 \|\boldsymbol{y} - \boldsymbol{x}\|_2 \mathrm{d}t \\
&\leq \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \int_0^1 Lt \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \mathrm{d}t \\
&= \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{L}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2, \qquad \text{(B.16)}
\end{aligned}
$$

where Cauchy–Schwarz inequality and the L-smoothness property are successively applied. The proof is completed. □

Lemma B.2. *If $f : \mathbb{R}^d \mapsto \mathbb{R}$ is convex and L-smooth, then for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, we have*

$$
\frac{1}{2L} \|\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x})\|_2^2 \leq f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle . \qquad \text{(B.17)}
$$

Proof. Fix any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$. By convexity and L-smoothness, for any $\boldsymbol{z} \in \mathbb{R}^d$, we have

$$
\begin{aligned}
f(\boldsymbol{x}) - f(\boldsymbol{y}) &= f(\boldsymbol{x}) - f(\boldsymbol{z}) + f(\boldsymbol{z}) - f(\boldsymbol{y}) \\
&\leq \langle \nabla f(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{z} \rangle + \langle \nabla f(\boldsymbol{y}), \boldsymbol{z} - \boldsymbol{y} \rangle + \frac{L}{2} \|\boldsymbol{z} - \boldsymbol{y}\|_2^2. \qquad \text{(Lemma B.1)} \qquad \text{(B.18)}
\end{aligned}
$$

Take $\boldsymbol{z} = \boldsymbol{y} - (\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x}))/L$ to minimize the right hand side, we get

$$
f(\boldsymbol{x}) - f(\boldsymbol{y}) \leq \langle \nabla f(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{y} \rangle - \frac{1}{2L} \|\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x})\|_2^2, \qquad \text{(B.19)}
$$

which completes the proof. □

We also need the following norm estimate on the product between matrices and probability-like vectors.

Lemma B.3. *For any matrix $\boldsymbol{G} := [\boldsymbol{g}_1, \cdots, \boldsymbol{g}_n] \in \mathbb{R}^{m \times n}$, and any vector $\boldsymbol{p} := [p_1, \cdots, p_n]^\top \in \mathbb{R}^n$ satisfying $\sum_{i=1}^n p_i \leq 1$ with $p_i \geq 0$ for all $i \in [n]$, we have*

$$
\|\boldsymbol{G}\boldsymbol{p}\|_2^2 = \left\| \sum_{i=1}^n p_i \boldsymbol{g}_i \right\|_2^2 \leq \sum_{i=1}^n p_i \|\boldsymbol{g}_i\|_2^2. \qquad \text{(B.20)}
$$

Proof. It is straightforward to verify that

$$
\begin{aligned}
\|\boldsymbol{G}\boldsymbol{p}\|_2^2 &= \left(\sum_{i=1}^n p_i \boldsymbol{g}_i \right)^\top \sum_{j=1}^n p_j \boldsymbol{g}_j = \sum_{i=1}^n \sum_{j=1}^n p_i p_j \boldsymbol{g}_i^\top \boldsymbol{g}_j \\
&\leq \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n p_i p_j (\|\boldsymbol{g}_i\|_2^2 + \|\boldsymbol{g}_j\|_2^2) \leq \sum_{i=1}^n p_i \|\boldsymbol{g}_i\|_2^2, \qquad \text{(B.21)}
\end{aligned}
$$

which completes the proof. □

Now we are ready to prove the main theorem (Theorem 3.3).

Proof. First, we prove the convergence rate. Given the SGD training dynamics

$$
\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta_t \sum_{j \in \mathcal{B}_t} p_j(t) \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}(t)),
$$

we have

$$
\begin{aligned}
\|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}^*\|_2^2 &= \|(\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)) + (\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*)\|_2^2 \\
&= \|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\|_2^2 + \|\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*\|_2^2 + 2 \langle \boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t), \boldsymbol{\theta}(t) - \boldsymbol{\theta}^* \rangle
\end{aligned}
$$

$$= \|\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*\|_2^2 + \left\| \eta_t \sum_{j \in \mathcal{B}_t} p_j(t) \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}(t)) \right\|_2^2 - 2 \left\langle \eta_t \sum_{j \in \mathcal{B}_t} p_j(t) \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}(t)), \boldsymbol{\theta}(t) - \boldsymbol{\theta}^* \right\rangle$$
$$=: \|\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*\|_2^2 + I_1 + I_2. \tag{B.22}$$

For I_1, by Lemma B.3 we have

$$I_1 = \eta_t^2 \left\| \sum_{j \in \mathcal{B}_t} p_j(t) \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}(t)) \right\|_2^2 \leq \eta_t^2 \sum_{j \in \mathcal{B}_t} p_j(t) \|\nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}(t))\|_2^2. \tag{B.23}$$

Due to the optimality of $\boldsymbol{\theta}^*$, we have $\nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}^*) = 0$ for any $i \in [n]$. Recall that $\ell_i(\cdot)$ is L_i-smooth, we derive by Lemma B.2 that

$$\begin{aligned} \|\nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}(t))\|_2^2 &= \|\nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}(t)) - \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}^*)\|_2^2 \\ &\leq 2L_j \left[\ell_j(\boldsymbol{\theta}(t)) - \ell_j(\boldsymbol{\theta}^*) - \langle \nabla \ell_j(\boldsymbol{\theta}^*), \boldsymbol{\theta}(t) - \boldsymbol{\theta}^* \rangle \right] \\ &= 2L_j \left[\ell_j(\boldsymbol{\theta}(t)) - \ell_j(\boldsymbol{\theta}^*) \right], \end{aligned} \tag{B.24}$$

which gives

$$I_1 \leq 2\eta_t^2 \sum_{j \in \mathcal{B}_t} L_j p_j(t) \left[\ell_j(\boldsymbol{\theta}(t)) - \ell_j(\boldsymbol{\theta}^*) \right]. \tag{B.25}$$

For I_2, by convexity we have

$$\begin{aligned} I_2 &= 2\eta_t \sum_{j \in \mathcal{B}_t} p_j(t) \left\langle \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}(t)), \boldsymbol{\theta}^* - \boldsymbol{\theta}(t) \right\rangle \\ &\leq 2\eta_t \sum_{j \in \mathcal{B}_t} p_j(t) (\ell_j(\boldsymbol{\theta}^*) - \ell_j(\boldsymbol{\theta}(t))) \\ &= -2\eta_t \sum_{j \in \mathcal{B}_t} p_j(t) \left[\ell_j(\boldsymbol{\theta}(t)) - \ell_j(\boldsymbol{\theta}^*) \right]. \end{aligned} \tag{B.26}$$

Therefore, we obtain

$$\|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}^*\|_2^2 \leq \|\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*\|_2^2 + 2\eta_t \sum_{j \in \mathcal{B}_t} (\eta_t L_j - 1) p_j(t) \left[\ell_j(\boldsymbol{\theta}(t)) - \ell_j(\boldsymbol{\theta}^*) \right]. \tag{B.27}$$

Let $L := \max_{i \in [n]} L_i$, and set $\eta_t \leq 1/(2L)$. Then we have

$$\|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}^*\|_2^2 \leq \|\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*\|_2^2 - \eta_t \sum_{j \in \mathcal{B}_t} p_j(t) \left[\ell_j(\boldsymbol{\theta}(t)) - \ell_j(\boldsymbol{\theta}^*) \right] \leq \|\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*\|_2^2, \tag{B.28}$$

which yields that the sequence $\left\{ \|\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*\|_2^2 \right\}_{t \in \mathbb{N}}$ is decreasing and bounded.[3] Take the expectation (conditioned on $(\boldsymbol{\theta}(s))_{s \leq t}$) over both sides of Equation (B.27), we have

$$\begin{aligned} \mathbb{E} \|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}^*\|_2^2 &\leq \|\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*\|_2^2 + 2\eta_t \sum_{j \in \mathcal{B}_t} (\eta_t L_j - 1) p_j(t) \left[\ell_j(\boldsymbol{\theta}(t)) - \ell_j(\boldsymbol{\theta}^*) \right] \\ &\leq \|\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*\|_2^2 - \eta_t \sum_{j \in \mathcal{B}_t} p_j(t) \left[\ell_j(\boldsymbol{\theta}(t)) - \ell_j(\boldsymbol{\theta}^*) \right] \\ &= \|\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*\|_2^2 - \eta_t \frac{1}{B} \sum_{j \in \mathcal{B}_t} \left[\ell_j(\boldsymbol{\theta}(t)) - \ell_j(\boldsymbol{\theta}^*) \right] \\ &\quad - \eta_t \sum_{j \in \mathcal{B}_t} \left(p_j(t) - \frac{1}{B} \right) \left[\ell_j(\boldsymbol{\theta}(t)) - \ell_j(\boldsymbol{\theta}^*) \right], \end{aligned} \tag{B.29}$$

[3] That is, all $\ell_i(\cdot)$ can be defined on a compact set, and are also bounded due to the smoothness/continuity. Let $M > 1/n$ such that $\ell_i(\cdot) \leq M$, then by the sampling scheme Equation (3.4), we have both $s_i(\cdot) \leq M$ and $w_i(\cdot) \leq M$.

and by law of total expectation,

$$\begin{aligned}\mathbb{E}\left\|\boldsymbol{\theta}(t+1)-\boldsymbol{\theta}^*\right\|_2^2 &\le \mathbb{E}\left\|\boldsymbol{\theta}(t)-\boldsymbol{\theta}^*\right\|_2^2-\eta_t\mathbb{E}\frac{1}{B}\sum_{j\in\mathcal{B}_t}\left[\ell_j(\boldsymbol{\theta}(t))-\ell_j(\boldsymbol{\theta}^*)\right]\\ &\quad-\eta_t\mathbb{E}\sum_{j\in\mathcal{B}_t}\left(p_j(t)-\frac{1}{B}\right)\left[\ell_j(\boldsymbol{\theta}(t))-\ell_j(\boldsymbol{\theta}^*)\right]\\ &=\mathbb{E}\left\|\boldsymbol{\theta}(t)-\boldsymbol{\theta}^*\right\|_2^2-\eta_t\left[\hat{L}_n(\boldsymbol{\theta}(t))-\hat{L}_n(\boldsymbol{\theta}^*)\right]\\ &\quad-\eta_t\mathbb{E}\sum_{j\in\mathcal{B}_t}\left(p_j(t)-\frac{1}{B}\right)\left[\ell_j(\boldsymbol{\theta}(t))-\ell_j(\boldsymbol{\theta}^*)\right].\end{aligned}\tag{B.30}$$

Then by telescoping sum, we obtain

$$\begin{aligned}\mathbb{E}\left\|\boldsymbol{\theta}(T)-\boldsymbol{\theta}^*\right\|_2^2 &\le \mathbb{E}\left\|\boldsymbol{\theta}(0)-\boldsymbol{\theta}^*\right\|_2^2-\sum_{t=0}^{T-1}\eta_t\left[\hat{L}_n(\boldsymbol{\theta}(t))-\hat{L}_n(\boldsymbol{\theta}^*)\right]\\ &\quad-\sum_{t=0}^{T-1}\eta_t\mathbb{E}\sum_{j\in\mathcal{B}_t}\left(p_j(t)-\frac{1}{B}\right)\left[\ell_j(\boldsymbol{\theta}(t))-\ell_j(\boldsymbol{\theta}^*)\right],\end{aligned}\tag{B.31}$$

which yields

$$\sum_{t=0}^{T-1}\eta_t\left[\hat{L}_n(\boldsymbol{\theta}(t))-\hat{L}_n(\boldsymbol{\theta}^*)\right]\le\mathbb{E}\left\|\boldsymbol{\theta}(0)-\boldsymbol{\theta}^*\right\|_2^2-\sum_{t=0}^{T-1}\eta_t R(t),\tag{B.32}$$

where $R(t):=\mathbb{E}\sum_{j\in\mathcal{B}_t}\left(p_j(t)-\frac{1}{B}\right)\left[\ell_j(\boldsymbol{\theta}(t))-\ell_j(\boldsymbol{\theta}^*)\right]$ denotes the remainder. Therefore

$$\sum_{t=0}^{T-1}\frac{\eta_t}{\sum_{s=0}^{T-1}\eta_s}\left[\hat{L}_n(\boldsymbol{\theta}(t))-\hat{L}_n(\boldsymbol{\theta}^*)\right]\le\frac{\mathbb{E}\left\|\boldsymbol{\theta}(0)-\boldsymbol{\theta}^*\right\|_2^2}{\sum_{s=0}^{T-1}\eta_s}-\sum_{t=0}^{T-1}\frac{\eta_t}{\sum_{s=0}^{T-1}\eta_s}R(t),\tag{B.33}$$

and by convexity

$$\hat{L}_n(\bar{\boldsymbol{\theta}}_T)-\hat{L}_n(\boldsymbol{\theta}^*)\le\frac{\mathbb{E}\left\|\boldsymbol{\theta}(0)-\boldsymbol{\theta}^*\right\|_2^2}{\sum_{s=0}^{T-1}\eta_s}-\sum_{t=0}^{T-1}\frac{\eta_t}{\sum_{s=0}^{T-1}\eta_s}R(t),\tag{B.34}$$

with $\bar{\boldsymbol{\theta}}_T:=\sum_{t=0}^{T-1}\frac{\eta_t}{\sum_{s=0}^{T-1}\eta_s}\boldsymbol{\theta}(t)$. For $\eta_t\equiv\eta=1/(2L)$, Equation (B.34) gives

$$\hat{L}_n\left(\frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{\theta}(t)\right)-\hat{L}_n(\boldsymbol{\theta}^*)\le\frac{2L\mathbb{E}\left\|\boldsymbol{\theta}(0)-\boldsymbol{\theta}^*\right\|_2^2}{T}-\frac{1}{T}\sum_{t=0}^{T-1}R(t).\tag{B.35}$$

Next, we provide some *sufficient* conditions to bound the remainder term $\frac{1}{T}\sum_{t=0}^{T-1}R(t)$ from below:

(i) Order consistency: For instance, in the sampling scheme Equation (3.4), there exists (β_1,β_2) such that

$$(w_i(t)-w_j(t))(\ell_i(\boldsymbol{\theta}(t))-\ell_j(\boldsymbol{\theta}(t)))\ge 0\tag{B.36}$$

for any $i,j\in[n]$ and $t\in\mathbb{N}$ (e.g. when $\beta_1\to 0^+$). That is, when one sample's loss is larger/smaller than that of the other, so does its weight. With this condition, for any $t\in\mathbb{N}$, we have

$$\begin{aligned}0&\le\sum_{i\in\mathcal{B}_t}\sum_{j\in\mathcal{B}_t}(p_i(t)-p_j(t))(\ell_i(\boldsymbol{\theta}(t))-\ell_j(\boldsymbol{\theta}(t)))\\ &=\sum_{i\in\mathcal{B}_t}\sum_{j\in\mathcal{B}_t}(p_i(t)\ell_i(\boldsymbol{\theta}(t))+p_j(t)\ell_j(\boldsymbol{\theta}(t))-p_i(t)\ell_j(\boldsymbol{\theta}(t))-p_j(t)\ell_i(\boldsymbol{\theta}(t)))\\ &=B\sum_{i\in\mathcal{B}_t}p_i(t)\ell_i(\boldsymbol{\theta}(t))+B\sum_{j\in\mathcal{B}_t}p_j(t)\ell_j(\boldsymbol{\theta}(t))-\sum_{i\in\mathcal{B}_t}p_i(t)\sum_{j\in\mathcal{B}_t}\ell_j(\boldsymbol{\theta}(t))-\sum_{i\in\mathcal{B}_t}\ell_i(\boldsymbol{\theta}(t))\sum_{j\in\mathcal{B}_t}p_j(t)\end{aligned}$$

$$= 2B \sum_{i\in\mathcal{B}_t} p_i(t)\ell_i(\boldsymbol{\theta}(t)) - 2\sum_{i\in\mathcal{B}_t} p_i(t) \sum_{i\in\mathcal{B}_t} \ell_i(\boldsymbol{\theta}(t)) = 2B \sum_{i\in\mathcal{B}_t} p_i(t)\ell_i(\boldsymbol{\theta}(t)) - 2\sum_{i\in\mathcal{B}_t} \ell_i(\boldsymbol{\theta}(t)), \tag{B.37}$$

which gives

$$\frac{1}{B}\sum_{i\in\mathcal{B}_t} \ell_i(\boldsymbol{\theta}(t)) \le \sum_{i\in\mathcal{B}_t} p_i(t)\ell_i(\boldsymbol{\theta}(t)). \tag{B.38}$$

Hence, by the fact that $\hat{L}_n(\boldsymbol{\theta}^*) = 0 \Leftrightarrow \ell_i(\boldsymbol{\theta}^*) = 0, \forall i \in [n]$, we get

$$R(t) = \mathbb{E}\left[\sum_{j\in\mathcal{B}_t} p_j(t)\ell_j(\boldsymbol{\theta}(t)) - \frac{1}{B}\sum_{j\in\mathcal{B}_t} \ell_j(\boldsymbol{\theta}(t))\right] \ge 0. \tag{B.39}$$

(ii) Positive correlation: For instance, in the sampling scheme Equation (3.4), there exists (β_1, β_2) such that

$$\mathrm{Cov}(\{w_i(t)\}_{i\in\mathcal{B}_t}, \{\ell_i(\boldsymbol{\theta}(t))\}_{i\in\mathcal{B}_t}) \ge 0 \tag{B.40}$$

for any $t \in \mathbb{N}$ (e.g. when $\beta_1 \to 0^+$). That is, sample's weights are positively correlated with their losses. With this condition, for any $t \in \mathbb{N}$, we have

$$\begin{aligned}
0 &\le \sum_{i\in\mathcal{B}_t} \left(w_i(t) - \frac{1}{B}\sum_{j\in\mathcal{B}_t} w_j(t)\right)\left(\ell_i(\boldsymbol{\theta}(t)) - \frac{1}{B}\sum_{j\in\mathcal{B}_t} \ell_j(\boldsymbol{\theta}(t))\right) \\
&= \sum_{j\in\mathcal{B}_t} w_j(t) \cdot \sum_{i\in\mathcal{B}_t} \left(p_i(t) - \frac{1}{B}\right)\left(\ell_i(\boldsymbol{\theta}(t)) - \frac{1}{B}\sum_{j\in\mathcal{B}_t} \ell_j(\boldsymbol{\theta}(t))\right) \\
&= \sum_{j\in\mathcal{B}_t} w_j(t) \cdot \left(\sum_{i\in\mathcal{B}_t} \left(p_i(t) - \frac{1}{B}\right)\ell_i(\boldsymbol{\theta}(t)) - \sum_{i\in\mathcal{B}_t} \left(p_i(t) - \frac{1}{B}\right)\cdot\frac{1}{B}\sum_{j\in\mathcal{B}_t} \ell_j(\boldsymbol{\theta}(t))\right) \\
&= \sum_{j\in\mathcal{B}_t} w_j(t) \cdot \sum_{i\in\mathcal{B}_t} \left(p_i(t) - \frac{1}{B}\right)\ell_i(\boldsymbol{\theta}(t)),
\end{aligned} \tag{B.41}$$

which also yields $R(t) \ge 0$.

Second, we prove the generalization gap. We have

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{z}}\left[\ell(\boldsymbol{z};\boldsymbol{\theta}(t))\right] - \sum_{i=1}^n p_i(t)\ell(\boldsymbol{z}_i;\boldsymbol{\theta}(t)) \\
&= \left(\mathbb{E}_{\boldsymbol{z}}\left[\ell(\boldsymbol{z};\boldsymbol{\theta}(t))\right] - \frac{1}{n}\sum_{i=1}^n \ell(\boldsymbol{z}_i;\boldsymbol{\theta}(t))\right) + \left(\frac{1}{n}\sum_{i=1}^n \ell(\boldsymbol{z}_i;\boldsymbol{\theta}(t)) - \sum_{i=1}^n p_i(t)\ell(\boldsymbol{z}_i;\boldsymbol{\theta}(t))\right) \\
&= \left(\mathbb{E}_{\boldsymbol{z}}\left[\ell(\boldsymbol{z};\boldsymbol{\theta}(t))\right] - \frac{1}{n}\sum_{i=1}^n \ell(\boldsymbol{z}_i;\boldsymbol{\theta}(t))\right) - \left(\sum_{i=1}^n p_i(t)\ell(\boldsymbol{z}_i;\boldsymbol{\theta}(t)) - \frac{1}{n}\sum_{i=1}^n \ell(\boldsymbol{z}_i;\boldsymbol{\theta}(t))\right) \\
&\le \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\left\{\mathbb{E}_{\boldsymbol{z}}\left[\ell(\boldsymbol{z};\boldsymbol{\theta})\right] - \frac{1}{n}\sum_{i=1}^n \ell(\boldsymbol{z}_i;\boldsymbol{\theta})\right\} - \left(\sum_{i=1}^n p_i(t)\ell(\boldsymbol{z}_i;\boldsymbol{\theta}(t)) - \frac{1}{n}\sum_{i=1}^n \ell(\boldsymbol{z}_i;\boldsymbol{\theta}(t))\right).
\end{aligned} \tag{B.42}$$

For the first term, according to Shalev-Shwartz & Ben-David (2014) (Theorem 26.5), for any $\delta > 0$, with probability of at least $1-\delta$, we have

$$\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\left\{\mathbb{E}_{\boldsymbol{z}}\left[\ell(\boldsymbol{z};\boldsymbol{\theta})\right] - \frac{1}{n}\sum_{i=1}^n \ell(\boldsymbol{z}_i;\boldsymbol{\theta})\right\} \le 2\mathrm{Rad}_{\mathcal{D}}(\boldsymbol{\Theta}) + 4M\sqrt{\frac{2\ln(4/\delta)}{n}}, \tag{B.43}$$

where $\mathrm{Rad}_{\mathcal{D}}(\boldsymbol{\Theta}) \overset{\text{def}}{=} \frac{1}{n}\mathbb{E}_{\boldsymbol{\sigma}\sim\{\pm1\}^n}\left[\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\sum_{i=1}^n \sigma_i\ell(\boldsymbol{z}_i;\boldsymbol{\theta})\right]$ denotes the (empirical) Rademacher complexity ($\{\sigma_i\}_{i=1}^n$ are i.i.d. variables with $\mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = 1/2$), and M is the upper bound of losses. This is exactly the generalization gap in the standard setting. For the second term, it is in fact a full-batch version of $-R(t)$ (see Equation (B.39)), and hence similarly non-positive given sufficient conditions like order consistency or positive correlation. The proof is completed. □

Remark 2. *We emphasize again that the conditions like order consistency Equation* (B.36) *or positive correlation Equation* (B.40) *proposed in the above proof are* provable *and* sufficient*: $\exists(\beta_1, \beta_2)$ s.t. Equation* (B.36) *or Equation* (B.40) *holds $\Rightarrow R(t) \geq 0 \Rightarrow \frac{1}{T}\sum_{t=0}^{T-1} R(t) \geq 0$, while the reverse does not hold. That is, to guarantee $\frac{1}{T}\sum_{t=0}^{T-1} R(t) \geq 0$, one can include more betas, where Equation* (B.36) *or Equation* (B.40) *does not necessarily hold (e.g. for $\{R(t)\}_{t\in\mathbb{N}}$ as an alternating series with positive elements larger than negative elements' absolute values). We plot the dynamics of $R(t)$ in Figure 7 (training ResNet-50 on CIFAR-100), numerically validating its positivity during the whole training process.*

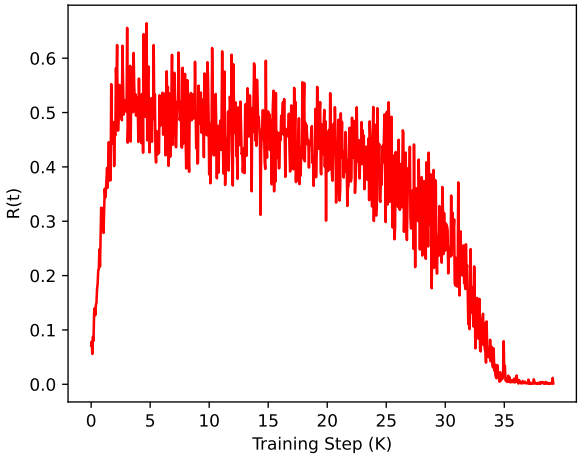


Figure 7: Dynamics of the per-step remainder $R(t) \geq 0$ when training ResNet-50 on CIFAR-100.

Remark 3. *Discussions on theoretical assumptions in Theorem 3.3:*

- *The convexity and L-smoothness conditions are widely-adopted and also standard in proving the convergence of (S)GD in optimization literature (e.g. Garrigos & Gower (2023)). Under the non-convex setting, typically the* convergence to only stationary points *can be theoretically guaranteed for general smooth functions (see e.g. Khaled & Richtárik (2023)) even for standard training (no data selection). We provide extensive experiments in Section 4 to numerically verify the convergence under practical scenarios.*
- *For the condition $\hat{L}_n(\boldsymbol{\theta}^*) = 0$, it simply means that the optimal training loss can be zero. There is empirical evidence to support this assumption for training neural networks (see e.g. Figure 1 (a) in Zhang et al. (2017)).*

B.4 Fourier Analysis on Evolved Sampling

In this section, we provide mathematical justifications for the sampling scheme Equation (3.4) by characterizing its frequency properties. To achieve this, we first view $\ell(t)(:= \ell(\boldsymbol{\theta}(t)))$, $s(t)$, $w(t)$ defined in the sampling scheme Equation (3.4) all as signals in time. For ease of notation, we omit the sample index i here. For a signal in time $f(t)$ (with appropriate regularities), we consider the Laplace transform $\mathcal{L}\{f\}(\omega) = \int_0^\infty e^{-\omega t} f(t)\mathrm{d}t$, $\omega \in \mathbb{C}$. Then, according to the Fourier analysis, $|\mathcal{L}\{f\}(\mathrm{i}\omega_0)|$ represents the magnitude of f's ω_0-frequency for $\omega_0 > 0$ ($\mathrm{i}^2 = -1$). We have the following result.

Theorem B.4. *Consider a continuous-time idealization of the sampling scheme Equation* (3.4)*:*

$$w(t) = s(t) + \frac{\beta_2 - \beta_1}{1 - \beta_2} s'(t), \qquad s'(t) = (1 - \beta_2)(\ell(t) - s(t)), \tag{B.44}$$

with $s(0) = 1/n$, and $\beta_1, \beta_2 \in (0, 1)$ as two hyper-parameters. Then we have

$$\mathcal{L}\{w\}(\omega) = \frac{(\beta_2 - \beta_1)\omega + (1 - \beta_2)}{\omega + (1 - \beta_2)} \mathcal{L}\{\ell\}(\omega) + O(1/n), \tag{B.45}$$

implying that the transfer function $H(\omega) = \frac{(\beta_2-\beta_1)\omega+(1-\beta_2)}{\omega+(1-\beta_2)}$ *satisfies*

$$|H(\mathrm{i}\omega_0)| \leq 1, \quad \lim_{\omega_0 \to +\infty} |H(\mathrm{i}\omega_0)| = |\beta_2 - \beta_1|. \tag{B.46}$$

Proof. Consider a continuous-time idealization of the sampling scheme Equation (3.4):

$$s(t) = \beta_2 s(t-1) + (1-\beta_2)\ell(t) \Rightarrow s(t) - s(t-1) = (1-\beta_2)(\ell(t) - s(t-1)) \tag{B.47}$$

$$\Rightarrow s'(t) = (1-\beta_2)(\ell(t) - s(t)), \tag{B.48}$$

with $\ell(t) := \ell(\boldsymbol{\theta}(t))$, and $\beta_2 \neq 0$. Similarly,

$$\begin{aligned} w(t) = \beta_1 s(t-1) + (1-\beta_1)\ell(t) &\Rightarrow w(t) - s(t) = (\beta_2 - \beta_1)(\ell(t) - s(t-1)) \\ &\Rightarrow w(t) = s(t) + (\beta_2 - \beta_1)\frac{s(t) - s(t-1)}{1-\beta_2} \quad \text{(by Equation (B.47))} \\ &\Rightarrow w(t) = s(t) + \frac{\beta_2 - \beta_1}{1-\beta_2} s'(t) \\ &\Rightarrow w(t) = (\beta_2 - \beta_1)\ell(t) + (1 - \beta_2 + \beta_1)s(t). \quad \text{(by Equation (B.48))} \end{aligned} \tag{B.49}$$

Since $\mathcal{L}\{\cdot\}$ is linear and satisfies $\mathcal{L}\{f'\}(\omega) = \omega\mathcal{L}\{f\}(\omega) - f(0)$, we have

$$\begin{aligned} \text{Equation (B.48)} \Rightarrow \mathcal{L}\{s'\}(\omega) &= (1-\beta_2)(\mathcal{L}\{\ell\}(\omega) - \mathcal{L}\{s\}(\omega)) = \omega\mathcal{L}\{s\}(\omega) - s(0) \\ \Rightarrow \mathcal{L}\{s\}(\omega) &= \frac{1-\beta_2}{\omega + (1-\beta_2)}\mathcal{L}\{\ell\}(\omega) + \frac{s(0)}{\omega + (1-\beta_2)}, \end{aligned} \tag{B.50}$$

and

$$\begin{aligned} \text{Equation (B.49)} \Rightarrow \mathcal{L}\{w\}(\omega) &= (\beta_2 - \beta_1)\mathcal{L}\{\ell\}(\omega) + (1 - \beta_2 + \beta_1)\mathcal{L}\{s\}(\omega) \\ &= \frac{(\beta_2 - \beta_1)\omega + (1-\beta_2)}{\omega + (1-\beta_2)}\mathcal{L}\{\ell\}(\omega) + \frac{(1 - \beta_2 + \beta_1)}{\omega + (1-\beta_2)}s(0), \quad \text{(by Equation (B.50))} \end{aligned} \tag{B.51}$$

$$\begin{aligned} &= \frac{(\beta_2 - \beta_1)\omega + (1-\beta_2)}{\omega + (1-\beta_2)}\mathcal{L}\{\ell\}(\omega) + \mathcal{L}\big\{(1 - \beta_2 + \beta_1)s(0) \cdot e^{-(1-\beta_2)t}\big\}(\omega) \\ &= \frac{(\beta_2 - \beta_1)\omega + (1-\beta_2)}{\omega + (1-\beta_2)}\mathcal{L}\{\ell\}(\omega) + O(1/n). \quad (\text{recall } s(0) = 1/n) \end{aligned} \tag{B.52}$$

Then, the transfer function is $H(\omega) = \frac{(\beta_2-\beta_1)\omega+(1-\beta_2)}{\omega+(1-\beta_2)}$, with

$$|H(\mathrm{i}\omega_0)| = \left|\frac{(\beta_2 - \beta_1)\mathrm{i}\omega_0 + (1-\beta_2)}{\mathrm{i}\omega_0 + (1-\beta_2)}\right| = \sqrt{\frac{(\beta_2 - \beta_1)^2\omega_0^2 + (1-\beta_2)^2}{\omega_0^2 + (1-\beta_2)^2}}, \tag{B.53}$$

and

$$|H(\mathrm{i}\omega_0)| \leq 1, \quad \lim_{\omega_0 \to +\infty} |H(\mathrm{i}\omega_0)| = |\beta_2 - \beta_1|. \tag{B.54}$$

The proof is completed. □

Based on Theorem B.4, we conclude that (i) for all frequencies in the loss signal $\ell(t) = \ell(\boldsymbol{\theta}(t))$, the weight signal $w(t)$ calculated by the sampling scheme Equation (3.4) does not enlarge them, hence is more stable in the frequency domain given oscillations in loss signals; (ii) for high frequencies in the loss signal $\ell(t) = \ell(\boldsymbol{\theta}(t))$, the weight signal $w(t)$ calculated by the sampling scheme Equation (3.4) reserves a $|\beta_2 - \beta_1|$-portion, which can be tuned via betas (frequency tuning). This suggests that the sampling scheme Equation (3.4) can not only stably capture the overall trend (low frequency), but also flexibly tune the portion of details (high frequency) in loss signals. See illustrations in Figure 8.

B.5 ES TO SOLVE A DRO PROBLEM

From another perspective, ES can be also reformulated as a solution to a distributionally robust optimization (DRO) problem, or more specifically the minimax optimization problem

$$\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \max_{\boldsymbol{p}\in\Delta^n} L_n(\boldsymbol{\theta};\boldsymbol{p}) := \sum_{i=1}^{n} p_i(\ell_i(\boldsymbol{\theta}) - \ell_i^{\text{ref}}), \tag{B.55}$$

where Δ^n denotes the $(n-1)$-dimensional probability simplex. This objective leads to a stronger requirement for robust performances on both typical and rare samples compared to the regular ERM (Shalev-Shwartz & Wexler (2016)). Different from traditional DRO, Equation (B.55) introduces a reference loss ℓ_i^{ref}, with the excess loss $\ell_i(\boldsymbol{\theta}) - \ell_i^{\text{ref}}$ measuring the improvement of the model on the i-th sample with respect to a reference model (typically *pre-trained*; see e.g. Oren et al. (2019); Xie et al. (2023a); Mindermann et al. (2022)). The second advantage of ES is to naturally leverage losses of historical models along the training dynamics as a proxy of the reference loss ℓ_i^{ref} in Equation (B.55), which can be continuously updated without explicitly (pre-)training additional models.

Specifically, we have the following proposition.

Proposition B.5. *Consider to solve the minimax objective Equation* (B.55) *via gradient ascent-descent*

$$\begin{cases} \boldsymbol{p}(t) \propto \boldsymbol{w}(t) := \boldsymbol{w}(t-1) + (1-\beta_1)(\boldsymbol{\ell}(\boldsymbol{\theta}(t)) - \boldsymbol{\ell}^{ref}(\boldsymbol{\theta}(1:t-1))), \\ \boldsymbol{\theta}(t+1) := \boldsymbol{\theta}(t) - \eta_t^{\boldsymbol{\theta}} \sum_{i=1}^{n} p_i(t) \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}(t)), \end{cases} \tag{B.56}$$

where the reference loss is defined as $\boldsymbol{\ell}^{ref}(\boldsymbol{\theta}(1:t)) := [\ell_i^{ref}(\boldsymbol{\theta}(1:t))]_{i\in[n]}$ *with* $\ell_i^{ref}(\boldsymbol{\theta}(1:t)) := \frac{1-2\beta_1+\beta_1\beta_2}{1-\beta_1}\ell_i(\boldsymbol{\theta}(t)) + \frac{\beta_1(1-\beta_2)^2}{1-\beta_1}\sum_{k=1}^{t-1}\beta_2^{t-1-k}\ell_i(\boldsymbol{\theta}(k)) + \frac{\beta_1(1-\beta_2)\beta_2^{t-1}}{n(1-\beta_1)}$, $i \in [n]$. *Then, the dynamics Equation* (B.56) *is consistent with gradient descent sampled with the sampling scheme Equation* (3.4)*.*

Proof. The problem Equation (B.55) can be solved in an alternative gradient descent-ascent manner:

$$\begin{aligned} \boldsymbol{\theta}(t+1) &= \boldsymbol{\theta}(t) - \eta_t^{\boldsymbol{\theta}} \sum_{i=1}^{n} p_i(t) \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}(t)), \\ w_i(t+1) &= w_i(t) + \eta_t^{\boldsymbol{w}}(\ell_i(\boldsymbol{\theta}(t+1)) - \ell_i^{\text{ref}}), \quad p_i(t) = \frac{w_i(t)}{\sum_j w_j(t)}. \end{aligned} \tag{B.57}$$

The sampling scheme Equation (3.4) updates the weights as

$$w_i(t+1) = w_i(t) + (1-\beta_1)(\ell_i(\boldsymbol{\theta}(t+1)) - \ell_i(\boldsymbol{\theta}(t))) + \beta_1(s_i(t) - s_i(t-1)). \tag{B.58}$$

By Equation (B.9), we get

$$s_i(t) - s_i(t-1) = -(1-\beta_2)\beta_2^{t-1}s_i(0) - (1-\beta_2)^2\sum_{k=1}^{t-1}\beta_2^{t-1-k}\ell_i(\boldsymbol{\theta}(k)) + (1-\beta_2)\ell_i(\boldsymbol{\theta}(t)),$$

hence

$$\begin{aligned} w_i(t+1) = w_i(t) &+ (1-\beta_1)(\ell_i(\boldsymbol{\theta}(t+1)) - \ell_i(\boldsymbol{\theta}(t))) - \beta_1(1-\beta_2)\beta_2^{t-1}s_i(0) \\ &- \beta_1(1-\beta_2)^2\sum_{k=1}^{t-1}\beta_2^{t-1-k}\ell_i(\boldsymbol{\theta}(k)) + \beta_1(1-\beta_2)\ell_i(\boldsymbol{\theta}(t)). \end{aligned} \tag{B.59}$$

Let

$$\ell_i^{\text{ref}} = \frac{1-2\beta_1+\beta_1\beta_2}{1-\beta_1}\ell_i(\boldsymbol{\theta}(t)) + \frac{\beta_1(1-\beta_2)^2}{1-\beta_1}\sum_{k=1}^{t-1}\beta_2^{t-1-k}\ell_i(\boldsymbol{\theta}(k)) + \frac{\beta_1(1-\beta_2)\beta_2^{t-1}}{1-\beta_1}s_i(0), \tag{B.60}$$

then we have

$$w_i(t+1) = w_i(t) + (1-\beta_1)(\ell_i(\boldsymbol{\theta}(t+1)) - \ell_i^{\text{ref}}), \tag{B.61}$$

which coincides with the update formula Equation (B.57) with $\eta_t^{\boldsymbol{w}} = 1-\beta_1$. The proof is completed. □

C More Details of Algorithms

This section presents more details of the ES(WP) sampling framework.

Annealing (optional). Notably, similar to the loss-weighted sampling scheme Equation (3.1) and its further variants, the sampling scheme Equation (3.4) also assigns different weights on the respective gradient of data samples, leading to a biased estimation on the true gradient $\nabla_{\boldsymbol{\theta}} \hat{L}_n(\cdot)$ (with uniform individual weights). Inspired by Qin et al. (2024), we adopt the *annealing* strategy, to perform normal training (with the standard batched sampling, no data selection) at the last few epochs. Besides, to get better initializations of the weights $\{w_i(\cdot)\}_{i\in[n]}$, we also apply the annealing strategy at the first few epochs.

Combining the sampling scheme Equation (3.4) with the annealing strategy, we obtain the **Evolved Sampling** (**ES**) framework (formalized in Algorithm 1).

Algorithm 1 **E**volved **S**ampling (**W**ith **P**runing)

Require: Dataset $\mathcal{D} = \{\boldsymbol{z}_i\}_{i=1}^n$, optimizer (e.g. Adam)
Require: Pruning ratio r, meta-batch size B, mini-batch size $b \leq B$, total epochs E, annealing epochs $(E_{a_{\text{start}}}, E_{a_{\text{end}}})$, hyper-parameters $\beta_1, \beta_2 \in (0, 1)$
  Initialize the scores/weights $\boldsymbol{s}(0) = \boldsymbol{w}(0) = \frac{1}{n}\mathbf{1}_n$, $t = 0$
  for $e = 0, 1, \cdots, E-1$ **do**
    if $E_{a_{\text{start}}} \leq e < E - E_{a_{\text{end}}}$ **then**
      Sample a sub-dataset $\mathcal{D}_e$ ($|\mathcal{D}_e| = (1-r)|\mathcal{D}|$) from $\mathcal{D}$ without replacement, according to the probability $p'_i(e) \propto w_i(e)$ ▷ *pruning*
    else
      Set $\mathcal{D}_e = \mathcal{D}$
    end if
    for $j = 0, 1, \cdots, \lceil \frac{|\mathcal{D}_e|}{B} \rceil - 1$ **do**
      Sample a meta-batch $\mathcal{B}_t$ ($|\mathcal{B}_t| = B$) uniformly from $\mathcal{D}_e$ without replacement
      Compute the loss $\ell_i(\boldsymbol{\theta}(t))$ for $\boldsymbol{z}_i \in \mathcal{B}_t$
      Update score: $s_i(e+1) \leftarrow \beta_2 s_i(e) + (1-\beta_2)\ell_i(\boldsymbol{\theta}(t))$ for $\boldsymbol{z}_i \in \mathcal{B}_t$
      Update the weight: $w_i(e+1) \leftarrow \beta_1 s_i(e) + (1-\beta_1)\ell_i(\boldsymbol{\theta}(t))$ for $\boldsymbol{z}_i \in \mathcal{B}_t$
      if $E_{a_{\text{start}}} \leq e < E - E_{a_{\text{end}}}$ **then**
        Sampling a mini-batch $\mathfrak{b}_t$ ($|\mathfrak{b}_t| = b$) from $\mathcal{B}_t$ without replacement, according to the probability $p_i(e+1) \propto w_i(e+1)$
        Update model: $\boldsymbol{\theta}(t+1) \leftarrow \text{optimizer}(\boldsymbol{\theta}(t); \mathfrak{b}_t)$
      else
        Update model: $\boldsymbol{\theta}(t+1) \leftarrow \text{optimizer}(\boldsymbol{\theta}(t); \mathcal{B}_t)$ ▷ *annealing*
      end if
      $t \leftarrow t+1$
    end for
  end for

Pruning (optional). Note that applying the sampling scheme Equation (3.4) to meta-batches (with the batch size B) in fact introduces data selection in a *batch* level, since one can always select a smaller batch (with the batch size $b < B$) out of the meta-batch, according to the sampling probability $p_i(t)$ defined in Equation (3.4). For more aggressive data pruning and enhanced data efficiency, we can further extend ES by involving the *set*-level data selection. That is, randomly pruning the *whole* dataset according to the probability proportional to the weights $\{w_i(e)\}_{i=1}^n$ at the beginning of the e-th epoch. This is formalized as **Evolved Sampling with Pruning** (**ESWP**) in Algorithm 1.

D More Experimental Results and Details

In this section, we present further experimental results and details. We run all the experiments with one NVIDIA A100 (80GB) with the mixed-precision training except the pre-training of ViT-Large/GPT-2-Small/GPT-2-XL on ImageNet-1K/OpenWebText/FineWeb. All the algorithms are

implemented based on PyTorch (Paszke et al. (2019)) and Timm (Wightman et al. (2019)). For InfoBatch, our implementation is adapted from Qin et al. (2024).

D.1 Illustrations of ES on Synthetic Signals

See Figure 8 for details.

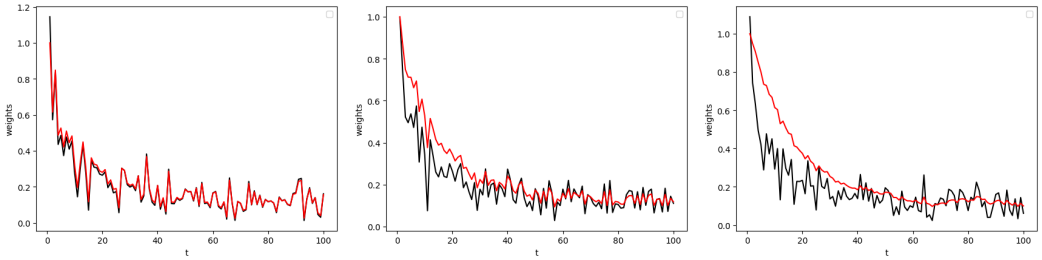


Figure 8: The output weights of different sampling schemes, where the black curves denote Equation (3.1), while the red curves represent Equation (3.4) (from left to right: $\beta_1 = 0.1, 0.5, 0.8$, and $\beta_2 \equiv 0.9$). Here, we draw the black curve as a decayed function with random perturbations, to mimic typical behaviors of loss curves in general machine learning tasks. It is shown that the sampling scheme Equation (3.1) is usually sensitive w.r.t. oscillations. However, when losses oscillate, the sampling scheme Equation (3.4) reacts moderately by not only reserving some portion of dynamical details of losses (high frequencies), but also remaining necessary robustness by capturing the overall trend (low frequencies), with the flexibility to trade off in between by tuning (β_1, β_2). See further theoretical analysis in Appendix B.4.

D.2 Visualization of Selected Samples by ES(WP)

See Figure 9 for details.

D.3 ES(WP) to Reduce Back Propagations

See Figure 10 for details.

D.4 More Baselines

Loss-based data selection. Here, we provide the taxonomy of multiple previous loss-based dynamic sampling methods in Table 8. ES(WP) first takes gradient information during training into account.

Table 8: Comparison of different loss-based dynamic sampling methods. The "history" denotes whether the method uses historical (loss) information along the training. The "grad" column stands for whether the method (implicitly) exploits gradient information during the training for sampling. The last column summarizes the ratio of samples used for back propagations (BPs) relative to the standard training. Here, r stands for the pruning ratio for *set*-level methods (pruning data samples of the whole epoch), and b/B represents the pruning ratio for *batch*-level methods (selecting a mini-batch $\mathfrak{b}$ (subset) from a meta-batch $\mathcal{B}$).

	set	batch	history	**grad**	ratio of samples for BPs
UCB (Raju et al. (2021))	✓		✓		$1-r$
KA (Thao Nguyen et al. (2023))	✓				$1-r$
InfoBatch (Qin et al. (2024))	✓				$1-r$
Loss (Katharopoulos & Fleuret (2017))		✓			b/B
Order (Kawaguchi & Lu (2020))		✓			b/B
ES (ours)		✓	✓	✓	b/B
ESWP (ours)	✓	✓	✓	✓	$\boldsymbol{(1-r)b/B}$

Gradient-based data selection. Here, we provide two representative examples of gradient-based data selection methods: GRAD-MATCH (Killamsetty et al. (2021a)) and VCAS (Wang et al. (2024b)).

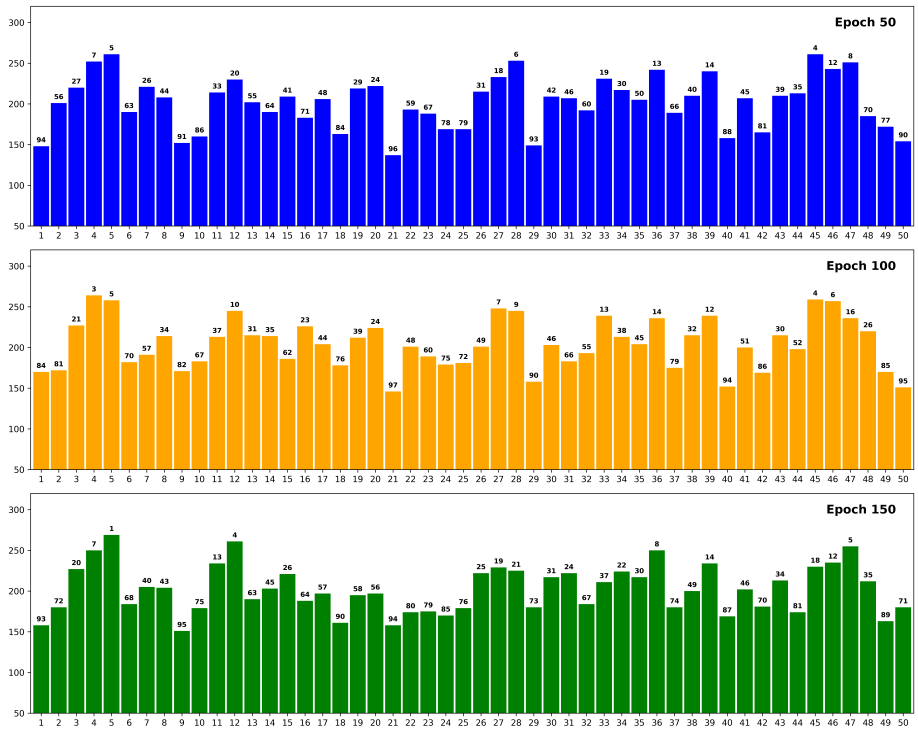


Figure 9: Visualization of the number of selected samples for BP of each class in ES(WP) (ResNet-50, Cifar-100), following Figure 6 in Thao Nguyen et al. (2023). Here, it shows the result of the first 50 classes. The number on top of each column shows the rank over 100 classes (a lower rank indicates a higher number of selected samples). It is shown that ES(WP) can automatically adjust selected samples at different training stages.

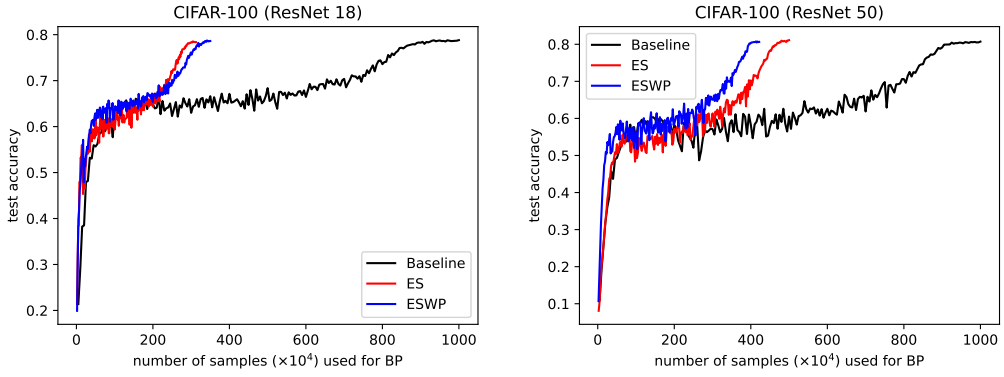


Figure 10: Learning dynamics of different data selection methods: Test accuracy versus the number of samples used for back propagations (BPs).

For GRAD-MATCH, it is shown in Figure 11 that GRAD-MATCH underperforms ESWP with clear gaps. For VCAS, Wang et al. (2024b) reports worse test accuracies and accelerations with clear gaps compared to ES(WP) (see Table 9).

In addition, we also test for potential gradient variance reduction of these data selection methods. See Figure 12 for details. It is observed that as training proceeds, the gradient variances of all methods basically decrease and eventually converge. Notably, after 25 epochs, ES/ESWP consistently

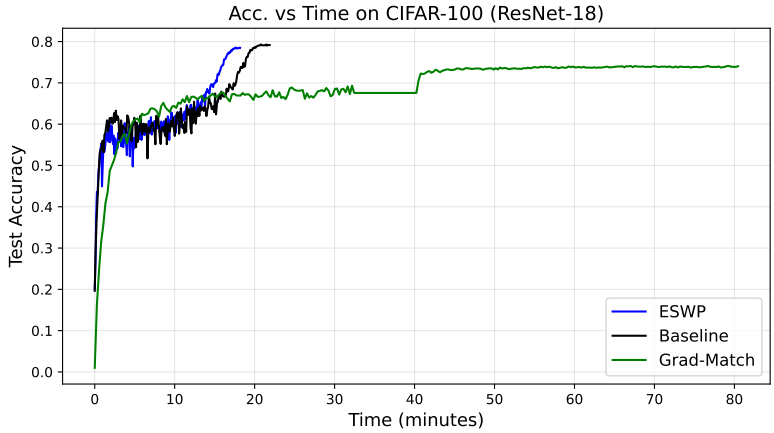


Figure 11: The comparison between GRAD-MATCH and ESWP when training ResNet-18 on CIFAR-100. ESWP shows superior learning performance and training acceleration with clear advantages.

Table 9: Test accuracies (%) and reduced wall-clock time of fine-tuning ViT-Large on ImageNet-1K. Results are from Table 3 in Wang et al. (2024b) and Table 1(b) in this work.

	VCAS	ES	ESWP
Accuracy	82.3	84.7	85.0
Time↓	21.1%	26.0%	40.7%

achieves much lower gradient variances than Baseline. This result further highlights the superiority of ES/ESWP from a variance reduction perspective.

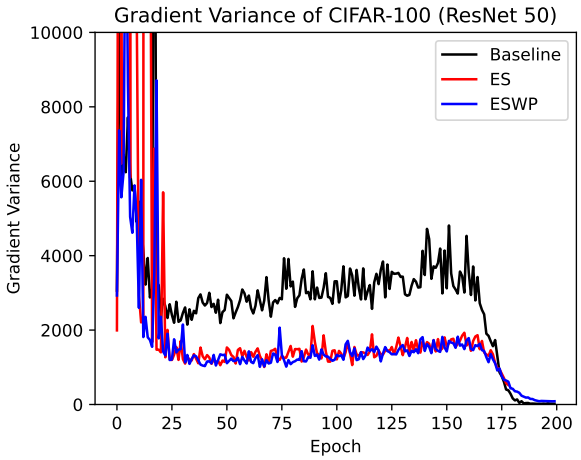
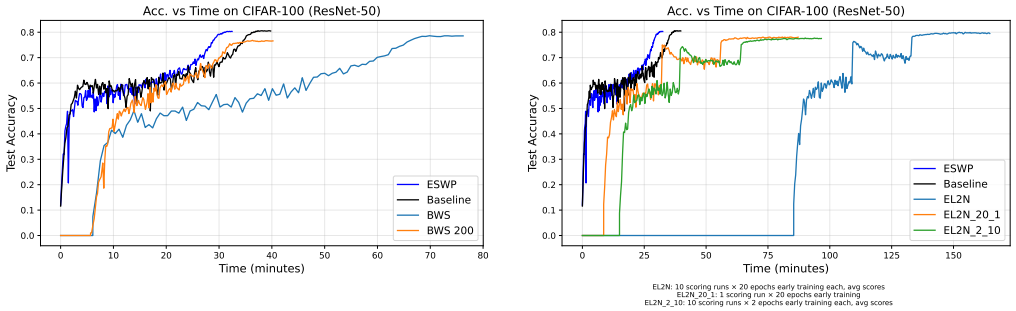


Figure 12: Gradient variances of different methods along the training process.

Static/early-training data selection. We further include two static/early-training data selection methods, namely BWS (Choi et al. (2024)) and EL2N (Paul et al. (2021)), which select certain "coresets" based on early-stage training before the formal training. That is, these static data selection methods require an additional pre-scoring stage before the formal training. (i) For BWS, we follow the official configurations in Choi et al. (2024) to train ResNet-50 for score computation and train a proxy model for 20 epochs. After selecting 80% of the original dataset (the same proportion as ESWP), we train on the selected coreset and evaluate two formal training settings: the official

400-epoch setup configured in Choi et al. (2024), and another 200-epoch setup to align with that of ESWP (with a batch size 256 and learning rate 0.2). The time of learning curves in Figure 13(a) counts both proxy and formal training stages (not include the requisite score stage). (ii) For EL2N, the pruning ratio is the same as that of ESWP (20%). In Figure 13(b), we have noted the scoring budget (epochs), which aligns with official configurations (2 or 20 early training epochs, 1 or 10 scoring runs). The scoring budgets are 200 epochs for EL2N, and 20 epochs for EL2N_20_1 and EL2N_2_10. The time of learning curves in Figure 13(b) stands for the total end-to-end runtime (including both early and formal training stages). It is shown that there are potential limitations of these early-training data selection methods, as they insufficiently consider the evolution of sample weights during the whole training process.



(a) BWS (b) EL2N

Figure 13: The comparison between ESWP and static/early-training data selection baselines, which underperform ESWP with clear gaps.

D.5 Experiments on CIFAR Datasets

For computer vision (CV) tasks, we train ResNet-18/50 (R-18/50) models on CIFAR-10/100 datasets, using SGD for 200 epochs, with $B = 128/256$ for ResNet-18/50 ($b/B = 50\%$ for ResNet-50).

For the experiments on the CIFAR-10/100 datasets, we use the SGD optimizer with the momentum 0.9 and weight decay 5×10^{-4}. We apply the OneCycle scheduler (Smith & Topin (2019)) with the cosine annealing. For CIFAR-10, the maximal learning rate is 0.2 for the baseline and *set*-level selection methods, while 0.05 for *batch*-level selection methods due to larger variances of stochastic gradients and 0.08 for ESWP. For CIFAR-100 trained with ResNet-18/50, the maximal learning rates for all the sampling methods are $0.05/0.2$, following Qin et al. (2024).

Robustness on randomness. We further plot the learning curves of ESWP versus Baseline (standard training without data selection) with multi-seed statistics over five random seeds when training ResNet-50 on CIFAR-100 in Figure 14. The results show that ESWP consistently achieves lossless accelerations across randomness.

D.6 Experiments of Full Fine-Tuning

Vision Transformer. We fine-tune ViT-Large model on ImageNet-1K with a meta-batch size $B = 256$ for 10 epochs, using the Adam optimizer with the OneCycle scheduler (Smith & Topin (2019)) with the cosine annealing and a maximal learning rate of 2×10^{-5}.

ALBERT. Following the setup in Xie et al. (2023b) (Table 8), we use the AdamW optimizer and the polynomial decay scheduler with warm up.

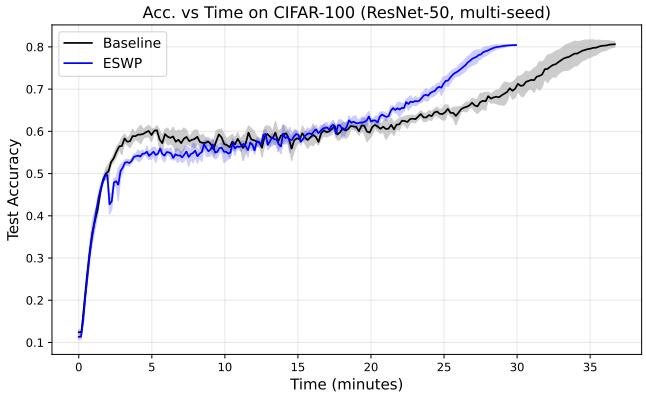


Figure 14: Learning curves of ESWP and Baseline when training ResNet-50 on CIFAR-100 over five random seeds. ESWP consistently achieves lossless accelerations over Baseline across randomness.

D.7 Experiments of Low-Resourced Fine-Tuning of Qwen2.5-Math-1.5B

We conduct experiments on a single A100 (40GB) GPU to investigate the low-resource regime. Our implementation builds upon the verl framework.[4] We set the batch sizes $B = 32, b = b_{\text{micro}} = 8$, and use the AdamW optimizer with a learning rate of 1×10^{-5}, which follows a cosine decay scheduler with a warmup ratio of 0.1. We set the total epoch as 10 and evaluate the model after 1K, 2K, 4K, 6K and 8K training steps. We use a temperature of 1.0, top_p=1, the default chat template and Chain-of-Thought (CoT) prompting for evaluation.

D.8 LoRA Fine-Tuning of Qwen3.5-9B-Base

We use the ms-swift framework[5] for text-only supervised fine-tuning of `Qwen/Qwen3.5-9B-Base` on 100k samples from NuminaMath-CoT. Hyperparameter settings are in Table 10. During active selection, ES scores $B = 16$ candidates per rank and samples $b = 4$ without replacement, using score microbatches of eight and $(\beta_1, \beta_2) = (0.2, 0.9)$.

Table 10: Training settings for the Qwen3.5-9B-Base LoRA experiment.

Setting	Value
LoRA rank	8
LoRA scaling parameter	32
LoRA dropout	0.05
Maximum sequence length	2,048 tokens
Optimizer	AdamW
Learning rate	10^{-5}
Learning-rate schedule	Cosine decay
Warmup ratio	0.1
Weight decay	0.1
AdamW coefficients	$(0.9, 0.95)$
AdamW epsilon	10^{-8}
Backward batch size per rank	4
Batch size	16

[4] `https://github.com/volcengine/verl`
[5] `https://github.com/modelscope/ms-swift`

D.9 Experiments of Pre-Training

ViT-Large. We conduct the MAE-based pre-training of ViT-Large on ImageNet-1K using $4\times$A100 GPUs. Following the setup in He et al. (2022), we train for 300 epochs with a 40-epoch warmup, base learning rate 1.5×10^{-4}, weight decay 0.05, and batch sizes $(B, b) = (256, 256)$ per GPU for ESWP, i.e., there is no batch-level data selection. In our implementation, the sampling procedure of ESWP is conducted by an additional round of synchronization. After pre-training, we fully fine-tune the model for 50 epochs with a 5-epoch warmup, using the standard batched sampling (no data selection) with the batch size $B = 256$ per GPU.

GPT-2. We conduct the pre-training of GPT-2-Small on OpenWebText using $8\times$A100 GPUs. Following the official configuration in nanoGPT,[6], we train for 600K iterations (including 2K warmup iterations) with the context length 1024, cosine learning rate decay to 6×10^{-5} with a peak learning rate 6×10^{-4}, and weight decay 0.1. For ES, we apply the sequence-level data selection with $(\beta_1, \beta_2) = (0.2, 0.9)$, the selection ratio $b/B = 0.5$, and 60K annealing steps (10%) before activating data selection. After pre-training, we fully fine-tune the model on GLUE without data selection, using a randomly initialized classifier/regression head, maximum sequence length 128, learning rate 5×10^{-6}, weight decay 0.01, and warmup ratio 0.06. We train for 3 epochs on SST-2, QNLI, QQP and MNLI-m, 5 epochs on MRPC, 10 epochs on CoLA and RTE, and 20 epochs on STS-B.

We pre-train GPT-2-XL using eight H100 (80GB) GPUs with a training sequence length of $6{,}144$ and $49{,}152$ update tokens per optimizer step. The hybrid optimizer applies Muon to hidden matrix parameters and Adam to the remaining parameters, with learning rates of 0.010 and 0.002, respectively, Adam coefficients $(0.8, 0.95)$, zero weight decay, and gradient clipping at 1. We use cosine learning-rate decay to 0.1 of the initial rates, without learning-rate warmup, and evaluate on 100M validation tokens with a sequence length of $8{,}192$. For ES, $(\beta_1, \beta_2) = (0.2, 0.9)$ and $b/B = 0.5$: each rank scores 64 candidate sequences and selects 32 per meta-batch, without dataset-level or token-level pruning. The initial ordinary-training phase lasts $610{,}352$ steps (approximately 30B tokens), followed by selection for the remaining 270B nominal source-token exposures, with no final ordinary-training phase. This yields approximately $30\text{B} + 0.5 \times 270\text{B} = 165\text{B}$ update tokens for ES. The ten-epoch budget denotes 300B nominal source-token exposures from the 30B-token corpus.

Downstream tasks. For downstream evaluation, we fully fine-tune the final checkpoints on GLUE, using a randomly initialized task head and learning rates in $\{10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$. We use global batch size 32, evaluation batch size 64, maximum sequence length 128, AdamW coefficients $(0.9, 0.999)$, weight decay 0.01, warmup ratio 0.06, gradient clipping at 1, and classifier dropout 0.1. The head and backbone share the same learning rate. We train for 3 epochs on SST-2, QNLI, QQP, and MNLI-m, 5 on MRPC, 10 on CoLA and RTE, and 20 on STS-B, selecting the best development score across learning rates and epochs. Each condition has one pre-training run and one fine-tuning seed; we do not estimate seed variance.

D.10 More Ablations

Trade-offs between performance and speed. We emphasize that batch sizes (b, B), the pruning ratio and annealing epochs in ES(WP) are all user-defined, and flexible to trade off between learning performance and training costs. We evaluate different values of b/B when fine-tuning ViT-Large on ImageNet-1K, and varied pruning ratios when training ResNet-18 on CIFAR-10. The results are illustrated in Figure 15 and Figure 16. It is shown that ES robustly achieves lossless acceleration when $b/B \geq 1/16$; while the selection is too aggressive ($b/B \leq 1/32$), the performance degrades as expected (Figure 15), due to the increase of variances in stochastic gradients. Also, there is a clear trade-off between the performance and speed shown in Figure 16, where setting the pruning ratio around $20\% \sim 30\%$ seems efficient. We further evaluate different annealing ratios ("ar"; i.e., the ratio of annealing over total epochs) when training ResNet-18 on CIFAR-100 (see Table 11), showcasing its robustness.

Table 11: Ablations on the annealing (default in bold).

ar	0.0	**0.05**	0.075	0.1
Acc. (%)	78.60	**78.79**	78.32	78.20

[6] https://github.com/karpathy/nanogpt

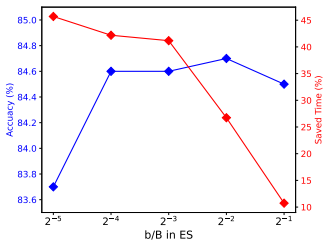


Figure 15: Performance-efficiency trade-offs regarding mini/meta-batch sizes.

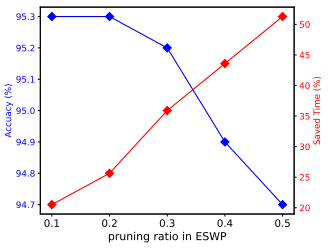


Figure 16: Performance-efficiency trade-offs regarding the pruning ratio.

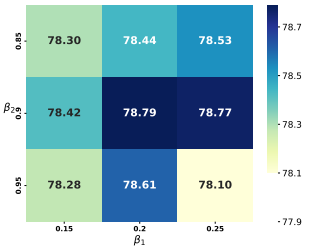


Figure 17: Local optimality of default betas.

D.11 Comparison Methods: Default Hyper-Parameters

For all the other data selection methods, we also use their default hyper-parameters in original papers in our experiments. Therefore, the comparisons and evaluations are fair in terms of hyper-parameters. We list the default hyper-parameters of all the other data selection methods as follows:

- InfoBatch (Qin et al. (2024)): pruning ratio $r = 0.5$, annealing ratio $1 - \delta = 0.125$;
- KAKURENBO (Thao Nguyen et al. (2023)): pruning ratio $r = 0.3$, confidence threshold $\tau = 0.7$;
- UCB (Raju et al. (2021)): pruning ratio $r = 0.3$, decay parameter $\beta = 0.8$, confidence bound $c = 1$;
- Loss (Katharopoulos & Fleuret (2017)), Order (Kawaguchi & Lu (2020)): the same batch sizes as ES.

E Future Directions

Studies in the future may include three aspects: (i) More mathematical analysis on the effect of data selection (Kolossov et al. (2024)); (ii) more specific applications, such as data selection on domain mixtures (Chen et al. (2023); Xie et al. (2023a)); (iii) more efficient and scalable implementations, such as data parallelism (You et al. (2017; 2020)). These directions are certainly worthy of explorations in the future.