Diversified Batch Selection for Training Acceleration

Feng Hong [1 †] **Yueming Lyu** [2 3] **Jiangchao Yao** [1 4] **Ya Zhang** [1 4] **Ivor W. Tsang** [2 3 5] **Yanfeng Wang** [1 4]

Abstract

The remarkable success of modern machine learning models on large datasets often demands extensive training time and resource consumption. To save cost, a prevalent research line, known as online batch selection, explores selecting informative subsets during the training process. Although recent efforts achieve advancements by measuring the impact of each sample on generalization, their reliance on additional reference models inherently limits their practical applications, when there are no such ideal models available. On the other hand, the vanilla reference-model-free methods involve independently scoring and selecting data in a sample-wise manner, which sacrifices the diversity and induces the redundancy. To tackle this dilemma, we propose Diversified Batch Selection (DivBS), which is reference-model-free and can efficiently select diverse and representative samples. Specifically, we define a novel selection objective that measures the group-wise orthogonalized representativeness to combat the redundancy issue of previous sample-wise criteria, and provide a principled selection-efficient realization. Extensive experiments across various tasks demonstrate the significant superiority of DivBS in the performance-speedup trade-off. The code is publicly available.

1. Introduction

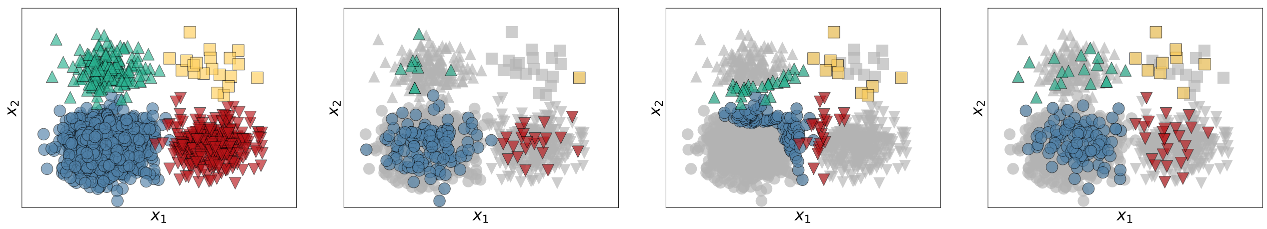
Deep learning, propelled by vast amounts of web-scraped data, has led to significant advancements in models such as GPT-4 (OpenAI, 2023), CLIP (Radford et al., 2021), SAM (Kirillov et al., 2023), and Stable Diffusion (Rombach et al., 2022). However, the time-intensive training process, lasting for weeks or even months, poses challenges with extended development cycles and increased resource consumption. Additionally, with a growing focus on data quality in deep learning systems, given the prevalence of low-quality, redundant, and biased data in real-world scenarios (Xie et al., 2023; Deng et al., 2023), there is an increasing need to select valuable training data for accelerating model training while maintaining the performance.

Recent studies (Mindermann et al., 2022; Deng et al., 2023) have achieved notable acceleration and convergence results by employing the online batch selection (Loshchilov & Hutter, 2015) paradigm, which involves selecting samples that are most conducive to model convergence at the current training stage. However, these reference-model-based methods rely on extra reference models, either trained from a considerable amount of holdout data (Mindermann et al., 2022) or a pre-trained zero-shot predictor (Deng et al., 2023). Obtaining such a reference model can be costly or challenging in certain scenarios, especially for large-scale pre-training tasks. On the other hand, reference-model-free online batch selection methods (Jiang et al., 2019; Katharopoulos & Fleuret, 2018b; Loshchilov & Hutter, 2015) prioritize challenging samples based on high loss or large gradient norm. Despite their practicality and efficiency, they often fall short in performance even compared to uniform selection (Mindermann et al., 2022; Deng et al., 2023).

In this paper, we focus on selecting a limited budget of crucial samples in a reference-model-free batch selection manner for training acceleration with negligible performance drop. We contend that existing reference-model-free selection methods adopt sample-wise strategies. Specifically, they independently apply predefined scoring criteria to all samples and conduct the score-based selection. Such *sample-wise selection* methods overlook the correlations and redundancies among samples and may lead to poor diversity, degrading the performance. When a sample is selected based on a high score, similar (or even identical) samples also receive similar scores. However, these samples contribute negligible new information to model training. As shown in Figure 1(c), samples selected based on train loss exhibit significant overlap and poor coverage of the original

[†] Work down during internship at CFAR, A*STAR [1] Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai, China [2] CFAR, Agency for Science, Technology and Research (A*STAR), Singapore [3] IHPC, Agency for Science, Technology and Research (A*STAR), Singapore [4] Shanghai AI Laboratory, Shanghai, China [5] College of Computing and Data Science, NTU, Singapore. Correspondence to: Jiangchao Yao <Sunarker@sjtu.edu.cn>, Yanfeng Wang <wangyanfeng@sjtu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).



(a) Full Data (b) Uniform (c) Train Loss (d) DivBS

Figure 1. Visualization of a toy motivating example, which is a 2D imbalanced four-class classification problem. Subfigure (a) represents all the training data. Subfigures (b), (c), and (d) depict the subsets selected by the Uniform, Train Loss, and DivBS methods, with 10% budget. For more details, please refer to Appendix C.1.

space. Zheng et al. (2023); Guo et al. (2022) have also discussed the negatives of such methods on subset diversity and performance in the context of the one-shot coreset selection, particularly with small budgets.

To tackle the diversity challenge, we propose a novel reference-model-free batch selection method, Diversified Batch Selection (DivBS). Our core concept encompasses two aspects: (1) data selection should consider the selected subset as a whole, rather than independently selecting data on a sample-wise basis; (2) when assessing the overall representativeness of a subset, the inter-sample redundancy should be eliminated. Motivated by this, we introduce a new objective for batch selection, aiming to maximize the overall orthogonalized representativeness of the subset after removing the inter-sample redundancy (Equations (1) and (2)). Through a principled simplification of the optimization problem (Proposition 3.1), we propose a greedy algorithm (Algorithm 1) that can theoretically achieve an approximate ratio of $1 - e^{-1}$ *w.r.t.* the optimal objective value (Proposition 3.3). We further streamline the selection process (Algorithm 2), empirically achieving substantially reduced time consumption with comparable performance. We summarize the contributions as follows:

- We explore prioritizing samples that enhance model convergence without extra reference models and highlight the diversity challenge faced by current sample-wise online batch selection methods.
- We propose to maximize the overall orthogonalized representativeness of the subset rather than independently sample-wise selection. Building on a greedy algorithm with a $1 - e^{-1}$ approximate ratio, we present a more efficient reference-model-free batch selection method, Diversified Batch Selection (DivBS).
- We conduct extensive experiments, covering image classification, imbalanced learning, semantic segmentation, cross-modal retrieval, and language model fine-tuning. The results consistently demonstrate the superiority of DivBS in accelerating training while maintaining the performance, *e.g.*, with 70% fewer iterations, classification accuracy drops by under 0.5% on average, segmentation mIoU decreases by under 1%, and cross-modal retrieval performance improves.

2. Background: Online Batch Selection

We consider a task of learning a deep model f_θ with parameters θ on training data $\mathcal{D}$ with stochastic gradient descent (SGD). At each training step, we can access a data batch $B = \{d_i\}_{i=1}^{N_B}$ with N_B data points from $\mathcal{D}$. In the online batch selection scenario, we need to conduct a smaller batch $S \subset B$ with a fixed budget of sample number $N_S < N_B$ to update the model. The next large batch is then pre-sampled from $\mathcal{D}$ without replacement of previously sampled points in a same epoch.

Let $U = g(B, \theta) = \{g(d_i, \theta)\}_{i=1}^{N_B}$ denote the features used for selection from B, where g denotes the mapping function from data points to the selection features given current model f_θ. Existing methods simplify the problem of selecting a subset from B into a sample ranking problem. Employing different scoring criteria $s(u), u \in U$, they select the top-N_S samples from B to conduct S. Loshchilov & Hutter (2015); Jiang et al. (2019) opt for high loss, where U contains the outputs and labels, and $s(\cdot)$ is the loss function; Katharopoulos & Fleuret (2018b) select samples with large gradient norm, where U is the sample-wise gradients, and $s(\cdot)$ is the norm function; Mindermann et al. (2022); Deng et al. (2023) leverage a reference model to compute the score, where U contains the training model outputs, the reference model outputs, and class labels, and $s(\cdot)$ is an approximate version of the generalization loss. These methods select data in a sample-wise manner, without considering the interactions and redundancy among samples when they collectively update the model within a batch.

Algorithm 1 The greedy algorithm.

```
**Input:** Batch $B$, current model $f_\theta$, budget number $N_S$
**Output:** Selected mini batch $S$
$S \leftarrow \emptyset$, $E \leftarrow \emptyset$, $\text{Sum} \leftarrow \sum_{u \in g(B,\theta)} u$
**repeat**
    $E_{\text{Cand}} \leftarrow \{\frac{g(d,\theta)-\sum_{e\in E}(e\cdot g(d,\theta))e}{\|g(d,\theta)-\sum_{e\in E}(e\cdot g(d,\theta))e\|}$ for $d \in B\}$
    // Candidate orthonormal basis
    $\text{idx} \leftarrow \arg\max_i |e_i \cdot \text{Sum}|, e_i \in E_{\text{Cand}}$
    $S \leftarrow S \cup \{d_{\text{idx}}\}$
    $E \leftarrow E \cup \{e_{\text{idx}}\}$
    $B \leftarrow B \setminus \{d_{\text{idx}}\}$
**until** $|S| = N_S$
```

Algorithm 2 DivBS.

```
**Input:** Batch $B$, current model $f_\theta$, budget number $N_S$
**Output:** Selected mini batch $S$
$S \leftarrow \emptyset$, $E \leftarrow \emptyset$, $\text{Sum} \leftarrow \sum_{u \in g(B,\theta)} u$
**repeat**
    $d \leftarrow \arg\max_{d \in B} |g(d,\theta) \cdot \text{Sum}|$
    $e \leftarrow \frac{g(d,\theta)-\sum_{e\in E}(e\cdot g(d,\theta))e}{\|g(d,\theta)-\sum_{e\in E}(e\cdot g(d,\theta))e\|}$
    $S \leftarrow S \cup \{d\}$
    $E \leftarrow E \cup \{e\}$
    $\text{Sum} \leftarrow \text{Sum} - (e \cdot \text{Sum})e$ // Subtracting
    the orthogonal components of the
    already selected samples from Sum.
    $B \leftarrow B \setminus \{d\}$
**until** $|S| = N_S$ or $\text{Sum} = 0$
```

3. Method: Diversified Batch Selection

3.1. Motivation

In Figure 1, we present a toy example involving an imbalanced four-class classification task and showcase subsets selected by different batch selection methods. We can observe that the uniform sampling method (Figure 1(b)) does not always achieve effective coverage of the original sample space, especially for the low-density (yellow and green) regions. The method of selecting challenging samples (Train Loss) (Figure 1(c)) results in high redundancy, with many points nearly overlapping. Additionally, the data distribution deviates significantly from the overall distribution, leading to inferior model convergence. We posit that this stems from current methods independently scoring and selecting data in a sample-wise manner. We propose that the overall evaluation of the selected samples should be conducted instead of sample-wise scoring, and the impact of inter-sample redundancy should be removed. It is evident that the subset selected by our proposed DivBS (Figure 1(d)) effectively covers the original sample space and significantly enhances the diversity of the selected samples.

3.2. Objective

Recall that our core ideas for addressing the diversity challenge of the reference-model-free batch selection are: (1) we should consider the representativeness of the selected subset as a whole, rather than evaluating data in a sample-wise manner; (2) we should eliminate the impact of inter-sample redundancy on the representativeness of the subset. Motivated by these, we define a new measure, which characterizes the group-wise orthogonalized representativeness of subset S with respect to B as:

$$r(S,B,\theta) = \max_{E \in \mathcal{E}(g(S,\theta))} \sum_{e \in E} \sum_{u \in g(B,\theta)} e \cdot u \tag{1}$$

Here $\mathcal{E}(g(S,\theta)) := \{E = \{e_1, \dots e_{|E|}\} | \forall e_i, e_j \in E, e_i \cdot e_j = \delta_{ij}, \text{span}(E) = \text{span}(g(S,\theta))\}$ denotes the set of all potential orthonormal bases for the subspace spanned by $g(S,\theta)$, where δ_{ij} is the Kronecker delta, taking the value 1 when $i = j$ and 0 otherwise, and $\text{span}(\cdot)$ denotes the subspace spanned by all elements in a set.

Generally, the design intuition of Eq. (1) involves removing inter-sample redundancy in subset S through orthogonalization. When calculating the contribution of an element in the subset S to r, we subtract the redundant components already presented by other elements in S and only consider its unique part distinct from others. This prevents redundant components between elements from contributing duplicate values to the orthogonalized representativeness r. In contrast, in sample-wise selection, redundant elements contribute duplicate scores because they are scored individually and then directly added up. Therefore, based on Eq. (1), we propose a new objective for online batch selection, aiming to choose a subset $S \subset B$ with the maximum orthogonalized representativeness $r(S,B,\theta)$, under a specified budget constraint $|S| \le N_S$.

$$\arg\max_{S \subset B} r(S,B,\theta), \quad \text{s.t.}\ |S| \le N_S. \tag{2}$$

3.3. Optimization

Directly optimizing Equation (2) encounters two immediate challenges: (1) directly traversing an infinite space of orthonormal bases is clearly impractical; (2) the candidate subset space with size $O({N_B}^{N_S})$ is also challenging to efficiently solve. In the following, we present our development to address these two challenges.

Simplify the optimization of the orthonormal basis $E \in \mathcal{E}(g(S,\theta))$. To tackle the first challenge, we introduce Proposition 3.1, which offers a simple computational form for $r(S,B,\theta)$ *w.r.t. any* orthonormal basis for $g(S,\theta)$.

Table 1. Final accuracy and wall-clock time of one epoch training of different methods on CIFAR-10 with 10% budget.

Method	Full	Uniform	Algorithm 1	Algorithm 2
Acc ↑	95.50%	92.06%	94.57%	94.65%
Time ↓	44.7s	13.2s	19.8s	13.9s

Proposition 3.1. *$\forall E \in \mathcal{E}(g(S,\theta))$, we have*

$$\sqrt{|E| \sum_{e \in E} (e \cdot \sum_{u \in g(B,\theta)} u)^2} = r(S,B,\theta).$$

The computational form for $r(S,B,\theta)$ in Proposition 3.1 enables us to freely choose the orthonormal basis without affecting the optimization of the overall objective. Therefore, we no longer need to consider the optimization of $E \in \mathcal{E}(g(S,\theta))$ and can directly optimize S to maximize $r(S,B,\theta)$. Please refer to Appendix B.1 for the proof.

Greedy algorithm and its approximation guarantee. Moreover, we seek to streamline the optimization for subset $S \subset B$. A straightforward approach involves sequentially and greedily selecting data that maximizes $r(S,B,\theta)$ according to the computational form in Proposition 3.1, as depicted in Algorithm 1.

To analyze the theoretical guarantee of Algorithm 1 for the optimization problem in Equation (2), we introduce an auxiliary function $r'(S,B,\theta) = \sqrt{\sum_{e\in E}(e \cdot \sum_{u \in g(B,\theta)} u)^2}$, $\forall E \in \mathcal{E}(g(S,\theta))$, which removes the $|E|$ term from $r(S,B,\theta)$ and also remains constant as $E \in \mathcal{E}(g(S,\theta))$ changes. In the following, we show Proposition 3.2, which provides some favorable properties of $r'(S,B,\theta)$.

Proposition 3.2. *Define r' as $r'(S,B,\theta) = \sqrt{\sum_{e\in E}(e \cdot \sum_{u \in g(B,\theta)} u)^2}$, $\forall E \in \mathcal{E}(g(S,\theta))$. $r'(S,B,\theta)$ is submodular (Bach, 2013), normalized, and monotone (referring to Definition B.1 for detailed definition).*

As per Proposition 3.2, $r'(S,B,\theta)$ is submodular, normalized, and monotone for $S \subset B$. Considering that Algorithm 1 is also a greedy algorithm for maximizing $r'(S,B,\theta)$, we can establish that Algorithm 1 has a $1-e^{-1}$ approximation ratio *w.r.t.* the optimal objective value for maximizing $r'(S,B,\theta)$ (Nemhauser et al., 1978) (also referring to Lemma B.3 in Appendix).

Furthermore, by establishing the connection between the optimal solutions for $r(S,B,\theta)$ and $r'(S,B,\theta)$, along with the solutions output by Algorithm 1, and their corresponding objective values, we present Proposition 3.3.

Proposition 3.3. *Algorithm 1 returns a $1-e^{-1}$ approximation for $\arg\max_{S\subset B} r(S,B,\theta), \text{s.t.} |S| \leq N_S$. That is, denote S^* as the optimum subset for Equation (2), and S' as the output of Algorithm 1, we have*

$$r(S',B,\theta) \geq (1-e^{-1}) r(S^*,B,\theta).$$

Proposition 3.3 provides the theoretical guarantee for Algorithm 1 regarding the optimization problem defined in Equation (2) with an approximation ratio of $1-e^{-1}$. This enables us to effectively transform a subset optimization problem into a sequential selection problem. Please refer to Appendix B.3 for the detailed proof.

3.4. Realization

More Efficient Selection Process. In Algorithm 1, the operation in line 5 involves subtracting the corresponding components of E (an orthonormal basis of already selected samples) in all elements of $U = g(B,\theta)$, and then normalizing them. We further simplify this step in Algorithm 2 by only subtracting the orthogonal components on the Sum term, as shown in line 9. Please refer to Appendix B.4 for a detailed plain-text description of Algorithms 1 and 2. Here, we make an approximation by neglecting the normalization operation in Algorithm 1's line 5; refer to Appendix B.5 for details. Basically, we simplify the process of orthogonalization of $|B|$ elements (line 5 of Algorithm 1) w.r.t. E to the orthogonalization of only 1 element and a subtraction operation (line 6 and 9 of Algorithm 2), considerably reducing the cost of selection. Empirically, we observed that Algorithm 2 substantially reduces the selection time compared to Algorithm 1 while achieving comparable performance, as shown in Table 1.

Choice of selection features U. For the features $U = \{g(d_i,\theta)\}_{i=1}^{N_B}$ used in the selection, we leverage the gradient of each sample at the final layer. This choice is motivated by the following reasons: (1) Gradients directly capture the influence of data on model training; (2) Gradients are applicable to various supervised and unsupervised tasks without relying on specific task requirements or annotations; (3) The overhead of computing gradients at the final layer is negligible compared to the overall training cost of the batch.

3.5. Discussion

Existing reference-model-free batch selection methods independently score and select data in a sample-wise manner. Consequently, they cannot avoid selecting highly scored but mutually redundant samples, leading to a lack of diversity. In contrast, our DivBS evaluates the selected subset as a whole and eliminates the impact of inter-sample redundancy, ensuring the diversity of the selected samples. Additionally, through theoretical analysis and approximation of the objective function, we provide an efficient selection process.

Table 2. Final accuracies (↑) of DivBS and various baseline methods on CIFAR-10, CIFAR-100 and Tiny ImageNet with different budget ratio 10%, 20%, 30%. Bold indicates the best results. Experiments show that DivBS consistently outperforms all baselines.

Method	CIFAR-10			CIFAR-100			Tiny ImageNet		
Full Data Training	95.50%			77.28%			56.76%		
Budget ratio	10%	20%	30%	10%	20%	30%	10%	20%	30%
Uniform	92.06%	93.76%	94.61%	70.61%	74.18%	75.98%	48.36%	51.71%	53.76%
Train Loss	92.73%	93.87%	94.54%	65.12%	69.34%	72.62%	37.12%	45.23%	47.72%
Grad Norm	65.23%	76.23%	82.34%	64.72%	69.23%	72.34%	37.24%	44.34%	48.24%
Grad Norm IS	92.51%	93.78%	94.41%	69.34%	72.71%	73.21%	42.79%	47.34%	50.23%
SVP	57.38%	73.87%	82.34%	31.23%	43.35%	50.73%	19.34%	28.97%	34.24%
Moderate-BS	92.32%	93.57%	94.36%	70.21%	74.35%	75.34%	48.92%	51.36%	54.23%
CCS-BS	92.61%	93.88%	94.81%	71.11%	74.42%	76.21%	49.18%	52.43%	54.17%
DivBS	**94.65%**	**94.83%**	**95.07%**	**73.11%**	**76.10%**	**77.21%**	**50.84%**	**55.03%**	**55.94%**

Table 3. Epochs (↓) required for DivBS and various baseline methods to reach given target test accuracies on CIFAR-100 with different budget ratio 10%, 20%, 30%. The target accuracies are set at 80% and 90% of the full dataset training accuracy (77.28%), equivalent to 62% and 69%, respectively. *NR* indicates that the target accuracy is not reached. Bold indicates the best results.

Budget	Target Acc	Uniform	Train Loss	Grad Norm	Grad Norm IS	SVP	Moderate-BS	CCS-BS	**DivBS**
10%	62%	150	172	174	163	*NR*	153	152	**132**
	69%	177	*NR*	*NR*	196	*NR*	179	174	**165**
20%	62%	118	153	155	143	*NR*	123	118	**83**
	69%	148	195	194	182	*NR*	150	147	**130**
30%	62%	113	147	149	133	*NR*	111	114	**52**
	69%	146	175	174	170	*NR*	148	145	**111**

4. Experiments

4.1. Experimental Setup

Datasets. We conduct experiments to evaluate our DivBS on CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 (Krizhevsky et al., 2009), and Tiny ImageNet (Le & Yang, 2015) for image classification. We then conduct experiments in the context of class imabalance, specifically the CIFAR-100-LT dataset with imbalance ratio 100, obtained through exponential sampling (Cui et al., 2019). We further conduct experiments on more tasks, including semantic segmentation, cross-modal retrieval, and fine-tuning language models, using datasets PASCAL VOC 2012 trainaug (Chen et al., 2018), Wikipedia (Rasiwasia et al., 2010; Hu et al., 2021), and E2E NLG Challenge (Novikova et al., 2017).

Baselines. In addition to uniform sampling, we compared our DivBS with various reference-model-free baseline methods, including training loss (Kawaguchi & Lu, 2020), gradient norm (Katharopoulos & Fleuret, 2018a), gradient norm with importance sampling (gradient norm IS) (Katharopoulos & Fleuret, 2018a), and Selection-via-Proxy (SVP) (Coleman et al., 2020). Additionally, we applied the selection strategies of Moderate (Xia et al., 2023) and CCS (Zheng et al., 2023) in the online batch selection paradigm, referred to as Moderate-BS and CCS-BS. Both strategies have demonstrated superior performance in one-shot coreset selection with high pruning rates (Zheng et al., 2023).

Implementation details. For image classification, we use 18-layer ResNet as the backbone. The standard data augmentations are applied as in Cubuk et al. (2020). Models are trained using SGD with momentum of 0.9 and weight decay of 0.005 as the optimizer. The initial learning rate is set to 0.1. We train the model for 200 epochs with the cosine learning-rate scheduling. Following Mindermann et al. (2022); Deng et al. (2023), we set the budget of batch sample number as $N_S = 32$, and the budget ratio as $\frac{N_S}{N_B} = 10\%$ unless specified otherwise. Our implementations for semantic segmentation, cross-modal retrieval, and language model fine-tuning are aligned with the details in Chen et al. (2018), Hu et al. (2021), and Hu et al. (2022), respectively.

4.2. Performance Evaluation on Image Classification

We first empirically evaluate DivBS on CIFAR-10, CIFAR-100 and Tiny ImageNet. We report the final accuracy of different methods with budget ratio $\frac{N_S}{N_B} = \{10\%, 20\%, 30\%\}$ in Table 2. We can observe that DivBS significantly outperforms all baselines under different budgets across the three datasets. Furthermore, it is notable that no baseline consistently outperforms uniform sampling. Techniques

Table 4. Final accuracies of DivBS and baselines and epochs required to reach given target accuracies on CIFAR-100-LT (imbalance ratio 100). The target accuracies are set at 60% and 80% of the full dataset training accuracy (42.80%), equivalent to 26% and 35%, respectively. *NR* indicates that the target Acc is not reached.

Full Data Training Acc: 42.80%				
Budget	Method	Final Acc ↑	Epochs to target Acc ↓	
			26%	35%
10%	Uniform	26.97%	177	*NR*
	Moderate-BS	26.35%	183	*NR*
	CCS-BS	27.43%	178	*NR*
	DivBS	**31.74%**	**145**	*NR*
20%	Uniform	38.50%	95	144
	Moderate-BS	37.78%	106	163
	CCS-BS	38.72%	92	140
	DivBS	**39.46%**	**69**	**118**
30%	Uniform	39.67%	54	113
	Moderate-BS	39.43%	60	118
	CCS-BS	40.28%	53	107
	DivBS	**42.41%**	**48**	**84**

like Train Loss and Grad Norm, which focus on selecting challenging samples, may exhibit decent performance in certain CIFAR-10 scenarios but suffer significant performance drops on more intricate datasets such as CIFAR-100 and Tiny ImageNet. Even with Moderate-BS and CCS-BS, which employ strategies like selecting samples with intermediate or diverse metrics, achieving results comparable to uniform sampling, they still notably lag behind DivBS.

Moreover, Table 3 illustrates the number of epochs needed to reach some given target accuracies. We can observe that some baselines like Grad Norm IS and CCS-BS that achieve slightly higher final accuracy compared to uniform sampling in Table 2, do not gain an advantage in terms of convergence speed. Remarkably, DivBS not only effectively boosts the final accuracy but also expedites the model's convergence.

4.3. Performance Evaluation under Class Imbalance

We further evaluate DivBS on a more challenging task of imbalanced classification, a scenario often encountered in the real world (Menon et al., 2021; Hong et al., 2023; Fan et al., 2022; 2023; 2024). Class imbalance poses a greater challenge to the diversity of data selection, as the already scarce samples from the tail classes are more likely to be completely absent in the selected subset. Specifically, we experiment on CIFAR-100-LT (imbalance ratio 100) (Cui et al., 2019; Zhou et al., 2022; 2023) with different budget ratios 10%, 20%, 30%. We report the final accuracies and the numbers of epochs required to reach the target accuracies in Table 4. DivBS consistently outperforms baselines in both final performance and convergence speed, demonstrating its ability to ensure data diversity even under class imbalance.

Table 5. Final mIoUs of DivBS and various baseline methods and epochs required to reach given target mIoUs on PASCAL VOC 2012 trainaug (Chen et al., 2018) with different budget ratio 10%, 20%, 30%. The target mIoUs are set at 80% and 90% of the full dataset training mIoU (70.80%), equivalent to 57% and 64%, respectively. *NR* indicates that the target accuracy is not reached.

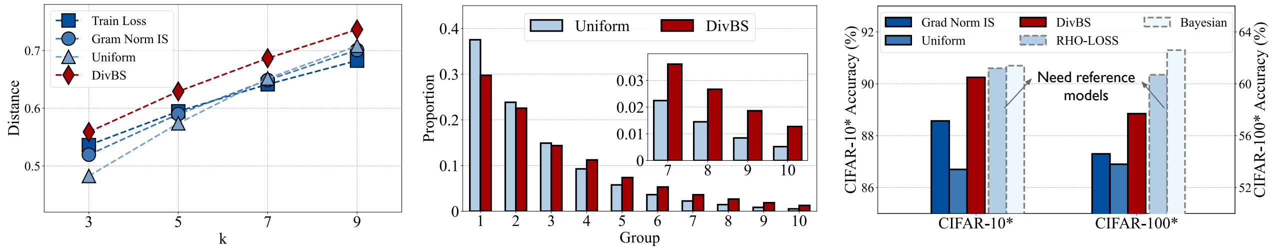
Full Data Training mIoU: 70.80%				
Budget	Method	Final mIoU ↑	Epochs to target mIoU ↓	
			57%	64%
10%	Uniform	63.72%	20	*NR*
	Moderate-BS	63.27%	23	*NR*
	CCS-BS	63.98%	21	*NR*
	DivBS	**65.45%**	**12**	**38**
20%	Uniform	67.07%	8	25
	Moderate-BS	66.83%	10	33
	CCS-BS	67.22%	8	26
	DivBS	**68.13%**	**7**	**19**
30%	Uniform	68.56%	8	19
	Moderate-BS	68.34%	9	22
	CCS-BS	68.47%	9	20
	DivBS	**69.85%**	**4**	**13**

Table 6. Mean Average Precision (MAP) scores for both image → text and text → image, along with their average on Wikipedia with different budget ratio 10%, 20%, 30%. Bold indicates the best results and red signifies improvements over full data training.

Budget	Methods	Img2Txt ↑	Txt2Img ↑	avg ↑
100%	Full Data Training	0.525	0.464	0.4945
10%	Uniform	0.508	0.457	0.4825
	DivBS	**0.512**	**0.462**	**0.487**
20%	Uniform	0.513	0.462	0.4875
	DivBS	**0.517**	**0.467**	**0.492**
30%	Uniform	0.518	0.464	0.491
	DivBS	**0.522**	**0.468**	**0.495**

4.4. Performance Evaluation on Semantic Segmentation

We compare the performance of different methods on PASCAL VOC 2012 trainaug dataset for semantic segmentation, which is a crucial and practical dense prediction task. Specifically, we train the DeepLabV3 (Chen et al., 2017) segmentation model with MobileNet (Howard et al., 2017) as the backbone for 50 epochs, aligning with the details provided in Chen et al. (2018). For Moderate-BS and CCS-BS, their original metrics, based on classification concepts, are not directly applicable to segmentation tasks. We replace them with training losses that similarly characterize the difficulty of samples, while retaining their selection strategies. We report the results in Table 5. Still, our proposed DivBS outperforms uniform sampling and other baseline methods in aspects of both the speed to reach the target mIoUs and the final mIoU for the semantic segmentation task.



(a) KNN distance

(b) Group Proportion

(c) CIFAR-10/100*

Figure 2. (a) Average mean feature cosine distance with the k-nearest neighbors for the selected data on CIFAR-10 (10% budget). (b) Properties of 10 groups in the selected data on CIFAR-100-LT. (c) Performance comparison on CIFAR-10* and CIFAR-100*. Note that Beyesian and RHO-LOSS requring reference models and also introduce additional overhead from using auxiliary models for inference.

Table 7. Performance of GPT-2 medium (M) with LoRA finetuning on the E2E NLG Challenge with budget ratio 20%.

GPT-2 M (LoRA)	Full Data Training	Uniform	**DivBS**
BLEU ↑	68.07%	66.14%	**66.87%**
NIST ↑	8.726	8.624	**8.685**
MET ↑	46.43%	44.26%	**44.57%**
ROUGE-L ↑	69.44%	66.90%	**67.83%**
CIDEr ↑	2.444	2.301	**2.343**

Table 8. Wall-clock time (↓) per epoch of different methods on CIFAR-10 and PASCAL VOC 2012 trainaug. The results are averaged over ten epochs. The subscript indicates the percentage of time saved compared to full data training.

CIFAR-10			
Full data training	Budget ratio	Uniform	**DivBS**
44.7s	10%	13.2s $_{\downarrow 70\%}$	13.9s $_{\downarrow 69\%}$
	20%	17.4s $_{\downarrow 61\%}$	18.5s $_{\downarrow 59\%}$
	30%	22.1s $_{\downarrow 51\%}$	23.6s $_{\downarrow 47\%}$
PASCAL VOC 2012 trainaug			
Full data training	Budget ratio	Uniform	**DivBS**
291.2s	10%	96.1s $_{\downarrow 67\%}$	100.9s $_{\downarrow 65\%}$
	20%	125.3s $_{\downarrow 58\%}$	132.4s $_{\downarrow 55\%}$
	30%	157.2s $_{\downarrow 46\%}$	166.9s $_{\downarrow 43\%}$

4.5. Performance Evaluation on Cross-Modal Retrieval

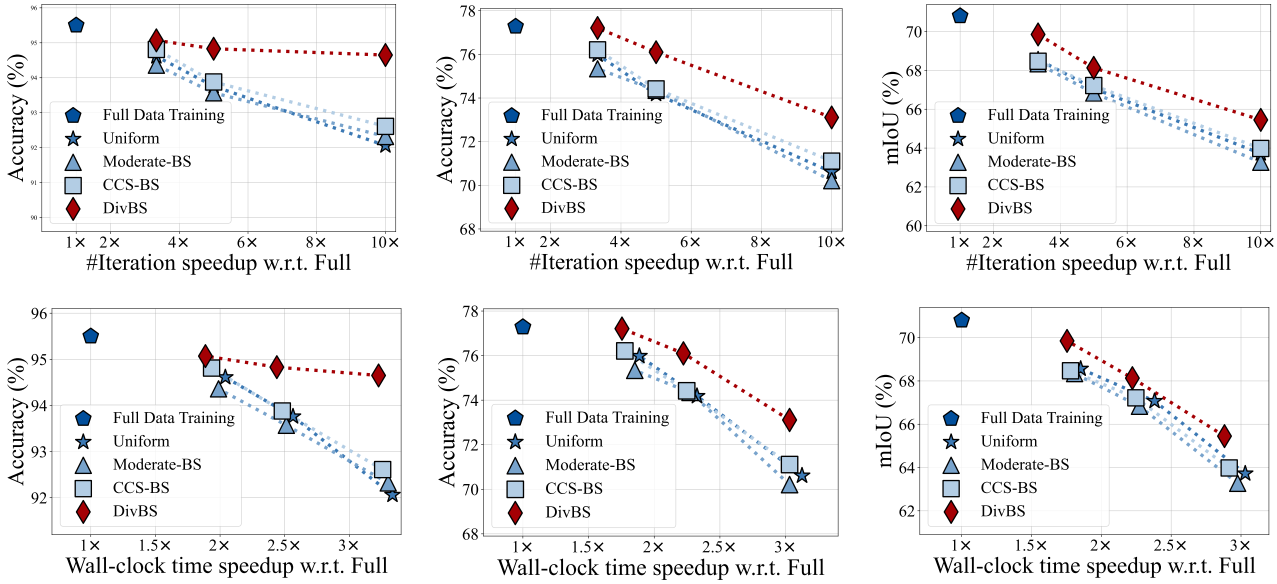
In addition to visual tasks, we extend our empirical research to evaluate DivBS in the cross-modal retrieval task. Specifically, we conduct experiments on Wikipedia with different budget ratio 10%, 20%, 30%. We employ an ImageNet-pretrained VGG-19 model (Simonyan & Zisserman, 2015) as the image backbone and a pre-trained Doc2Vec model (Lau & Baldwin, 2016) for text. All other implementation details align with Hu et al. (2021). In Table 6, we report the Mean Average Precision (MAP) score for both $\text{image} \rightarrow \text{text}$ and $\text{text} \rightarrow \text{image}$ retrieval, along with their average. It can be observed that our DivBS consistently outperforms uniform sampling in the cross-modal retrieval task. More impressively, DivBS even surpasses full-data training in terms of $\text{text} \rightarrow \text{image}$ MAP and average MAP at 20% and 30% budget ratios. This indicates that our DivBS remains effective in selecting diverse and high-quality subsets in the cross-modal retrieval task.

4.6. Performance Evaluation on LM Finetuning

We also validate the effectiveness of our DivBS on the language model finetuning task. Specifically, we finetune the GPT-2 Medium (M) model (Radford et al., 2019) using LoRA (Hu et al., 2022) on the E2E NLG Challenge (Novikova et al., 2017), which is widely used dataset for natural language generation evaluations. We finetune GPT-2 M for 5 epochs with a minibatch size of 4, aligning with the remaining implementation details in Hu et al. (2022). We report the results of full data training, uniform sampling and our DivBS with budget ratio 20% in Table 7. We can observe that our DivBS consistently outperforms uniform sampling across all 5 metrics, further demonstrating its universality across different tasks and training paradigms.

4.7. Further Analysis

Properties of the Selected Data. In Figure 2(a), we compare the average feature cosine distances to the k-nearest neighbors ($k = 1, 3, 5, 7, 9$) of samples selected by different methods on CIFAR-10. DivBS stands out with the largest KNN distances, highlighting the reduced redundancy and broader coverage of the selected samples. We then arrange the classes of CIFAR-100-LT in descending order of sample number, grouping every ten classes together, denoted as groups 1-10. In Figure 2(b), we illustrate the proportion of samples selected by uniform sampling and our DivBS for each group. It is evident that DivBS consistently increases the proportion of tail samples, demonstrating the effective enhancement of diversity in the selected subset.



(a) CIFAR-10

(b) CIFAR-100

(c) PASCAL VOC 2012 trainaug

Figure 3. Performance (↑) *v.s.* speedup (↑) on (a) CIFAR-10, (b) CIFAR-100, and (c) PASCAL VOC 2012 trainaug. The **upper** panel displays the relationship between the performance (accuracy or mIoU) of different methods and the speedup *w.r.t.* the number of training iterations. The **lower** panel illustrates the relationship between the performance and the speedup *w.r.t.* the wall-clock time.

Table 9. Final accuracies (↑) on CIFAR-100 with different budget ratio 10%, 20%, 30% when using SGD and AdamW as optimizer.

SGD			
Full data training	Budget ratio	Uniform	**DivBS**
77.28%	10%	70.61%	**73.11%**
	20%	74.18%	**76.10%**
	30%	75.98%	**77.21%**
AdamW			
Full data training	Budget ratio	Uniform	**DivBS**
70.50%	10%	66.72%	**69.07%**
	20%	68.93%	**70.43%**
	30%	69.09%	**70.47%**

Wall-clock time. Compared to uniform sampling, our DivBS incurs some additional overhead in the selection process. While our research primarily investigates data selection strategies favorable for model convergence, we still empirically measure and compare the practical impact of different methods on training duration. In Table 8, we report the wall-clock time per epoch on CIFAR-10 image classification and PASCAL VOC 2012 trainaug semantic segmentation. We can observe that the time proportion of uniform sampling compared to full time training is greater than the corresponding budget ratio. This discrepancy arises because batch selection only reduces the data used for network updates while operations like loading data, model validation, and saving model files still require the same amount of time. Our DivBS introduces additional overhead of less than 5% total time compared to uniform sampling. There is potential to further reduce this overhead using hardware techniques or parallelization methods, as discussed in Section 5.

Robustness with different optimizers. We validate the robustness of our DivBS under different optimizers. Table 9 showcases the performance of our DivBS on CIFAR-100 using both SGD and AdamW. DivBS consistently outperforms the baseline across various budget ratios. Moreover, with both optimizers, DivBS exhibits minimal performance loss at 30% budget compared to full dataset training.

Narrow the gap with methods involving extra reference models. In Figure 2(c), we compare our DivBS with RHO-LOSS (Mindermann et al., 2022) and Bayesian (Deng et al., 2023), which utilize extra reference models for selection, on CIFAR-10* and CIFAR-100* with 10% budget. The implementation details are strictly aligned with those of RHO-LOSS and Bayesian. CIFAR-10/100* are versions of CIFAR-10/100, with only half of the data retained (Mindermann et al., 2022). Our DivBS significantly narrows the gap with methods that utilize extra reference models.

Trade-off between performance and speedup. While our primary aim is to reduce training costs while preserving performance, there inherently exists a trade-off between model performance and acceleration effects. In Figure 3, we present the trade-off between the performance and speedup (*w.r.t.* training iterations and wall-clock time) of various methods on CIFAR-10, CIFAR-100, and PASCAL VOC 2012 trainaug datasets. Points located in the upper-right corner of the subfigures indicate superior performance coupled with enhanced acceleration effects. Notably, our DivBS excels in achieving a superior performance-speedup trade-off.

5. Related Work

Coreset selection (Mirzasoleiman et al., 2020; Xin et al., 2024), also known as data pruning, aims to create a smaller subset (coreset) of the original data that captures essential patterns for efficient model training. Various metrics like the entropy score (Coleman et al., 2020), EL2N score (Paul et al., 2021), forgetting score (Toneva et al., 2019), and classification margin (Pleiss et al., 2020), are used to measure individual differences among data points. Yet, selecting samples with the highest scores can lead to diversity issues, especially at high pruning rates, resulting in performance degradation (Xia et al., 2023). Zheng et al. (2023); Xia et al. (2023) propose strategies of selecting samples with intermediate scores or with diverse scores, yielding promising results under high pruning rates. However, coreset selection faces limitations in prioritizing samples with diverse properties during different training stages. Moreover, the acceleration benefits are noticeable only when the coreset is repeatedly used to train various models, as the data selection relies on a full data trained model.

Curriculum learning (Bengio et al., 2009) seeks to enhance model performance with minimal computational costs by prioritizing "easy" samples before uniformly training on the entire dataset (Jiang et al., 2015; Sinha et al., 2020; Zhou et al., 2020). Although curriculum learning can improve model convergence, it may not efficiently reduce training expenses. And they fall short in addressing the challenge of skipping redundant points that have already been learned.

Online batch selection speeds up model training by using only a portion of data in each batch. Jiang et al. (2019); Katharopoulos & Fleuret (2018b); Loshchilov & Hutter (2015) prioritize hard samples based on criteria like training loss or gradient norm, but these can hinder early-stage model convergence and be sensitive to outliers. Mindermann et al. (2022) and Deng et al. (2023) achieve notable speedup by leveraging additional reference models to select valuable samples. However, their practical applications are restricted by the availability of well-performing reference models. Compared to prior methods, which score and select data in a sample-wise manner, our reference-model-free DivBS, excels in selecting high-quality and diverse samples by optimizing the overall orthogonalized representativeness of the subset after removing inter-sample redundancy.

Acceleration of the Selection Process. Except for uniform sampling, online batch selection methods generally require an additional forward pass for each batch. Jouppi et al. (2017) achieve nearly 10× acceleration in forward propagation by leveraging low-precision cores on GPUs or TPUs, grounded in the observation that forward propagation exhibits higher tolerance to low precision. Alain et al. (2015) utilize a group of workers to asynchronously execute forward propagation and selection while the main process trains on the recently chosen data, thereby saving time of selection processes. Selection can also be cheaper by reusing features, gradients, or losses computed in previous epochs (Loshchilov & Hutter, 2015). Though our research scope is limited to the effects of different selection strategies, exploring the integration of these techniques for maximum acceleration is a promising and noteworthy avenue.

6. Conclusion

We investigate the diversity challenge that may arise from existing batch selection methods independently scoring and selecting data in a sample-wise manner. We propose Diversified Batch Selection (DivBS), which selects diverse and representative subsets by optimizing the overall orthogonalized representativeness after removing inter-sample redundancy, thereby accelerating model training. Extensive experiments validate the superiority of our DivBS in performance-speedup tradeoff across various tasks.

Acknowledgement

This work is supported by the National Key R&D Program of China (No. 2022ZD0160702), STCSM (No. 22511106101, No. 22511105700, No. 21DZ1100100), 111 plan (No. BP0719010) and National Natural Science Foundation of China (No. 62306178). LY is supported by Career Development Fund (CDF) of the Agency for Science, Technology and Research (A*STAR) (No: C233312007).

Impact Statement

Investigating the impact of data selection on the performance of minority group data is of paramount importance, especially for some social applications, such as medical diagnosis, auto-driving, and criminal justice. Some data selection algorithms may prioritize minority groups, due to their slower learning pace and the challenging nature of instances within these groups (Yang et al., 2023; Hong et al., 2024). Conversely, other algorithms may downplay the importance of rare groups, as excluding them has a minimal impact on the overall performance of the training set. Our method is designed to guarantee the diversity of selected data, thereby somewhat safeguarding the priority of minority group data within the selection process, as demonstrated in experiments under class imbalance.

References

Agarwal, S., Arora, H., Anand, S., and Arora, C. Contextual diversity for active learning. In *ECCV*, pp. 137–153. Springer, 2020.

Alain, G., Lamb, A., Sankar, C., Courville, A., and Bengio, Y. Variance reduction in sgd by distributed importance sampling. *ArXiv preprint*, abs/1511.06481, 2015.

Bach, F. R. Learning with submodular functions: A convex optimization perspective. *Found. Trends Mach. Learn.*, 6 (2-3):145–373, 2013.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *ICML*, volume 382, pp. 41–48, 2009.

Chen, L., Papandreou, G., Schroff, F., and Adam, H. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

Chen, L., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, volume 11211 of *Lecture Notes in Computer Science*, pp. 833–851. Springer, 2018.

Coleman, C., Yeh, C., Mussmann, S., Mirzasoleiman, B., Bailis, P., Liang, P., Leskovec, J., and Zaharia, M. Selection via proxy: Efficient data selection for deep learning. In *ICLR*. OpenReview.net, 2020.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. Randaugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, 2020.

Cui, Y., Jia, M., Lin, T., Song, Y., and Belongie, S. J. Class-balanced loss based on effective number of samples. In *CVPR*, pp. 9268–9277. Computer Vision Foundation / IEEE, 2019.

Das, A. M., Bhatt, G., Bhalerao, M. M., Gao, V. R., Yang, R., and Bilmes, J. Accelerating batch active learning using continual learning techniques. *TMLR*, 2023. ISSN 2835-8856.

Deng, Z., Cui, P., and Zhu, J. Towards accelerated model training via bayesian data selection. In *NeurIPS*, 2023.

Fan, Z., Wang, Y., Yao, J., Lyu, L., Zhang, Y., and Tian, Q. Fedskip: Combatting statistical heterogeneity with federated skip aggregation. In *ICDM*, pp. 131–140. IEEE, 2022.

Fan, Z., Yao, J., Han, B., Zhang, Y., Wang, Y., et al. Federated learning with bilateral curation for partially class-disjoint data. In *NeurIPS*, volume 36, 2023.

Fan, Z., Hu, S., Yao, J., Niu, G., Zhang, Y., Sugiyama, M., and Wang, Y. Locally estimated global perturbations are better than local perturbations for federated sharpness-aware minimization. In *ICML*, 2024.

Guo, C., Zhao, B., and Bai, Y. Deepcore: A comprehensive library for coreset selection in deep learning. In Strauss, C., Cuzzocrea, A., Kotsis, G., Tjoa, A. M., and Khalil, I. (eds.), *DESA*, pp. 181–195. Springer International Publishing, 2022. ISBN 978-3-031-12423-5.

Hong, F., Yao, J., Zhou, Z., Zhang, Y., and Wang, Y. Long-tailed partial label learning via dynamic rebalancing. In *ICLR*. OpenReview.net, 2023.

Hong, F., Yao, J., Lyu, Y., Zhou, Z., Tsang, I., Zhang, Y., and Wang, Y. On harmonizing implicit subpopulations. In *ICLR*. OpenReview.net, 2024.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *ICLR*. OpenReview.net, 2022.

Hu, P., Peng, X., Zhu, H., Zhen, L., and Lin, J. Learning cross-modal retrieval with noisy labels. In *CVPR*, pp. 5403–5413. Computer Vision Foundation / IEEE, 2021.

Jiang, A. H., Wong, D. L.-K., Zhou, G., Andersen, D. G., Dean, J., Ganger, G. R., Joshi, G., Kaminsky, M., Kozuch, M. A., Lipton, Z. C., and Pillai, P. Accelerating deep learning by focusing on the biggest losers. *ArXiv*, abs/1910.00762, 2019.

Jiang, L., Meng, D., Yu, S., Lan, Z., Shan, S., and Hauptmann, A. G. Self-paced learning with diversity. In *NeurIPS*, pp. 2078–2086, 2014.

Jiang, L., Meng, D., Zhao, Q., Shan, S., and Hauptmann, A. G. Self-paced curriculum learning. In *AAAI*, pp. 2694–2700. AAAI Press, 2015.

Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., et al. In-datacenter performance analysis of a tensor processing unit. In *ISCA*, pp. 1–12, 2017.

Katharopoulos, A. and Fleuret, F. Not all samples are created equal: Deep learning with importance sampling. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2530–2539. PMLR, 2018a.

Katharopoulos, A. and Fleuret, F. Not all samples are created equal: Deep learning with importance sampling. In

Dy, J. and Krause, A. (eds.), *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2525–2534. PMLR, 10–15 Jul 2018b.

Kawaguchi, K. and Lu, H. Ordered SGD: A new stochastic optimization framework for empirical risk minimization. In *AISTATS*, volume 108 of *Proceedings of Machine Learning Research*, pp. 669–679. PMLR, 2020.

Killamsetty, K., Sivasubramanian, D., Ramakrishnan, G., and Iyer, R. K. GLISTER: generalization based data subset selection for efficient and robust learning. In *AAAI*, pp. 8110–8118. AAAI Press, 2021.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollar, P., and Girshick, R. Segment anything. In *ICCV*, pp. 4015–4026, October 2023.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kulesza, A., Taskar, B., et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

Lau, J. H. and Baldwin, T. An empirical evaluation of doc2vec with practical insights into document embedding generation. In *Rep4NLP@ACL 2016*, pp. 78–86. Association for Computational Linguistics, 2016.

Le, Y. and Yang, X. S. Tiny imagenet visual recognition challenge. 2015.

Loshchilov, I. and Hutter, F. Online batch selection for faster training of neural networks. *ArXiv*, abs/1511.06343, 2015.

Menon, A. K., Jayasumana, S., Rawat, A. S., Jain, H., Veit, A., and Kumar, S. Long-tail learning via logit adjustment. In *ICLR*. OpenReview.net, 2021.

Mindermann, S., Brauner, J. M., Razzak, M. T., Sharma, M., Kirsch, A., Xu, W., Höltgen, B., Gomez, A. N., Morisot, A., Farquhar, S., and Gal, Y. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15630–15649. PMLR, 17–23 Jul 2022.

Mirzasoleiman, B., Bilmes, J. A., and Leskovec, J. Coresets for data-efficient training of machine learning models. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6950–6960. PMLR, 2020.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.*, 14(1):265–294, 1978.

Novikova, J., Dusek, O., and Rieser, V. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 201–206. Association for Computational Linguistics, 2017.

OpenAI. GPT-4 technical report. *ArXiv*, abs/2303.08774, 2023.

Paul, M., Ganguli, S., and Dziugaite, G. K. Deep learning on a data diet: Finding important examples early in training. In *NeurIPS*, pp. 20596–20607, 2021.

Pleiss, G., Zhang, T., Elenberg, E. R., and Weinberger, K. Q. Identifying mislabeled data using the area under the margin ranking. In *NeurIPS*, 2020.

Qin, Z., Wang, K., Zheng, Z., Gu, J., Peng, X., xu Zhao Pan, Zhou, D., Shang, L., Sun, B., Xie, X., and You, Y. Infobatch: Lossless training speed up by unbiased dynamic data pruning. In *ICLR*, 2024.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Rasiwasia, N., Pereira, J. C., Coviello, E., Doyle, G., Lanckriet, G. R. G., Levy, R., and Vasconcelos, N. A new approach to cross-modal multimedia retrieval. In *Proceedings of the 18th International Conference on Multimedia*, pp. 251–260. ACM, 2010.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., Chen, X., and Wang, X. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10684–10695, June 2022.

Sener, O. and Savarese, S. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Sinha, S., Garg, A., and Larochelle, H. Curriculum by smoothing. In *NeurIPS*, 2020.

Toneva, M., Sordoni, A., des Combes, R. T., Trischler, A., Bengio, Y., and Gordon, G. J. An empirical study of example forgetting during deep neural network learning. In *ICLR*. OpenReview.net, 2019.

Tremblay, N., Barthelmé, S., and Amblard, P.-O. Determinantal point processes for coresets. *J. Mach. Learn. Res.*, 20:168–1, 2019.

van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86): 2579–2605, 2008.

Wei, K., Iyer, R. K., and Bilmes, J. A. Submodularity in data subset selection and active learning. In Bach, F. R. and Blei, D. M. (eds.), *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1954–1963. JMLR.org, 2015.

Xia, X., Liu, J., Yu, J., Shen, X., Han, B., and Liu, T. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *ICLR*. OpenReview.net, 2023.

Xie, S. M., Santurkar, S., Ma, T., and Liang, P. Data selection for language models via importance resampling. In *NeurIPS*, 2023.

Xin, Z., Jiawei, D., Yunsong, L., Weiying, X., and Zhou, J. T. Spanning training progress: Temporal dual-depth scoring (tdds) for enhanced dataset pruning. In *CVPR*. Computer Vision Foundation / IEEE, 2024.

Yang, Y., Zhang, H., Katabi, D., and Ghassemi, M. Change is hard: A closer look at subpopulation shift. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 39584–39622. PMLR, 2023.

Zheng, H., Liu, R., Lai, F., and Prakash, A. Coverage-centric coreset selection for high pruning rates. In *ICLR*. OpenReview.net, 2023.

Zhou, T. and Bilmes, J. A. Minimax curriculum learning: Machine teaching with desirable difficulties and scheduled diversity. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Zhou, T., Wang, S., and Bilmes, J. A. Curriculum learning by optimizing learning dynamics. In Banerjee, A. and Fukumizu, K. (eds.), *AISTATS*, volume 130 of *Proceedings of Machine Learning Research*, pp. 433–441. PMLR, 2021.

Zhou, Y., Yang, B., Wong, D. F., Wan, Y., and Chao, L. S. Uncertainty-aware curriculum learning for neural machine translation. In *ACL*, pp. 6934–6944, Online, 2020. Association for Computational Linguistics.

Zhou, Z., Yao, J., Wang, Y., Han, B., and Zhang, Y. Contrastive learning with boosted memorization. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27367–27377. PMLR, 2022.

Zhou, Z., Yao, J., Hong, F., Zhang, Y., Han, B., and Wang, Y. Combating representation learning disparity with geometric harmonization. In *NeurIPS*, 2023.

A. Notations

In Table 10, we summarize the notations used in this paper.

Table 10. Summary of notations

Category	Notation	Description
Data and Sets	$\mathcal{D}$	Training data set
	B	(Large) training data batch
	N_B	Size of B
	d	a data point
	$S \subset B$	A smaller subset of B
	N_S	Budget of the size of S
	$U = g(B, \theta) = \{g(d_i, \theta)\}_{i=1}^{N_B}$	Features used for selection from B
	$u \in U$	A feature in U
	$\mathcal{E}(g(S, \theta))$	$\{E = \{e_i, \dots e_{\lvert E \rvert}\} \mid \forall e_i, e_j \in E, e_i \cdot e_j = \delta_{ij}, \mathrm{span}(E) = \mathrm{span}(g(S, \theta))\}$, Set of all potential orthonormal bases for the subspace spanned by $g(S, \theta)$
	$E \in \mathcal{E}(g(S, \theta))$	A orthonormal base for $g(S, \theta)$
	$e \in E$	A unit vector in E
	$j, k \in B \setminus S$	Two data points not in S
Model and functions	f_θ	a deep model with parameters θ
	θ	Model parameters
	$g(B, \theta)$, $g(d, \theta)$	Mapping function from data points to the selection features, given model parameters θ
	$s(u)$	Scoring function for sample-wise selection
	$r(S, B, \theta)$	$\max_{E \in \mathcal{E}(g(S,\theta))} \sum_{e \in E} \sum_{u \in g(B,\theta)} e \cdot u$, the orthogonalized representativeness of subset S with respect to B
	δ_{ij}	Kronecker delta, taking the value 1 when $i = j$ and 0 otherwise
	$\mathrm{span}(\cdot)$	Subspace spanned by all elements in a set
	$r'(S, B, \theta)$	$\sqrt{\sum_{e \in E}(e \cdot \sum_{u \in g(B,\theta)} u)^2}$, $\forall E \in \mathcal{G}_E(g(S, \theta))$, an auxiliary function introduced to study algorithm performance
	$F(\cdot)$	A general set function
	$\hat{\beta}(x)$	$\sqrt{a + x} - \sqrt{x}$ with $a > 0$
Others	e	Euler's Number, approximately equal to 2.71828

B. Technical Details

B.1. Proof of Proposition 3.1

Proof of Proposition 3.1.
Based on Equation (1):

$$\begin{aligned} r(S,B,\theta) &= \max_{E\in\mathcal{E}(g(S,\theta))} \sum_{e\in E} \sum_{u\in g(B,\theta)} e\cdot u \\ &= \max_{E\in\mathcal{E}(g(S,\theta))} (\sum_{e\in E} e)\cdot(\sum_{u\in g(B,\theta)} u) \\ &= \max_{E\in\mathcal{E}(g(S,\theta))} (\sum_{e\in E} e)\cdot(\sum_{e\in E}(e\cdot \sum_{u\in g(B,\theta)} u)e) \end{aligned} \tag{3}$$

where $\sum_{e\in E}(e\cdot\sum_{u\in g(B,\theta)} u)e$ is the projection of $\sum_{u\in g(B,\theta)} u$ onto the subspace spanned by $g(S,\theta)$, and it remains constant with variations of E, with a length of $\sqrt{\sum_{e\in E}(e\cdot\sum_{u\in g(B,\theta)} u)^2}$. The length of $\sum_{e\in E} e$ is a constant $\sqrt{|E|}$, and it can take any direction as E varies. Therefore, when E changes to align the directions of $\sum_{e\in E}(e\cdot\sum_{u\in g(B,\theta)} u)e$ and $\sum_{e\in E} e$, the term $(\sum_{e\in E} e)\cdot(\sum_{e\in E}(e\cdot\sum_{u\in g(B,\theta)} u)e)$ achieves its maximum value $\sqrt{\sum_{e\in E}(e\cdot\sum_{u\in g(B,\theta)} u)^2}\times\sqrt{|E|} = \sqrt{|E|\sum_{e\in E}(e\cdot\sum_{u\in g(B,\theta)} u)^2}$, which remains constant as E changes in $\mathcal{E}(g(S,\theta))$. Thus, $\forall E\in\mathcal{E}(g(S,\theta))$, we have

$$\sqrt{|E|\sum_{e\in E}(e\cdot\sum_{u\in g(B,\theta)} u)^2} = r(S,B,\theta). \tag{4}$$

□

B.2. Proof of Proposition 3.2

Definition B.1 (Proposition 2.3 in Bach (2013))**.** The set-function F is submodular if and only if for all $S\subset B$, and $j,k\in B\setminus S$, we have

$$F(S\cup\{k\}) - F(S) \geq F(S\cup\{j,k\}) - F(S\cup\{j\}).$$

And the function is called normalized if $F(\emptyset)=0$ and monotone if and only if $F(S')\leq F(S), \forall S'\subset S$.

Proof of Proposition 3.2.
Given $r'(S,B,\theta)=\sqrt{\sum_{e\in E}(e\cdot\sum_{u\in g(B,\theta)} u)^2}, \forall E\in\mathcal{E}(g(S,\theta))$. Note that r' remains constant for arbitrary choice of the basis E spanned the same subspace. For all $S\subset B$, and $j,k\in B\setminus S$, we discuss this problem case by case.

(1) $g(j,\theta)$ is in the subspace spanned by $g(S,\theta)$, *i.e.*, $g(j,\theta)-\sum_{e\in E}(e\cdot g(j,\theta))e=0$. we have

$$\begin{aligned} &r'(S,B,\theta) = r'(S\cup\{j\},B,\theta) \\ &r'(S\cup\{k\},B,\theta) = r'(S\cup\{j,k\},B,\theta) \end{aligned} \tag{5}$$

Thus, we have

$$r'(S\cup\{k\},B,\theta) - r'(S,B,\theta) = r'(S\cup\{j,k\},B,\theta) - r'(S\cup\{j\},B,\theta). \tag{6}$$

(2) $g(j,\theta)$ is not in the subspace spanned by $g(S,\theta)$, *i.e.*, $g(j,\theta)-\sum_{e\in E}(e\cdot g(j,\theta))e\neq 0$. Define e_j as

$$e_j = \frac{g(j,\theta)-\sum_{e\in E}(e\cdot g(j,\theta))e}{\|g(j,\theta)-\sum_{e\in E}(e\cdot g(j,\theta))e\|}, \tag{7}$$

we have $E\cup\{e_j\}\in\mathcal{E}(g(S\cup\{j\},\theta))$.

(2a) $g(k,\theta)$ is in the subspace spanned by $g(S \cup \{j\}, \theta)$. We have

$$\begin{aligned} &r'(S \cup \{j,k\}, B, \theta) - r'(S \cup \{j\}, B, \theta) = 0 \\ &r'(S \cup \{k\}, B, \theta) - r'(S, B, \theta) \geq 0 = r'(S \cup \{j,k\}, B, \theta) - r'(S \cup \{j\}, B, \theta) \end{aligned} \tag{8}$$

(2b) $g(k,\theta)$ is not in the subspace spanned by $g(S \cup \{j\}, \theta)$. Define e_k as

$$e_k = \frac{g(k,\theta) - \sum_{e \in E}(e \cdot g(k,\theta))e - (e_j \cdot g(k,\theta))e_j}{\|g(k,\theta) - \sum_{e \in E}(e \cdot g(k,\theta))e - (e_j \cdot g(k,\theta))e_j\|}, \tag{9}$$

we have

$$\begin{aligned} &E \cup \{e_k\} \in \mathcal{E}(g(S \cup \{k\}, \theta)) \\ &E \cup \{e_j, e_k\} \in \mathcal{E}(g(S \cup \{j,k\}, \theta)) \end{aligned} \tag{10}$$

Then

$$\begin{aligned} r'(S, B, \theta) &= \sqrt{\sum_{e \in E}(e \cdot \sum_{u \in g(B,\theta)} u)^2} \\ r'(S \cup \{k\}, B, \theta) &= \sqrt{\sum_{e \in E}(e \cdot \sum_{u \in g(B,\theta)} u)^2 + (e_k \cdot \sum_{u \in g(B,\theta)} u)^2} \\ r'(S \cup \{j\}, B, \theta) &= \sqrt{\sum_{e \in E}(e \cdot \sum_{u \in g(B,\theta)} u)^2 + (e_j \cdot \sum_{u \in g(B,\theta)} u)^2} \\ r'(S \cup \{j,k\}, B, \theta) &= \sqrt{\sum_{e \in E}(e \cdot \sum_{u \in g(B,\theta)} u)^2 + (e_j \cdot \sum_{u \in g(B,\theta)} u)^2 + (e_k \cdot \sum_{u \in g(B,\theta)} u)^2} \end{aligned} \tag{11}$$

Note that the function $\hat{\beta}(x) = \sqrt{a+x} - \sqrt{x}$ with $a > 0$ is a decreasing function w.r.t. x. Let $a = (e_k \cdot \sum_{u \in g(B,\theta)} u)^2$, $x_1 = \sum_{e \in E}(e \cdot \sum_{u \in g(B,\theta)} u)^2$ and $x_2 = \sum_{e \in E}(e \cdot \sum_{u \in g(B,\theta)} u)^2 + (e_j \cdot \sum_{u \in g(B,\theta)} u)^2$, we have $x_1 < x_2$ and $\hat{\beta}(x_1) > \hat{\beta}(x_2)$. It follows that

$$r'(S \cup \{k\}, B, \theta) - r'(S, B, \theta) > r'(S \cup \{j,k\}, B, \theta) - r'(S \cup \{j\}, B, \theta) \tag{12}$$

In summary, for all $S \subset B$, and $j, k \in B \setminus S$, we have

$$r'(S \cup \{k\}, B, \theta) - r'(S, B, \theta) \geq r'(S \cup \{j,k\}, B, \theta) - r'(S \cup \{j\}, B, \theta) \tag{13}$$

And it's obvious that $r'(S, B, \theta)$ is normalized and monotone.

□

B.3. Proof of Proposition 3.3

Lemma B.2 (Nemhauser et al. (1978))**.** *Greedy maximization of a monotone, submodular function returns a set with value within a factor of* $1 - e^{-1}$ *from the optimum set with the same size.*

Lemma B.3. *Algorithm 1 returns a* $1 - e^{-1}$ *approximation for* $\arg\max_{S \subset B} r'(S, B, \theta), \text{s. t.} |S| \leq N_S$*. That is, denote* S^{**} *as the optimum subset for maximizing* $r'(S, B, \theta)$*, and* S' *is the output of Algorithm 1, we have*

$$r'(S', B, \theta) \geq (1 - e^{-1}) r'(S^{**}, B, \theta).$$

Proof of Lemma B.3.
Note that Algorithm 1 is also a greedy maximization of $r'(S, B, \theta)$, given Proposition 3.2 and Lemma B.2, the conclusion can be drawn immediately.

□

Proof of Proposition 3.3.
Denote S^* as the optimum subset for maximizing $r(S, B, \theta)$, S^{**} as the optimum subset for maximizing $r'(S, B, \theta)$, and S' as the output of Algorithm 1. Let E^*, E^{**}, E' be orthonormal bases corresponding to S^*, S^{**}, S', respectively. We can get

$$|E^*| = |E^{**}| = |E'| = \min(\text{rank}(B), N_S). \tag{14}$$

Otherwise, If $|E| < \min(\text{rank}(B), N_S)$, there must be at least one selected d_i that does not contribute to E, and an unselected d_j that can contribute a new element to E. Therefore, replacing d_i with d_j further increases the objective value, both from a global and a sequential greedy perspective.

Based on Lemma B.3, we have

$$\begin{aligned} r(S', B.\theta) &= \sqrt{\min(\text{rank}(B), N_S)} r'(S', B.\theta) \\ &\geq (1 - e^{-1}) \sqrt{\min(\text{rank}(B), N_S)} r'(S^{**}, B, \theta) \\ &\geq (1 - e^{-1}) \sqrt{\min(\text{rank}(B), N_S)} r'(S^{*}, B, \theta) \quad \text{(Definition of } S^{**}) \\ &= (1 - e^{-1}) r(S^*, B, \theta) \end{aligned} \tag{15}$$

□

B.4. Plain-text Description of Algorithms 1 and 2

Algorithm 1:

1. Initialize the selected subset S to an empty set and the corresponding orthonormal basis E to an empty set. Denote the sum of all features of elements of batch B as Sum: $S \leftarrow \emptyset$, $E \leftarrow \emptyset$, $\text{Sum} \leftarrow \sum_{u \in g(B,\theta)} u$.

2. Add a sample to S that maximizes the current $r(S, B, \theta) = \sqrt{|E| \sum_{e \in E} (e \cdot \sum_{u \in g(B,\theta)} u)^2}$, where $E \in \mathcal{E}(g(S, \theta))$.
   (a) Compute the contribution of each candidate sample when individually added to S to the orthonormal basis E: $E_{\text{Cand}} \leftarrow \frac{g(d,\theta) - \sum_{e \in E} (e \cdot g(d,\theta)) e}{|g(d,\theta) - \sum_{e \in E} (e \cdot g(d,\theta)) e|}$ for $d \in B$
   (b) Identify the sample that maximizes r: $\text{idx} \leftarrow \arg\max_i |e_i \cdot \text{Sum}|, e_i \in E_{\text{Cand}}$
   (c) Update E, S, B: $S \leftarrow S \cup \{d_{\text{idx}}\}$, $E \leftarrow E \cup \{e_{\text{idx}}\}$, $B \leftarrow B \setminus \{d_{\text{idx}}\}$

3. Repeat Step 2 until $|S| = N_S$.

Algorithm 2:

1. Initialize the selected subset S to an empty set and the corresponding orthonormal basis E to an empty set. Initialize Sum as the sum of all features of elements of batch B: $\leftarrow \emptyset$, $E \leftarrow \emptyset$, $\text{Sum} \leftarrow \sum_{u \in g(B,\theta)} u$.

2. Add a sample to S that approximately maximizes the current r, referring to Appendix B.5 for the approximation.
   (a) Select the sample d to be added at the current step: $d \leftarrow \arg\max_{d\in B} |g(d,\theta)\cdot \mathrm{Sum}|$
   (b) Compute the contribution of sample d to the current orthonormal basis: $e \leftarrow \frac{g(d,\theta)-\sum_{e\in E}(e\cdot g(d,\theta))e}{|g(d,\theta)-\sum_{e\in E}(e\cdot g(d,\theta))e|}$
   (c) Update E, S, B: $S \leftarrow S \cup \{d\}$, $E \leftarrow E \cup \{e\}$, $B \leftarrow B \setminus \{d\}$

B.5. Approximation from Algorithm 1 to Algorithm 2

In Algorithm 1, we select d_{greedy} as follows :

$$d_{greedy} = \arg\max_{d\in B} |\frac{g(d,\theta)-\sum_{e\in E}(e\cdot g(d,\theta))e}{\|g(d,\theta)-\sum_{e\in E}(e\cdot g(d,\theta))e\|}\cdot \sum_{u\in g(B,\theta)} u| \tag{16}$$

where E represents an orthogonal basis corresponding to the already selected samples (line 7,8 in Algorithm 1). If we disregard the normalization term $\|g(d,\theta)-\sum_{e\in E}(e\cdot g(d,\theta))e\|$, then

$$\begin{aligned}
&|g(d,\theta)-\sum_{e\in E}(e\cdot g(d,\theta))e\cdot \sum_{u\in g(B,\theta)} u| \\
&=|(g(d,\theta)-\sum_{e\in E}(e\cdot g(d,\theta))e)\cdot(\sum_{u\in g(B,\theta)} u-\sum_{e\in E}(e\cdot \sum_{u\in g(B,\theta)} u)e+\sum_{e\in E}(e\cdot \sum_{u\in g(B,\theta)} u)e)| \\
&=|(g(d,\theta)-\sum_{e\in E}(e\cdot g(d,\theta))e)\cdot(\sum_{u\in g(B,\theta)} u-\sum_{e\in E}(e\cdot \sum_{u\in g(B,\theta)} u)e| \\
&=|g(d,\theta)\cdot(\sum_{u\in g(B,\theta)} u-\sum_{e\in E}(e\cdot \sum_{u\in g(B,\theta)} u)e|
\end{aligned} \tag{17}$$

The third and fourth lines follow from the fact that for any $e_i \in E$, we have $(g(d,\theta)-\sum_{e\in E}(e\cdot g(d,\theta))e)\cdot e_i = 0$, and $(\sum_{u\in g(B,\theta)} u-\sum_{e\in E}(e\cdot\sum_{u\in g(B,\theta)} u)e)\cdot e_i = 0$. Note that $\arg\max_{d\in B}|g(d,\theta)\cdot(\sum_{u\in g(B,\theta)} u-\sum_{e\in E}(e\cdot\sum_{u\in g(B,\theta)} u)e)|$ corresponds to line 5 of Algorithm 2 given $\mathrm{Sum}=\sum_{u\in g(B,\theta)} u-\sum_{e\in E}(e\cdot\sum_{u\in g(B,\theta)} u)e$ (line 6,8,9 of Algorithm 2), we can employ Algorithm 2 to approximate Algorithm 1.

C. Supplement for Experiments

C.1. Toy Example (Figure 1)

In Figure 1, we visualize a toy motivating example involving a four-class classification task among red, blue, green and yellow points. Specifically, we sample 1000 red points, 300 blue points, 150 green points, and 20 yellow points from following normal distributions: $N([0,0],[1,1])$, $N([5,0],[1,1])$, $N([0,5],[1,1])$, and $N([5,5],[1,1])$, respectively. We use a two-layer MLP with 100 hidden neurons as the model for the toy study. We construct a batch B using all available training data. The toy models are trained using Adam with learning rate 0.001 for 100 epochs. The budget ratio is set to 10%.

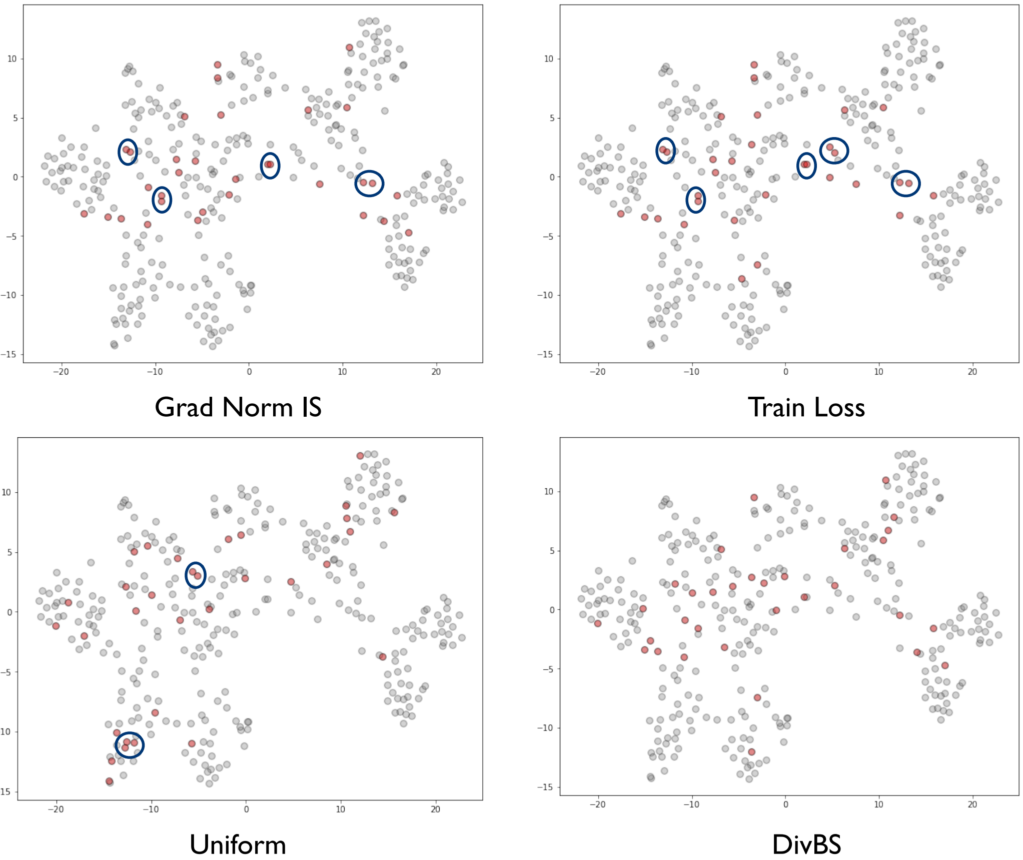
C.2. T-SNE Visualization

In Figure 4, we visualize subsets selected by different methods from the same batch of data on CIFAR-10. The batch size in all the experiments is set to 320. 10%-budget, i.e., 32 samples are selected. The t-SNE (van der Maaten & Hinton, 2008) visualization of the last layer features is shown in Figure 4. The red points represent the selected samples, and the gray points represent the full data. We have circled highly redundant samples. It is evident that baseline methods tend to select redundant samples, wasting data capacity, while our DivBS effectively avoids such issues.

C.3. Comparison with InfoBatch (Qin et al., 2024)

In this section, we compare the method InfoBatch (Qin et al., 2024). As it doesn't originally operate with a fixed budget (and the average budget exceeds our setup), we adapt it by using a percentile threshold based on the guidance[1] from the

[1] https://github.com/NUS-HPC-AI-Lab/InfoBatch/issues/16#issuecomment-1903467666



Grad Norm IS

Train Loss

Uniform

DivBS

Figure 4. T-SNE visualization of data selected by different methods on CIFAR-10 with 10% budget. Circles highlight redundant samples.

official repository to fix the budget ratio. The results are presented in Table 11. Note that, the results of InfoBatch are significantly lower than those of Uniform Sampling. This is due to the rescaling operation in InfoBatch, which can lead to unstable training, especially under the small budget. For example, with a budget of 10%, if we set the percentile threshold for infobatch to 95%, then infobatch needs to assign a weight of $\frac{0.95}{0.1-0.05} = 19$ times to the selected low-loss samples, which clearly destabilizes the training process. In contrast, the default threshold for InfoBatch is the mean, and it only clips 50% of samples below the mean, assigning a weight of 2 to low-loss samples. This may suggest that, rather than weighting losses of samples, simply selecting a subset of samples may be safer for accelerating model training.

C.4. Comparison with K-means++ Initialization Method

Given that K-means++ initialization method is also build to sample diverse centers, we conduct comparison between DivBS and k-means++ initialization method in Table 12. K-means++ initialization has shown comparable results to Uniform sampling, achieving some improvement at 10% and 20% budget ratios. This can be attributed to K-means++ initialization placing some emphasis on diversity, which may become more important as the budget decreases. However, DivBS still outperforms K-means++ initialization significantly, possibly due to the following reasons: 1) K-means++ initialization, as a heuristic algorithm, probabilistically samples points based on their distances from previously selected points, without a stable performance guarantee; 2) K-means++ initialization only considers distances between subset elements without considering the representativeness of the selected subset for the entire batch. In contrast, our proposed objective simultaneously considers both the diversity within the subset and the representativeness of the subset.

C.5. Error Bars

We provide error bars of Table 2 in Table 13.

Table 11. Accuracies on CIFAR-100 for different budget ratios.

Budget ratio	10%	20%	30%
Uniform	70.61%	74.18%	75.98%
InfoBatch	43.56%	68.22%	74.31%
DivBS	**73.11%**	**76.10%**	**77.21%**

Table 12. Final accuracies of DivBS and K-means++ initialization method on CIFAR-100 for different budget ratios. Full Data Training achieves 77.28% accuracy.

Budget ratio	10%	20%	30%
Uniform	70.61%	74.18%	75.98%
k-means++ initialization	71.14%	74.35%	75.76%
DivBS	**73.11%**	**76.10%**	**77.21%**

D. Other Sampling Methods Involving Diversity or Submodularity

Curriculum learning involving redundancy and diversity: self-paced learning with diversity (SPLD) (Jiang et al., 2014) is the first work to introduce diversity into curriculum learning, formalizing preferences for simple and diverse samples as a universal regularization term. MCL (Zhou & Bilmes, 2018) proposes that early training should focus on a small set of diverse samples, while later stages should prioritize training on larger, more challenging, and more homogeneous samples. Similar to MCL, DoCL (Zhou et al., 2021) promotes diversity through regularization using a submodular function.

Submodular coreset selection: Craig (Mirzasoleiman et al., 2020) attempt to find an coreset that approximates the gradients of the full dataset. They achieve this by transforming the gradient matching problem into the maximization of a monotone submodular. GLISTER (Killamsetty et al., 2021) formulates coreset selection as a mixed discrete-continuous bi-level optimization problem. It aims to select a subset of the training data that maximizes the log-likelihood on a held-out validation set. Additionally, GLISTER establishes connections to submodularity.

Submodular active learning: Wei et al. (2015) discusses the connection between likelihood functions and submodularity. It demonstrates that under a cardinality constraint, maximizing the likelihood function is equivalent to maximizing submodular functions for Naive Bayes or Nearest Neighbor classifiers. This naturally provides a powerful tool for coreset selection. By introducing submodularity into Naive Bayes and Nearest Neighbor classifiers, they propose a novel framework for active learning called Filtered Active Submodular Selection (Fass). CAL (Das et al., 2023) integrates continual learning techniques into active learning to alleviate the high training costs associated with active learning. Similarly, it employs submodular functions to regularize the sampling points.

Diversity-aware active learning: some sampling methods also focus on diversity of the chosen elements in the active learning area (Ren et al., 2021), where a model proactively chooses and queries the most informative data points for annotation, aiming to enhance its performance with minimal labeled examples. For example, Sener & Savarese (2018) theoretically formalize the data selection process as a k-Center problem and introduce the CoreSet algorithm, while Agarwal et al. (2020) substitute the Euclidean distance with context-aware KL-divergence. Determinantal Point Process (DPP) (Kulesza et al., 2012; Tremblay et al., 2019) is also an effective sampling method for preventing redundancy.

Discussion: the majority of sampling methods involving diversity come with a high computational cost, requiring at least $O(N^2)$ or $O(N^3)$ to calculate set properties (such as pairwise distances or determinants) and $O(N^2)$ or $O(N^3)$ to perform the sampling process, where N is the number of all candidate elements. As a result, they are suitable only for small-scale offline sampling and are not applicable for large-scale online selection. In contrast, our method has been demonstrated to be sufficiently lightweight, enabling its application in accelerating training within the online batch selection paradigm.

E. Limitation and Future Work

Online batch selection methods require an additional forward pass for selecting a subset in each batch, which somewhat limits the upper bound of acceleration, especially when the budget is small. Similar to previous research of online batch

Table 13. Final accuracies (↑, mean ± std) of DivBS and various baseline methods on CIFAR-10, CIFAR-100 and Tiny ImageNet with different budget ratio 10%, 20%, 30%. Bold indicates the best results. Experiments show that DivBS consistently outperforms all baselines.

Method	CIFAR-10			CIFAR-100			Tiny ImageNet		
Full Data Training	95.50%			77.28%			56.76%		
Budget ratio	10%	20%	30%	10%	20%	30%	10%	20%	30%
Uniform	92.06% ± 0.19%	93.76% ± 0.14%	94.61% ± 0.19%	70.61% ± 0.34%	74.18% ± 0.37%	75.98% ± 0.31%	48.36% ± 0.23%	51.71% ± 0.28%	53.76% ± 0.32%
Train Loss	92.73% ± 0.22%	93.87% ± 0.16%	94.54% ± 0.21%	65.12% ± 0.36%	69.34% ± 0.31%	72.62% ± 0.37%	37.12% ± 0.25%	45.23% ± 0.30%	47.72% ± 0.25%
Grad Norm	65.23% ± 0.17%	76.23% ± 0.20%	82.34% ± 0.24%	64.72% ± 0.29%	69.23% ± 0.34%	72.34% ± 0.28%	37.24% ± 0.21%	44.34% ± 0.26%	48.24% ± 0.31%
Grad Norm IS	92.51% ± 0.20%	93.78% ± 0.25%	94.41% ± 0.16%	69.34% ± 0.35%	72.71% ± 0.30%	73.21% ± 0.26%	42.79% ± 0.23%	47.34% ± 0.28%	50.23% ± 0.13%
SVP	57.38% ± 0.15%	73.87% ± 0.08%	82.34% ± 0.22%	31.23% ± 0.27%	43.35% ± 0.32%	50.73% ± 0.26%	19.34% ± 0.17%	28.97% ± 0.22%	34.24% ± 0.27%
Moderate-BS	92.32% ± 0.18%	93.57% ± 0.23%	94.36% ± 0.18%	70.21% ± 0.23%	74.35% ± 0.18%	75.34% ± 0.24%	48.92% ± 0.23%	51.36% ± 0.20%	54.23% ± 0.32%
CCS-BS	92.61% ± 0.21%	93.88% ± 0.16%	94.81% ± 0.11%	71.11% ± 0.36%	74.42% ± 0.31%	76.21% ± 0.37%	49.18% ± 0.26%	52.43% ± 0.21%	54.17% ± 0.36%
DivBS	**94.65%** ± 0.27%	**94.83%** ± 0.22%	**95.07%** ± 0.27%	**73.11%** ± 0.12%	**76.10%** ± 0.37%	**77.21%** ± 0.23%	**50.84%** ± 0.28%	**55.03%** ± 0.23%	**55.94%** ± 0.15%

selection, our scope is also limited to the effects of different selection strategies. Exploring the integration of techniques discussed in Section 5 such as hardware acceleration for forward pass, parallelization, or leveraging historical training information to avoid additional data loading and forward pass for maximum acceleration is a promising and noteworthy avenue.