OPT2025: 17th Annual Workshop on Optimization for Machine Learning

Balanced Locality-Sensitive Hashing for Online Data Selection

Hoang Phan HVP2011@NYU.EDU
Yijun Dong YD1319@NYU.EDU
Andrew Gordon Wilson ANDREWGW@CIMS.NYU.EDU
Qi Lei QL518@NYU.EDU
New York University

Abstract

Training contemporary foundation models is becoming an astronomical-scale, compute-limited optimization (instead of generalization) problem where heterogeneous data arrive in a stream whose storage is prohibitive, and a central question is how to spend gradient steps on more informative data that brings better convergence. We study online data selection as a variance reduction tool for stochastic optimization, and propose a balanced locality-sensitive hashing (LSH) sampler that is one-pass, simple, and lightweight. Our method has linear complexity in the batch size and gradient dimension and is insensitive to hyperparameters, making it a practical choice for streaming, compute-constrained training. Through extensive experiments on image/text classification and fine-tuning Llama 3 on mixed math corpora, we show that our method matches or exceeds the performance of strong diversity and uncertainty baselines with significantly better efficiency. Gradient similarity analyses further confirm that our selected subsets closely approximate full-data gradients, demonstrating both efficiency and effectiveness in diverse online data selection.

1. Introduction

With the unprecedented data and model sizes, foundation model pretraining increasingly operates in a compute-capped, data-rich regime where heterogeneous examples arrive continually in a stream, far exceeding what any fixed compute budget can process to convergence [74]. In this setting, each update step is a scarce resource in a nonstationary, streaming stochastic optimization problem. A core question is how to allocate computational resources to informative samples so that stochastic gradients remain low-variance and unbiased, yielding fast descent per unit compute.

Redundancy in streaming batches inflates gradient variance and thereby slows down convergence. Diversity-aware selection strategies (*e.g.*, k-center and k-means) counter this by spreading updates across the representation space. However, iterative clustering is too slow to run at every step in high-dimensional optimization problems. We seek a near one-pass, sublinear-overhead mechanism that preserves the benefits of diversity without paying the cost of heavy optimization.

We propose a novel method for online data selection based on Random Projection Locality Sensitive Hashing (LSH) to enforce diversity in a computationally efficient manner. LSH is a randomized hashing technique that maps similar items to the same bucket with high probability. Our method uses random projection LSH on the data's latent representations to partition the feature space into buckets, and then samples a small, roughly equal number of points from each bucket. In this way, each selection is forced to include data from across the feature space, limiting the chance of over-sampling any single dense region. Crucially, we introduce simple modifications to

standard LSH to balance the bucket sizes and avoid buckets with overwhelming numbers of points dominating the selection.

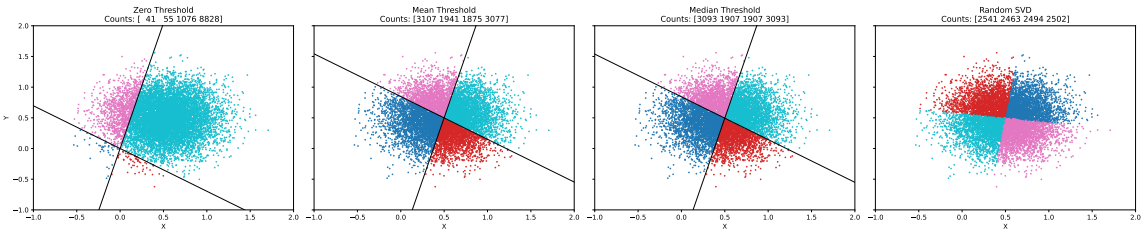

Figure 1: **Effect of threshold choice on LSH bucket balance**. Locality-sensitive hashing partitioning of 1000 random 2D points using two random hyperplanes and three different binarization thresholds. Points are colored by their 4-bucket assignment, and black lines indicate the hyperplanes under each scheme. (Left) Original LSH (through the origin) yields a highly imbalanced distribution. (Middle-left) Mean Threshold (shifted to the average projection) produces a more even split. (Middle-right) Median Threshold (shifted to the median projection) achieves near-uniform occupancy. (Right) Random SVD partition baseline yields roughly uniform splits.

In summary, our contributions are depicted as follows:

- We introduce a data selection algorithm by hashing data into buckets and samples across buckets to improve diversity. We address the inherent bucket size imbalance in vanilla LSH with lightweight adjustments that keep the method one-pass and efficient.
- The proposed method is significantly faster and cheaper than clustering-based diversity sampling (*e.g.*, k-center and k-means), while achieving comparable diversity. We provide complexity analysis and empirical timing to demonstrate the efficiency gains.
- Through experiments on large-scale image and text datasets, and on challenging math benchmarks, we show that our approach maintains or improves model performance relative to other baselines. Notably, it yields competitive accuracy while the gradient of the selected subset closely matches the full data gradient, indicating excellent representativeness.

Due to the space limit, we defer the detailed discussion on related works to Appendix A.

2. Why Balanced LSH Helps: An Optimization Perspective

For the optimization regime that online data selection generally lies in, the convergence of the stochastic optimizer is largely influenced by *variance and bias* of random gradient estimates. This section focuses on the simple yet popular stochastic gradient descent (SGD) and motivates our online data selection strategy via an illustrative toy example.

Effect of variance and bias on the convergence of SGD. Given a differentiable objective function $f : \mathbb{R}^d \to \mathbb{R}$, we consider solving the optimization problem $f_* = \min_{\boldsymbol{\theta}\in\mathbb{R}^d} f(\boldsymbol{\theta})$ via SGD. Initializing at any $\boldsymbol{\theta}_0 \in \mathbb{R}^d$ with $f(\boldsymbol{\theta}_0) - f_* = F > 0$, SGD iteratively updates $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta_t \mathbf{g}(\boldsymbol{\theta}_t, \xi_t)$ with some learning rate $\eta_t > 0$. $\mathbf{g}(\boldsymbol{\theta}_t, \xi_t) \in \mathbb{R}^d$ is a random estimate of $\nabla f(\boldsymbol{\theta}_t)$, with randomness from the independent random variable ξ_t, such that $\mathbf{g}(\boldsymbol{\theta}_t, \xi_t) = \nabla f(\boldsymbol{\theta}_t) + \mathbf{b}(\boldsymbol{\theta}_t) + \mathbf{v}(\boldsymbol{\theta}_t, \xi_t)$, where (i) $\mathbb{E}[\mathbf{v}(\boldsymbol{\theta}, \xi)] = 0$ (for all $\boldsymbol{\theta} \in \mathbb{R}^d$) represents variance; and (ii) $\mathbf{b}(\boldsymbol{\theta})$ represents bias.

The convergence of SGD with random (possibly biased) gradient estimates has been extensively studied in optimization literature [1, 8, 33]. For example, [1, Lemma 3] shows that when f is L-

smooth, *i.e.*, $f(\boldsymbol{\theta}') \leqslant f(\boldsymbol{\theta}) + \nabla f(\boldsymbol{\theta})^\top(\boldsymbol{\theta}' - \boldsymbol{\theta}) + \frac{L}{2}\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|_2^2$ for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$, given a gradient estimate $\mathbf{g}(\boldsymbol{\theta}, \xi)$ with (i) (c_v, σ^2)-bounded variance, *i.e.*, there exist $c_v, \sigma \geqslant 0$ such that

$$\mathbb{E}[\|\mathbf{v}(\boldsymbol{\theta}, \xi)\|_2^2] \leqslant c_v \|\nabla f(\boldsymbol{\theta}) + \mathbf{b}(\boldsymbol{\theta})\|_2^2 + \sigma^2 \quad \forall\, \boldsymbol{\theta} \in \mathbb{R}^d, \tag{1}$$

and (ii) (c_b, β^2)-bounded bias, *i.e.*, there exist $c_b \in [0, 1)$ and $\beta \geqslant 0$ such that

$$\|\mathbf{b}(\boldsymbol{\theta})\|_2^2 \leqslant c_b \|\nabla f(\boldsymbol{\theta})\|_2^2 + \beta^2 \quad \forall\, \boldsymbol{\theta} \in \mathbb{R}^d, \tag{2}$$

after any $T \geqslant \frac{2FL(c_v+1)^2}{\sigma^2}$ steps of SGD with constant learning rates $\eta_t = \eta = \sqrt{\frac{2F}{TL\sigma^2}}$, the average gradient norm square converges in expectation: $\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(\boldsymbol{\theta}_t)\|_2^2] \leqslant \frac{2}{1-c_b}\sqrt{\frac{2FL\sigma^2}{T}} + \frac{\beta^2}{1-c_b}$. In other words, with biased gradient estimates (*i.e.*, $c_b > 0$ or $\beta > 0$), the squared gradient norm of SGD is only guaranteed to converge to a $(\frac{\beta^2}{1-c_b})$-neighborhood of a stationary point. Meanwhile, with unbiased gradient estimates (*i.e.*, $c_b = \beta = 0$), the variance characterized by c_v, σ^2 determines the convergence rate to a stationary point of f.

Balanced LSH is unbiased with lower variance than uniform sampling. As a simple motivating example, we consider a linear regression problem over $(\mathbf{x}, y) \sim \mathcal{D}(\boldsymbol{\theta}_*)$ such that $y = \mathbf{x}^\top\boldsymbol{\theta}_* + z$ for some unknown ground truth $\boldsymbol{\theta}_* \in \mathbb{R}^d$ and an independent label noise $z \sim \mathcal{N}(0, \sigma_y^2)$. Let $\mathcal{D}$ be the marginal distribution of $\mathbf{x}$ and define $\boldsymbol{\Sigma} = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[\mathbf{x}\mathbf{x}^\top]$, $\mathbf{M} = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[(\mathbf{x}\mathbf{x}^\top)^2]$. We aim to learn $\boldsymbol{\theta}_*$ with squared loss, $\boldsymbol{\theta}_* = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d}\{f(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}(\boldsymbol{\theta}_*)}[\frac{1}{2}(\mathbf{x}^\top\boldsymbol{\theta} - y)^2]\}$, by applying SGD on subsampled online data batches.

Example 1 (Toy thought experiment: variance reduction via balanced LSH) *At each SGD step, we receive a batch of* n i.i.d. *samples* $\{(\mathbf{x}_i, y_i) \sim \mathcal{D}(\boldsymbol{\theta}_*)\}_{i=1}^n$ *and select a subset of* k *samples, indexed by* $S \subset [n]$ *(*$|S| = k$*), from it. Denote* $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]^\top \in \mathbb{R}^{n\times d}$, $\mathbf{y} = [y_1, \cdots, y_n] \in \mathbb{R}^n$, *and let* $\mathbf{X}_S \in \mathbb{R}^{k\times d}, \mathbf{y}_S \in \mathbb{R}^k$ *contain the selected subset. Given any current parameter* $\boldsymbol{\theta}$*, we aim to find a suitable* S *such that the gradient estimate based on the subsampled batch,* $\mathbf{g}(\boldsymbol{\theta}, (\mathbf{X}, \mathbf{y}, S)) = \frac{1}{k}\mathbf{X}_S^\top(\mathbf{X}_S\boldsymbol{\theta} - \mathbf{y}_S)$*, is close to the true gradient* $\nabla f(\boldsymbol{\theta})$*. We compare two selection strategies for* S *in terms of the variance and bias of* $\mathbf{g}(\boldsymbol{\theta}, (\mathbf{X}, \mathbf{y}, S))$*:*

(i) Uniform sampling draws k *samples* $\{i_1, \cdots, i_k\} \subset [n]$ *from* $(\mathbf{X}, \mathbf{y})$ *each with probability* $1/n$*. The resulting gradient estimate* $\mathbf{g}(\boldsymbol{\theta}, (\mathbf{X}, \mathbf{y}, S^{\text{uni}}))$ *is unbiased (*i.e. *satisfies* (2) *with* $c_b = \beta = 0$*) and satisfies* (1) *with* $\sigma^2 = \frac{\sigma_y^2}{k}\operatorname{tr}(\boldsymbol{\Sigma})$ *and* $c_v^{\text{uni}} = \frac{1}{k}\|\boldsymbol{\Sigma}^\dagger(\mathbf{M} - \boldsymbol{\Sigma}^2)\boldsymbol{\Sigma}^\dagger\|_2$ *where* $\boldsymbol{\Sigma}^\dagger$ *denotes the pseudoinverse of* $\boldsymbol{\Sigma}$*.*

(ii) Balanced LSH sampling, ideally, partitions the support of $\mathcal{D}$ *evenly into* $b \in \mathbb{N}$ *buckets,* $\operatorname{supp}(\mathcal{D}) = \cup_{\iota=1}^b \mathcal{X}_\iota$ *with* $\Pr(\mathbf{x} \in \mathcal{X}_\iota) = 1/b$ *and* $\mathcal{X}_\iota \cap \mathcal{X}_\nu = \emptyset$ *for all* $\iota, \nu \in [b]$*,* $\iota \neq \nu$*, and draws* k/b *samples uniformly from each bucket (assuming* n, k *are both divisible by* b *for simplicity). Let* $\boldsymbol{\Sigma}_\iota = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[\mathbf{x}\mathbf{x}^\top | \mathbf{x} \in \mathcal{X}_\iota]$ *for all* $\iota \in [b]$*. The resulting gradient estimate* $\mathbf{g}(\boldsymbol{\theta}, (\mathbf{X}, \mathbf{y}, S^{\text{lsh}}))$ *satisfies* (2) *with* $c_b = \beta = 0$ *and* (1) *with* $\sigma^2 = \frac{\sigma_y^2}{k}\operatorname{tr}(\boldsymbol{\Sigma})$ *and* $c_v^{\text{lsh}} = \frac{1}{k}\|\boldsymbol{\Sigma}^\dagger(\mathbf{M} - \frac{1}{b}\sum_{\iota=1}^b \boldsymbol{\Sigma}_\iota^2)\boldsymbol{\Sigma}^\dagger\|_2$*.*

Notice that $c_v^{\text{lsh}} \leqslant c_v^{\text{uni}}$ *since* $\frac{1}{b}\sum_{\iota=1}^b \boldsymbol{\Sigma}_\iota^2 \succeq \boldsymbol{\Sigma}^2 = (\sum_{\iota=1}^b \frac{1}{b}\boldsymbol{\Sigma}_\iota)^2$ *by Jensen's inequality. In particular, the gain of balanced LSH over uniform sampling,* $c_v^{\text{uni}} - c_v^{\text{lsh}}$*, due to the Jensen gap tends to be substantial when* $\boldsymbol{\Sigma}_1, \cdots, \boldsymbol{\Sigma}_b$ *are distinct, which is facilitated by the locality-sensitive sampling. While both uniform and balanced LSH sampling are unbiased with the same additive variance* σ^2*, balanced LSH tends to achieve lower* c_v *in* (1) *and therefore better convergence under SGD.*

We defer the detailed analysis of Example 1 to the supplementary material, while highlighting key insights from Example 1 in the following remark.

Remark 1 (Unbiasedness from balanced sampling, lower variance from LSH) *On one hand, the unbiasedness of* $\mathbf{g}(\boldsymbol{\theta}, (\mathbf{X}, \mathbf{y}, S^{\text{lsh}}))$ *in Example 1 comes from balanced sampling over evenly partitioned buckets. In contrast, sampling evenly from imbalanced buckets (*e.g.*, based on plain LSH or k-means) leads to biased gradient estimates and usually compromises the performance in practice (see Figure 1 and results in Section 4). On the other hand, balanced LSH achieves lower variance than uniform sampling intuitively by enforcing locality diversity in the selected samples.*

3. Locality-Sensitive Hashing (LSH) Algorithm for Paritioning Data

We consider an online data selection setting where, at time t, a mini-batch of n unlabeled or labeled examples $\mathcal{D}_t = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ arrives, and we aim to choose a fixed-size subset $\mathcal{S}_t \subset \mathcal{D}_t$, $|\mathcal{S}_t| = k < n$, to train (or fine-tune) our model.

Given a batch of n embedding vectors $X \in \mathbb{R}^{n \times d}$, we fix the number of buckets b and hyperplanes m. We sample $W \in \mathbb{R}^{m \times d}$ with *i.i.d.* entries $w_{j\ell} \sim \mathcal{N}(0, 1)$ and compute the projected embedding $P = X\,W^\top \in \mathbb{R}^{n \times m}$. We then choose a threshold vector $\mathbf{t} \in \mathbb{R}^m$ by

$$t_j = \begin{cases} 0, & \text{(zero)}\,, \\ \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} P_{ij}, & \text{(mean)}\,, \\ \text{median}\big\{P_{1j}, \ldots, P_{nj}\big\}, & \text{(median)}\,. \end{cases}$$

Each entry in the projected embedding is binarized as $B_{ij} = [\,P_{ij} > t_j\,] \in \{0, 1\}$, where each row $B_{i:}$ is interpreted as an m-bit integer $h_i = \sum_{j=1}^m B_{ij}\, 2^{j-1}$, and the final bucket assignment is $b_i = h_i \bmod b, \quad i = 1, \ldots, n.$.

Furthermore, in order to balance the number of datapoints across bucket, we also propose another strategy that compute an approximate rank-d randomized SVD $X \approx U\,\Sigma\,V^\top$, whiten via $\tilde{X} = X\,V\,\Sigma^{-1}$, set $P = \tilde{X}$, and then apply the same zero-threshold binarization and hashing steps above. Overall our proposed algorithm is depicted in Algorithm 1 where we divide the incoming minibatch of data X into b different buckets.

Bucket-aware sampling. Once each example has been assigned to a bucket, we build the selection batch of size k with a simple round-robin procedure. First, we gather all buckets that still contain unpicked items. Then, until we have k samples or all buckets are empty, we randomly shuffle the list of non-empty buckets. Then, for each bucket in this shuffled order, we remove one data point at random from the bucket and add it to our batch. If the bucket becomes empty, drop it from further consideration. This round-robin random extraction draws approximately one sample per bucket per pass, ensuring diversity across buckets.

4. Experiments

Due to space constraints, some experimental details and additional results are deferred to the appendix. Here we present the main results and ablation studies demonstrating the efficiency of the proposed algorithm. Empirically, we found that LSH with a mean or median threshold or LSH-SVD perform similarly in practice, thus we stick with LSH-median throughout the experiment.

Vision Classification on ImageNet-1K. We fine-tune a ResNet-50 [31] backbone on the ImageNet-1K [18] training set (1.28 M images, 1 000 classes) using Adam [39] ($\beta_1 = 0.9,\ \beta_2 = 0.999$) with a learning rate of 10^{-3} for one epoch to replicate the online data selection setup. Mini-batches consist of 256 images, of which 128 are selected by the data-selection algorithm at each step.

Product Title Text Classification. We experiment on a large-scale product-titles dataset (≈5 M examples) from Kaggle [6]. BERT-base-uncased [20] is fine-tuned for one epoch with Adam (same settings as above) at a learning rate of 10^{-3}. Input titles are truncated or padded to 128 tokens; total batch size is 1024, with 256 examples selected per step and we report accuracy on a held-out validation. For both image and text classification, we randomly initialize a classification head and optimize the network with cross-entropy loss.

Table 1: Performance comparison on ImageNet and Massive Product Text classification dataset for various selection methods. The best method for each experiment is indicated in **bold**. Only geometry-based methods outperform the random selection baseline in some scenarios.

	Random	MaxLoss	MinLoss	GradNorm	Entropy	LeastConfidence	Herding	KcenterGreedy	K-means	LSH
Image	33.62	28.11	22.23	33.28	30.87	32.29	28.58	31.08	34.59	**35.30**
Text	33.96	28.54	11.71	30.02	30.83	32.23	34.22	34.71	34.58	**35.32**

We report the results for our proposed method and baselines in Table 1, from which we observe consistent improvement over the random selection baseline on both text and image classification setups. In particular, LSH-based sampling method improves over random ones by 1.7% on ImageNet and 1.3% on Massive Product Text, respectively. Moreover, our proposed method achieves the best performance and outperforms the second-best baseline, K-means, by 0.8%. It is also worth noting that, while geometry-based data selection methods can obtain higher end task performance in some scenarios, none of the comparable methods can outperform in both settings, which aligns with prior observation [29].

Table 2: Performance of different data selection strategies across downstream tasks. We utilize the lm-evaluation-harness library to calculate the performance on different math benchmarks.

Task	Pretrained (%)	Random (%)	MaxLoss (%)	CL (%)	MidPerplexity (%)	LeastConfidence (%)	MTLD (%)	RewardScore (%)	K-means (%)	LSH (%)
gsm8k	26.08	32.68	31.16	32.98	33.89	33.89	33.06	34.80	33.28	**34.72**
gsm8k CoT	29.87	38.06	36.16	36.77	39.20	39.42	38.97	37.60	38.36	**40.56**
gsm Plus	15.59	20.96	20.32	20.49	20.07	20.15	20.49	20.15	20.49	**21.38**
MATH	19.84	**22.35**	21.17	20.46	20.66	20.46	21.17	20.31	20.50	21.69
MathQA	34.77	35.95	36.11	34.30	34.97	35.64	35.71	35.04	35.98	**37.02**
minerva_math	6.92	**8.66**	6.52	7.52	6.74	6.52	6.44	7.52	8.64	8.20
agieval_math	6.80	7.30	8.70	7.60	7.90	7.60	7.71	6.50	7.60	**9.00**
svamp	55.47	**59.53**	59.20	57.53	57.53	57.53	58.53	56.86	57.46	59.20
NumGLUE	48.00	51.43	51.43	48.57	48.57	49.71	48.57	49.71	52.57	**53.71**
asdiv	61.65	62.46	61.49	62.14	**63.27**	62.62	62.78	61.65	62.46	**63.27**
avg	30.50	33.94	33.23	32.84	33.28	33.35	33.34	33.01	33.73	**34.88**

Fine-Tuning language models for Math Reasoning. We merge threecorpora of math word problems-MathInstruct [67], Numi-Math 1.5, [43] and ORCA-Math [53] - and fine-tune the Llama-3.2-3B [68] model for 5000 steps using AdamW [46] ($\beta_1 = 0.9,\ \beta_2 = 0.999$) at a learning rate of 2×10^{-5}. We use mini-batches of 128 examples, selecting 64 per step. Evaluation is performed zero-shot on GSM8K [15] and GSM8K-CoT: [72], GSM-Plus [45], MathQA [3], MATH[32], Minerva-Math

[42], AGIEval-Math [76], SVAMP [54], NumGLUE[52] and ASDiv [50] dataset, measuring exact-match and solution-accuracy to quantify reasoning performance.

In Table 2, we test the performance of the fine-tuned Llama 3.2 3B model on different math tasks. Our method outperforms other baselines on average, especially on NumGLUE dataset, where its performance exceeds the random baseline by more than 2%. Overall, fine-tuning Llama using our method helps improve the performance of the pretrained model by 4.38% and also outperforms K-means by 1%. It is interesting to note that the RewardScore baseline, which employs a Math reward model to grade examples at each minibatch, could not outperform the random baseline in this setting.

Running-time comparison We first quantify the overhead of our selection strategy relative to common diversity baselines. Figure 3 shows the per-iteration selection time for each method: Balanced LSH with both mean and median thresholding incurs just about 1.2 ms per iteration, as low as MaxLoss, LeastConfidence and Entropy, while K-means clustering requires approximately 1.2 ms and herding even takes approximately 1.33 ms (20% overhead compared to ours).

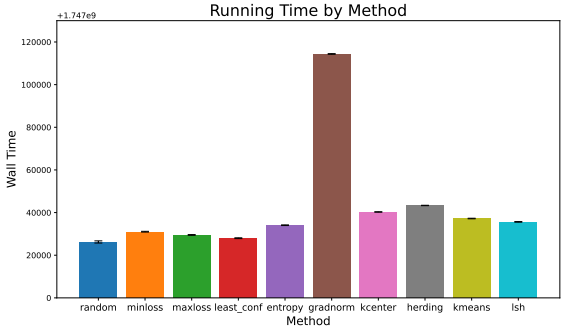

Figure 2: Total running time for each method, including both selection and training time.

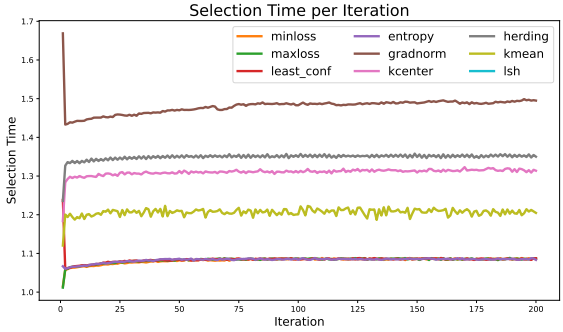

Figure 3: Selection time spent each iteration for different data selecitonmethods.

Correspondingly, when including both selection and training time in Figure 2, Balanced LSH reduces total wall-clock time by an order of magnitude compared to K-means, without sacrificing downstream accuracy. It can be seen from the figure that geometry-based methods incur small overhead compared to logit-based methods. Meanwhile, Gradnorm is the most expensive one as it computes per-sample gradient to rank the importance of individual samples.

5. Conclusion

In this paper, we introduce a novel geometry-driven online data selection method that projects data representations onto random hyperplanes to form hash buckets of roughly equal size via simple, data-adaptive thresholding. This lightweight procedure requires only a few dot-products per example—no costly clustering or optimization—yet enforces diversity by uniformly allocating selection budget across buckets and sampling at random within each. In experiments on different benchmarks, Balanced LSH matches or outperforms strong baselines while speeding up the selection time compared to other geometry-aware methods. With its minimal computational overhead and simplicity, our proposed method offers a practical, scalable method for diversity-aware data selection.

References

[1] Ahmad Ajalloeian and Sebastian U Stich. On the convergence of sgd with biased gradients. *arXiv preprint arXiv:2008.00051*, 2020.

[2] Zeyuan Allen-Zhu, Yuanzhi Li, Aarti Singh, and Yining Wang. Near-optimal discrete optimization for experimental design: A regret minimization approach. *arXiv preprint arXiv:1711.05174*, 2017.

[3] Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1245. URL https://aclanthology.org/N19-1245/.

[4] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM–SIAM Symposium on Discrete Algorithms (SODA)*, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[5] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM–SIAM Symposium on Discrete Algorithms (SODA)*, pages 1027–1035, 2007.

[6] Mateusz Asaniczka. Product titles text classification. https://www.kaggle.com/datasets/asaniczka/product-titles-text-classification, 2021.

[7] Francis Bach, Simon Lacoste-Julien, and Guillaume Obozinski. On the equivalence between herding and conditional gradient algorithms. *arXiv preprint arXiv:1203.4523*, 2012.

[8] Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.

[9] Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. Instruction mining: Instruction data selection for tuning large language models. In *Proceedings of the Workshop on Instruction Mining (COLM)*, 2024. URL https://openreview.net/pdf?id=wF6k0aWjAu.

[10] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical science*, pages 273–304, 1995.

[11] Samprit Chatterjee and Ali S Hadi. Influential observations, high leverage points, and outliers in linear regression. *Statistical science*, pages 379–393, 1986.

[12] Feng Chen, Xiaoming Li, and Yuan Zhao. Magicpig: Efficient attention approximation via locality-sensitive hashing. In *Proceedings of the 38th Conference on Neural Information Processing Systems*, 2024.

[13] Yifan Chen, Ethan N Epperly, Joel A Tropp, and Robert J Webber. Randomly pivoted cholesky: Practical approximation of a kernel matrix with few entry evaluations. *Communications on Pure and Applied Mathematics*, 78(5):995–1041, 2025.

[14] Yutian Chen and Max Welling. Parametric herding. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 97–104, 2010. URL https://proceedings.mlr.press/v9/chen10a.html.

[15] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, and et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[16] Michael B Cohen, Cameron Musco, and Christopher Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1758–1777. SIAM, 2017.

[17] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262, 2004.

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[19] Amit Deshpande and Santosh Vempala. Adaptive sampling and fast low-rank matrix approximation. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 292–303. Springer, 2006.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, 2019. URL https://arxiv.org/abs/1810.04805.

[21] Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 137–143. ACM, 2001.

[22] Yijun Dong and Per-Gunnar Martinsson. Simpler is better: a comparative study of randomized pivoting algorithms for cur and interpolative decompositions. *Advances in Computational Mathematics*, 49(4):66, 2023.

[23] Yijun Dong, Chao Chen, Per-Gunnar Martinsson, and Katherine Pearce. Robust blockwise random pivoting: Fast and accurate adaptive interpolative decomposition. *arXiv preprint arXiv:2309.16002*, 2023.

[24] Yijun Dong, Xiang Pan, Hoang Phan, and Qi Lei. Randomly pivoted v-optimal design: Fast data selection under low intrinsic dimension. In *Workshop on Machine Learning and Compression, NeurIPS 2024*, 2024.

[25] Yijun Dong, Viet Hoang Phan, Xiang Pan, and Qi Lei. Sketchy moment matching: Toward fast and provable data selection for finetuning. *Advances in Neural Information Processing Systems*, 37:43367–43402, 2024.

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

[27] Ethan N Epperly, Joel A Tropp, and Robert J Webber. Embrace rejection: Kernel matrix approximation by accelerated randomly pivoted cholesky. *arXiv preprint arXiv:2410.03969*, 2024.

[28] Valerii Vadimovich Fedorov. *Theory of optimal experiments*. Elsevier, 2013.

[29] Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning, 2022. URL https://arxiv.org/abs/2204.08499.

[30] Jujie He, Tianwen Wei, Rui Yan, Jiacai Liu, Chaojie Wang, Yimeng Gan, Shiwen Tu, Chris Yuhao Liu, Liang Zeng, Xiaokun Wang, Boyang Wang, Yongcong Li, Fuxiang Zhang, Jiacheng Xu, Bo An, Yang Liu, and Yahui Zhou. Skywork-o1 open series. https://huggingface.co/Skywork, November 2024. URL https://huggingface.co/Skywork.

[31] 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*, pages 770–778, 2016.

[32] Dan Hendrycks, Collin Burns, Saurav Kadavath, and et al. Measuring mathematical problem solving with the math dataset. In *NeurIPS*, 2021.

[33] Bin Hu, Peter Seiler, and Laurent Lessard. Analysis of biased stochastic gradient descent using sequential semidefinite programs. *Mathematical programming*, 187:383–408, 2021.

[34] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2106.09685.

[35] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, pages 604–613, 1998.

[36] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1988.

[37] Leonard Kaufman and Peter J. Rousseeuw. *Clustering by Means of Medoids*, pages 405–416. North-Holland, Amsterdam, 1987.

[38] Krishnateja Killamsetty, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *AAAI Conference on Artificial Intelligence*, volume 35, pages 8110–8118, 2021.

[39] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (Poster)*, 2015. URL https://arxiv.org/abs/1412.6980.

[40] 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.

[41] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12. ACM, 1994.

[42] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Solving quantitative reasoning problems with language models. *arXiv preprint arXiv:2206.14858*, 2022.

[43] Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. https://huggingface.co/AI-MO/NuminaMath-1.5, 2024.

[44] Mu Li, Gary L Miller, and Richard Peng. Iterative row sampling. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 127–136. IEEE, 2013.

[45] Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers. *arXiv preprint arXiv:2402.19255*, 2024.

[46] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations*, 2019.

[47] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

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

[49] Andrea Martino, Andrea Ghiglietti, Francesca Ieva, and Anna M. Paganoni. A k-means procedure based on a mahalanobis type distance for clustering multivariate functional data. *arXiv preprint arXiv:1708.00386*, 2017.

[50] Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing English math word problem solvers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.92. URL https://aclanthology.org/2020.acl-main.92/.

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

[52] Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.246. URL https://aclanthology.org/2022.acl-long.246/.

[53] Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. Orca-math: Unlocking the potential of slms in grade school math, 2024.

[54] Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168/.

[55] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *arXiv preprint arXiv:2107.07075*, 2021.

[56] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ac56f8fe9eea3e4a365f29f0f1957c55-Abstract.html.

[57] Katherine Pearce, Chao Chen, Yijun Dong, and Per-Gunnar Martinsson. Adaptive parallelizable algorithms for interpolative decompositions via partially pivoted lu. *Numerical Linear Algebra with Applications*, 32(1):e70002, 2025.

[58] Friedrich Pukelsheim. *Optimal design of experiments*. SIAM, 2006.

[59] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[60] Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *2006 47th annual IEEE symposium on foundations of computer science (FOCS'06)*, pages 143–152. IEEE, 2006.

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

[62] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*, 2018.

[63] Sarath Shekkizhar, Mohamed Soliman Ahmed Soliman Farghl, Animesh Nandi, Mohammadhossein Bateni, Sasan Tavakkol, and Neslihan Bulut. Data sampling using locality sensitive hashing for large scale graph learning, February 6 2025. US Patent App. 18/794,578.

[64] Atsushi Shimizu, Xiaoou Cheng, Christopher Musco, and Jonathan Weare. Improved active learning via dependent leverage score sampling. *arXiv preprint arXiv:2310.04966*, 2023.

[65] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[66] 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 (ICLR)*, 2019.

[67] Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. Openmathinstruct-1: A 1.8 million math instruction tuning dataset. *Advances in Neural Information Processing Systems*, 37:34737–34774, 2024.

[68] Hugo Touvron, Thibaut Lavril, Gautier Izacard, and et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[69] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Semi-supervised hashing for large-scale search. *IEEE transactions on pattern analysis and machine intelligence*, 34(12):2393–2406, 2012.

[70] Yining Wang, Adams Wei Yu, and Aarti Singh. On computationally tractable selection of experiments in measurement-constrained regression models. *Journal of Machine Learning Research*, 18(143):1–41, 2017.

[71] Yiping Wang, Yifang Chen, Wendan Yan, Kevin Jamieson, and Simon Shaolei Du. Variance alignment score: A simple but tough-to-beat data selection method for multimodal contrastive learning. *arXiv preprint arXiv:2402.02055*, 2024.

[72] Jason Wei, Xuezhi Wang, Dale Schuurmans, and et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

[73] Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*. PMLR, 2015.

[74] Lechao Xiao. Rethinking conventional wisdom in machine learning: From generalization to scaling. *arXiv preprint arXiv:2409.15156*, 2024.

[75] Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Long is more for alignment: A simple but tough-to-beat baseline for instruction fine-tuning. *arXiv preprint arXiv:2402.04833*, 2024.

[76] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, and et al. Agieval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2299–2314, 2024.

Appendix A. Related Works

In this section, we review prior approaches for data subset selection, broadly categorized into the unsupervised geometry-based and supervised uncertainty-based methods, and then revisit some applications of locality sensitive hashing in recent deep learning literature.

Unsupervised geometry-based methods. Unsupervised geometry-based coreset selection builds on the intuition that nearby points in a suitable feature space are likely redundant. For example, herding [7, 14] selects data greedily to minimize the distance between the centers of the coreset and the original dataset. The k-Center Greedy algorithm [61] approximates the NP-hard minimax facility-location problem by greedily selecting the farthest unseen point. For submodular objectives, Filtered Active Submodular Selection (FASS) [73] combines uncertainty filtering with greedy maximization of a submodular diversity utility. Beyond greedy selection, leverage score sampling and its variations [11, 16, 44, 60, 64] sample *individual data points* by quantifying their importance based on the geometric spread. The optimal experimental design (OED) criteria [10, 28, 58] provide importance weights for *subsets of data* based on their joint geometric coverage of the entire dataset, selection via which can be conducted through optimization [2, 70]. The underlying intuition of OED further inspires selection strategies based on covariance alignment [24, 25, 71]. Adaptive sampling that expands the coreset progressively conditioned on previous selections, either sample-wisely [13, 19, 22] or batch-wisely [23, 27, 57], shows competitive performance on regression tasks with low noise [25].

Supervised uncertainty-based methods. Leveraging label information, uncertainty sampling ranks points by model-prediction ambiguity (e.g. least confidence, margin, or entropy) and selects the most uncertain examples [73]. Such strategies have been widely used in active learning and coreset construction for classification tasks. By gauging the closest points to the decision boundaries via adversarial perturbations, DeepFool-based active learning selects data with the smallest adversarial perturbation distance [26]. The sample-wise loss or error rate during training provides an intuitive measure of the uncertainty. Samples that contribute most to training loss, measured via forgetting events [66], early-epoch gradient, or error norms (GraNd/EL2N) [56], are considered more informative and retained in the coreset. Alternative to loss, model gradients provide more information regarding the uncertainty. For example, CRAIG [51] chooses weighted subsets that closely approximate the full-batch gradient in a submodular fashion, guaranteeing convergence rates comparable to full data. In addition, uncertainty-based data selection can be conducted via bilevel optimization by formulating subset selection as an outer problem over data weights and model training based on the subset as an inner problem. For instance, GLISTER [38] jointly optimizes data subset and validation-set log-likelihood under a mixed discrete-continuous bilevel formulation.

Projection-based locality sensitive hashing. A common way to achieve LSH is leveraging random projections. [35] first introduced random-projection-based LSH for approximate nearest-neighbor search in high-dimensional spaces. Subsequently, [17] generalized LSH to use p-stable distributions—enabling efficient hashing under ℓ_p norms. Semi-supervised extensions such as [69] integrate labeled information to learn hash functions that preserve semantic similarity.

More recent work applies LSH sampling beyond retrieval: [63] employs LSH for scalable graph learning by sampling node neighborhoods, while [12] uses LSH-based strata to accelerate attention approximation in large-language-model decoding. Despite its sublinear query time and minimal preprocessing, vanilla LSH often yields highly imbalanced bucket sizes. We address this by intro-

ducing a lightweight, data-adaptive thresholding mechanism that ensures each bucket contributes uniformly to the final coreset sample.

Appendix B. Algorithm Description

Algorithm 1: Balanced LSH Indexing for Diversity-Aware Online Selection

Data: Embedding matrix $X \in \mathbb{R}^{n \times d}$, embedding dim. d, #hyperplanes m, #buckets b, threshold strategy $\tau \in \{\text{zero}, \text{mean}, \text{median}, \text{svd}\}$
Result: Bucket assignments $(\text{bucket}_1, \ldots, \text{bucket}_n)$
Sample random hyperplanes $\{ \mathbf{w}_j \}_{j=1}^{m} \sim \mathcal{N}(\mathbf{0}, I_d)$ and set $W \in \mathbb{R}^{m \times d}$ with rows $\mathbf{w}_j$;
for $j \leftarrow 1$ **to** m **do**
  `powers[j]` $\leftarrow 2^{j-1}$; `// bit weights`
end
if $\tau = \text{svd}$ **then**
  $\tilde{X} \leftarrow X - \frac{1}{n}\mathbf{1}\mathbf{1}^\top X$; `// center the batch`
  $(U, \Sigma, V) \leftarrow \text{RandSVD}(\tilde{X},\, k = d)$;
  $\tilde{X} \leftarrow \tilde{X}\, V\, \Sigma^{-1}$; `// whiten embeddings`
  $P \leftarrow \tilde{X}_{:,\,1:m}$; `// use first` m `components`
  $\mathbf{t} \leftarrow \mathbf{0} \in \mathbb{R}^m$;
else
  $P \leftarrow X\, W^\top \in \mathbb{R}^{n \times m}$; `// batch projections`
  if $\tau = \text{zero}$ **then**
    $\mathbf{t} \leftarrow \mathbf{0} \in \mathbb{R}^m$;
  else if $\tau = \text{mean}$ **then**
    $\mathbf{t} \leftarrow \frac{1}{n} \sum_{i=1}^{n} P_{i:}$;
  else // $\tau = \text{median}$
    $\mathbf{t} \leftarrow \text{median}(P, \text{axis} = 0)$;
  end
end
$B \leftarrow [\, P_{ij} > t_j \,] \in \{0, 1\}^{n \times m}$; `// bit matrix`
;
for $i \leftarrow 1$ **to** n **do**
  $h_i \leftarrow \sum_{j=1}^{m} B_{ij} \cdot \texttt{powers}[j]$;
  $\text{bucket}_i \leftarrow h_i \bmod b$;
end
return $(\text{bucket}_1, \ldots, \text{bucket}_n)$;

Appendix C. Analysis of 1

Proof [Analysis of 1] For any given $\boldsymbol{\theta} \in \mathbb{R}^d$, we first observe that

$$\begin{aligned}\nabla f(\boldsymbol{\theta}) =& \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}(\boldsymbol{\theta}_*)}[\mathbf{x}(\mathbf{x}^\top\boldsymbol{\theta} - y)] = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}(\boldsymbol{\theta}_*)}[\mathbf{x}(\mathbf{x}^\top\boldsymbol{\theta} - \mathbf{x}^\top\boldsymbol{\theta}_*)] \\ =& \mathbb{E}_{\mathbf{x}\sim\mathcal{D}(\boldsymbol{\theta}_*)}[\mathbf{x}\mathbf{x}^\top](\boldsymbol{\theta} - \boldsymbol{\theta}_*) = \boldsymbol{\Sigma}(\boldsymbol{\theta} - \boldsymbol{\theta}_*).\end{aligned} \tag{3}$$

Meanwhile, for n *i.i.d.* samples $(\mathbf{X}, \mathbf{y})$ drawn from $\mathcal{D}(\boldsymbol{\theta}_*)$, $\mathbf{g}(\boldsymbol{\theta}, (\mathbf{X}, \mathbf{y})) = \frac{1}{n}\mathbf{X}^\top(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})$ is unbiased,

$$\begin{aligned}\mathbb{E}_{(\mathbf{X},\mathbf{y})}\left[\mathbf{g}(\boldsymbol{\theta}, (\mathbf{X}, \mathbf{y}))\right] =& \mathbb{E}_{(\mathbf{X},\mathbf{y})}\left[\frac{1}{n}\mathbf{X}^\top\mathbf{X}(\boldsymbol{\theta} - \boldsymbol{\theta}_*) - \frac{1}{n}\mathbf{X}^\top\mathbf{z}\right] \\ =& \mathbb{E}_{\mathbf{X}}\left[\frac{1}{n}\mathbf{X}^\top\mathbf{X}\right](\boldsymbol{\theta} - \boldsymbol{\theta}_*) = \boldsymbol{\Sigma}(\boldsymbol{\theta} - \boldsymbol{\theta}_*) = \nabla f(\boldsymbol{\theta})\end{aligned} \tag{4}$$

with variance

$$\begin{aligned}\mathcal{V}_n(\theta) =& \mathbb{E}_{(\mathbf{X},\mathbf{y})}\left[\|\mathbf{g}(\boldsymbol{\theta}, (\mathbf{X}, \mathbf{y})) - \nabla f(\boldsymbol{\theta}\|_2^2\right] \\ =& \mathbb{E}_{(\mathbf{X},\mathbf{y})}\left[\left\|\left(\frac{1}{n}\mathbf{X}^\top\mathbf{X} - \boldsymbol{\Sigma}\right)(\boldsymbol{\theta} - \boldsymbol{\theta}_*)\right\|_2^2 + \left\|\frac{1}{n}\mathbf{X}^\top\mathbf{z}\right\|_2^2\right] \\ =& (\boldsymbol{\theta} - \boldsymbol{\theta}_*)^\top\left(\mathbb{E}_{\mathbf{X}}\left[\left(\frac{1}{n}\mathbf{X}^\top\mathbf{X}\right)^2\right] - \boldsymbol{\Sigma}^2\right)(\boldsymbol{\theta} - \boldsymbol{\theta}_*) + \frac{\sigma_y^2}{n}\operatorname{tr}(\boldsymbol{\Sigma}).\end{aligned}$$

Notice that

$$\begin{aligned}\mathbb{E}_{\mathbf{X}}\left[\left(\frac{1}{n}\mathbf{X}^\top\mathbf{X}\right)^2\right] =& \frac{1}{n^2}\left(\sum_{i=1}^n \mathbb{E}_{\mathbf{x}_i}\left[(\mathbf{x}_i\mathbf{x}_i^\top)^2\right] + \sum_{i=1}^n\sum_{j\neq i}^n \mathbb{E}_{\mathbf{x}_i}\left[\mathbf{x}_i\mathbf{x}_i^\top\right]\mathbb{E}_{\mathbf{x}_j}\left[\mathbf{x}_j\mathbf{x}_j^\top\right]\right) \\ =& \frac{1}{n^2}\left(n\mathbf{M} + n(n-1)\boldsymbol{\Sigma}^2\right) = \frac{1}{n}\mathbf{M} + \left(1 - \frac{1}{n}\right)\boldsymbol{\Sigma}^2.\end{aligned} \tag{5}$$

Therefore, the variance of $\mathbf{g}(\boldsymbol{\theta}, (\mathbf{X}, \mathbf{y})) = \frac{1}{n}\mathbf{X}^\top(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})$ can be simplified as

$$\mathcal{V}_n(\theta) = \frac{1}{n}\|\boldsymbol{\theta} - \boldsymbol{\theta}_*\|_{\mathbf{M}-\boldsymbol{\Sigma}^2}^2 + \frac{\sigma_y^2}{n}\operatorname{tr}(\boldsymbol{\Sigma}) \tag{6}$$

Uniform sampling. Subsampling S from $[n]$ uniformly at random effectively leads to a set of k *i.i.d.* samples from $\mathcal{D}(\boldsymbol{\theta}_*)$. Therefore, analogous to the derivation of (4) and (6), we have

$$\mathbb{E}_{(\mathbf{X},\mathbf{y})}\left[\mathbf{g}(\boldsymbol{\theta}, (\mathbf{X}, \mathbf{y}, S^{\text{uni}}))\right] = \nabla f(\boldsymbol{\theta}),$$

and variance

$$\mathbb{E}_{(\mathbf{X},\mathbf{y})}\left[\mathbb{E}_{S^{\text{uni}}}\left[\left\|\mathbf{g}(\boldsymbol{\theta}, (\mathbf{X}, \mathbf{y}, S^{\text{uni}})) - \nabla f(\boldsymbol{\theta}\right\|_2^2\right]\right] = \mathcal{V}_k(\theta) = \frac{1}{k}\|\boldsymbol{\theta} - \boldsymbol{\theta}_*\|_{\mathbf{M}-\boldsymbol{\Sigma}^2}^2 + \frac{\sigma_y^2}{k}\operatorname{tr}(\boldsymbol{\Sigma}).$$

Since $\frac{1}{k}\|\boldsymbol{\theta} - \boldsymbol{\theta}_*\|_{\mathbf{M}-\boldsymbol{\Sigma}^2}^2 \leqslant \frac{1}{k}\left\|\boldsymbol{\Sigma}^\dagger(\mathbf{M} - \boldsymbol{\Sigma}^2)\boldsymbol{\Sigma}^\dagger\right\|_2 \|\boldsymbol{\Sigma}(\boldsymbol{\theta} - \boldsymbol{\theta}_*)\|_2^2$, (1) holds for uniform sampling with $c_v^{\text{uni}}, \sigma^2$ in Example 1

Balanced LSH sampling. We first show the unbiasedness of $\mathbf{g}(\boldsymbol{\theta}, (\mathbf{X}, \mathbf{y}, S^{\text{lsh}}))$ with S^{lsh} from balanced LSH sampling. Since $\Pr(\mathbf{x} \in \mathcal{X}_\iota) = 1/b$ for all $\iota \in [b]$ with balanced buckets, we have

$$\begin{aligned}\mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top \mathbf{X}_{S^{\text{lsh}}}\right]\right] =& \frac{1}{k}\sum_{\iota=1}^{b}\sum_{i=1}^{n/b}\frac{k/b}{n/b}\mathbb{E}_{\mathbf{x}}\left[\mathbf{x}\mathbf{x}^\top | \mathbf{x} \in \mathcal{X}_\iota\right] \\ =& \sum_{\iota=1}^{b}\frac{1}{b}\mathbb{E}_{\mathbf{x}}\left[\mathbf{x}\mathbf{x}^\top | \mathbf{x} \in \mathcal{X}_\iota\right] = \boldsymbol{\Sigma}.\end{aligned} \tag{7}$$

Therefore,

$$\begin{aligned}\mathbb{E}_{(\mathbf{X},\mathbf{y})}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\mathbf{g}(\boldsymbol{\theta}, (\mathbf{X}, \mathbf{y}, S^{\text{lsh}}))\right]\right] =& \mathbb{E}_{(\mathbf{X},\mathbf{y})}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top \mathbf{X}_{S^{\text{lsh}}}(\boldsymbol{\theta} - \boldsymbol{\theta}_*) - \frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top \mathbf{z}_{S^{\text{lsh}}}\right]\right] \\ =& \mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top \mathbf{X}_{S^{\text{lsh}}}\right]\right](\boldsymbol{\theta} - \boldsymbol{\theta}_*) \\ =& \boldsymbol{\Sigma}(\boldsymbol{\theta} - \boldsymbol{\theta}_*) = \nabla f(\boldsymbol{\theta}).\end{aligned}$$

For the variance of $\mathbf{g}(\boldsymbol{\theta}, (\mathbf{X}, \mathbf{y}, S^{\text{lsh}}))$ with S^{lsh}, we observe that

$$\begin{aligned}&\mathbb{E}_{(\mathbf{X},\mathbf{y})}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\left\|\mathbf{g}(\boldsymbol{\theta}, (\mathbf{X}, \mathbf{y}, S^{\text{lsh}})) - \nabla f(\boldsymbol{\theta})\right\|_2^2\right]\right] \\ =& \mathbb{E}_{(\mathbf{X},\mathbf{y})}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\left\|\left(\frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top \mathbf{X}_{S^{\text{lsh}}} - \boldsymbol{\Sigma}\right)(\boldsymbol{\theta} - \boldsymbol{\theta}_*) - \frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top \mathbf{z}_{S^{\text{lsh}}}\right\|_2^2\right]\right] \\ =& \mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\left\|\left(\frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top \mathbf{X}_{S^{\text{lsh}}} - \boldsymbol{\Sigma}\right)(\boldsymbol{\theta} - \boldsymbol{\theta}_*)\right\|_2^2\right]\right] + \mathbb{E}_{(\mathbf{X},\mathbf{z})}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\left\|\frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top \mathbf{z}_{S^{\text{lsh}}}\right\|_2^2\right]\right],\end{aligned}$$

where the second term can be simplified by recalling (7),

$$\mathbb{E}_{(\mathbf{X},\mathbf{z})}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\left\|\frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top \mathbf{z}_{S^{\text{lsh}}}\right\|_2^2\right]\right] = \frac{\sigma_y^2}{k}\operatorname{tr}\left(\mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top \mathbf{X}_{S^{\text{lsh}}}\right]\right]\right) = \frac{\sigma_y^2}{k}\operatorname{tr}(\boldsymbol{\Sigma});$$

and the first term can be decomposed as

$$\begin{aligned}&\mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\left\|\left(\frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top \mathbf{X}_{S^{\text{lsh}}} - \boldsymbol{\Sigma}\right)(\boldsymbol{\theta} - \boldsymbol{\theta}_*)\right\|_2^2\right]\right] \\ =& (\boldsymbol{\theta} - \boldsymbol{\theta}_*)^\top\left(\mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\left(\frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top \mathbf{X}_{S^{\text{lsh}}}\right)^2\right]\right] + \boldsymbol{\Sigma}^2\right)(\boldsymbol{\theta} - \boldsymbol{\theta}_*) \\ &- (\boldsymbol{\theta} - \boldsymbol{\theta}_*)^\top\left(\mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\left(\frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top \mathbf{X}_{S^{\text{lsh}}}\right)\boldsymbol{\Sigma} + \boldsymbol{\Sigma}\left(\frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top \mathbf{X}_{S^{\text{lsh}}}\right)\right]\right]\right)(\boldsymbol{\theta} - \boldsymbol{\theta}_*) \\ =& (\boldsymbol{\theta} - \boldsymbol{\theta}_*)^\top\left(\mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\left(\frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top \mathbf{X}_{S^{\text{lsh}}}\right)^2\right]\right] - \boldsymbol{\Sigma}^2\right)(\boldsymbol{\theta} - \boldsymbol{\theta}_*),\end{aligned}$$

where the last equality comes from (7). Let $\mathbf{M}_\iota = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[(\mathbf{x}\mathbf{x}^\top)^2|\mathbf{x}\in\mathcal{X}_\iota]$ for all $\iota\in[b]$ such that $\mathbf{M} = \frac{1}{b}\sum_{\iota=1}^b \mathbf{M}_\iota$; and recall $\mathbf{\Sigma}_\iota = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[\mathbf{x}\mathbf{x}^\top|\mathbf{x}\in\mathcal{X}_\iota]$ for all $\iota\in[b]$ such that $\mathbf{\Sigma} = \frac{1}{b}\sum_{\iota=1}^b \mathbf{\Sigma}_\iota$. Notice that with balanced LSH,

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\left(\frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top\mathbf{X}_{S^{\text{lsh}}}\right)^2\right]\right] = \mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\left(\sum_{\iota=1}^b\frac{1}{b}\sum_{i=1}^{k/b}\frac{b}{k}\mathbf{x}_{\iota,i}\mathbf{x}_{\iota,i}^\top\right)^2\right]\right]\\
=&\frac{1}{b^2}\mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\sum_{\iota=1}^b\left(\sum_{i=1}^{k/b}\frac{b}{k}\mathbf{x}_{\iota,i}\mathbf{x}_{\iota,i}^\top\right)^2 + \sum_{\iota=1}^b\sum_{\nu\neq\iota}^b\left(\sum_{i=1}^{k/b}\frac{b}{k}\mathbf{x}_{\iota,i}\mathbf{x}_{\iota,i}^\top\right)\left(\sum_{i=1}^{k/b}\frac{b}{k}\mathbf{x}_{\nu,i}\mathbf{x}_{\nu,i}^\top\right)\right]\right]\\
=&\frac{1}{b^2}\left(\sum_{\iota=1}^b\mathbb{E}_{\mathbf{X}}\left[\left(\sum_{i=1}^{k/b}\frac{b}{k}\mathbf{x}_{\iota,i}\mathbf{x}_{\iota,i}^\top\right)^2\right] + \sum_{\iota=1}^b\sum_{\nu\neq\iota}^b\mathbb{E}_{\mathbf{X}}\left[\sum_{i=1}^{k/b}\frac{b}{k}\mathbf{x}_{\iota,i}\mathbf{x}_{\iota,i}^\top\right]\mathbb{E}_{\mathbf{X}}\left[\sum_{i=1}^{k/b}\frac{b}{k}\mathbf{x}_{\nu,i}\mathbf{x}_{\nu,i}^\top\right]\right)\\
=&\frac{1}{b^2}\left(\sum_{\iota=1}^b\left(\frac{b}{k}\mathbf{M}_\iota + \left(1-\frac{b}{k}\right)\mathbf{\Sigma}_\iota^2\right) + \sum_{\iota=1}^b\sum_{\nu\neq\iota}^b\mathbf{\Sigma}_\iota\mathbf{\Sigma}_\nu\right) \quad \text{(By (5))}\\
=&\frac{1}{k}\left(\mathbf{M} - \frac{1}{b}\sum_{\iota=1}^b\mathbf{\Sigma}_\iota^2\right) + \left(\frac{1}{b}\sum_{\iota=1}^b\mathbf{\Sigma}_\iota\right)^2 = \frac{1}{k}\left(\mathbf{M} - \frac{1}{b}\sum_{\iota=1}^b\mathbf{\Sigma}_\iota^2\right) + \mathbf{\Sigma}^2.
\end{aligned}
$$

Thus,

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\left\|\left(\frac{1}{k}\mathbf{X}_{S^{\text{lsh}}}^\top\mathbf{X}_{S^{\text{lsh}}} - \mathbf{\Sigma}\right)(\boldsymbol{\theta}-\boldsymbol{\theta}_*)\right\|_2^2\right]\right]\\
=&\frac{1}{k}(\boldsymbol{\theta}-\boldsymbol{\theta}_*)^\top\left(\mathbf{M} - \frac{1}{b}\sum_{\iota=1}^b\mathbf{\Sigma}_\iota^2\right)(\boldsymbol{\theta}-\boldsymbol{\theta}_*),
\end{aligned}
$$

and therefore the variance of $\mathbf{g}(\boldsymbol{\theta},(\mathbf{X},\mathbf{y},S^{\text{lsh}}))$ can be expressed as

$$
\begin{aligned}
&\mathbb{E}_{(\mathbf{X},\mathbf{y})}\left[\mathbb{E}_{S^{\text{lsh}}}\left[\left\|\mathbf{g}(\boldsymbol{\theta},(\mathbf{X},\mathbf{y},S^{\text{lsh}})) - \nabla f(\boldsymbol{\theta})\right\|_2^2\right]\right]\\
=&\frac{1}{k}(\boldsymbol{\theta}-\boldsymbol{\theta}_*)^\top\left(\mathbf{M} - \frac{1}{b}\sum_{\iota=1}^b\mathbf{\Sigma}_\iota^2\right)(\boldsymbol{\theta}-\boldsymbol{\theta}_*) + \frac{\sigma_y^2}{k}\operatorname{tr}(\mathbf{\Sigma}),
\end{aligned}
$$

which satisfies (1) with $c_v^{\text{lsh}}, \sigma^2$ in Example 1. ■

Appendix D. Experimental details

Hardware configuration. All experiments were conducted on high-performance machines equipped with Intel Xeon CPUs and NVIDIA GPUs, selected to accommodate varying computational needs and optimize job priority scheduling across different tasks. Specifically, we utilized three machine configurations: (1) Intel Xeon Platinum 8268 @ 2.90GHz with 377 GiB RAM and an NVIDIA Tesla V100-PCIE-32GB GPU, (2) Intel Xeon Platinum 8268 @ 2.90GHz with 377 GiB RAM and

an NVIDIA Quadro RTX 8000 (48GB), and (3) Intel Xeon Platinum 8380 @ 2.30GHz with 1.0 TiB RAM and an NVIDIA A100-SXM4-80GB GPU. Although different GPU types were used to balance workload priorities, we ensured that all running comparisons across baselines were performed on the same hardware configuration for a given model and dataset to eliminate hardware-induced variability and maintain consistency and fairness in evaluation.

Training configuration. Our experiments are implemented in the Hugging Face Transformers library using Low-Rank Adaptation (LoRA) to inject trainable rank-decomposition matrices into the Transformer layers, freezing the original weights and only updating the adapters [34]. The LoRA rank was set to $r = 128$ and the scaling coefficient to $\alpha = 1.0$ [34], we applied a dropout rate of 0.1 on the adapter outputs [65]. All trainable parameters were optimized with the AdamW optimizer at a constant learning rate of 2×10^{-5} for 5000 iterations [47].

Datasets overview. The Product Titles Text Classification [6] dataset curated by asaniczka on Kaggle provides a large-scale collection of raw product titles scraped from Amazon marketplaces in the USA, Canada, and the UK. Each entry pairs a product title string with its corresponding category label, enabling straightforward supervised learning experiments. With over 5 million individual title–category pairs spanning 700+ distinct product categories, this dataset stands out as one of the most extensive public resources for real-world e-commerce text classification tasks.

Baselines Overview. To evaluate our Balanced LSH sampling, we compare against a diverse suite of baselines spanning geometry-based, uncertainty-based, loss-based, and instruction-quality methods. **Geometry-based approaches include**: parametric herding, which greedily matches dataset moments to approximate the data distribution [14]; k-center greedy, which iteratively selects the farthest point to solve a minimax facility-location objective [62]; k-means clustering, which chooses cluster centroids as representatives under a k-means++ initialization [5]. **Uncertainty-based methods** select samples that the current model finds most ambiguous. Entropy sampling measures the Shannon entropy of the predicted class distribution [41], while Least Confidence picks examples with the lowest maximum predicted probability [41]. These aim to reduce model uncertainty by targeting confusing inputs. **Loss-based strategies leverage training dynamics**: max-loss and min-loss select samples with highest or lowest training loss norms, respectively, hypothesizing that high-loss examples are informative and low-loss ones easier to learn [55]. GradNorm method refines this by ranking samples according to the norm of per-sample gradients approximating their impact on optimization [55]. **Instruction-quality** indicators quantify the richness of instruction–response pairs. Completion length (CL) selects the longest outputs, exploiting the observation that richer responses correlate with higher quality [75]. Mid-perplexity prunes by keeping examples with intermediate perplexity under a reference model, balancing difficulty and familiarity [48]. Finally, MTLD measures lexical diversity, and reward-score uses the learned preference model Skywork-o1-Open-PRM-Qwen-2.5-1.5B to rank samples by their quality [9, 30].

Appendix E. Potential limitations

Our approach is tailored to fine-tuning scenarios where we can leverage pretrained embeddings (e.g., ResNet or Llama). However, this reliance on already-available representations means that Balanced LSH Sampling is less suitable for setups where embeddings must be learned from scratch or where pretrained models do not exist. Besides, even though the LSH hashing and bucket-balancing steps introduce minimal overhead, every selection step still requires a forward pass to compute those

embeddings. Thus, this can erode efficiency in low-latency or resource-constrained settings (e.g. training LLMs with a small minibatch size or with gradient accumulation).

Appendix F. Additional experimental results

Bucket-count sensitivity Next, we vary the number of hash bucket $b \in \{16, 32, 64, 128, 256\}$ and plot downstream classification accuracy on and product-title text (BERT-base-uncased [20], distilbert-base-uncased [59] and ALBERT-base-v2 [40]) in Figure 5. From which, we observe the consistency of the model performance across different values of b, indicating the robustness of this hyperparameter in our proposed method.

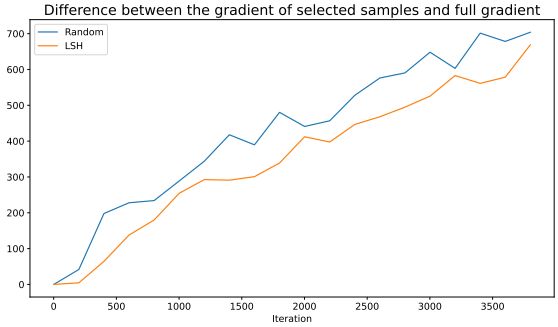

Figure 4: ℓ_2 **norm the difference** between the full-batch gradient and the gradient computed on each selected minibatch.

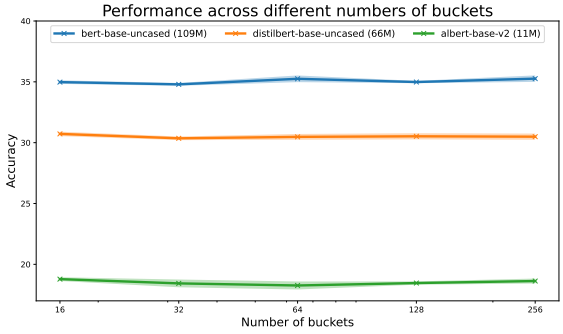

Figure 5: Performance across number of buckets on three different transformer encoder-decoder backbones.

Evolution of the gradient similarity with the full batch training In Figure 4, we plot the evolution of the ℓ_2 norm between the true full-batch gradient and the minibatch gradient produced by each selection strategy over the first 4,000 training iterations. Compared to random sampling (orange), our LSH-based selection method (red) consistently yields a lower approximation error, demonstrating that it more faithfully captures the true gradient direction. This tighter alignment means that, at every step, the direction in which the model parameters are updated more closely matches what would have been chosen using the entire dataset. In practice, this more faithful gradient estimation accelerates convergence and reduces the variance of parameter updates, ultimately leading to more stable and efficient model training.

F.1. Performance of K-means Variants on Different Bucket Sizes

In Figures 6 and 7, we measure the performance of different variants of K-means: Elliptical K-means [49], K-means++ [4], k-medians clustering [36], Partitioning Around Medoids (PAM) [37] , spherical K-means [21] and our proposed method on the same number of buckets. While K-means variants often take longer to run, Elliptical K-means, K-means++, k-medians, and spherical K-means obtain 34.22, 34.35, 34.44 and 34.35 accuracy scores, respectively, on the Massive Product Text classification benchmark. Thus, none of them improves over the original K-means.

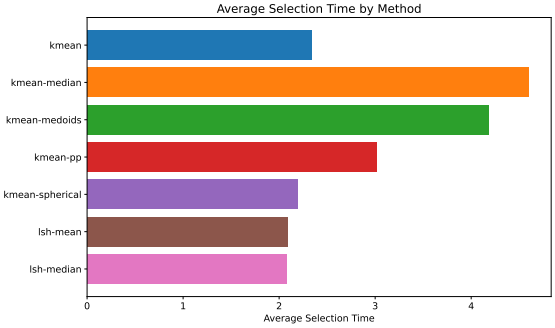

Figure 6: Average selection time of LSH-based sampling methods and K-means's variants.

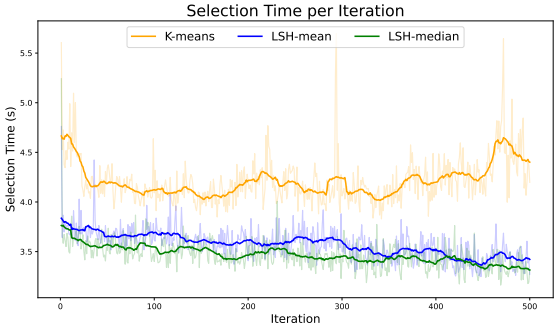

Figure 7: Selection time of LSH vs K-means on ImageNet dataset.