DATASET PRUNING FROM FIRST PRINCIPLES: A LABEL-FREE LINEAR PROGRAMMING APPROACH

Anonymous authors
Paper under double-blind review

ABSTRACT

Dataset pruning reduces a large training set to a representative subset while preserving model performance. Existing geometry-based methods typically assume that nearby points in embedding space share similar properties. Rather than imposing this assumption, we derive geometric selection criteria by reformulating unbiased subset selection as a variance minimization problem. Unbiasedness ensures that unweighted subset averages recover full-dataset averages in expectation, including losses and gradients at fixed model parameters. Specifically, we characterize a family of unbiased subset selection algorithms as a high-dimensional polytope. In this context, minimizing the expected sampling variance is a linear objective. Differences in sampling variance, averaged over rigid motions, admit closed-form pairwise expressions. Because the polytope has high dimension, directly applying standard linear programming is impractical. We instead use these expressions to construct an efficient vertex walk that optimizes an approximation of the variance objective while preserving unbiasedness, yielding a method that requires neither labels nor model training during selection. Across CIFAR-10, MNIST, and CelebA benchmarks, our method matches or exceeds uniform sampling in mean test accuracy at every evaluated budget and outperforms competing geometric methods in several settings, particularly at small selection budgets. Beyond dataset pruning, the same framework reduces stochastic-gradient variance by increasing diversity within mini-batches while keeping the batch size unchanged.

1 INTRODUCTION

Training large deep learning models on abundant data raises a basic question: given a fixed budget, which samples should we train on? The budget may be compute, as in self-supervised pretraining, where scaling laws favor larger models and datasets (Kaplan et al., 2020; Hoffmann et al., 2022) at substantial computational cost (Strubell et al., 2019), and where a well-chosen subset can outperform a larger random one (Sorscher et al., 2022). It may also be annotation, as in remote sensing, where unlabeled archives are abundant but expert labels are expensive (Tuia et al., 2011). Dataset pruning, also known as coreset selection, chooses a small, representative subset of the data. Addressing both cases requires selection that is label-free, so that annotation can be deferred until after selection, and training-free, so that selection does not itself consume the compute it is meant to save. We study methods that satisfy both requirements.

Without labels or training signals, selection must rely on the geometry of the embedding space alone. Existing geometry-based methods typically assume that nearby points in embedding space share similar properties (Moser et al., 2026). This assumption is a heuristic rather than a derived principle: it treats nearby points as redundant, which biases the distribution of the selected subset and can push accuracy below uniform random sampling (Guo et al., 2022). We drop this assumption and instead derive a geometric selection criterion from the *principle of no preferred orientation*.

We seek a subset whose unweighted averages are unbiased, minimum-variance estimates of full-dataset averages of an unknown downstream function, such as a loss or gradient. Since we know nothing about how downstream functions are aligned with the data, we assume no preferred orientation. Under this assumption, we average the mean squared estimation error over all rigid motions of the function, or equivalently of the data (Figure 1, left). For unbiased selections, this averaged error is exactly the expected sampling variance. On toroidal and spherical embedding spaces, this averaging

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

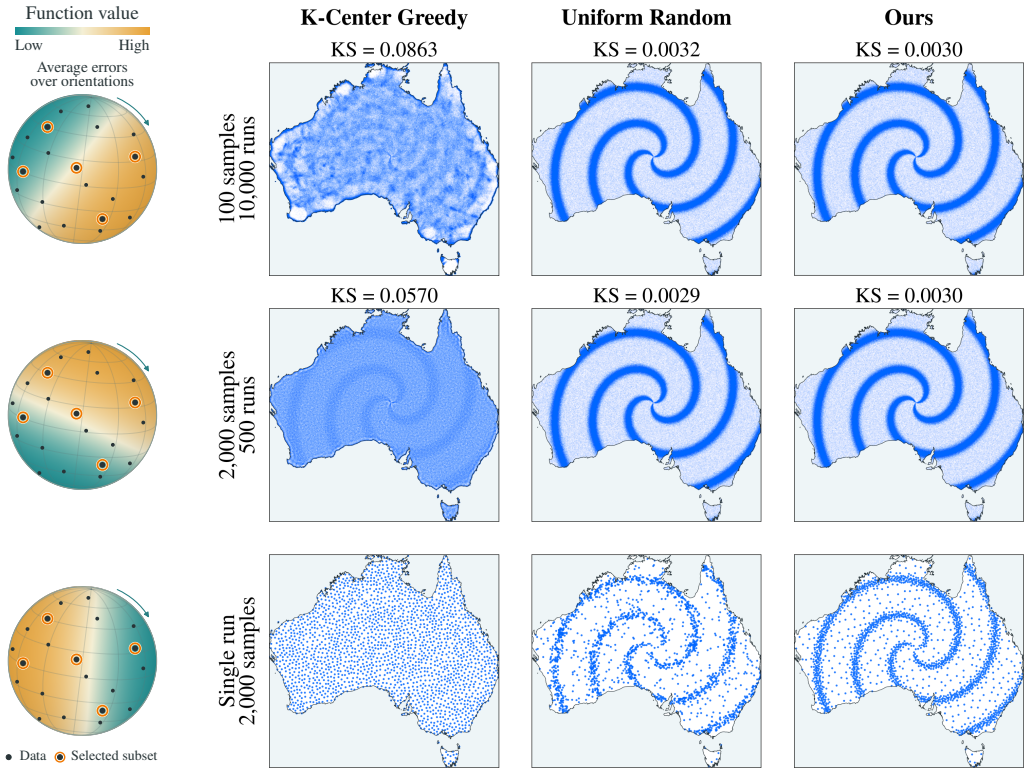


Figure 1: Left: orientation averaging. The data and the selected subset stay fixed while the function rotates, which is equivalent to rotating the data. The three orientations are illustrative; the estimation error is averaged over all of them. Right: spatial distributions of points selected from a vortex-shaped candidate distribution over Australia, shown in an equal-area projection. The top and middle rows pool 100 and 2,000 selected points from 10,000 and 500 runs, respectively, giving one million selected points per panel; the bottom row shows the first 2,000-point run. KS denotes the pooled two-dimensional Kolmogorov–Smirnov discrepancy against a vortex reference sample (lower is better; not a p-value).

yields closed-form expressions previously studied in the context of blue-noise sampling for computer graphics (Pilleboue et al., 2015). We then use these formulas to build a novel unbiased coreset selection method.

These two geometries cover most embedding spaces used in practice. Many encoders are trained with cosine similarity and output ℓ_2-normalized features, so their embeddings lie on the unit sphere; this is the case for contrastive image encoders (Chen et al., 2020; Wang & Isola, 2020), sentence embeddings for language (Reimers & Gurevych, 2019), and multimodal models such as CLIP (Radford et al., 2021). Any other embedding distribution with bounded support, or with rapidly decaying tails such as the approximately Gaussian latent codes of a variational autoencoder (Kingma & Welling, 2014), can be enclosed in a sufficiently large box. Identifying opposite faces of this box places the data on a flat torus while leaving the pairwise distances between data points unchanged.

Our method rests on one reformulation: we treat a selection algorithm as a probability distribution over subsets of the training set (Section 3). Unbiased algorithms that select k elements then form a convex polytope, and the orientation-averaged estimation error is a linear function over it, so unbiased minimum-variance selection is a linear program. The program has one coordinate per subset, far too many for standard solvers, so we instead develop an efficient vertex walk that decreases a variance surrogate while staying on the polytope, and hence unbiased.

We evaluate on CIFAR-10 and MNIST within the widely used DeepCore benchmarking framework, whose comparative study found uniform random selection to remain a strong baseline that competing methods did not consistently outperform (Guo et al., 2022; Moser et al., 2026). In our experiments, our method matches or exceeds uniform selection in mean accuracy at every evaluated budget and

outperforms the compared geometric methods at small budgets on CIFAR-10 and MNIST (Section 6). Beyond dataset pruning, the same framework applies to mini-batch construction for stochastic gradient descent, where optimized batches show lower mean gradient variance than random batches at the same batch size (Section 6.2).

1.1 CONTRIBUTIONS

1. **Formulation.** We derive unbiased selection criteria from the principle of no preferred orientation and obtain closed-form pairwise variance expressions on the sphere and the torus (Section 4).
2. **Algorithm.** We give a label-free, training-free vertex walk that lowers a variance surrogate while preserving unbiasedness (Section 3).
3. **Experiments.** On CIFAR-10, MNIST, and CelebA, our method matches or exceeds uniform sampling at every budget and outperforms the compared geometric methods at small budgets on CIFAR-10 and MNIST (Section 6).
4. **Mini-batches.** The same framework reduces gradient variance in mini-batch construction at fixed batch size (Section 6.2).

2 RELATED WORK

We refer to Moser et al. (2026) for a recent taxonomy that organizes coreset selection into training-free, training-oriented, and label-free approaches. The DeepCore benchmark (Guo et al., 2022) compares many of these methods on CIFAR-10 and ImageNet and finds that uniform random selection remains a strong baseline that many sophisticated methods fail to beat consistently. Uniform sampling is unbiased; we keep this property by construction and search for lower variance within it.

Training-oriented methods score points with signals collected during training: gradient norms early in training (Paul et al., 2021), gradient matching (Mirzasoleiman et al., 2020; Killamsetty et al., 2021a), and bilevel formulations (Killamsetty et al., 2021b). These methods can reach higher accuracy than training-free selection, but they require labels and at least one training pass over the full dataset, which is precisely the cost that training-free and label-free selection is designed to avoid.

Closest to our setting are selection rules that operate directly on an embedding of the data, without labels or training runs. Herding (Welling, 2009; Chen et al., 2010) greedily selects points so that the subset mean tracks the full-data mean; k-center greedy (Sener & Savarese, 2018) minimizes a covering radius over the feature space; and Sorscher et al. (2022) prune by distance to self-supervised cluster prototypes. Related geometric criteria additionally exploit labels or training signals: moderate coresets keep points at intermediate distances from class centers (Xia et al., 2023), and coverage-centric selection stratifies the budget over difficulty scores to preserve coverage (Zheng et al., 2023). All of these methods encode some version of the heuristic that nearby points are redundant, and all of them bias the distribution of the selected subset, which can drive accuracy below uniform sampling, particularly at small budgets and under class imbalance (Guo et al., 2022; Zheng et al., 2023). In contrast, we derive the geometric intuition from first principles and search only among selections that are provably unbiased.

Unbiasedness is a classical requirement in the subset-sampling literature. Sensitivity-based coresets attach importance weights to the selected points so that the weighted subset loss is an unbiased estimator of the full loss (Feldman & Langberg, 2011; Bachem et al., 2017), and determinantal point processes yield Monte Carlo estimators with faster-than-standard convergence rates (Bardenet & Hardy, 2020). Kernel thinning (Dwivedi & Mackey, 2024) recursively splits the data into balanced blocks using a kernel-dependent randomized procedure, then selects a block with small maximum mean discrepancy and refines it greedily. The last two steps can bias the selection. We make the opposite choice: optimize the partition, then return a uniformly chosen block. Unlike the weighted estimators above, our selections are unweighted, and unbiasedness enters as a linear constraint that turns the set of selection algorithms into a convex polytope.

Mini-batch construction is a second setting where unbiased subset averages matter, since stochastic gradient descent relies on uniformly sampled batches being unbiased. Importance sampling draws examples non-uniformly and reweights their gradients to stay unbiased (Needell et al., 2014; Zhao

& Zhang, 2015; Katharopoulos & Fleuret, 2018), at the cost of per-example gradient information. Stratified sampling draws each batch proportionally from clusters of similar examples (Zhao & Zhang, 2014), and repulsive point processes make similar examples less likely to share a batch, whether determinantal (Zhang et al., 2017) or Poisson disk (Zhang et al., 2019), at the cost of a batch gradient that is, in general, unbiased only for a re-weighted objective. Example ordering is a related lever: random reshuffling improves on sampling with replacement (Mishchenko et al., 2020), and GraB uses herding on stale gradients to order examples so that partial sums of gradients track the full gradient (Lu et al., 2022). Our batches are the blocks of an unbiased partition recomputed at each epoch from the embedding geometry alone. Like stratified and repulsive sampling, this needs no gradient information, but the partition minimizes our variance surrogate rather than a fixed clustering or a repulsion heuristic, and a uniformly chosen batch has an unbiased gradient for the original objective at fixed model parameters. Like reshuffling, every example is used once per epoch.

Finally, our closed forms are related to the variance analysis of sampling patterns in computer graphics, where the error of Monte Carlo integration under randomly transformed point sets admits spectral expressions on the torus and on the sphere (Pilleboue et al., 2015; Öztireli, 2016; Singh et al., 2019). The Poisson disk batches of Zhang et al. (2019) come from the same machinery: the variance of the stochastic gradient is expressed through the first- and second-order product densities of a point process over the data. These works analyze continuous point processes; we transport the machinery to the discrete setting of selecting subsets of a fixed dataset, average the estimation error over orientations of the unknown function, and couple the resulting closed forms with the polytope characterization of unbiased selections to obtain a practical algorithm that is unbiased by construction.

3 Problem formulation

3.1 Dataset pruning

Let $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{N}$ be a training dataset, consisting of N independent samples from the joint probability distribution $P(X, Y)$. We define $x_i \in \mathcal{X}$ as the inputs and $y_i \in \mathcal{Y}$ as the ground-truth labels. Dataset pruning aims to find a subset $\mathcal{S} \subset \mathcal{T}$ such that a model $M_{\theta_\mathcal{S}} \in \mathcal{M}$ trained on $\mathcal{S}$ and with parameters $\theta_\mathcal{S}$ achieves a good generalization error (Moser et al., 2026):

$$\mathcal{S}^* := \underset{\mathcal{S}\subset\mathcal{T},\, \frac{|\mathcal{S}|}{|\mathcal{T}|}\leq 1-\alpha}{\arg\min} \; \mathbb{E}_{(x,y)\sim P}\big[\mathcal{L}(x, y; M_{\theta_\mathcal{S}})\big],$$

where $\mathcal{S}^*$ is the optimal subset, $\alpha \in (0, 1)$ is known as the pruning ratio, and $\mathcal{L}$ is the loss function. In practical applications, a subset $\mathcal{S} \subset \mathcal{T}$ is considered good if $|\mathcal{S}| \ll |\mathcal{T}|$ and $\mathbb{E}_{(x,y)\sim P}\big[\mathcal{L}(x, y; M_{\theta_\mathcal{S}})\big] \approx \mathbb{E}_{(x,y)\sim P}\big[\mathcal{L}(x, y; M_{\theta_\mathcal{T}})\big]$, that is, if the model trained on the subset generalizes nearly as well as the model trained on the full dataset.

3.2 Unbiased selection

We want selection algorithms that are unbiased, meaning that subset averages equal full-dataset averages in expectation. Under this constraint, the mean squared error of any quantity we estimate from the subset, whether a loss, a gradient, or an accuracy, reduces to its variance, which is therefore the quantity to minimize. To make this precise, we model a selection algorithm p as a discrete probability measure on the subsets of $\mathcal{T}$. The set of all such measures is $\mathcal{P} := \big\{q \in [0,1]^{2^{|\mathcal{T}|}} : \sum_{S\subseteq\mathcal{T}} q_S = 1\big\}$, a probability simplex with $2^{|\mathcal{T}|}$ coordinates. If $p \in \mathcal{P}$, then p_S is the probability that p assigns to subset S. Unbiasedness is then a condition on p:

Definition 1. A selection algorithm $p \in \mathcal{P}$ is *unbiased* if for every function $f : X \to V$ from an arbitrary set $X \supset \mathcal{T}$ to a vector space V over $\mathbb{R}$,

$$\mathbb{E}_{S\sim p}\left[\frac{1}{|S|}\sum_{x\in S} f(x)\right] = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} f(t).$$

Two restricted classes of unbiased algorithms will be central to what follows.

Definition 2. An unbiased selection algorithm $p^{(k)} \in \mathcal{P}$ *selects* k *elements* if $\forall S \subseteq \mathcal{T},\ p_S^{(k)} > 0 \implies |S| = k$. The set of all such algorithms is denoted by $\mathcal{U}_k$.

Definition 3. An algorithm $p^{(k)} \in \mathcal{U}_k$ is a *uniform k-selection* algorithm, or simply a *k-selection*, if $\forall S \subseteq \mathcal{T},\ p_S^{(k)} \in \{0, k/|\mathcal{T}|\}$.

It follows that a k-selection exists only if k divides $|\mathcal{T}|$, and that every k-selection amounts to breaking $\mathcal{T}$ into $|\mathcal{T}|/k$ blocks of size k and selecting one of them uniformly at random.

4 From orientation averages to practical objectives

This section shows how averaging over all orientations yields a practical objective. Since nothing is known about how the downstream function is aligned with the data, no orientation is preferred, and the variance is averaged over all orientations of the function, or equivalently of the data (Figure 1, left). On the sphere, a change of orientation is a rigid motion; on the flat torus, it is a rotation followed by a translation, with points wrapped back onto the torus. Either way the domain is mapped to itself, so the average never involves values of the function on unrelated regions outside it. Averaging over these transformations turns differences in variance between selection algorithms into sums over pairs of selected points (Theorems A.1 and A.2, proved in the Appendix), and these pairwise expressions motivate surrogate objectives that depend only on the geometry of the selection.

Let $f \in L^2(\mathcal{X} \to \mathbb{R})$ be a function whose full-dataset average is to be estimated by subset averages. For an unbiased selection algorithm $p \in \mathcal{P}$, the variance coincides with the mean squared error and is defined by

$$V_p := \mathbb{E}_{S\sim p}\left[\left(\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} f(t) - \frac{1}{|S|}\sum_{x\in S} f(x)\right)^2\right].$$

After a change of orientation g of the points, the variance becomes

$$V_p(g) := \mathbb{E}_{S\sim p}\left[\left(\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} (f\circ g)(t) - \frac{1}{|S|}\sum_{x\in S} (f\circ g)(x)\right)^2\right].$$

By Theorems A.1 and A.2, for two unbiased selection algorithms $p, q \in \mathcal{P}$, averaging the difference $V_p(g) - V_q(g)$ over the uniform distribution $\mathcal{G}$ of orientations on the torus or the sphere gives

$$\mathbb{E}_{g\sim\mathcal{G}}\big[V_p(g) - V_q(g)\big] = \mathbb{E}_{S\sim p}\left[\frac{1}{|S|^2}\sum_{x,y\in S} v_{\mathcal{X}}\big(h_{\mathcal{X}}(x,y)\big)\right] - \mathbb{E}_{S\sim q}\left[\frac{1}{|S|^2}\sum_{x,y\in S} v_{\mathcal{X}}\big(h_{\mathcal{X}}(x,y)\big)\right],$$

in which

$$v_{\mathbb{T}^d}(r) := \sum_{j\in\mathbb{Z}^d} \left|\hat{f}(j)\right|^2 \frac{\Gamma(d/2)}{(\pi r\,\|j\|_2)^{d/2-1}} J_{d/2-1}(2\pi r\,\|j\|_2),\ h_{\mathbb{T}^d}(x,y) := \|x-y\|_2 \tag{1}$$

$$v_{S^d}(t) := \sum_{\ell\in\mathbb{N}} \|f_\ell\|^2_{L^2(\sigma)}\, P_\ell^{d+1}(t),\ h_{S^d}(x,y) := x\cdot y. \tag{2}$$

Here $\hat{f}(j)$ is the j-th Fourier coefficient of f, J_α is the Bessel function of the first kind, σ is the normalized surface measure on S^d, with $\sigma(S^d) = 1$, f_ℓ is the degree-ℓ spherical harmonic component of f, and P_ℓ^{d+1} is the generalized Legendre polynomial of dimension $d+1$ and degree ℓ.

In both cases, the orientation-averaged variance difference is a linear combination of oscillating functions of the pairwise geometry (Figure 2). The spectral weights $|\hat{f}(j)|^2$ and $\|f_\ell\|^2_{L^2(\sigma)}$ depend on the unknown function f, but the oscillating factors they multiply depend only on the distance between pairs of selected points on the torus, or on their inner product on the sphere. The envelopes of these factors identify regions of low correlation: large separations on the torus, and inner products near zero, corresponding to approximately perpendicular points, on the sphere. Placing pairs in these regions keeps every term of the combination small regardless of f, which motivates a variance surrogate built from a geometry-only pairwise score $\tilde{v}_{\mathcal{X}}$:

$$\widetilde{V}_{\mathcal{X}}(p) := \mathbb{E}_{S\sim p}\left[\frac{1}{|S|^2}\sum_{x,y\in S} \tilde{v}_{\mathcal{X}}(x,y)\right]. \tag{3}$$

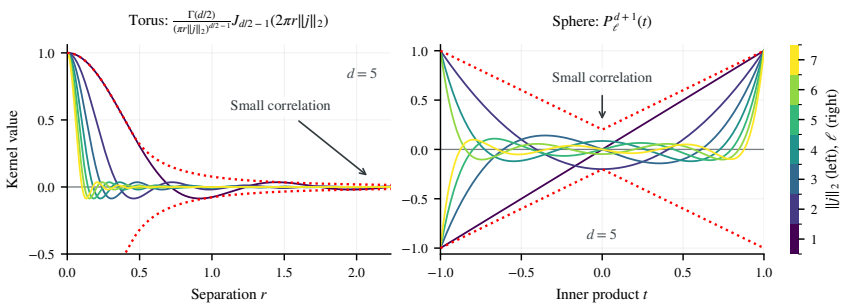


Figure 2: Left: the normalized Bessel factor as a function of the pairwise separation r. Right: the normalized Gegenbauer polynomial $P_\ell^{d+1}(t)$ as a function of the inner product t, both for $d = 5$. Colors identify the frequency norm $\|j\|_2$ (left) or the degree ℓ (right), from 1 to 7; the constant mode is omitted. Dotted red curves show the envelope of each family, and arrows mark the regions of low correlation that motivate the surrogate.

The difference $\widetilde{V}_{\mathcal{X}}(p) - \widetilde{V}_{\mathcal{X}}(q)$ then stands in for the orientation-averaged variance difference $\mathbb{E}_{g\sim\mathcal{G}}[V_p(g) - V_q(g)]$. The experiments use the inverse-distance score $\tilde{v}_{\mathcal{X}}(x, y) = \frac{1}{\|x-y\|+0.01}$, which favors large separations, following the torus motivation, although exact upper bounds could also be used. On the sphere, scores that favor distant points also favor approximately perpendicular pairs when pairwise inner products are nonnegative. We observed this positive-inner-product regime for almost all sampled pairs of IMDb embeddings.

5 Our method

We partition $\mathcal{T}$ into $m = |\mathcal{T}|/k$ blocks $S_1, \ldots, S_m$, each containing k points.[1] Selecting one of these blocks uniformly at random defines a k-selection p (see Definition 3), which corresponds to a vertex of $\mathcal{U}_k$. Swapping two points from different blocks preserves the partition structure and therefore defines another k-selection. Algorithm 1 exploits this property by repeatedly proposing swaps between blocks and accepting them only when they decrease the variance surrogate from Section 4:

$$\widetilde{V}_{\mathcal{X}}(p) = \frac{1}{mk^2} \sum_{i=1}^{m} \sum_{x,y\in S_i} \tilde{v}_{\mathcal{X}}(x, y). \tag{4}$$

Lemma A.3 shows that each k-selection of the training set corresponds to a vertex of the polytope of unbiased selection algorithms with selection budget k. It also shows that the expected sum of pairwise kernel evaluations is a linear objective in the subset-selection probabilities. Thus, our swap-based method moves among extreme points of the feasible polytope, each of which is optimal for some linear objective.

As a faster alternative to iterative swaps, we encode each embedding as an N_b-bit vector whose scaled Hamming distances approximate the original ℓ_1 distances. We maintain per-bit counts for each bucket, allowing us to compute a point's total Hamming distance to its members without storing pairwise distances. Using these distances as a surrogate, we greedily assign points to distant buckets while ensuring that each bucket ends with exactly k points. Selecting one completed bucket uniformly at random then preserves unbiasedness and yields a partition-induced vertex of $\mathcal{U}_k$. Excluding encoding, assignment costs $O(mN_b|\mathcal{T}|)$, which is linear in dataset size for a fixed selection fraction and number of bits. This greedy alternative gives up some fidelity to our derived variance surrogate, and may perform fewer comparisons, in exchange for faster execution (see Section 6).

[1] We consider $k < |\mathcal{T}|$. If k does not divide $|\mathcal{T}|$, we first discard $|\mathcal{T}| \bmod k$ points uniformly at random and apply the procedure to the retained dataset. The resulting selection remains unbiased for the original dataset.

Algorithm 1 Swap-based partition optimization

Require: Dataset $\mathcal{T}$, subset size k with $k \mid |\mathcal{T}|$, variance surrogate $\tilde{v}_{\mathcal{X}}$, number of iterations L
Output: A subset of k points

$m \leftarrow |\mathcal{T}| / k$
Precompute $V_{x,y} \leftarrow \tilde{v}_{\mathcal{X}}(x, y)$ for all $x, y \in \mathcal{T}$
Partition $\mathcal{T}$ into blocks $S_1, \ldots, S_m$, each stored as a list of k points
for $\ell = 1, \ldots, L$ **do**
    Sample i, j uniformly without replacement from $\{1, \ldots, m\}$
    Sample r, s independently and uniformly from $\{1, \ldots, k\}$
    $\Delta \leftarrow -\sum_{q=1}^{k} \left(V_{S_i[r], S_i[q]} + V_{S_j[s], S_j[q]} \right)$
    Swap $S_i[r]$ and $S_j[s]$
    $\Delta \leftarrow \Delta + \sum_{q=1}^{k} \left(V_{S_i[r], S_i[q]} + V_{S_j[s], S_j[q]} \right)$
    if $\Delta \geq 0$ **then**
        Swap $S_i[r]$ and $S_j[s]$ back
    end if
end for
Sample i uniformly from $\{1, \ldots, m\}$
return S_i

6 Experiments

6.1 Dataset pruning

We evaluate our method on CIFAR-10, MNIST, CelebA (Liu et al., 2015), and IMDb (Maas et al., 2011) at selection fractions from 0.005 to 0.3 of the training set. We compare against three training-free, label-free baselines: uniform sampling, herding (Welling, 2009; Chen et al., 2010), and k-center greedy (Sener & Savarese, 2018). We use the DeepCore framework (Guo et al., 2022) without class-balanced selection quotas, since enforcing them requires labels. Herding uses a corrected implementation, since the original DeepCore implementation contains a bug. Tables report mean test accuracy in percent with 95% Wald confidence-interval half-widths; boldface marks the highest mean in each column, not a statistically significant difference.

On CIFAR-10 and MNIST, the embedding-based methods use features from a ResNet18 trained for 10 epochs on the full labeled training set, following DeepCore's protocol for benchmark comparability. A freshly initialized ResNet18 is then trained on the selected subset for 200 epochs. Training uses stochastic gradient descent (SGD) with a batch size of 128 and a learning rate of 0.1. Each setting uses 30 jobs. Each job comprises five selection-and-training runs; we average their best-epoch test accuracies within each job and report means and confidence intervals across jobs.

CelebA and IMDb use pretrained embeddings. On CelebA, we use the latent codes of a variational autoencoder (VAE) (Kingma & Welling, 2014), whose regularized, approximately Gaussian embedding fits the toroidal setting. On IMDb, a film-review dataset, we use Qwen3-Embedding-8B embeddings (Zhang et al., 2025) to evaluate the spherical and label-free setting. Among one million uniformly sampled pairs of distinct IMDb training embeddings, all inner products were positive and over 99% exceeded 0.2 (Figure 3). Thus, increasing pairwise separation favors movement toward orthogonality in the observed regime. Test accuracy is that of a multilayer perceptron (MLP) trained on the selected subset, averaged over 100 independent runs per setting on CelebA and 200 on IMDb. Additional experimental details are given in Appendix B.

On all four datasets we use the surrogate score $\tilde{v}_{\mathcal{X}}(x, y) = 1/(\|x - y\|_2 + 0.01)$ and run Algorithm 1 for $65{,}536k$ proposed swaps, reduced to $2{,}048k$ only at fraction 0.3 on CIFAR-10 and MNIST. The fast variant, reported as Ours (fast), replaces Euclidean distances by Hamming distances between 64k-bit binary representations of the embeddings. Tables 1, 2, and 3 show that our method is particularly competitive at small budgets on CIFAR-10 and MNIST, whereas k-center greedy achieves the highest mean accuracy at larger budgets on these datasets. On CelebA and IMDb, our method has the highest mean accuracy at every budget except the smallest, where herding leads. Unlike the other methods, ours matches or exceeds uniform sampling in mean accuracy at every evaluated budget.

Table 1: Test accuracy (%) on CIFAR-10 and MNIST.

Method / selected fraction	0.005	0.01	0.05	0.1	0.3
CIFAR-10					
Herding	24.65 ± 0.30	30.63 ± 0.39	56.66 ± 0.48	74.76 ± 0.34	90.61 ± 0.07
Ours	31.30 ± 0.25	**38.30** ± 0.28	**63.84** ± 0.32	**76.91** ± 0.26	90.59 ± 0.06
Ours (fast)	**31.30** ± 0.28	38.20 ± 0.28	63.63 ± 0.41	76.48 ± 0.21	90.67 ± 0.05
Uniform	30.63 ± 0.54	38.20 ± 0.51	63.22 ± 0.69	76.35 ± 0.36	90.49 ± 0.11
k-center greedy	28.95 ± 0.27	34.93 ± 0.23	60.19 ± 0.51	75.73 ± 0.35	**91.03** ± 0.07
MNIST					
Herding	64.25 ± 1.13	82.83 ± 0.67	98.22 ± 0.05	99.17 ± 0.02	99.56 ± 0.01
Ours	**88.44** ± 0.40	**92.83** ± 0.25	98.34 ± 0.05	99.13 ± 0.02	99.53 ± 0.01
Ours (fast)	87.87 ± 0.45	92.57 ± 0.21	98.26 ± 0.04	99.12 ± 0.01	99.53 ± 0.01
Uniform	86.39 ± 0.78	92.41 ± 0.35	98.20 ± 0.07	99.09 ± 0.03	99.53 ± 0.01
k-center greedy	68.93 ± 1.28	86.21 ± 0.51	**98.96** ± 0.02	**99.49** ± 0.01	**99.65** ± 0.00

Table 2: CelebA mean test attribute accuracy (%).

Method / selected fraction	0.005	0.01	0.05	0.1	0.3
Herding	**82.89** ± 0.04	85.04 ± 0.02	88.56 ± 0.01	89.52 ± 0.01	90.62 ± 0.00
Ours	82.65 ± 0.07	**85.08** ± 0.03	**88.64** ± 0.01	**89.60** ± 0.01	**90.66** ± 0.01
Ours (fast)	82.67 ± 0.06	85.02 ± 0.03	88.63 ± 0.01	89.58 ± 0.01	90.64 ± 0.01
Uniform	82.57 ± 0.07	84.95 ± 0.03	88.58 ± 0.02	89.57 ± 0.01	90.65 ± 0.01
k-center greedy	80.18 ± 0.03	81.80 ± 0.07	88.27 ± 0.01	89.46 ± 0.01	90.62 ± 0.00

Table 3: IMDb test accuracy (%).

Method / fraction	0.005	0.01	0.05
Herding	**92.73** ± 0.01	93.46 ± 0.01	94.47 ± 0.02
Ours	92.44 ± 0.09	93.53 ± 0.05	94.50 ± 0.02
Ours (fast)	92.52 ± 0.09	**93.54** ± 0.06	**94.53** ± 0.02
Uniform	92.03 ± 0.12	93.33 ± 0.06	94.45 ± 0.02
kCenterGreedy	84.76 ± 0.61	91.03 ± 0.08	94.29 ± 0.02

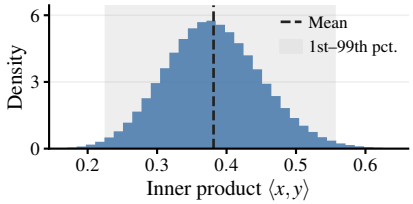


Figure 3: IMDb inner products.

Figure 4 shows the recorded runtimes for 4096-dimensional embeddings. The largest run of our method, with 2^{22} embeddings, takes 153.1 seconds, whereas both baselines complete at most 2^{18} embeddings within the 300-second limit of the sweep.

6.2 Mini-batch variance reduction

We evaluate mini-batch construction on CelebA's 40 attributes using ResNet-18 trained with AdamW (initial learning rate 10^{-3}, weight decay 10^{-4}) and cosine annealing ($T_{\max} = 30$, $\eta_{\min} = 0$). At each epoch, we use the previous epoch's 512-dimensional pooled features, extracted in evaluation mode, as the selection features for batch optimization. Before each reported epoch, we compare random and optimized partitions of the same samples at fixed model parameters and batch size 128. For each scheme, we estimate variance as the mean squared Euclidean distance of the mini-batch loss gradients from their mean. We report means ± 95% Wald confidence-interval half-widths across 256 paired runs. Table 4 shows lower mean gradient variance for optimized batches than for random batches at every evaluated epoch.

7 Limitations and Future Work

The linear objective of Lemma A.3 has one variable per subset, so Algorithm 1 settles for a local search over uniform k-selections. It accepts only improving swaps, uses a fixed iteration budget with no optimality guarantee, and precomputes all pairwise scores at quadratic cost. Faster or more accurate optimizers, of which the fast variant is a first step, are directions for future work.

The inverse-distance surrogate is motivated by the envelopes of the exact kernels in Theorems A.1 and A.2, but decreasing it does not guarantee a reduction in the true variance. When a single spectral

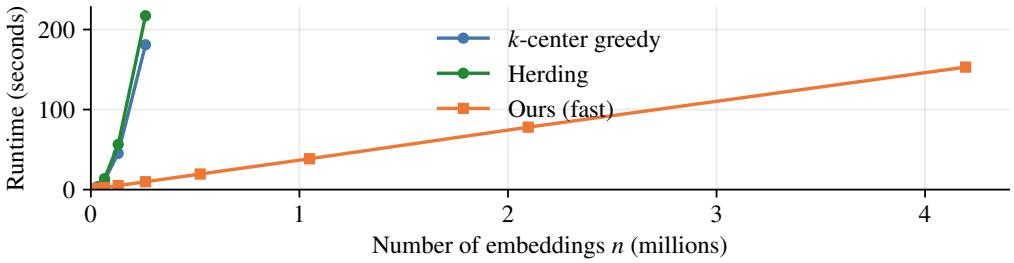


Figure 4: Recorded runtime versus input size (linear axes). We select $n/128$ points from standard-normal, 4096-dimensional float32 embeddings on an NVIDIA GeForce RTX 5090. We use binary representations with 64k bits.

Table 4: Gradient variance of random and optimized mini-batches at fixed batch size.

Epoch	Random variance	Optimized variance	Reduction (%)
2	0.20397 ± 0.00901	0.17367 ± 0.00751	14.85
9	0.23544 ± 0.00827	0.20531 ± 0.00644	12.80
16	0.24042 ± 0.00839	0.21829 ± 0.00651	9.20
23	0.10832 ± 0.00475	0.10365 ± 0.00407	4.32
30	0.00800 ± 0.00018	0.00794 ± 0.00017	0.78

component dominates, the sign of its geometric factor matters: the Bessel factor alternates sign with separation, and even-degree sphere factors favor perpendicular rather than antipodal pairs, so pushing pairs apart can increase the variance. Estimating the dominant components of f from a few labels, and using the exact kernels, could close this gap.

When the data lie on a submanifold of the domain, as images do in an embedding space, most motions carry them off it. The orientation average then evaluates the downstream function, such as the loss of an image classifier, at points that are not images, and the closed forms are no longer informative about the actual variance. To address this, we could use a submersion that maps the data onto a lower-dimensional sphere or torus that they fill, so that no motion carries them off it.

Unbiasedness is a design choice. For particular functions, a biased selection can have lower estimation error than any unbiased one, for instance a smooth f estimated from the evenly spread points of k-center greedy. The price is a selected subset whose distribution differs from that of the data, as in the k-center greedy column of Figure 1 (right). This is undesirable when the subset must represent the population, for instance under class imbalance.

Future work. Beyond the directions above, the Haar average extends to other homogeneous spaces, such as hyperbolic and projective embeddings and products of spheres, where harmonic analysis yields analogous pairwise kernels. Weighted selections enlarge the polytope and connect the framework to sensitivity-based coresets. On the application side, the annotation-budget setting that motivates label-free selection, such as remote sensing, remains to be tested.

8 Conclusion

Label-free, training-free dataset pruning has relied on the heuristic that nearby points are redundant, which biases the selected subset. This paper replaces the heuristic with a principle. If no orientation of the downstream function relative to the data is preferred, the objective is the sampling variance averaged over all orientations, and the search space is the polytope of unbiased selection algorithms. On the torus and the sphere, the averaged variance has closed pairwise forms in which the unknown function enters only through spectral weights, and the envelopes of the geometric factors give a surrogate that a vertex walk optimizes without labels or training. The resulting selections keep the unbiasedness of uniform sampling and match or exceed its mean accuracy at every budget on CIFAR-10, MNIST, CelebA, and IMDb with the largest gains at small budgets. The same partitions lower gradient variance in mini-batch construction at fixed batch size.

AI USE STATEMENT

In this work, we did not use generative AI tools for any task with required disclosure: generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proofs, writing proofs, proposing or refining hypotheses, designing or giving feedback on the research methodology or experiments, implementing methods, translation, cleaning or reformatting datasets, qualitative or thematic data analysis, and interpreting results.

We used generative AI tools for the following tasks with recommended disclosure. In writing, they corrected grammar and spelling and gave feedback on the quality of the text. They helped identify relevant literature. They helped create and format the figures and plots. Most of the code in the project was written by hand; generative AI assisted in optimizing CUDA kernels and in orchestrating (but not design) tests.

We have reviewed all AI-assisted work. Every AI-suggested text edit was reviewed by the authors before being accepted; every AI-suggested reference was read and verified against its original source before citation; and the AI-assisted CUDA kernels and test orchestration were verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All theoretical results are stated with their assumptions and proved in full in the Appendix: Theorems A.1 and A.2 give the closed-form variance differences on the torus and the sphere, and Lemma A.3 establishes the polytope and linear-objective structure the method relies on. The definitions of unbiased selection and of k-selections are in Section 3, and Algorithm 1 specifies the selection procedure completely, with the surrogate score used in all experiments given in Section 4. The experimental setup in Section 6 lists the datasets, selection fractions, baselines, training framework and hyperparameters, numbers of runs and seeds, and the reporting conventions for accuracies and confidence intervals. The remaining dataset-pruning settings and the CelebA autoencoder details are given in Appendices B.1 and B.2, and the mini-batch experiments are described in Section 6.2. The source code for the selection algorithm, the corrected herding baseline, the benchmarks, and the scripts that generate every figure will be made available to reviewers through an anonymous repository during the discussion period, and released publicly afterwards.

REFERENCES

Olivier Bachem, Mario Lucic, and Andreas Krause. Practical coreset constructions for machine learning, 2017.

Rémi Bardenet and Adrien Hardy. Monte carlo with determinantal point processes. *The Annals of Applied Probability*, 30(1), 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Yutian Chen, Max Welling, and Alexander J. Smola. Super-samples from kernel herding. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.

Raaz Dwivedi and Lester Mackey. Kernel thinning. *Journal of Machine Learning Research*, 25(152): 1–77, 2024.

Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, 2011.

H Groemer. Geometric applications of Fourier series and spherical harmonics. *ENCYCLOPEDIA OF MATHEMATICS AND ITS APPLICATIONS*.

Chengcheng Guo, Bo Zhao, and Yanbing Bai. DeepCore: A Comprehensive Library for Coreset Selection in Deep Learning, 2022.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2525–2534, 2018.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. GRAD-MATCH: Gradient matching based data subset selection for efficient deep model training. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021a.

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

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2015. URL https://openaccess.thecvf.com/content_iccv_2015/html/Liu_Deep_Learning_Face_ICCV_2015_paper.html.

Yucheng Lu, Wentao Guo, and Christopher De Sa. GraB: Finding provably better data permutations than random reshuffling. In *Advances in Neural Information Processing Systems 35*, 2022.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, 2011. URL https://aclanthology.org/P11-1015/.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik. Random reshuffling: Simple analysis with vast improvements. In *Advances in Neural Information Processing Systems 33*, 2020.

Brian B. Moser, Arundhati S. Shanbhag, Stanislav Frolov, Federico Raue, Joachim Folz, and Andreas Dengel. A Coreset Selection of Coreset Selection Literature: Introduction and Recent Advances, 2026.

Deanna Needell, Nathan Srebro, and Rachel Ward. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. In *Advances in Neural Information Processing Systems 27*, 2014.

A. Cengiz Öztireli. Integration with stochastic point processes. *ACM Transactions on Graphics*, 35 (5), 2016.

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 34 (NeurIPS)*, 2021.

Adrien Pilleboue, Gurprit Singh, David Coeurjolly, Michael Kazhdan, and Victor Ostromoukhov. Variance analysis for Monte Carlo integration. *ACM Transactions on Graphics*, 34(4), 2015.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

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

Gurprit Singh, Cengiz Öztireli, Abdalla G. M. Ahmed, David Coeurjolly, Kartic Subr, Oliver Deussen, Victor Ostromoukhov, Ravi Ramamoorthi, and Wojciech Jarosz. Analysis of sample correlations for Monte Carlo rendering. *Computer Graphics Forum*, 38(2):473–491, 2019.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: Beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, 2022.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Devis Tuia, Michele Volpi, Loris Copa, Mikhail Kanevski, and Jordi Muñoz-Marí. A survey of active learning algorithms for supervised remote sensing image classification. *IEEE Journal of Selected Topics in Signal Processing*, 5(3):606–617, 2011.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2009.

Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *International Conference on Learning Representations (ICLR)*, 2023.

Cheng Zhang, Hedvig Kjellström, and Stephan Mandt. Determinantal point processes for mini-batch diversification. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

Cheng Zhang, Cengiz Öztireli, Stephan Mandt, and Giampiero Salvi. Active mini-batch sampling using repulsive point processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5741–5748, 2019.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025. URL `https://arxiv.org/abs/2506.05176`.

Peilin Zhao and Tong Zhang. Accelerating minibatch stochastic gradient descent using stratified sampling, 2014.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1–9, 2015.

Haizhong Zheng, Rui Liu, Fan Lai, and Atul Prakash. Coverage-centric coreset selection for high pruning rates. In *International Conference on Learning Representations (ICLR)*, 2023.

A Appendix

A.1 Theorems

Theorem A.1 (Torus). *Let $\mathcal{T} \subset \mathbb{R}^d/\mathbb{Z}^d = \mathbb{T}^d$ be a finite set of independent samples from μ, and let $f \in L^2(\mathbb{T}^d \to \mathbb{R})$. Identify torus points with their canonical representatives in $[0,1)^d$, and let $w : \mathbb{R}^d \to [0,1)^d$ denote coordinatewise wrapping modulo one. For a translation $t \in \mathbb{T}^d$ and a rotation $R \in SO(d)$, define*

$$g_{t,R}(x) := w(Rx + t).$$

Thus, rotation of the fixed representatives is followed by translation and periodic wrapping. The distances below are Euclidean distances between these representatives. Write $g = g_{t,R}$. For an unbiased selection algorithm $p \in \mathcal{P}$ and a nonempty subset $S \subseteq \mathcal{T}$, define

$$V(g,S) := \left[\int_{\mathbb{T}^d} (f \circ g)(x) \ \mathrm{d}\mu(x) - \frac{1}{|S|} \sum_{x \in S} (f \circ g)(x) \right]^2, \text{ and}$$

$$V_p(g) := \mathbb{E}_{S \sim p}[V(g,S)] = \sum_{\emptyset \neq S \subseteq \mathcal{T}} p_S V(g,S).$$

it follows that $V(I,S)$ is the realized sampled variance of $S \subseteq \mathcal{T}$, and thus $V_p(I)$ is the mean of the realized sampled variances of the unbiased selection algorithm p (see Definition 1).

If $p, q \in \mathcal{P}$ are unbiased selection algorithms and $\mathcal{G}$ is the distribution of $g_{t,R}$ induced by independently sampling t and R from the normalized Haar measures on $\mathbb{T}^d$ and $SO(d)$, respectively, then

$$\mathbb{E}_{g \sim \mathcal{G}} \left[V_p(g) - V_q(g) \right] = \mathbb{E}_{S \sim p} \left[\frac{1}{|S|^2} \sum_{x,y \in S} v_{\mathbb{T}^d}\big(\|x - y\|_2 \big) \right] - \mathbb{E}_{S \sim q} \left[\frac{1}{|S|^2} \sum_{x,y \in S} v_{\mathbb{T}^d}\big(\|x - y\|_2 \big) \right]$$

in which

$$v_{\mathbb{T}^d}(r) = \sum_{j \in \mathbb{Z}^d} \left| \hat{f}(j) \right|^2 \frac{\Gamma(d/2)}{(\pi \|j\|_2 r)^{d/2-1}} J_{d/2-1}(2\pi \|j\|_2 r)$$

and J_α is the Bessel function of the first kind. The normalized Bessel factor is defined by its continuous extension, with value one when $\|j\|_2 r = 0$.

Proof. Expanding the square yields

$$\mathbb{E}_{g \sim \mathcal{G}}\big[V(g,S)\big] = \mathbb{E}_{g \sim \mathcal{G}} \left[\left[\int_{\mathbb{T}^d} (f \circ g)(x) \ \mathrm{d}\mu(x) - \frac{1}{|S|} \sum_{a \in S} (f \circ g)(a) \right]^2 \right] \tag{5}$$

$$= \mathbb{E}_{g \sim \mathcal{G}} \left[c_S(g) \right] + \frac{1}{|S|^2} \mathbb{E}_{g \sim \mathcal{G}} \left[\left[\sum_{a \in S} (f \circ g)(a) \right]^2 \right]. \tag{6}$$

Note that for almost every g,

$$\mathbb{E}_{S \sim p}[c_S(g)] = \left(\int_{\mathbb{T}^d} (f \circ g)(x) \ \mathrm{d}\mu(x) \right)^2 - 2 \int_{\mathbb{T}^d} (f \circ g)(x) \ \mathrm{d}\mu(x) \sum_{\emptyset \neq S \subseteq \mathcal{T}} \frac{p_S}{|S|} \sum_{a \in S} (f \circ g)(a)$$

It follows from our definition of unbiased selection algorithm that

$$\mathbb{E}_{S\sim p}[c_S(g)] = \left(\int_{\mathbb{T}^d} (f\circ g)(x)\ \mathrm{d}\mu(x)\right)^2 - 2\int_{\mathbb{T}^d} (f\circ g)(x)\ \mathrm{d}\mu(x)\frac{1}{|\mathcal{T}|}\sum_{z\in\mathcal{T}}(f\circ g)(z),$$

which is independent on the algorithm p and thus

$$\mathbb{E}_{S\sim p}[c_S(g)] = \mathbb{E}_{S\sim q}[c_S(g)].$$

One can conclude that both terms cancel out and that we can concentrate on the second element of Equation 6. Let us define $\mathcal{R}$ as the uniform distribution over rotation matrices in $SO(d)$. Write $\widetilde{f} = f\circ w$ for the periodic extension of f. Then, it follows that

$$\frac{1}{|S|^2}\mathbb{E}_{g\sim\mathcal{G}}\left[\left[\sum_{a\in S}(f\circ g)(a)\right]^2\right] = \frac{1}{|S|^2}\sum_{a,b\in S}\mathbb{E}_{g\sim\mathcal{G}}\left[(f\circ g)(a)(f\circ g)(b)\right],$$

in which

$$\mathbb{E}_{g\sim\mathcal{G}}\left[(f\circ g)(a)(f\circ g)(b)\right] = \mathbb{E}_{R\sim\mathcal{R}}\left[\int_{[0,1)^d}\widetilde{f}(Ra+t)\widetilde{f}(Rb+t)\ \mathrm{d}t\right].$$

Using the substitution $y = Ra + t$, it follows that $\mathrm{d}y = \mathrm{d}t$. The translated domain $Ra + [0,1)^d$ can be replaced by $[0,1)^d$ because $\widetilde{f}(y)\widetilde{f}(R(b-a)+y)$ is $\mathbb{Z}^d$-periodic, and we can proceed by applying Parseval's theorem

$$\begin{aligned}
\mathbb{E}_{R\sim\mathcal{R}}\left[\int_{[0,1)^d}\widetilde{f}(Ra+t)\widetilde{f}(Rb+t)\ \mathrm{d}t\right] &= \mathbb{E}_{R\sim\mathcal{R}}\left[\int_{[0,1)^d}\widetilde{f}(y)\widetilde{f}(R(b-a)+y)\ \mathrm{d}y\right] \\
&= \mathbb{E}_{R\sim\mathcal{R}}\left[\sum_{j\in\mathbb{Z}^d}\mathcal{F}[f](j)\overline{\mathcal{F}[\widetilde{f}(\cdot+R(b-a))](j)}\right] \\
&= \mathbb{E}_{R\sim\mathcal{R}}\left[\sum_{j\in\mathbb{Z}^d}\left|\hat{f}(j)\right|^2 e^{-2\pi ij\cdot R(b-a)}\right] \\
&= \sum_{j\in\mathbb{Z}^d}\left|\hat{f}(j)\right|^2\mathbb{E}_{R\sim\mathcal{R}}\left[e^{-2\pi ij\cdot R(b-a)}\right].
\end{aligned}$$

Here $\hat{f}(j) = \int_{[0,1)^d} f(y)e^{-2\pi ij\cdot y}\ \mathrm{d}y$; the interchange of expectation and summation follows from $\sum_j\left|\hat{f}(j)\right|^2 < \infty$.

For $d = 1$, the formula simplifies to

$$\sum_{j\in\mathbb{Z}}\left|\hat{f}(j)\right|^2\mathbb{E}_{R\sim\mathcal{R}}\left[e^{-2\pi ijR(b-a)}\right] = \sum_{j\in\mathbb{Z}}\left|\hat{f}(j)\right|^2\cos\left(2\pi j\left\|b-a\right\|_2\right).$$

Indeed, $SO(1) = \{1\}$ and $\left|\hat{f}(-j)\right|^2 = \left|\hat{f}(j)\right|^2$ since f is real.

Otherwise, $\left|S^{d-2}\right|$ is defined and it follows that (with $\alpha = d/2 - 1$)

$$
\begin{aligned}
\mathbb{E}_{R\sim\mathcal{R}}\left[e^{-2\pi i j\cdot R(b-a)}\right] =& \frac{1}{|S^{d-1}|}\int_{S^{d-1}} e^{-2\pi i j\cdot x\|b-a\|_2}\ \mathrm{d}x \\
=& \frac{1}{|S^{d-1}|}\int_0^\pi e^{-2\pi i\|b-a\|_2\|j\|_2\cos\theta}\left|S^{d-2}\right|\sin(\theta)^{d-2}\ \mathrm{d}\theta \\
=& \frac{\frac{2\pi^{(d-1)/2}}{\Gamma((d-1)/2)}}{\frac{2\pi^{d/2}}{\Gamma(d/2)}}\int_0^\pi e^{-2\pi i\|b-a\|_2\|j\|_2\cos\theta}\sin(\theta)^{d-2}\ \mathrm{d}\theta \\
=& \frac{\Gamma(d/2)}{\Gamma((d-1)/2)\sqrt{\pi}}\int_0^\pi e^{-2\pi i\|b-a\|_2\|j\|_2\cos\theta}\sin(\theta)^{d-2}\ \mathrm{d}\theta \\
=& \frac{\Gamma(\alpha+1)}{\Gamma(\alpha+1/2)\sqrt{\pi}}\int_0^\pi e^{-2\pi i\|b-a\|_2\|j\|_2\cos\theta}\sin(\theta)^{2\alpha}\ \mathrm{d}\theta \\
=& \left.\frac{\Gamma(d/2)}{\left(\frac{t}{2}\right)^{d/2-1}}J_{d/2-1}(t)\right|_{t=2\pi\|j\|_2\|b-a\|_2}.
\end{aligned}
$$

□

Theorem A.2 (Sphere)**.** *Let $\mathcal{T}\subset S^d$ be a finite set of independent samples from μ, $g: S^d\to S^d$ a rigid motion and $f\in L^2(S^d\to\mathbb{R})$. Let σ be the normalized surface measure on S^d, with $\sigma(S^d)=1$, and let f_ℓ be the degree-ℓ spherical harmonic component of f. Its squared norm is*

$$\|f_\ell\|^2_{L^2(\sigma)} := \int_{S^d}|f_\ell(x)|^2\ \mathrm{d}\sigma(x).$$

For an unbiased selection algorithm $p\in\mathcal{P}$ and a nonempty subset $S\subseteq\mathcal{T}$, define

$$V(g,S) := \left[\int_{S^d}(f\circ g)(x)\ \mathrm{d}\mu(x) - \frac{1}{|S|}\sum_{x\in S}(f\circ g)(x)\right]^2, \text{ and}$$

$$V_p(g) := \mathbb{E}_{S\sim p}[V(g,S)] = \sum_{\emptyset\neq S\subseteq\mathcal{T}} p_S V(g,S).$$

it follows that $V(I,S)$ is the realized sampled variance of $S\subseteq\mathcal{T}$, and thus $V_p(I)$ is the mean of the realized sampled variances of the unbiased selection algorithm p (see Definition 1).

If $p,q\in\mathcal{P}$ are unbiased selection algorithms and $\mathcal{G}$ is the uniform distribution over the isometries in S^d, then

$$
\begin{aligned}
\mathbb{E}_{g\sim\mathcal{G}}\left[V_p(g)-V_q(g)\right] = \mathbb{E}_{S\sim p}&\left[\frac{1}{|S|^2}\sum_{x,y\in S} v_{S^d}(x\cdot y)\right] \\
&- \mathbb{E}_{S\sim q}\left[\frac{1}{|S|^2}\sum_{x,y\in S} v_{S^d}(x\cdot y)\right]
\end{aligned}
$$

in which

$$v_{S^d}(t) := \sum_{\ell\in\mathbb{N}}\|f_\ell\|^2_{L^2(\sigma)}\,P_\ell^{d+1}(t),$$

and P_ℓ^{d+1} is the generalized Legendre polynomial of dimension $d+1$ and degree ℓ, that can alternatively be defined for $d\geq 2$ as:

$$P_\ell^{d+1}(t) = \frac{C_\ell^{(\lambda)}(t)}{C_\ell^{(\lambda)}(1)}$$

in which $C_\ell^{(\lambda)}$ are the Gegenbauer polynomials, and $\lambda=(d-1)/2$. For $d=1$, we use the limiting definition

$$P_\ell^2(t) = \cos(\ell\arccos t), \qquad t\in[-1,1].$$

Proof. Expanding the square yields

$$\mathbb{E}_{g\sim\mathcal{G}}\big[V(g,S)\big] = \mathbb{E}_{g\sim\mathcal{G}}\left[\left[\int_{S^d}(f\circ g)(x)\ \mathrm{d}\mu(x) - \frac{1}{|S|}\sum_{a\in S}(f\circ g)(a)\right]^2\right] \tag{7}$$

$$= \mathbb{E}_{g\sim\mathcal{G}}\left[c_S(g)\right] + \frac{1}{|S|^2}\mathbb{E}_{g\sim\mathcal{G}}\left[\left[\sum_{a\in S}(f\circ g)(a)\right]^2\right]. \tag{8}$$

As shown in Theorem A.1, unbiasedness gives

$$\mathbb{E}_{S\sim p}[c_S(g)] = \mathbb{E}_{S\sim q}[c_S(g)]$$

for almost every g. Thus these terms cancel out in $V_p(g) - V_q(g)$, and we can focus on the second element of Equation 8.

$$\frac{1}{|S|^2}\mathbb{E}_{g\sim\mathcal{G}}\left[\left[\sum_{a\in S}(f\circ g)(a)\right]^2\right] = \frac{1}{|S|^2}\sum_{a,b\in S}\mathbb{E}_{Q\sim\mathcal{G}}\left[f(Qa)f(Qb)\right]. \tag{9}$$

Let R be any rotation such that $Ra = a$. Since all isometries on the sphere are orthogonal transformations, we will denote g with the letter Q:

$$\begin{aligned}\alpha :=& \mathbb{E}_{Q\sim\mathcal{G}}\left[f(Qa)f(Qb)\right]\\ =& \mathbb{E}_{Q\sim\mathcal{G}}\left[f(Qa)f(QRb)\right]\\ =& \mathbb{E}_{Q\sim\mathcal{G}}\left[\frac{1}{|C_{Qa}|}\int_{x\in C_{Qa}} f(Qa)f(x)\ \mathrm{d}x\right],\end{aligned}$$

in which $C_y = \{|y|\, x : x \in S^d \wedge x\cdot y/|y| = a\cdot b\}$. At $a\cdot b = 1$, each normalized slice integral below is interpreted as evaluation at y; at $a\cdot b = -1$, it is interpreted as evaluation at $-y$. Note that now the expected value only depends on Qa rather than Q, and it follows that

$$\begin{aligned}\alpha =& \frac{1}{|S^d|}\int_{y\in S^d}\frac{1}{|C_y|}\int_{x\in C_y} f(y)f(x)\ \mathrm{d}x\ \mathrm{d}y\\ =& \frac{1}{|S^d|}\int_{y\in S^d} f(y)\frac{1}{|C_y|}\int_{x\in C_y} f(x)\ \mathrm{d}x\ \mathrm{d}y\\ =& \frac{1}{|S^d|}\int_{y\in S^d} f(y)\overline{h(y)}\ \mathrm{d}y\\ =& \langle f, h\rangle_{L^2(\sigma)}.\end{aligned}$$

Let A_ℓ denote the space of degree-ℓ spherical harmonics on S^d, identified with their homogeneous harmonic extensions to $\mathbb{R}^{d+1}$.

By expressing f as the direct sum

$$f = \widehat{\bigoplus}_{\ell\in\mathbb{N}} f_\ell,\ f_\ell \in A_\ell,$$

it follows that

$$h(y) = \frac{1}{|C_y|} \int_{x \in C_y} f(x) \, \mathrm{d}x \tag{10}$$

$$= \frac{1}{|C_y|} \int_{x \in C_y} \sum_{\ell \in \mathbb{N}} f_\ell(x) \, \mathrm{d}x \tag{11}$$

$$= \sum_{\ell \in \mathbb{N}} \frac{1}{|C_y|} \int_{x \in C_y} f_\ell(x) \, \mathrm{d}x. \tag{12}$$

We want to show that

$$h_\ell(y) := \frac{1}{|C_y|} \int_{x \in C_y} f_\ell(x) \, \mathrm{d}x$$

is in A_ℓ.

1. $h_\ell(\lambda y) = \lambda^\ell h_\ell(y)$

   For $\lambda \neq 0$,

$$\begin{aligned} h_\ell(\lambda y) &= \frac{1}{|C_{\lambda y}|} \int_{x \in C_{\lambda y}} f_\ell(x) \, \mathrm{d}x \\ &= \frac{1}{|\lambda|^{d-1} \, |C_y|} \int_{x \in C_y} f_\ell(\lambda x) \, |\lambda|^{d-1} \, \mathrm{d}x \\ &= \lambda^\ell h_\ell(y). \end{aligned}$$

2. $\Delta h_\ell = 0$

   This is a consequence of the Funk–Hecke theorem Groemer, since $\forall y \in S^d$, $h_\ell(y) = \beta_\ell(a \cdot b) \; f_\ell(y)$ for some function β_ℓ: $[-1, 1] \to \mathbb{R}$ that will be characterized latter. The previous item can be used to extend the equality to $\mathbb{R}^{d+1}$, thus proving that $h_\ell(y) \in A_\ell$.

$$\therefore h = \widehat{\bigoplus}_{\ell \in \mathbb{N}} h_\ell, h_\ell \in A_\ell$$

Then, we can use the orthogonality of the spherical harmonics and the Funk–Hecke theorem to conclude that

$$\begin{aligned} \alpha &= \sum_{\ell \in \mathbb{N}} \langle f_\ell, h_\ell \rangle_{L^2(\sigma)} \\ &= \sum_{\ell \in \mathbb{N}} \|f_\ell\|^2_{L^2(\sigma)} \; \beta_\ell(a \cdot b) \\ &= \sum_{\ell \in \mathbb{N}} \|f_\ell\|^2_{L^2(\sigma)} \, P_\ell^{d+1}(a \cdot b), \end{aligned}$$

in which P_ℓ^{d+1} is the generalized Legendre polynomial of dimension $d + 1$ and degree ℓ. □

A selection algorithm $p \in \mathcal{P}$ is unbiased in the sense of Definition 1 if and only if

$$p_\emptyset = 0, \qquad \sum_{\substack{S \subseteq \mathcal{T} \\ x \in S}} \frac{p_S}{|S|} = \frac{1}{|\mathcal{T}|} \quad \text{for every } x \in \mathcal{T}.$$

Indeed, the empty subset must have zero probability for the subset average to be defined. Expanding the expectation over nonempty subsets gives

$$\mathbb{E}_{S\sim p}\left[\frac{1}{|S|}\sum_{x\in S} f(x)\right] = \sum_{x\in\mathcal{T}} f(x) \sum_{\substack{S\subseteq\mathcal{T}\\ x\in S}} \frac{p_S}{|S|}.$$

The stated conditions therefore imply unbiasedness; conversely, applying unbiasedness to the indicator function of each individual point gives these conditions. For subsets of fixed size $k \geq 1$, they reduce to equal inclusion probabilities:

$$\Pr_{S\sim p^{(k)}}(x \in S) = \sum_{\substack{S\subseteq\mathcal{T}\\ x\in S}} p_S^{(k)} = \frac{k}{|\mathcal{T}|} \quad \text{for every } x \in \mathcal{T}.$$

These linear constraints yield the polytope characterization below.

Lemma A.3. *Let $\binom{\mathcal{T}}{k} = \left\{S_1, \cdots, S_{\binom{|\mathcal{T}|}{k}}\right\}$ be an enumeration of the subsets of $\mathcal{T}$ with size k. Note that any unbiased selection algorithm of size k can be uniquely identified as*

$$p^{(k)} \in [0,1]^{\binom{|\mathcal{T}|}{k}}, \text{ in which}$$
$$p_i^{(k)} = p_{S_i}^{(k)}.$$

Similarly, any loss function $\mathcal{L} : \binom{\mathcal{T}}{k} \to \mathbb{R}$ can be uniquely identified as

$$\mathcal{L} \in \mathbb{R}^{\binom{|\mathcal{T}|}{k}}, \text{ in which}$$
$$\mathcal{L}_i = \mathcal{L}(S_i).$$

It follows from the definition of expectation that

$$\mathbb{E}_{S\sim p^{(k)}}[\mathcal{L}(S)] = \left\langle p^{(k)}, \mathcal{L}\right\rangle_{\mathbb{R}^{\binom{|\mathcal{T}|}{k}}} = p^{(k)} \cdot \mathcal{L}.$$

Thus,

$$p^* = \arg\min_{p^{(k)}\in\mathcal{U}_k} \mathbb{E}_{S\sim p^{(k)}}[\mathcal{L}(S)], \tag{13}$$

is a linear programming problem. If k divides $|\mathcal{T}|$ and $p^{(k)}$ is a uniform k-selection, then $p^{(k)}$ is a vertex of the convex polytope $\mathcal{U}_k$.

Proof. Enumerate $\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\}$. Let

$$B := \left[c(S_1) \quad c(S_2) \quad \cdots \quad c\left(S_{\binom{|\mathcal{T}|}{k}}\right)\right] \in \mathbb{R}^{|\mathcal{T}|\times\binom{|\mathcal{T}|}{k}},$$

in which $c(S_j)$ are the binary column vectors

$$c(S_j)_i := \begin{cases} 1 \text{ if } t_i \in S_j \\ 0 \text{ if } t_i \notin S_j \end{cases}.$$

Using this definition, we can rewrite Equation 13 as the following optimization problem:

Given $\mathcal{L} \in \mathbb{R}^{\binom{|\mathcal{T}|}{k}}$, minimize

$$p^{(k)} \cdot \mathcal{L},$$

in which $p^{(k)} \in \mathbb{R}^{\binom{|\mathcal{T}|}{k}}$ is subject to being a probability measure on $\binom{\mathcal{T}}{k}$:

$$\forall 1 \leq i \leq \binom{|\mathcal{T}|}{k}, 0 \leq p_i^{(k)} = p_{S_i}^{(k)} \tag{14}$$

$$\sum_{i=1}^{\binom{|\mathcal{T}|}{k}} p_i^{(k)} = 1 \tag{15}$$

and being unbiased, i.e.

$$\forall t \in \mathcal{T}, \sum_{\substack{1 \leq i \leq \binom{|\mathcal{T}|}{k} \\ t \in S_i}} \frac{p_i^{(k)}}{k} = \frac{1}{|\mathcal{T}|}. \tag{16}$$

From Equation 16 and non-negativity, it follows immediately that

$$\forall i,\, p_i^{(k)} \leq \frac{k}{|\mathcal{T}|}.$$

Thus, we can rewrite Equations 14-16 as follows:

$$0 \leq p_i^{(k)} \leq \frac{k}{|\mathcal{T}|} \tag{17}$$

$$\mathbf{1}^\mathsf{T} p^{(k)} = 1 \tag{18}$$

$$Bp^{(k)} = \frac{k}{|\mathcal{T}|}\mathbf{1} \tag{19}$$

which indeed is a convex polytope $\mathcal{U}_k \subset \mathbb{R}^{\binom{|\mathcal{T}|}{k}}$. Since in real-world applications the number of dimensions would surpass the number of atoms in the universe, traditional linear programming techniques are not applicable. However, we can find vertices of $\mathcal{U}_k$ by solving the surrogate problem below.

For the vertex claim, assume $k \mid |\mathcal{T}|$. For $v \in \mathbb{R}^{\binom{|\mathcal{T}|}{k}}$, maximize

$$\|v\|_2^2$$

Subject to $v \in \mathcal{V}$, a convex polytope defined by the following two equations

$$0 \leq v_i \leq \frac{k}{|\mathcal{T}|} \tag{20}$$

$$\mathbf{1}^\mathsf{T} v = 1 \tag{21}$$

The problem has at least one solution, since $\mathcal{V} \neq \emptyset$ is a compact set and $\|v\|_2^2$ is continuous. We will show that if v^* is a solution to the problem, then $v_i^* \in \left\{0, \frac{k}{|\mathcal{T}|}\right\}$. Suppose by hypothesis that u maximizes $\|u\|_2^2$, but

$$\exists i_0, 0 < u_{i_0} < \frac{k}{|\mathcal{T}|}.$$

Since $k \mid |\mathcal{T}|$ and $\mathbf{1}^\mathsf{T} u = 1$, it follows that

$$\exists i_1 \neq i_0, 0 < u_{i_1} < \frac{k}{|\mathcal{T}|}.$$

This implies that u is not a solution, since

$$u'_i = \begin{cases} u_i & \text{if } i \neq i_0 \wedge i \neq i_1 \\ \min\left\{\frac{k}{|\mathcal{T}|}, u_{i_0} + u_{i_1}\right\} & \text{if } i = i_0 \\ \max\left\{u_{i_0} + u_{i_1} - \frac{k}{|\mathcal{T}|}, 0\right\} & \text{if } i = i_1 \end{cases}$$

satisfies the Equations 20 and 21, and $\|u'\|_2^2 > \|u\|_2^2$. Thus, there are $\binom{|\mathcal{T}|}{\frac{|\mathcal{T}|}{k}}$ solutions $v^* \in \mathcal{V}$ that maximize the distance to the origin, which includes all uniform k-selections (see Definition 3). Note that among these solutions, only the uniform k-selections belong to $\mathcal{U}_k$.

Since squared Euclidean norm is strictly convex, these maximizers are vertices of $\mathcal{U}_k$. □

B Additional experimental details

B.1 Dataset-pruning settings

CIFAR-10 and MNIST. We use DeepCore (Guo et al., 2022) with ResNet18 and the standard training and test splits. All results in Table 1 use selection without class-balanced quotas. For the embedding-based methods, a randomly initialized ResNet18 is trained on the full labeled training set for 10 epochs with cross-entropy loss. Selection uses its penultimate-layer features, extracted in evaluation mode after the final epoch. Thus, the selection rules require neither labels nor additional training given the embeddings, but producing these embeddings involves supervised training. Evaluation uses a separate, freshly initialized ResNet18 trained on the selected subset for 200 epochs.

Preprocessing and optimization. CIFAR-10 images are normalized channelwise with means $(0.4914, 0.4822, 0.4465)$ and standard deviations $(0.2470, 0.2435, 0.2616)$; MNIST uses mean 0.1307 and standard deviation 0.3081. Downstream CIFAR-10 training additionally uses random crops with four-pixel reflection padding and random horizontal flips. Embedding training, feature extraction, and test evaluation use no augmentation; MNIST uses no augmentation in either training stage. Both training stages use SGD with batch size 128, momentum 0.9, Nesterov acceleration, weight decay 5×10^{-4}, and cross-entropy loss. Embedding training uses a constant learning rate of 0.1; downstream training uses an initial learning rate of 0.1 with cosine annealing to 10^{-4} over 200 epochs, updated after each mini-batch.

B.2 CelebA experimental details

VAE embeddings. We use an ADM-style residual and self-attention VAE pretrained on aligned CelebA images. RGB images are converted to floating-point values in $[0, 1]$ and resized bilinearly to 128×128. The model has approximately 130 million parameters and a 512-dimensional Gaussian latent space. Its residual blocks use GroupNorm and SiLU, with four-head self-attention at spatial resolutions 32×32 and below; the decoder ends with a sigmoid. The VAE is trained for 140 epochs with batch size 32, learning rate 10^{-4}, and zero weight decay. The objective combines a Gaussian reconstruction likelihood with standard deviation 0.1, KL regularization with a free-bits threshold of 0.5, and a VGG perceptual loss weighted by 0.5. Selection and downstream classification use the deterministic 512-dimensional posterior means, extracted with the VAE in evaluation mode, rather than sampled latent vectors.

Downstream classification. The selection pool contains 40.5k embedding vectors; selection fractions are relative to this pool, and the selected vectors are used for downstream training. Evaluation uses a separate 20.2k-vector test split. The classifier is an MLP with widths $512 \to 256 \to 128 \to 20$. Each hidden linear layer is followed by LayerNorm, ReLU, and dropout with probability 0.2; the output layer produces 20 logits. We train for 20 epochs with Adam, a constant learning rate of 10^{-3}, batch size 256, and binary cross-entropy with logits.

The targets are Male, Young, Attractive, Smiling, Heavy Makeup, Wearing Lipstick, Blond Hair, Black Hair, Brown Hair, Gray Hair, Bald, Bangs, Eyeglasses, Goatee, Mustache, No Beard, Wearing

Hat, Wearing Earrings, Pale Skin, and Chubby. Labels are mapped from $\{-1, 1\}$ to $\{0, 1\}$, and predictions use a sigmoid threshold of 0.5. Mean test attribute accuracy averages the per-attribute test accuracies over these 20 targets. Each run contributes its highest mean test attribute accuracy over the 20 training epochs, and the table averages these values over 100 runs per setting.

B.3 IMDB EXPERIMENTAL DETAILS

Data and embeddings. We use the standard labeled IMDb splits: 25,000 training reviews and 25,000 test reviews, each containing equal numbers of positive and negative examples. The unlabeled split is not used. Reviews are embedded with the pretrained Qwen3-Embedding-8B model (Zhang et al., 2025), using the original review text without an added task-specific instruction and a truncation limit of 8,192 tokens. The resulting 4,096-dimensional, unit-ℓ_2-norm embeddings are stored in float32 and used unchanged for selection and downstream classification, with no additional normalization, dimensionality reduction, or augmentation. Selection operates only on the training split, without class-balanced quotas or additional embedding training.

Classifier and optimization. For each selected subset, we train a freshly initialized MLP with widths $4096 \to 128 \to 128 \to 2$, ReLU activations after both hidden layers, and no dropout or normalization layers. The output consists of two sentiment logits, trained with cross-entropy loss. Training runs for 200 epochs using SGD with batch size 128, momentum 0.9, Nesterov acceleration, and weight decay 5×10^{-4}. The learning rate starts at 0.1 and follows cosine annealing to 10^{-4} over 200 epochs, updated after each mini-batch. Training examples are shuffled each epoch.

Epoch selection and reporting. We evaluate accuracy on the full test split after every epoch. Each run contributes its highest test accuracy over the 200 training epochs; no separate validation split is used for epoch selection. The main IMDb comparison uses 200 selection-and-training runs per method–fraction setting. Table 3 reports their mean accuracy and 95% Wald confidence-interval half-width, computed as $1.96s/\sqrt{200}$, where s is the sample standard deviation of the per-run best-epoch accuracies.