ECHO: Effective Coreset-Driven Learning via Hierarchical Optimizations

Alec F. Diallo
School of Informatics
University of Edinburgh, UK
alec.frenn@ed.ac.uk

Weihe Li
School of Informatics
University of Edinburgh, UK
weihe.li@ed.ac.uk

Paul Patras
School of Informatics
University of Edinburgh, UK
paul.patras@ed.ac.uk

***Abstract*—Despite driving record performance, the increasing reliance of deep learning on ever-larger datasets has led to prohibitively high storage and management costs that threaten continued progress. While coreset selection offers a promising solution to this challenge, existing methods often rely on expensive iterative optimization procedures or fail to select samples that allow strong generalization across tasks. In this work, we introduce ECHO, a coreset construction and augmentation strategy that leverages the relational properties inherent to a dataset to find its most representative samples. Unlike prior methods, our approach constructs a structured graph that encodes intrinsic dataset patterns, based on which influential samples are identified and augmented to maximize generalization performance. Extensive experiments across five benchmark datasets and against eighteen different coreset selection baselines show that ECHO achieves up to 60% accuracy gains under extreme compression, while being orders of magnitude faster than state-of-the-art alternatives. These results establish a new benchmark for data-efficient learning, particularly under tight coreset budgets, and showcase the benefits of structured coreset selection for effective generalization.**

***Index Terms*—Geometric Coreset Selection, Data Efficiency, Training Optimization, Machine Learning.**

I. Introduction

In recent years, Deep Learning (DL) has revolutionized the field of artificial intelligence, enabling substantial breakthroughs in various domains such as image recognition, natural language processing, and autonomous driving [1]–[3]. This success predominantly arises from the ability of neural networks to learn complex patterns from vast quantities of data [4], driving advancements in hardware and algorithm designs to cope with the associated increase in computational demands [5]. Further, this continued and growing reliance of DL on large volumes of data presents major challenges in terms of storage requirements, scalability, and training time, posing barriers to adoption in environments with constrained processing capabilities, such as in startups and non-profit organizations [6]. Consequently, beyond the well-documented challenges of costly data labeling and annotation [7], there is a pressing need for innovative strategies that minimize the required amount of training data without compromising model performance.

Coreset selection has emerged as a promising and increasingly necessary technique for addressing these challenges [8], [9]. A coreset, defined as a representative subset of a dataset, aims to approximate the full dataset with much fewer samples, effectively reducing the computational burden while retaining the accuracy and generalizability of models trained on the full dataset [10]. Despite this potential, existing coreset selection methods face several limitations: they often fail to capture diverse, rare, or boundary samples that are critical for generalization; depend on labels, training dynamics, gradients, or external (pre-trained) models that may be unavailable or sub-optimal; require prior knowledge of specific downstream tasks; or assume static data distributions, leaving them ill-suited to streaming or evolving data [11]. These limitations significantly restrict the practical utility of these methods, especially when downstream tasks are unknown or when sample importance estimates misalign with the downstream objectives.

In this paper, we introduce *ECHO* (Effective Coreset-driven Hierarchical Optimization), a novel coreset selection method that harnesses the geometric and relational properties inherent to datasets. Unlike traditional methods that rely on statistical or random sampling strategies, our approach examines the structural composition of the dataset, identifying and preserving its most significant patterns and variabilities. Specifically, through the use of Proximally-Connected (PC) Graphs [12] (which abstractly represent datasets only in terms of relationships between their closest or most similar data points), we first partition the data into locality-aware regions and, within each region, select the most influential samples based on their spatial relationships.

To maximize sample diversity while keeping computational costs in check, our approach augments each coreset with a support graph, whereby every node is attributed its top-k support vectors, i.e., the samples farthest from that node's center. These support vectors act as anchors that span the extremal geometry of each node's neighborhood, which are often underrepresented or ignored in traditional coreset selection methods. By prioritizing boundary refinement, our approach ensures maximal coreset diversity and informativeness, thereby preserving the robust generalization of the models.

As illustrated in Figure 1, our proposed approach requires no task-specific knowledge, rendering it universally applicable and highly adaptable to diverse learning problems. By focusing on intrinsic data structure and explicitly modeling extreme variations, ECHO produces coresets that faithfully capture the essence of the original dataset, enabling effective downstream training even in small-data regimes. To the best of our knowledge, ECHO is the first approach to explicitly leverage

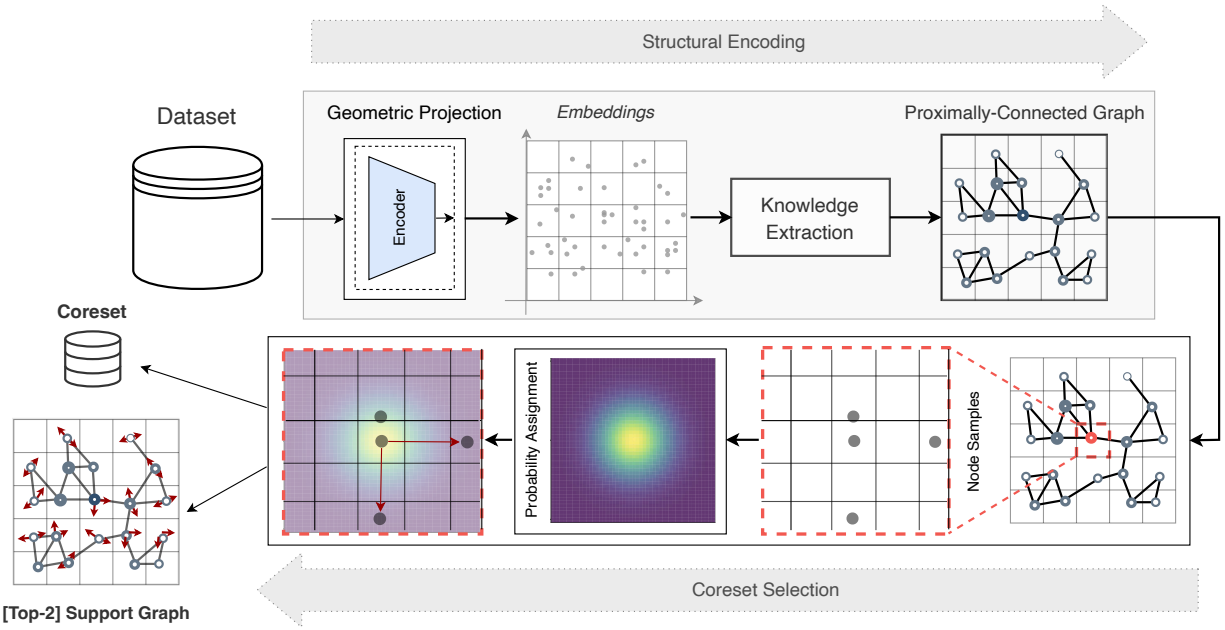

Fig. 1: **Structural Coreset Selection and Augmentation with ECHO.** From the initial dataset, low-dimensional embeddings are generated via Geometric Projection using an encoder network, enabling the construction of Proximally-Connected Graphs that encode structural relationships. Within each graph node (a grid cell containing nearby embeddings), a probability assignment heat-map is computed to prioritize central representatives, while simultaneously the top-k support vectors (samples farthest from the node center) are identified as extremal anchors. Iterating through each node of the graph, starting from those with highest degree centrality to capture densely connected regions first, we preserve both core and boundary-defining information, maximizing diversity and faithfully capturing the dataset's decision boundaries under strict computational and memory constraints.

hierarchical structural augmentations to systematically refine decision boundaries, allowing simultaneous optimization of sample diversity and computational efficiency. By strategically capturing boundary-defining samples that significantly influence model decision-making, our approach sets a new standard for coreset representativeness and generalization capability, overcoming the fundamental limitations that have long constrained (and still constrain) existing methods.

In summary, our main contributions are as follows:

- **Structural selection:** We propose a universally applicable coreset selection method that preserves the structural integrity of datasets by leveraging their geometric and relational characteristics, eliminating the need for task-specific information or external (pre-trained) models.
- **Structural augmentation:** We introduce support graphs, attributing to each node its top-k extremal anchors, explicitly refining decision boundaries. This strategy maximizes sample diversity while tightly controlling computational and memory requirements, ensuring minimal performance degradation relative to training with the full dataset.
- **Empirical validation:** We comprehensively evaluate ECHO across five benchmark datasets against eighteen state-of-the-art coreset selection baselines, demonstrating up to 60% accuracy gains under extreme coreset budgets, while requiring orders of magnitude less computation time.

II. Related Work

The challenge of efficient data selection has inspired a broad spectrum of coreset selection and data pruning strategies, each balancing representativeness, informativeness, and scalability.

Early **representation and diversity-based methods**, such as Herding [13], k-Center Greedy [14], and submodular maximization frameworks including Fass [15], Similar [16], and Prism [17], focus on maximizing coverage or minimizing redundancy in feature space. Recent advances incorporate Contextual Diversity (CD) [18] and geometric-proxy objectives [10], [11] to better capture the manifold's structure, while gradient-matching variants such as CRAIG [11] and GradMatch [19] explicitly align the optimization path of the coreset with that of the full dataset. Though effective in dense regions, these methods typically still neglect extremal or rare samples critical to decision boundaries, and can be computationally burdensome when extended to high-resolution or large-scale datasets.

Training-dynamics methods, drawing on active learning and model-aware selection, evaluate the influence of individual samples on learning outcomes. Classic uncertainty sampling (Least Confidence, Entropy, Margin) [20] is augmented by adversarial and contrastive approaches (DeepFool [21], CAL [22]) to emphasize decision boundaries, while dynamics-based methods, including Forgetting Events [23], GraNd / EL2N [24], and TDDS [25], identify influential or persistently hard examples through loss, gradient norms, or temporal scoring.

Bilevel optimization (Glister [26], Retrieve [27]) further refines selection by optimizing validation performance, but at the expense of significant computational overhead. Proxy-based methods [28] simplify this by using less complex models or heuristics to guide the selection process, speeding up decisions at the risk of potentially missing deeper insights from more sophisticated models. The need for gradient evaluations and repeated model updates makes these methods impractical in settings with limited computational resources or unlabeled data.

More recently, **label-free and foundation model-driven techniques** have emerged to address settings with scarce annotation or to scale data selection for large models. Embedding-based strategies such as ZCore and STAFF exploit pre-trained or foundation models to extract representations for zero-shot subset selection or coreset scoring [29], [30]. Along the same line, ELFS applies deep clustering on high-quality, pre-trained vision embeddings, and then uses those to generate pseudo-labels that drive its coreset scoring, thereby achieving strong performance without labels [31]. These advances, though promising, can inherit the computational cost and biases of the foundation models they use or rely on external (pre-trained) models whose embeddings may not perfectly align with the downstream objectives.

Our approach, **ECHO**, departs from all three categories. It directly leverages geometric and relational structure in data, requiring no labels, pre-trained models, or gradient-based scoring. By explicitly augmenting its coreset with support vectors, our approach captures both the central trends as well as the boundary-defining samples, enabling state-of-the-art generalization with minimal assumptions.

III. Preliminaries

A. Notations

Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^D$ denote a dataset of N samples in a D-dimensional space. Given an encoder $E_\theta : \mathbb{R}^D \to \mathbb{R}^d$ parametrized by θ, with $d \ll D$, mapping each $\mathbf{x}_i$ to a low-dimensional embedding $\mathbf{z}_i = E_\theta(\mathbf{x}_i) \in \mathbb{R}^d$, we denote by $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\} \subset \mathbb{R}^d$ the set of all embeddings.

All distances in $\mathbb{R}^d$ are measured using the Euclidean (ℓ_2) norm, and are denoted by $\|\cdot\|$.

To partition the embedding space, we impose a uniform, axis-aligned grid of side length $w = 1/\lceil 2\ \ln(1+N)\rceil$, where $\lceil x\rceil$ denotes the smallest integer greater than or equal to x. This choice ensures that grid resolution grows sub-linearly with dataset size, balancing spatial detail and computational cost. Each non-empty grid cell is denoted C_j for $j = 1, \ldots, M$, where M is the total number of cells that contain at least one embedding. Without loss of generality, each C_j is a hypercube

$$C_j = \big\{\, \mathbf{z} \in \mathbb{R}^d \mid k_\ell\, w \le z^{(\ell)} < (k_\ell + 1)\, w,\ \forall\, \ell = 1, \ldots, d \big\}$$

for integer indices $\mathbf{k} = (k_1, \ldots, k_d) \in \mathbb{Z}^d$.

We write $\mathcal{Z}_j = \{\mathbf{z}_i \in \mathcal{Z} \mid \mathbf{z}_i \in C_j\}$ to denote the set of embeddings that fall into C_j, and $m_j = |\mathcal{Z}_j| \ge 1$ to denote its cardinality. Within each cell C_j, the centroid is defined as

$$\mathbf{c}_j = \frac{1}{m_j} \sum_{\mathbf{z} \in \mathcal{Z}_j} \mathbf{z},$$

capturing the center of mass of embeddings within the cell.

To define the support vectors directly in the original space, we also denote by $\mathcal{X}_j = \big\{\, \mathbf{x}_i \in \mathcal{X} \;\big|\; E_\theta(\mathbf{x}_i) \in C_j \big\}$, the set of all original samples in cell j, and denote their average by

$$\boldsymbol{\mu}_j \;=\; \frac{1}{m_j} \sum_{\mathbf{x} \in \mathcal{X}_j} \mathbf{x}.$$

For each $\mathbf{x}_{j,k} \in \mathcal{X}_j$, we define its displacement vector from the cell mean:

$$\mathbf{d}_{j,k} \;=\; \mathbf{x}_{j,k} \;-\; \boldsymbol{\mu}_j, \qquad \|\mathbf{d}_{j,k}\| \;=\; \big\|\mathbf{x}_{j,k} - \boldsymbol{\mu}_j\big\|.$$

These displacements can then be used to identify the boundary samples for each cell. To this end, we sort the displacements vectors by descending norm via a permutation $\pi_j \colon \{1, \ldots, m_j\} \to \{1, \ldots, m_j\}$ that satisfies

$$\|\mathbf{d}_{j,\pi_j(1)}\| \;\ge\; \|\mathbf{d}_{j,\pi_j(2)}\| \;\ge\; \cdots \;\ge\; \|\mathbf{d}_{j,\pi_j(m_j)}\|.$$

Then, for cell C_j, the set of support vectors of size s is defined as:

$$\begin{aligned} S_j^{(s)} \;&=\; \big\{\, \mathbf{d}_{j,\pi_j(1)},\, \mathbf{d}_{j,\pi_j(2)},\, \ldots,\, \mathbf{d}_{j,\pi_j(s)} \big\} \\ &=\; \big\{\, \mathbf{x}_{j,\pi_j(1)} - \boldsymbol{\mu}_j,\; \ldots,\; \mathbf{x}_{j,\pi_j(s)} - \boldsymbol{\mu}_j \big\}. \end{aligned}$$

B. Proximally-Connected Graphs

An undirected graph $\mathcal{G} = (V, E)$ is said to be proximally connected if its vertices $V = \{v_1, \ldots, v_M\}$ correspond to non-empty, non-overlapping grid cells $C_1, \ldots, C_M$ (each with centroid $\mathbf{c}_j \in \mathbb{R}^d$), and its edge set E satisfies the following two requirements, as originally defined [12].

First, whenever two grid cells C_i and C_j are adjacent, i.e., share a $(d-1)$-dimensional face, there is an edge $\{v_i, v_j\} \in E$ whose weight is given by $w_{i,j} \;=\; \|\mathbf{c}_i - \mathbf{c}_j\|$.

Second, if an edge $\{v_i, v_j\} \in E$ does not arise from such a grid adjacency, then its removal must increase the number of connected components of $\mathcal{G}$ by exactly one; equivalently, denoting by $\kappa(\mathcal{G})$ the number of connected components of a graph $\mathcal{G}$, we have

$$\kappa\big(\mathcal{G}(V, E \setminus \{v_i, v_j\})\big) \;=\; \kappa(\mathcal{G}(V, E)) + 1.$$

Moreover, if $\mathcal{G}(V, E \setminus \{v_i, v_j\})$ splits into precisely two components X (containing v_i) and Y (containing v_j), then the edge weight $w_{i,j}$ must equal the minimal distance between any vertex in X and any vertex in Y:

$$\|\mathbf{c}_i - \mathbf{c}_j\| \;=\; \min_{\substack{v_u \in X \\ v_w \in Y}} \|v_u - v_w\|.$$

In other words, no edge in E can be replaced by a strictly shorter inter-component connection without violating this proximal-connectivity constraint.

Because every non-adjacent edge is chosen precisely when it is the shortest link between the two sets it would otherwise separate, $\mathcal{G}$ is connected and contains no redundant bridging edges. Finally, the degree of each vertex v_j is defined as

$$\deg(v_j) \;=\; \big|\{\, v_k : \{v_j, v_k\} \in E\}\big|.$$

In this setting, cells corresponding to vertices of larger degree lie in more densely interconnected regions.

C. Coreset Selection Problem

Given a labeled dataset with n samples and their corresponding labels $\mathcal{X} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$, our objective is to find a small subset $\mathcal{S} \subseteq \mathcal{X}$ of size m that retains the essential characteristics and utility of $\mathcal{X}$ for any downstream task. Specifically, the goal is to ensure that models trained on $\mathcal{S}$ perform on par with models trained on the full dataset $\mathcal{X}$. Therefore, we aim to construct $\mathcal{S}$ such that:

$$\forall f \in \mathcal{F},\ U(f(\mathcal{X})) \approx U(f(\mathcal{S})),$$

where $\mathcal{F}$ represents any downstream models or tasks, and $U(f(\mathcal{X}))$ denotes a utility measure (e.g., accuracy, loss) for f.

The coreset $\mathcal{S}$ is defined as the solution to the following optimization objective:

$$\begin{aligned} \text{Minimize} \quad & |U(f(\mathcal{X})) - U(f(\mathcal{S}))|, \quad \forall f \in \mathcal{F}, \\ \text{subject to} \quad & |\mathcal{S}| = m \qquad \text{where} \quad m \ll n, \\ & \mathcal{S} \subseteq \mathcal{X}. \end{aligned}$$

The challenge lies in identifying a coreset that balances size reduction and performance preservation across all downstream tasks, particularly when task-specific information is unavailable.

IV. Methodology

Coreset construction, at its core, is a question of structure: what minimal subset of a dataset preserves the form and function of the whole? ECHO builds on the intuition that, for effective learning, the information that shapes decision boundaries consists of both central tendencies and edge cases, i.e., the prototypical samples as well as the rare, the outlying, the structurally distinct samples. To capture this duality, we frame our generalization objective as identifying the precise geometry of the data: how regions concentrate, how they transition, and where boundaries emerge.

To this end, we first focus on embedding datasets into a geometry-preserving latent space, exposing its underlying structure, and then organize these embeddings into a spatially coherent representation that allows us to identify both the representative samples and also their surrounding supports (samples that define the reach and complexity of each local region). The following subsections detail each component of our approach, beginning with the geometric projection step that anchors the process.

A. Geometric Projection

The effectiveness of our structural coreset selection relies on embedding the data into a space where meaningful relationships between samples are preserved while ensuring computational efficiency. Therefore, this latent space must preserve local structure in a way that is faithful to the original space, while facilitating efficient spatial reasoning.

To this end, we define a parametric encoder $E_\theta : \mathbb{R}^D \to \mathbb{R}^d$, with $d \ll D$, which transforms each data point $\mathbf{x}_i \in \mathcal{X}$ into a lower-dimensional embedding $\mathbf{z}_i = E_\theta(\mathbf{x}_i) \in \mathbb{R}^d$. For simplicity and computational efficiency, we model this function as an MLP, with the embedding dimension set to two and the embedded values rescaled to be in the $0-1$ range, such that the latent space directly forms the basis for all subsequent steps in our pipeline, including spatial partitioning, graph construction, and coreset selection.

Encoding Objective. To ensure that the set of all embeddings, $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\} \subset \mathbb{R}^d$, reflects the intrinsic structure of $\mathcal{X}$, we train our encoder to explicitly preserve the relative spatial structure of the original dataset. That is, points that are similarly situated in $\mathcal{X}$ should remain similarly situated in $\mathcal{Z}$, up to scale and translation. To formalize this, we define a structural loss that penalizes discrepancies in average pairwise distances between the original and embedded spaces.

For a set of data points $\mathbf{X}$ and their corresponding embeddings $\mathbf{Z} = E_\theta(\mathbf{X})$, let $D^{\mathbf{X}}$ and $D^{\mathbf{Z}}$ denote their respective pairwise Euclidean distance matrices:

$$D^{\mathbf{X}}_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|, \qquad D^{\mathbf{Z}}_{ij} = \|\mathbf{z}_i - \mathbf{z}_j\|, \tag{1}$$

and let their normalized values be defined as:

$$\widetilde{D}^{\mathbf{X}}_{ij} = \frac{1}{\max_{k,\ell} D^{\mathbf{X}}_{k\ell}}\, D^{\mathbf{X}}_{ij}, \qquad \widetilde{D}^{\mathbf{Z}}_{ij} = \frac{1}{\max_{k,\ell} D^{\mathbf{Z}}_{k\ell}}\, D^{\mathbf{Z}}_{ij}. \tag{2}$$

This normalization avoids the need for strict point-wise alignment across the high-dimensional and low-dimensional spaces, and allows the embedding to emphasize local consistency.

Following this, the encoder's parameters (θ) are learned by solving the following minimization problem:

$$\min_{\theta} \sum_{i=1}^{N} \left(\frac{1}{N} \sum_{j=1}^{N} \widetilde{D}^{\mathbf{X}}_{ij} \;-\; \frac{1}{N} \sum_{j=1}^{N} \widetilde{D}^{\mathbf{Z}}_{ij} \right)^2. \tag{3}$$

This objective encourages the encoder to preserve the average local distance profile of each sample, ensuring that the latent space retains the structural characteristics of the original dataset.

After training, the encoder is fixed and its resulting latent space, characterized by $\mathbf{Z}$, enables geometry-aware operations at significantly reduced dimensional and computational cost. In particular, this supports uniform grid partitioning, efficient neighborhood computations, and our graph-based modeling of regional connectivity.

Moreover, because the embeddings are learned without supervision and solely with respect to structural fidelity, our encoder remains agnostic to downstream task while ensuring that the coreset selection process that follows is grounded in the intrinsic geometry of the data itself, rather than external modeling assumptions or class labels.

B. Structural Mapping and Coreset Selection

With the dataset embedded into a geometry-preserving latent space $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\} \subset \mathbb{R}^d$, we construct a compact subset that preserves its global structure and regional variations. This is achieved by first encoding $\mathcal{Z}$ as a proximally-connected graph over discrete regions, and then selecting coreset samples by traversing this graph in order of structural importance.

Structural Mapping. Following the construction steps of PC graphs as described in Section III-B, we partition the latent space into grid cells $\{C_1, \ldots, C_M\}$, each containing a set

of embeddings $\mathcal{Z}_j = \{\mathbf{z}_{j,1}, \ldots, \mathbf{z}_{j,m_j}\}$ and associated with a centroid $\mathbf{c}_j \in \mathbb{R}^d$. These cells are abstracted into vertices $V = \{v_1, \ldots, v_M\}$ of the graph $\mathcal{G} = (V, E)$, where edges reflect spatial adjacency or proximal connectivity.

Coreset Selection. To extract a subset $\mathcal{Q} \subset \mathcal{X}$ of fixed budget $B \ll N$ that maximally preserves the geometric and topological properties of the original dataset, we leverage the structure of $\mathcal{G}$ to identify and prioritize regions of high structural importance.

Each node $v_j \in V$ is assigned a priority based on its degree in the graph, $\deg(v_j)$, which reflects the density and centrality of its corresponding region in the latent space. We impose a prioritization scheme in which the nodes are sorted into a fixed ordering $(v_{\sigma(1)}, \ldots, v_{\sigma(M)})$ such that

$$\deg(v_{\sigma(1)}) \geq \deg(v_{\sigma(2)}) \geq \cdots \geq \deg(v_{\sigma(M)}), \tag{4}$$

where σ is a permutation on $\{1, \ldots, M\}$. This ordering determines the sequence in which cells are visited during coreset construction.

Within each cell C_j, we assign a sampling probability distribution over its points, centered at the centroid $\mathbf{c}_j$. For $\mathbf{z}_{j,k} \in \mathcal{Z}_j$, the selection probability is defined as

$$p_{j,k} = \frac{\varepsilon + \exp\left(-\|\mathbf{z}_{j,k} - \mathbf{c}_j\|^2/\tau\right)}{\sum_{\ell=1}^{m_j} \varepsilon + \exp\left(-\|\mathbf{z}_{j,\ell} - \mathbf{c}_j\|^2/\tau\right)}, \tag{5}$$

with temperature parameter $\tau > 0$ controlling the spread of the distribution, and $\varepsilon > 0$ preventing a 0-probability assignment to any sample. While the structure of the PC graph itself introduces diversity in the coreset, this density emphasizes points closest to the local centroid, which best represent the internal structure of each region.

The coreset $\mathcal{Q}$ can then be constructed through an iterative, round-based traversal of the ordered node list. At each round $r \in \mathbb{N}^+$, we visit nodes $v_{\sigma(1)}, \ldots, v_{\sigma(M)}$ in sequence, and from each cell $C_{\sigma(t)}$, select the sample with the r-th highest selection probability that has not already been added to the coreset. This process is repeated until exactly B samples have been selected. More formally, let $\pi_j : \{1, \ldots, m_j\} \to \{1, \ldots, m_j\}$ be the permutation that orders $\mathcal{Z}_j$ by descending $p_{j,k}$, so that

$$p_{j,\pi_j(1)} \geq p_{j,\pi_j(2)} \geq \cdots \geq p_{j,\pi_j(m_j)}, \quad \text{and} \tag{6}$$

$$\mathcal{Q} = \left\{ \mathbf{x}_{\sigma(t),\pi_{\sigma(t)}(r)} \;\middle|\; 1 \leq r \leq R,\, 1 \leq t \leq M \right\}, \tag{7}$$

where $\mathbf{z}_{j,k} = E_\theta(\mathbf{x}_{j,k})$, and R is the minimal number of rounds required such that $|\mathcal{Q}| = B$. Any cell containing zero non-selected points is skipped in later rounds.

This strategy spreads the selection budget across structurally salient areas first, and then proceeds uniformly through progressively less central samples, resulting in a coreset that is compact, diverse, and topologically faithful.

C. Structural Augmentation

Although the PC graph's structure already introduces diversity and ensures coverage across distinct regions, the resulting coreset $\mathcal{Q}$ can still under-represent local boundary geometry that are essential for defining decision margins. To bridge this gap, we propose a localized geometry-aware augmentation strategy that leverages boundary-adjacent samples, thereby reflecting each region's true empirical extent and improving boundary resolution for effective downstream learning.

Concretely, within each occupied cell C_j, recall the support set notation from Section III-A:

$$S_j^{(s)} \;=\; \left\{\mathbf{d}_{j,\pi_j(1)},\, \ldots,\, \mathbf{d}_{j,\pi_j(s)}\right\}, \tag{8}$$

where $\{\mathbf{d}_{j,\pi_j(r)}\}_{r=1}^{s}$ are the s displacement vectors of maximal norm from the cell center $\boldsymbol{\mu}_j$. These vectors capture the principal directions of local spread, and hence the empirical boundary, of the samples in $\mathcal{X}_j$. To construct our support set, we define

$$\mathcal{Q}^{\text{aug}} \;=\; \bigcup_{j=1}^{M} \left\{ \boldsymbol{\mu}_j \;+\; \gamma\, \mathbf{d} \;\middle|\; \mathbf{d} \in S_j^{(s)} \right\} \tag{9}$$

$$=\; \bigcup_{j=1}^{M} \left\{ \boldsymbol{\mu}_j \;+\; \gamma\, \mathbf{d}_{j,\pi_j(k)} \;\middle|\; k = 1, \ldots, s \right\}, \tag{10}$$

where $\gamma > 0$ is a scaling factor that modulates how far each support point extends beyond the local mean.

Combining $\mathcal{Q}$ and $\mathcal{Q}^{\text{aug}}$ produces a compact yet expressive summary of the dataset's geometry, ensuring that subsequent models have access to both prototypical and extremal information, which are critical for robust decision-making. In practice, while it is possible to use all available support vectors to maximize generalization, we limit the number of support vectors to a random subset of predetermined size, thereby minimizing computational overhead while still enabling structurally grounded generalization.

D. Structure-Driven Learning

Equally important as the question of which data points to select is the question of how to learn from them effectively. While the coreset $\mathcal{Q}$ and its support $\mathcal{Q}^{\text{aug}}$ extend coverage to all important regions, training on the entire union $\mathcal{Q} \cup \mathcal{Q}^{\text{aug}}$ risks being computationally expensive. Therefore, we employ a stochastic augmentation regime where at each iteration, training is performed on the core samples and a fixed-size random augmentation subset $\mathcal{A} \subset \mathcal{Q}^{\text{aug}}$, yielding $\widetilde{\mathcal{Q}} = \Psi(\mathcal{Q} \cup \mathcal{A})$, where Ψ denotes the corresponding interpolation operator. This allows us to maintain structural diversity while preserving computational efficiency.

To achieve this, we train the downstream model f_ω by minimizing a joint objective that combines the task-specific loss with a contrastive loss designed to reflect the structural affinities induced by the latent geometry. For any labeled sample $(x, y) \in \mathcal{X}$, let $\mathcal{L}_{\text{task}}(f_\omega(\mathbf{x}), y)$ denote the per-sample loss for the downstream task, such as cross-entropy for classification or mean squared error for regression. Then, the task loss over the batch $\widetilde{\mathcal{Q}}$ is given by

$$\mathcal{L}_{\text{task}}(\omega) = \frac{1}{|\widetilde{\mathcal{Q}}|} \sum_{(\mathbf{x}_i, y_i) \in \widetilde{\mathcal{Q}}} \mathcal{L}_{\text{task}}\big(f_\omega(\mathbf{x}_i),\, y_i\big), \tag{11}$$

ensuring the model learns to predict targets across both central and boundary-aware samples.

To encourage structurally aligned latent representations, we define a contrastive loss over all labeled pairs in $\widetilde{\mathcal{Q}}$. Let $\mathcal{P} = \{(i,j)\colon y_i = y_j,\, i \neq j\}$ and $\mathcal{N} = \{(i,j)\colon y_i \neq y_j\}$ be the sets of positive and negative pairs, respectively, then

$$\mathcal{L}_{\text{structure}}(\omega) = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \left\| f_\omega(\mathbf{x}_i) - f_\omega(\mathbf{x}_j) \right\|^2 + \frac{1}{|\mathcal{N}|} \sum_{(i,j)\in\mathcal{N}} \max\bigl(0,\; m - \|f_\omega(\mathbf{x}_i) - f_\omega(\mathbf{x}_j)\|\bigr)^2, \quad (12)$$

where $m > 0$ is a margin hyper-parameter (set to 1 in our experiments). This loss encourages compactness within classes and separation between classes, in alignment with the structure revealed by our PC graph.

Finally, the learning objective consists of minimizing:

$$\min_{\omega} \quad \mathcal{L}_{\text{task}}(\omega) \;+\; \lambda\, \mathcal{L}_{\text{structure}}(\omega), \quad (13)$$

where $\lambda > 0$ balances task performance and geometric regularity. This structure-driven approach ensures that learning is guided not only by semantic supervision, but also by the geometric relationships revealed during coreset construction.

Algorithm 1 summarizes all computational steps of ECHO[1].

Algorithm 1. Hierarchical Optimizations with ECHO

Input: Dataset $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, coreset budget B, support count s, augmentation count K, model f_ω, training iterations T, learning rate η.

Output: Trained model f_ω

▷ Embed data using structure-preserving encoder

$\mathbf{z}_i \leftarrow E_\theta(\mathbf{x}_i)$ for all $i = 1, \ldots, N$

Partition $\mathcal{Z} = \{\mathbf{z}_i\}_{i=1}^N$ into uniform grid cells $\{C_j\}$

▷ Construct PC graph over centroids $\mathbf{c}_j$

Build Proximally-Connected graph $\mathcal{G} = (V, E)$

▷ Rank cells by node degree

$\sigma \leftarrow \text{argsort}_j\left(-\deg(v_j)\right)$

▷ Sample coreset points

foreach $r = 1, 2, \ldots$ *until* $|\mathcal{Q}| = B$ **do**

  $\mathcal{Q} \leftarrow \mathcal{Q} \cup \left\{ \mathbf{x}_{\sigma(t),\pi_{\sigma(t)}(r)} \;\middle|\; t = 1, \ldots, M \right\}$

  $\mathcal{Q} \leftarrow \{\, q_i \in \mathcal{Q} \mid i = 1, \ldots, B \,\}$

▷ Compute structural augmentations

foreach *cell* C_j **do**

  $S_j^{(s)} = \left\{ \mathbf{d} \in \mathcal{X}_j - \boldsymbol{\mu}_j \;\middle|\; \mathbf{d} \text{ among top } s \text{ by } \|\mathbf{d}\| \right\}$

  $\mathcal{Q}^{\text{aug}} \leftarrow \mathcal{Q}^{\text{aug}} \cup \left\{ \boldsymbol{\mu}_j + \gamma\, \mathbf{d} \;\middle|\; \forall\, \mathbf{d} \in S_j^{(s)} \right\}$

▷ Train with stochastic augmentations

foreach $t = 1, 2, \ldots, T$ **do**

  Sample $\mathcal{A} \subset \mathcal{Q}^{\text{aug}}$ with $|\mathcal{A}| = K$

  $\widetilde{\mathcal{Q}} \leftarrow \Psi(\mathcal{Q} \cup \mathcal{A})$

  ▷ Compute task loss

  $\mathcal{L}_{\text{task}}(\omega) \leftarrow \frac{1}{|\widetilde{\mathcal{Q}}|} \sum_{(\mathbf{x}_i, y_i)\in\widetilde{\mathcal{Q}}} \mathcal{L}_{\text{task}}(f_\omega(\mathbf{x}_i), y_i)$

  ▷ Compute contrastive loss

  $\mathcal{L}_{\text{structure}}(\omega) \leftarrow \frac{1}{|\mathcal{P}|}\sum_{(i,j)\in\mathcal{P}} \|f_\omega(\mathbf{x}_i) - f_\omega(\mathbf{x}_j)\|^2 + \frac{1}{|\mathcal{N}|}\sum_{(i,j)\in\mathcal{N}} \max(0, m - \|f_\omega(\mathbf{x}_i) - f_\omega(\mathbf{x}_j)\|)^2$

  ▷ Update model

  $\omega \leftarrow \omega - \eta\, \nabla_\omega \left[\mathcal{L}_{\text{task}} + \lambda\, \mathcal{L}_{\text{structure}}\right]$

return f_ω

E. Complexity Analysis

Let $N = |\mathcal{X}|$ denote the dataset size, and assume fixed constants for embedding dimension d, input dimension D, coreset budget B, augmentation budget K, and support size s. We further assume that the encoder E performs a single linear mapping from $\mathbb{R}^D$ to $\mathbb{R}^d$. The embedding step applies E to all N points, requiring $O(ND)$ operations. Grid assignment for each $\mathbf{z}_i \in \mathbb{R}^d$ takes $O(1)$ time, so partitioning the embedded space incurs at most $O(N)$ cost.

Within each occupied cell C_j, the displacement vectors $\mathbf{d}_{j,k}$ are computed and sorted by norm. Since each sample appears in exactly one cell and $\sum_j m_j = N$, the maximum sorting cost is $O\left(\sum_{j=1}^M m_j \log m_j\right) = O(N \log N)$, which dominates the linear-time mean and displacement computation.

The PC graph $\mathcal{G}$ is constructed over the $M \leq N$ non-empty grid cells. Each vertex connects to at most its immediate neighbors plus one non-adjacent vertex to enforce proximal connectivity. Thus, the number of edges in $\mathcal{G}$ is constrained by $O(M)$, and constructing the graph can be performed in $O(M \log M) = O(N \log N)$. Empirically, in the 2D space, this allows up to a $\mathbf{9}\times$ runtime speed-up in graph parsing, compared to traditional fully-connected graphs.

Subsequent steps, including cell-wise sampling for coreset selection, support vector extraction, and augmentation, operate over a fixed number of elements per cell or batch and incur at most $O(N)$ additional cost. Downstream training proceeds over batches of size $O(1)$ and is decoupled from N.

Therefore, the overall computational complexity is

$$O(ND + N \log N), \quad (14)$$

with the ND term arising from the encoder and the $N \log N$ term from intra-cell sorting and PC graph construction. In the asymptotic regime, $D = O(1)$, this simplifies to $O(N \log N)$, resulting in near-linear scalability with respect to dataset size.

V. Experiments

We evaluate ECHO across a range of standard benchmarks to assess its effectiveness in selecting and learning from compact, structure-preserving coresets. Specifically, we aim to answer two central questions:

[RQ1] Can preserving the geometric profile of data enable models trained on highly compressed subsets to match the performance of full-data baselines?

[1]Source code: https://github.com/Mobile-Intelligence-Lab/ECHO

[RQ2] To what extent do hierarchical optimizations affect the runtime efficiency of coreset selection?

A. Experimental Setup

Datasets and Model Architectures.
To evaluate the generality and robustness of ECHO, we conduct experiments on five widely-used benchmark datasets[2], selected to span a range of dataset sizes and data complexity.

- **MNIST** [32]: $70,000$ grayscale images of handwritten digits (28x28 pixels), with $60,000$ training and $10,000$ test samples across 10 classes.
- **FashionMNIST** [33]: $70,000$ grayscale images (28x28) of clothing items, also split into $60,000$ training and $10,000$ test samples over 10 categories.
- **SVHN** [34]: A real-world digit dataset containing $73,257$ training and $26,032$ test color images (3x32x32), with challenging intra-class variation across 10 classes.
- **CIFAR-10** [35]: $60,000$ color images (3x32x32) spanning 10 object classes, with a $50,000$ / $10,000$ train-test split.
- **CIFAR-100** [35]: Similar to CIFAR-10 in format but with increased granularity: 100 object classes, each with 500 training and 100 test samples, totaling $50,000$ train and $10,000$ test images.

We use a LeNet-style architecture, as implemented in the DeepCore library [36], for our evaluations on the MNIST and FashionMNIST datasets, and adopt the standard ResNet-18 architecture for SVHN, CIFAR-10, and CIFAR-100.

Comparison Methods.
We compare ECHO against eighteen established and state-of-the-art methods, capturing the full methodological landscape of modern coreset selection. Following the taxonomy introduced in Section II, we group competing methods into three core families, alongside a random baseline:

- Representation and diversity-based: Herding, k-Center Greedy, Facility Location, Graph Cut, Contextual Diversity, Craig, and GradMatch.
- Training dynamics-based: Least Confidence, Entropy, Margin, DeepFool, Contrastive Active Learning, Forgetting Events, GraNd, TDDS, and Glister.
- Label-free or foundation model-driven: ELFS.
- Baseline: Random sampling.

All methods except TDDS and ELFS are implemented using the DeepCore library[3] to ensure consistency and reproducibility. For TDDS and ELFS, we use their official implementations[4][5].

Experimental Settings.
All experiments are conducted in Python using the PyTorch framework [37] for model implementation. For each experiment, we consider coreset sizes of 0.1%, 1%, 5%, 10%, 20%, 30%, and 50% of the full datasets. We train all models on each coreset for 200 epochs, including ELFS (replacing its default $40,000$ training iterations to enable fair comparisons), and perform evaluations against the entire test sets. For coreset selection methods that involve a pre-training phase, including our encoder, the pre-training is performed for 40 epochs. However, we retain ELFS's original training schedule of 200 epochs for its clustering heads, as to not degrade the quality of its selected coresets' samples. Experiments are conducted on a parallel computing cluster, using one Nvidia TITAN X GPU with 24 GB memory.

Except for ELFS training iterations, we use the default hyper-parameters for all baseline methods as provided in the DeepCore library or their official repositories. All models are trained using Stochastic Gradient Descent (SGD) with a learning rate of 0.1, a cosine decay scheduler with momentum of 0.9 and weight decay of 5×10^{-4}, and a batch size of 256.

For ECHO, we fix the number of support vectors per cell to $s = 2$, the support scaling parameter to $\gamma = 1$, and the geometric regularization weight to $\lambda = 1/2$. These values are held constant across all datasets and coreset sizes, allowing us to demonstrate the robustness of our approach without the need for per-task tuning.

B. Comparison against Existing Methods

Addressing our first question, we evaluate ECHO against eighteen coreset selection baselines on CIFAR-10 across a range of coreset budgets, from extreme compression (0.1%) to moderate size (50%). All methods are used to select a subset from the training data, on which a ResNet-18 model is trained from scratch. Test accuracy is then measured on the full test set, with results averaged over five independent runs.

As shown in Table I, ECHO consistently outperforms prior methods across all budget levels. At 0.1% of the data, where we have 50 samples per class, our approach achieves 65.6% test accuracy, over 40 percentage points higher than all alternatives. Notably, ECHO maintains a strong lead even at higher budgets, reaching 95.2% accuracy at 50% of the training set, essentially recovering full-data performance with only half the data.

Fidelity Score. To evaluate how consistently each method preserves performance across coreset sizes, we introduce the *Integrated Fidelity* score: a metric that balances accuracy and compression to capture the overall fidelity of a selection strategy. For a given coreset selection method m, let $\mathcal{B} \subset (0, 1]$ be the set of tested coreset selection ratios, and let $\mathrm{Acc}_m(b)$ denote the test accuracy achieved at budget $b \in \mathcal{B}$. Let $\mathrm{Acc}_{\mathrm{full}}$ be the accuracy of the model trained on the full dataset. Then, we define the integrated fidelity score as:

$$\mathrm{IFS}(m) = \sum_{b \in \mathcal{B}} w(b) \cdot \frac{\mathrm{Acc}_m(b)}{\mathrm{Acc}_{\mathrm{full}}},$$
$$\text{where} \quad w(b) = \frac{e^{-b}}{\sum_{b_j \in \mathcal{B}} e^{-b_j}}. \tag{15}$$

This formulation rewards methods that achieve high accuracy at lower budgets, reflecting the central goal of coreset learning. A higher IFS indicates stronger overall fidelity to the full-data model across the compression spectrum.

[2]Datasets available in torchvision: https://docs.pytorch.org/vision/stable/index.html
[3]https://github.com/PatrickZH/DeepCore
[4]https://github.com/zhangxin-xd/Dataset-Pruning-TDDS
[5]https://github.com/eltsai/elfs

TABLE I: **Performance of different coreset selection methods on the CIFAR-10 dataset.** ResNet-18 models are trained on coresets produced by the different methods and evaluated on the full test set. Performance is measured as the percentage of correctly classified samples in the test set (classification accuracy) vs. a baseline accuracy (full training set) of 95.6%. Averages and standard deviations are reported over 5 runs. The Integrated Fidelity Score is also reported for each method, indicating how consistently each method preserves the baseline accuracy across all budget sizes. The top-performing method (by fidelity score) and its three closest contenders are highlighted.

Method	Coreset Budget							Fidelity Score
	0.1%	1%	5%	10%	20%	30%	50%	
Random	21.5±0.7	36.8±0.9	64.7±2.0	74.4±4.9	82.5±4.6	87.4±3.5	91.4±2.2	0.65
Herding [13] 2009	20.0±2.3	34.8±3.3	50.8±3.4	62.6±3.0	74.2±3.2	80.4±2.9	88.3±1.9	0.58
k-Center Greedy [14] 2017	18.2±0.5	30.6±1.4	49.8±2.2	74.9±2.4	85.6±1.8	90.2±1.1	93.4±0.6	0.62
Facility Location [38] 2021	22.4±1.8	39.3±1.4	60.6±2.2	74.4±2.1	85.2±1.7	91.3±0.6	93.9±0.3	0.66
Graph Cut [38] 2021	24.2±1.8	42.8±1.4	65.5±1.1	76.6±1.2	84.1±1.1	87.7±0.9	93.3±0.5	0.68
CD [18] 2020	14.9±1.4	23.1±2.1	37.6±2.4	58.9±1.9	80.8±2.2	90.4±1.0	94.0±0.5	0.55
Craig [11] 2020	22.5±1.1	31.0±1.1	44.5±3.3	61.1±3.8	79.4±3.2	88.5±1.4	93.2±0.9	0.59
GradMatch [19] 2021	17.1±2.0	30.9±1.1	47.1±0.9	61.7±2.4	80.0±2.6	87.4±1.8	93.0±1.1	0.58
Least Confidence [20] 2019	14.2±1.1	19.9±2.0	36.0±2.1	57.2±3.4	81.8±2.6	90.1±1.8	94.5±0.2	0.54
Entropy [20] 2019	14.5±1.6	21.0±1.4	35.1±3.0	57.2±3.1	81.4±2.9	89.6±1.8	94.3±0.4	0.54
Margin [20] 2019	17.3±1.1	26.9±1.8	43.6±3.5	59.4±3.2	81.7±3.0	90.0±0.9	93.9±0.3	0.57
DeepFool [21] 2018	17.3±0.9	27.1±2.6	43.0±3.2	60.6±3.4	83.2±2.8	90.0±0.5	94.0±0.3	0.58
CAL [22] 2021	22.3±2.1	37.5±1.8	60.1±1.4	71.2±0.8	80.7±1.0	86.4±2.2	89.3±0.6	0.64
Forgetting [23] 2018	21.2±0.6	35.2±1.5	52.0±2.4	66.9±1.5	86.1±1.3	91.5±0.4	94.0±0.1	0.63
GraNd [24] 2021	17.5±1.7	26.2±1.5	40.1±2.2	52.9±2.8	78.3±2.7	91.1±1.0	94.6±0.4	0.55
TDDS [25] 2024	10.3±6.9	24.3±2.6	48.4±2.6	68.1±2.3	82.4±1.9	87.6±1.4	91.8±1.6	0.57
Glister [26] 2021	19.8±2.1	32.6±2.3	50.9±1.9	66.0±3.3	84.7±1.1	90.8±0.3	93.9±0.5	0.62
ELFS [31] 2025	17.3±2.8	32.2±2.1	57.2±1.9	78.4±3.2	86.2±2.4	87.1±1.1	88.8±0.6	0.63
ECHO (Ours)	**65.6**±1.3	**75.2**±2.6	**86.4**±3.1	**89.4**±2.8	**92.5**±1.4	**93.8**±0.5	**95.2**±0.2	**0.88**

ECHO attains an integrated fidelity score of 0.88, substantially outperforming all baselines. Its next-best contenders, namely Graph Cut (0.68), Facility Location (0.66), and random selection (0.65), trail by a wide margin, particularly at the low end of the budget range. These results demonstrate that preserving and learning from the geometric structures of datasets is not only sufficient for retaining predictive performance under compression, but also is decisively effective.

C. Validation across Datasets

For this experiment, we focus on the three top-contender baseline methods (Graph Cut, Facility Location, and Random) as comparison points, enabling focused and representative analysis while highlighting the structural advantages of ECHO.

As shown in Fig. 2, ECHO consistently outperforms all baselines across five datasets of increasing complexity: MNIST, FashionMNIST, SVHN, CIFAR-10, and CIFAR-100. At the extreme 0.1% budget, it achieves up to 60% accuracy gains over the best baseline (e.g., SVHN and CIFAR-100), and maintains clear margins as the budget increases.

Importantly, ECHO adapts well across both low-variance (e.g., MNIST) and high-variance (e.g., CIFAR-100) datasets, demonstrating strong generalization without task-specific tuning. This validates the universality of its structure-driven selection strategy, making it a reliable method for dataset compression across various real-world settings.

D. Ablation Study

To isolate the contributions of ECHO's key design choices, we conduct an ablation study focusing on two parameters: the number of support vectors per region (s) and the structural regularization (weighted by λ). The results, presented in Figure 3, compare each variant against our default implementation setting (i.e., $s = 2$, $\lambda = 1/2$), as well as the top-three contender baselines identified earlier.

Impact of Support Vectors. Setting $s = 0$ (i.e., removing support vectors entirely) leads to a significant drop in performance, particularly at smaller coreset budgets (e.g., $\sim$35% drop at 0.1%). This validates the importance of boundary-aware augmentations: without support vectors, the coreset captures regional centers but fails to represent local geometric spread critical for decision boundary formation. By contrast, $s = 1$ recovers a substantial portion of the lost accuracy, and $s = 2$ further closes the gap to full performance, indicating diminishing but meaningful returns from modeling local extent.

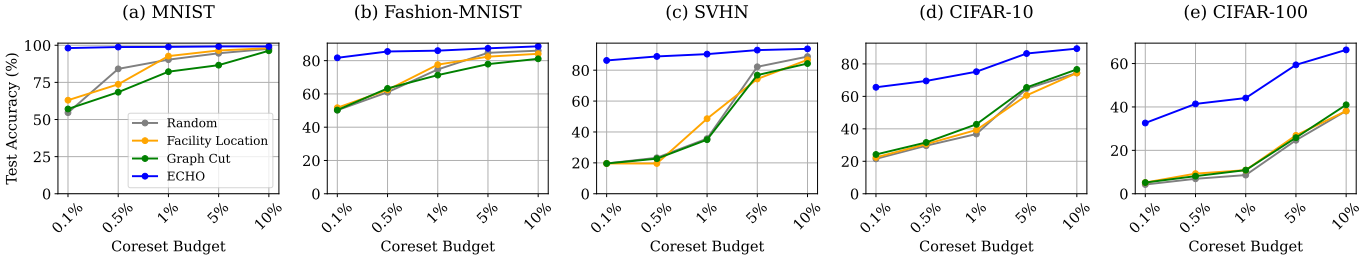

Fig. 2: Comparison of ECHO and its top-three contenders (Random, FL, GC) across five datasets.

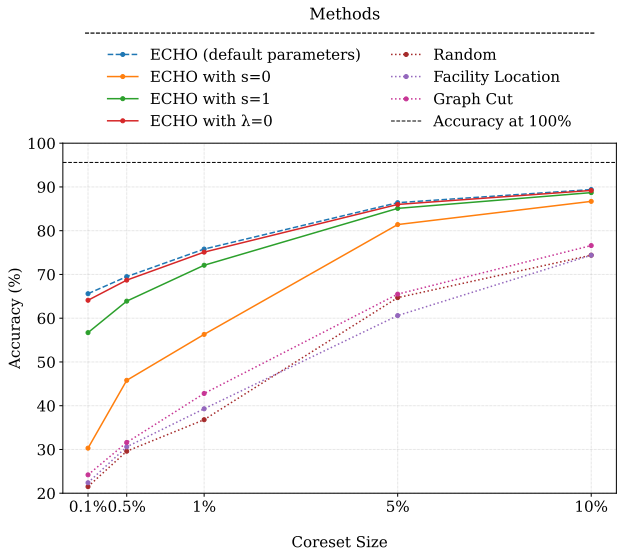

Fig. 3: Effect of support vector count per PC graph node (s) and use of structural loss (λ) on the effectiveness of ECHO. Comparisons are made with ECHO (default implementation, i.e., s=2, λ=1/2), Random, Facility Location, and Graph Cut.

Effect of Structural Regularization. When λ is set to zero, effectively removing the contrastive loss that enforces geometric consistency during learning, performance uniformly degrades across all budgets. While the degradation is less steep than omitting support vectors, the decline still averages 3–5% across the board, underscoring the benefits of geometry-aligned learning even after structurally informed selection.

Comparison with Baselines. Across all tested ablations, even the weakest variant of ECHO outperforms traditional methods such as Facility Location, Graph Cut, and Random selection, especially at low budget regimes. This highlights that the inductive bias introduced by our proximity-aware graph structure and spatial partitioning is itself a strong prior, even in the absence of augmentation or contrastive refinement. These results demonstrate that each component of ECHO contributes complementary value: region-centric selection ensures coverage; support vectors resolve geometric edges; and structural learning improves generalization.

E. Runtime Analysis

Beyond accuracy, the utility of any coreset selection method depends critically on its computational efficiency, particularly when applied to large-scale datasets or in low-latency settings. In Table II, we compare the end-to-end runtime of ECHO against its top-performing contenders: Facility Location (FL), Graph Cut (GC), and the Random baseline. Each method is evaluated over a range of coreset budgets (from 0.1% to 50%), with average runtimes reported across five runs.

TABLE II: Runtime of ECHO vs its top-three contenders. Averages and standard deviations are reported over 5 runs.

Method	Setup Phase	Runtime (in seconds) / Coreset Budget			
		0.1%	1%	10%	50%
Random	-	0.0057	0.0055	0.0055	0.0057
FL [38]	222.1±0.5	29.5±0.1	121.4±0.1	997.8±0.5	3887.1±1.3
GC [38]	221.9±0.8	22.1±0.1	22.8±0.3	48.9±0.5	499.4±0.5
ECHO	77.9±0.3	4.10±0.1	4.23±0.1	5.53±0.3	6.50±0.2

Despite incorporating a structural encoder, spatial partitioning, PC graph construction, priority-based selection, and support-vector augmentation, ECHO is remarkably efficient. Its total coreset selection time remains under 7 seconds even at 50% budget, over 550× faster than Facility Location and 75× faster than Graph Cut at the same budget level.

This efficiency stems from two of our key design choices:

(1) our use of a low-dimensional embedding space (2D), which enables rapid grid partitioning and graph construction at near-linear time (see Section IV-E), and

(2) the amortized cost of structural augmentation, which remains essentially constant across coreset sizes. Notably, ECHO's runtime increases only marginally with budget size (from 4.1s at 0.1% to 6.5s at 50%), whereas Facility Location scales super-linearly, and Graph Cut's runtime grows by over 20× across the same range.

Even when factoring in the setup phase (including encoder training), ECHO still completes in just ∼78 seconds, compared to ∼222 seconds for Facility Location and Graph Cut. This confirms that hierarchical optimizations (structurally grounded coreset selection) do not need to come at the cost of runtime.

VI. Discussion & Conclusion

This work introduces ECHO, a structure-driven coreset selection framework that unifies geometric encoding and principled augmentation to preserve data utility under extreme

compression. Through extensive comparisons across five different datasets against eighteen coreset selection baselines, we show that ECHO consistently achieves state-of-the-art performance. With up to 60% accuracy gains at very low coreset budgets, our approach outperforms all benchmarked methods, while remaining orders of magnitude faster in runtime.

Our findings demonstrate that by treating structure as the foundation for learning, we can seamlessly enable robust generalization from very few data samples.

Future work will explore applications and extensions of ECHO to unsupervised settings, different data modalities, and data streaming scenarios, where structure-aware learning may further enhance data efficiency and robustness. Although our results demonstrate strong generalization, the effectiveness of our proposed approach may in practice be influenced by embedding quality, suggesting that adaptive or higher-dimensional embeddings may further improve performance.

REFERENCES

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[3] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2021.

[4] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 843–852.

[5] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 44th annual international symposium on computer architecture*, 2017, pp. 1–12.

[6] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "Deep learning's diminishing returns: The cost of improvement is becoming unsustainable," *Ieee Spectrum*, vol. 58, no. 10, pp. 50–55, 2021.

[7] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM computing surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.

[8] P. K. Agarwal, S. Har-Peled, K. R. Varadarajan *et al.*, "Geometric approximation via coresets," *Combinatorial and computational geometry*, vol. 52, no. 1, pp. 1–30, 2005.

[9] D. Feldman, M. Schmidt, and C. Sohler, "Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering," *SIAM Journal on Computing*, vol. 49, no. 3, pp. 601–657, 2020.

[10] O. Bachem, M. Lucic, and A. Krause, "Practical coreset constructions for machine learning," *arXiv preprint arXiv:1703.06476*, 2017.

[11] B. Mirzasoleiman, J. Bilmes, and J. Leskovec, "Coresets for data-efficient training of machine learning models," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6950–6960.

[12] A. F. Diallo and P. Patras, "Deciphering clusters with a deterministic measure of clustering tendency," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[13] M. Welling, "Herding dynamical weights to learn," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 1121–1128.

[14] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," *arXiv preprint arXiv:1708.00489*, 2017.

[15] K. Wei, R. Iyer, and J. Bilmes, "Submodularity in data subset selection and active learning," in *International conference on machine learning*. PMLR, 2015, pp. 1954–1963.

[16] S. Kothawade, N. Beck, K. Killamsetty, and R. Iyer, "Similar: Submodular information measures based active learning in realistic scenarios," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 685–18 697, 2021.

[17] S. Kothawade, V. Kaushal, G. Ramakrishnan, J. Bilmes, and R. Iyer, "Prism: A unified framework of parameterized submodular information measures for targeted data subset selection and summarization," in *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*, 2022.

[18] S. Agarwal, H. Arora, S. Anand, and C. Arora, "Contextual diversity for active learning," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*. Springer, 2020, pp. 137–153.

[19] K. Killamsetty, S. Durga, G. Ramakrishnan, A. De, and R. Iyer, "Grad-match: Gradient matching based data subset selection for efficient deep model training," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5464–5474.

[20] C. Coleman, C. Yeh, S. Mussmann, B. Mirzasoleiman, P. Bailis, P. Liang, J. Leskovec, and M. Zaharia, "Selection via proxy: Efficient data selection for deep learning," *arXiv preprint arXiv:1906.11829*, 2019.

[21] M. Ducoffe and F. Precioso, "Adversarial active learning for deep networks: a margin based approach," *arXiv preprint arXiv:1802.09841*, 2018.

[22] K. Margatina, G. Vernikos, L. Barrault, and N. Aletras, "Active learning by acquiring contrastive examples," *arXiv preprint arXiv:2109.03764*, 2021.

[23] M. Toneva, A. Sordoni, R. T. d. Combes, A. Trischler, Y. Bengio, and G. J. Gordon, "An empirical study of example forgetting during deep neural network learning," *arXiv preprint arXiv:1812.05159*, 2018.

[24] M. Paul, S. Ganguli, and G. K. Dziugaite, "Deep learning on a data diet: Finding important examples early in training," *Advances in Neural Information Processing Systems*, vol. 34, pp. 20 596–20 607, 2021.

[25] X. Zhang, J. Du, Y. Li, W. Xie, and J. T. Zhou, "Spanning training progress: Temporal dual-depth scoring (tdds) for enhanced dataset pruning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26 223–26 232.

[26] K. Killamsetty, D. Sivasubramanian, G. Ramakrishnan, and R. Iyer, "Glister: Generalization based data subset selection for efficient and robust learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 9, 2021, pp. 8110–8118.

[27] K. Killamsetty, X. Zhao, F. Chen, and R. Iyer, "Retrieve: Coreset selection for efficient and robust semi-supervised learning," *Advances in neural information processing systems*, vol. 34, pp. 14 488–14 501, 2021.

[28] N. Sachdeva, C.-J. Wu, and J. McAuley, "Svp-cf: Selection via proxy for collaborative filtering data," *arXiv preprint arXiv:2107.04984*, 2021.

[29] B. A. Griffin, J. Marks, and J. J. Corso, "Zero-shot coreset selection: Efficient pruning for unlabeled data," *arXiv preprint arXiv:2411.15349*, 2024.

[30] X. Zhang, J. Zhai, S. Ma, C. Shen, T. Li, W. Jiang, and Y. Liu, "Speculative coreset selection for task-specific fine-tuning," *arXiv preprint arXiv:2410.01296*, 2024.

[31] H. Zheng, E. Tsai, Y. Lu, J. Sun, B. R. Bartoldson, B. Kailkhura, and A. Prakash, "Elfs: Label-free coreset selection with proxy training dynamics," in *The Thirteenth International Conference on Learning Representations*, 2025.

[32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[33] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[34] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng *et al.*, "Reading digits in natural images with unsupervised feature learning," in *NIPS workshop on deep learning and unsupervised feature learning*, vol. 2011, no. 2. Granada, 2011, p. 4.

[35] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[36] C. Guo, B. Zhao, and Y. Bai, "Deepcore: A comprehensive library for coreset selection in deep learning," in *International Conference on Database and Expert Systems Applications*. Springer, 2022, pp. 181–195.

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

[38] R. Iyer, N. Khargoankar, J. Bilmes, and H. Asanani, "Submodular combinatorial information measures with applications in machine learning," in *Algorithmic Learning Theory*. PMLR, 2021, pp. 722–754.