TopoPrune: Robust Data Pruning via Unified Latent Space Topology

Arjun Roy
Purdue University
roy208@purdue.edu

Prajna G. Malettira
Purdue University
pmaletti@purdue.edu

Manish Nagaraj
Purdue University
mnagara@purdue.edu

Kaushik Roy
Purdue University
kaushik@purdue.edu

Abstract

Geometric data pruning methods, while practical for leveraging pretrained models, are fundamentally unstable. Their reliance on extrinsic geometry renders them highly sensitive to latent space perturbations, causing performance to degrade during cross-architecture transfer or in the presence of feature noise. We introduce TopoPrune, a framework which resolves this challenge by leveraging topology to capture the stable, intrinsic structure of data. TopoPrune operates at two scales, (1) utilizing a *topology-aware manifold approximation* to establish a global low-dimensional embedding of the dataset. Subsequently, (2) it employs *differentiable persistent homology* to perform a local topological optimization on the manifold embeddings, ranking samples by their structural complexity. We demonstrate that our *unified dual-scale topological approach* ensures high accuracy and precision, particularly at significant dataset pruning rates (e.g., 90%). Furthermore, through the inherent stability properties of topology, TopoPrune is (a) exceptionally robust to noise perturbations of latent feature embeddings and (b) demonstrates superior transferability across diverse network architectures. This study demonstrates a promising avenue towards stable and principled topology-based frameworks for robust data-efficient learning.*

1 Introduction

The computational demands of training modern deep learning systems have escalated dramatically due to the scale of contemporary models and datasets. This growth has made training and fine-tuning computationally prohibitive, creating a need for data-efficient learning strategies. Data pruning is one such strategy which sub-samples a large dataset into a smaller, representative subset (or coreset) that preserves the learning characteristics of the full dataset. Thereby enabling rapid model training, efficient fine-tuning, and reduced storage costs, all while minimizing degradation in final model performance.

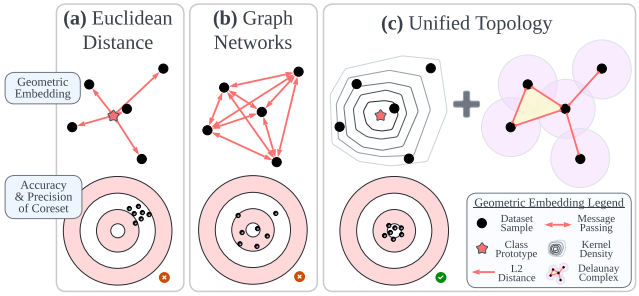


Figure 1: **Topological data selection yields higher-performing and more stable coresets. (a)** Euclidean-based selection is precise but achieves lower accuracy. **(b)** Graph-based methods achieve higher accuracy but are highly variable. **(c)** Our topological approach achieves both high accuracy and precision.

Broadly, coreset selection methods fall into three major categories. *Optimization-based* methods select a coreset whose loss landscape [51, 66] or gradient dynamics [67, 50, 91] align with the full dataset.

*Code is available at https://github.com/ar-roy/topoprune

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

While effective, such approaches are hampered by computationally intensive second-order [80] or bilevel optimization [10]. *Score-based* methods rank samples using training dynamics [95, 38, 109] or uncertainty estimations [77, 45, 44, 21]. However, these scores are intrinsically biased by a models inductive prior, capacity bottleneck, and unique learning trajectory. Furthermore, requiring access to training dynamics makes both optimization and score-based methods incompatible with the growing ecosystem of pretrained models, where only final, static representations are accessible.

To overcome this constraint, *geometry-based* coreset selection methods can operate on static embeddings from pretrained models. Approaches range from representations based on the penultimate-layer distances [100], distributional similarity via optimal transport [101], Wasserstein distance [103], or geometric reconstruction error [105]. While avoiding costly training analysis, these methods rely on metrics that are sensitive to the extrinsic geometry of the latent space, a vulnerability we term "geometric brittleness" [76]. This brittleness leads to two shortcomings: (1) over-prioritization of dense regions at the expense of informative sparse distribution tails [108], and (2) an instability across network architectures or under direct embedding perturbation. This is most apparent in Euclidean-distance metrics [100] and message-passing or spectral graph methods [61, 102, 25], which are sensitive to latent-space changes (see Fig. 1).

In this work, we introduce TopoPrune, a novel framework that resolves the challenge of geometric brittleness by leveraging topology [84, 76], which studies properties of space that are preserved under continuous deformations such as stretching and bending. As a canonical illustration, a coffee mug and a torus (donut) are topologically homeomorphic, one can be continuously deformed into the other because they share the same fundamental invariant (a single hole). By focusing on these stable, intrinsic invariants rather than transient, extrinsic geometric measurements (such as distance or curvature), we can analyze the latent space of datasets with a stable, topological metric. This structural focus allows TopoPrune to achieve exceptional robustness against embedding perturbations; whether caused by isotropic noise, anisotropic distortion, or shifts across varying network architectures [22, 89]. This enables the use of proxy models [23] or off-the-shelf pretrained models for coreset generation without retraining.

Our framework first establishes a **global structure** by using topology-aware manifold approximation [64, 99] to project high-dimensional features into standardized low-dimensional embeddings. While this global structure can group similar samples, it fails to distinguish which samples to prioritize within a localized region. Existing methods often resort to random sampling within localized regions [108] or use geometric heuristics like message-passing [61]. To better complement this global view with **local structure**, we then employ differentiable persistent homology [83, 16, 70] to assess a sample's structural relevancy, relative to its immediate neighbors. Persistent homology tracks the "birth" and "death" (persistence) of topological structures at multiple scales (derived from a filtration of simplicial complexes). For our application, we perform an optimization that *maximizes the persistence (the lifespan of the features) of local topological structures* constructed from a multiparameter filtration of the manifold projected Delaunay complex [59, 2]. This process iteratively repositions samples to an optimal configuration that enhances topological stability, directly measuring a sample's contribution to the structural complexity of its local neighborhood.

Our approach makes the following contributions:

- We introduce TopoPrune, a novel coreset selection framework that defines sample importance through a dual-scale topological analysis. It combines a *global manifold projection* with a *local persistence* score derived from a differentiable persistent homology optimization to identify structurally critical samples with higher accuracy and precision compared to previous coreset methods, particularly in extreme compression regimes (e.g., 90% pruning).
- We show that TopoPrune is resilient to representation degradation, maintaining superior robustness and coreset quality across three distinct perturbation paradigms: isotropic latent noise, pixel-level input noise, and structured image corruptions.
- We demonstrate that TopoPrune establishes robust cross-architecture transferability, consistently yielding high-quality coresets regardless of the transfer direction, whether utilizing diverse proxy embeddings (e.g., from ResNet to ViT) for a fixed target model or a single proxy to train a diverse set of target models. This flexibility permits the use of computationally inexpensive or off-the-shelf pretrained models for coreset generation without costly retraining.

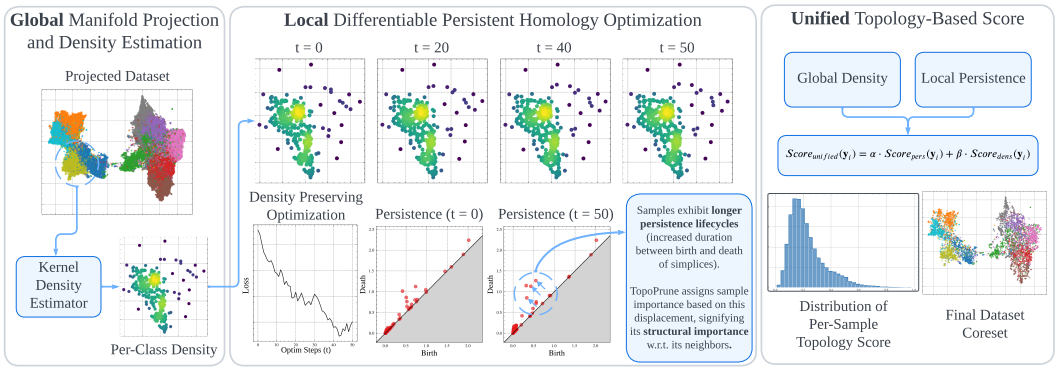


Figure 2: **An overview of TopoPrune**. **(Left)** A topology-aware projection visualizes the *global* data manifold. **(Middle)** Within each class, a density-preserving persistent homology optimization derives a *local* persistence score per sample. The color map indicates high (yellow) to low (blue) density. **(Right)** The final coreset is constructed via stratified sampling on a unified score combining *global density* and *local persistence*. This not only prioritizes the most topologically informative samples but also faithfully represents the density distribution of the original dataset.

2 Background and Related Work

2.1 Topology at Two Scales

While many modern topological algorithms inherently model both the global and local structure of data simultaneously, our work decouples these concepts into two distinct stages. For the purposes of this paper, we define global topology as the manifold structure of the entire dataset, which we capture as a low-dimensional embedding. We then define local topology as the fine-grained structure arising from the interactions between samples and their immediate neighbors, which we analyze using persistent homology.

Low-Dimensional Manifold Approximations. A critical step in high-dimensional analysis is creating a low-dimensional data representation. Linear methods like PCA [78] are efficient but preserve only global variance while missing non-linear structures. t-SNE [98] excels at preserving fine-grained local neighborhoods but distorts global structure. Topology-aware manifold approximation and projection (MAP) methods such as UMAP [64], PaCMAP [99], and DensMAP [73] model high-dimensional data as a fuzzy topological structure to preserve both fine-grained local connectivity and large-scale global relationships.

Learning-based alternatives such as Topological Autoencoders (TopoAE) [69] and Representation Topological Divergence (RTD) [96] offer powerful global topology preservation. However, they require computationally expensive training and often yield less distinct class separation on complex datasets. Topology-preserving objectives have similarly been applied to neural network embedding spaces in knowledge distillation [52], semi-supervised continual learning [31], and vision-language models [107, 106], each of which also requires training. Therefore, we prioritize algorithmic solutions like UMAP for efficiency and alignment with our training-free objective. These MAP-based methods have proven highly effective for interpreting the complex representations learned by deep models across numerous domains, from single-cell genomics [6] to clustering in dictionary learning [33, 34]. As shown by [26], these techniques produce embeddings with compact and well-defined clusters, enhancing the impact of downstream analysis.

By preserving nearest-neighbor structure from the high-dimensional space, MAP-based methods capture the underlying topological connectivity of the data manifold. This preserves notions of "prototypicality" (samples in highly connected regions) and "atypicality" (samples in sparse regions) after projection (see Section A.3 for a qualitative explanation on prototypicality and Section B.1 for an investigation across projection techniques).

Interactions of Samples and their Nearest Neighbors. Understanding the local interactions between a sample and its neighbors is crucial for determining its importance. A prevalent approach is the use of Graph Neural Networks (GNNs), which propagate information between nodes on a graph typically defined by nearest-neighbor relationships. In GNNs, a sample's importance is quantified through message-passing that aggregates features from its local neighborhood [61]. Other

methods use graph-level structural entropy combined with Shapley values and blue noise sampling [20] to select a diverse coreset [102]. FAST [25] likewise describes topology through the spectrum (symmetric normalized Laplacian) of a nearest-neighbor graph, which it uses to regularize distribution matching. However, these approaches operate on a single, fixed graph and can be sensitive to the geometric hyperparameters used in its construction.

In contrast, persistent homology [29, 111] offers a fundamentally different and more robust framework. Instead of analyzing a single graph, it studies the evolution of higher-order topological structures (e.g., connected components, loops, voids) across a multiparameter filtration of simplicial complexes [13, 14, 11]. This provides a complete summary of the data's shape at all scales simultaneously. A key advantage of persistent homology is its proven stability [22]. The persistence diagram of a dataset is guaranteed to change only slightly in response to small perturbations of the input data, making it a robust descriptor of local structure [97, 68]. Inclusion of these robust geometric descriptors has been widely used for understanding feature embeddings in machine learning applications such as monitoring generalization in networks over training [8] and exploring the topology of latent embeddings throughout network layers [72, 32]. For a detailed overview on the underlying construction of simplicial complexes and persistent homology, please see Section A.1.

While traditional persistent homology provides a powerful descriptive tool, its integration into modern deep learning pipelines has been limited as it is not inherently differentiable. Recent advances in differentiable persistent homology have overcome this barrier by enabling the backpropagation of gradients from the persistence diagram back to the coordinates of the input data points [16, 70]. This allows for the direct optimization of the data's topological features within a gradient-based framework. The work of [83] provides a fast and stable computational framework for these gradients, even for the more expressive case of multiparameter persistent homology. By leveraging this, rather than simply describing static local topology, we can perform an optimization to actively enhance it.

3 Methodology

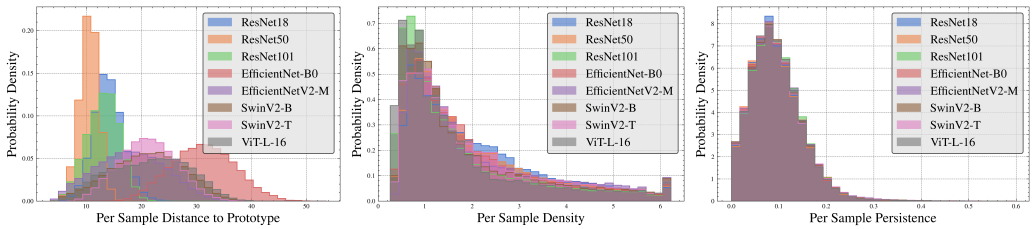
Our proposed method, TopoPrune, constructs a coreset by analyzing the data's topological structure at two distinct scales. (1) *Global Manifold Embedding*, projects the original high-dimensional embeddings into a standardized low-dimensional space. This ensures a stable, global view of the data's overall structure. (2) *Local Topological Interaction*, which employs differentiable multiparameter persistent homology to probe the local structure formed by samples and their closest neighbors. Together, these two topological scales are used to derive an importance score for each sample based on global density and local persistent homology, contributing to a unified topological measurement for selecting individual samples (see Fig. 2).

3.1 Global Structure: Dataset Representation with Topological Manifold Embedding

Given a well-trained deep model, denoted by $f(\cdot)$, we can express it as a composition of a feature extractor $h(\cdot)$ and a classifier $g(\cdot)$, such that $f(\cdot) = g(h(\cdot))$. Here, $h(\cdot)$ represents the network up to the *penultimate layer*, which maps an input data point $\mathbf{x}$ to a high-dimensional feature embedding $\mathbf{z} = h(\mathbf{x}) \in \mathbb{R}^D$. The full dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ can thus be transformed into a high-dimensional feature set $Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$. While this high-dimensional space Z contains rich semantic information, its extrinsic geometry is often complex and architecture-dependent. To obtain a stable and standardized representation, we project Z onto a low-dimensional manifold using topology-based manifold approximation and projection techniques [64, 99, 73].

This process involves two main stages. First, a topological representation of the high-dimensional data is constructed as a fuzzy simplicial set. This structure captures the data's shape by assigning a membership strength (p_{ij}), to the potential connections between each point and its neighbors, where the "fuzzy" aspect represents the belief that a certain simplex exists in the true underlying manifold. Subsequently, a low-dimensional embedding is learned $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, where $\mathbf{y}_i \in \mathbb{R}^d$ and $d \ll D$, whose own fuzzy simplicial set (q_{ij}) is similarly defined. The final low-dimensional representation Y is found by optimizing the positions of the points $\{\mathbf{y}_i\}$ to minimize a cross-entropy loss between the high-dimensional (p_{ij}) and low-dimensional (q_{ij}) pairwise similarities:

$$\mathcal{L}_{\text{proj}}(Y) = \sum_{i \neq j} \left[p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right) + (1 - p_{ij}) \log\left(\frac{1 - p_{ij}}{1 - q_{ij}}\right) \right] \tag{1}$$



(a) Euclidean Distance to Prototype (b) Global Manifold Kernel Density (c) Local Persistence

Figure 3: **Topological metrics are more consistent across networks**. Which translates directly to better coreset performance. Metric distributions become progressively more uniform as we move from **(a)** unstable Euclidean distances, to **(b)** density estimation from global topological projection, and finally to **(c)** local persistence.

This process yields a standardized manifold embedding that preserves the data's intrinsic shape. Through a detailed investigation into different manifold approximation and projection techniques presented in Section B.1 we use UMAP [64] as it creates uniform manifold embeddings across network architectures (see Section C.6 exploring UMAP hyperparameters). On this low-dimensional manifold we compute **Density Score**, a class-conditional rarity metric for each sample using a Kernel Density Estimator (KDE). Rather than using density directly, we adopt its negative log-transform, the self-information (or surprisal) of the sample under the estimated density [24, 88]. For an event x with probability $p(x)$, the self-information $-\log p(x)$ quantifies how unexpected, or informative, that event is. High-probability events convey little information, while rare events convey much. Applied to a sample's class-conditional density, this transforms the score into a monotonic measure of structural rarity that increases with informativeness. For a sample $\mathbf{y}_i$ belonging to class c:

$$\text{Score}_{\text{dens}}(\mathbf{y}_i) = -\log\left(\frac{1}{N_c h}\sum_{\mathbf{y}_j \in Y_c} K\left(\frac{\mathbf{y}_i - \mathbf{y}_j}{h}\right)\right) \tag{2}$$

where $N_c = |Y_c|$ is the total number of samples in class c, K is a Gaussian kernel, and h is the bandwidth. This score allows us to distinguish samples in low-density (atypical) regions of the manifold from those in high-density (prototypical) regions.

3.2 Local Structure: Sample Neighborhoods with Persistence-Based Optimizer

The global manifold embedding provides a low-dimensional representation that faithfully approximates the global structure of the data manifold. While this ensures a stable, high-level representation, a purely global perspective is insufficient for identifying the most informative samples, particularly in extreme compression regimes (e.g., 90% pruning). Samples co-located in the embedding space provide highly redundant information during model optimization, yielding diminishing marginal returns for gradient updates [85, 87]. Because these tightly clustered samples inherently exhibit nearly identical density scores, relying exclusively on density-based selection within a local neighborhood fails to resolve structural importance, effectively degenerating into local uniform sampling [61, 108].

To resolve this local ambiguity and capture fine-grained structure, we leverage persistent homology not as a static descriptor, but as a dynamic topological optimization process. The objective of this process is to iteratively adjust the position of each point within its class manifold to maximize persistence life-cycles (increasing the duration between birth and death of topological features within the simplicial complex). This is performed independently for each class $c \in \{1, \ldots, C\}$ to analyze the specific intra-class structure. For each class, we begin with its point cloud from the global manifold embedding $Y_c = \{\mathbf{y}_i \mid \text{label}(\mathbf{y}_i) = c\}$ and construct a Delaunay filtration [2] on Y_c, which circumvents prohibitive combinatorial explosions typically associated with the standard Vietoris-Rips (VR) complex [75, 68]. By restricting VR-style weights to the Delaunay skeleton, we accelerate computation while maintaining stability guarantees in our low-dimensional setting.

Similar to work from [83] we define a differentiable objective, $\mathcal{L}_{\text{pers}}(Y_c)$, whose gradient, $-\nabla_{Y_c}\mathcal{L}_{\text{pers}}$, points in the direction of steepest ascent, *maximally increasing the persistence life-cycle of samples*. This objective is formulated using a multi-parameter filtration considering two parameters: (1) the class-manifold Delaunay filtration (Del_{Y_c}) and (2) the class-manifold Kernel Density Estimator

($\hat{f} = KDE_{Y_c}$). The persistence of this two-parameter filtration is summarized using the Hilbert decomposition signed measure, of homology degree 1 (H_1), denoted $\mu^{Hil}_{H_1(Del_{Y_c},\hat{f})}$ [60]. We ablate the choice of filtration and homology degree in Section C.3. This descriptor represents the persistence diagram as a finite collection of positive point masses (representing feature births) and negative point masses (representing feature deaths) in the parameter space of (distance, density). Our objective is to maximize persistence life-cycles by maximizing the Optimal Transport (OT) distance between this signed measure and the zero measure, $\mathbf{0}$ [15]. The zero measure acts as a trivial baseline with no mass. OT distances reduce to the internal matching cost between positive (birth) and negative (death) point masses, capturing the total topological persistence. The differentiable objective for a given class c is therefore defined as:

$$\mathcal{L}_{\text{pers}}(Y_c) = \text{OT}(\mu^{Hil}_{H_1(Del_{Y_c},\hat{f})}, \mathbf{0}) \tag{3}$$

The optimization seeks a new point configuration Y_c' that maximizes this objective, solved iteratively via gradient ascent (see Section C.2 exploring optimization steps). This formulation *resolves topological ambiguity by separating features from the diagonal, while preserving the original density of the class manifold*, as the density is recomputed at each epoch and is an integral part of the objective. We then define the **Persistence Score** for each sample $\mathbf{y}_i$ belonging to class c as the magnitude of its total displacement during its class-specific optimization, where $\mathbf{y}_i$ is the initial position and $\mathbf{y'}_i$ is the final, optimized position.:

$$\text{Score}_{\text{pers}}(\mathbf{y}_i) = \|\mathbf{y}_i - \mathbf{y'}_i\|_2, \quad \text{for } \mathbf{y}_i \in Y_c, \mathbf{y'}_i \in Y_c' \tag{4}$$

Interpreting this notion of local dataset structure. A high persistence score quantifies the topological ambiguity a sample introduces in its class manifold. Density-preservation during optimization is vital, ensuring our search for structurally important samples does not distort global representativeness. The magnitude of corrective displacement therefore serves as a direct, dynamic measure of a sample's contribution to the topological complexity of its class.

3.3 Comprehensive Score with Global and Local Dataset Structures

To construct a highly effective coreset, we formulate a comprehensive sample importance metric that unifies global density and local persistence. As $\text{Score}_{\text{dens}}$ measures density surprisal and $\text{Score}_{\text{pers}}$ measures topological complexity, both signals quantify a sample's structural informativeness. To ensure neither metric implicitly dominates the formulation due to scale differences, both signals are independently normalized to the $[0, 1]$ range. This unified score is computed as a weighted combination, where $\mathcal{N}(\cdot)$ denotes min-max normalization, and hyperparameters $\alpha, \beta \in [0, 1]$ modulate the influence of local topological complexity versus global distributional rarity (see Fig. 4 for visualization):

$$\text{Score}_{\text{unified}}(\mathbf{y}_i) = \alpha \cdot \mathcal{N}(\text{Score}_{\text{pers}}(\mathbf{y}_i)) + \beta \cdot \mathcal{N}(\text{Score}_{\text{dens}}(\mathbf{y}_i)) \tag{5}$$

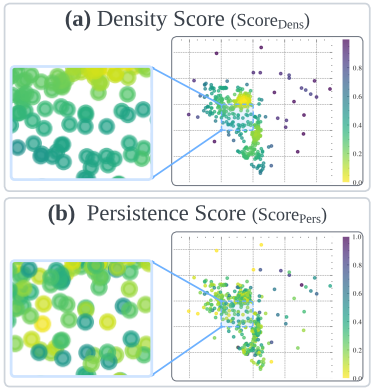


Figure 4: **Dual-scale scoring. (a)** Global Density Score provides structural prior but assigns saturated values within localized regions. **(b)** Local Persistence Score provides fine-grained neighborhood structure, isolating topological anchors.

We empirically validate the complementary nature of these metrics through three analyses provided in Section C.1. First, a joint distribution analysis across 10 network architectures reveals a consistently low linear correlation between persistence and density (average Pearson $r = 0.102$), confirming that the two scores capture independent structural properties. Second, qualitative analysis of target class embeddings illustrates that proximate samples with identical global densities frequently exhibit divergent local persistence scores. Finally, an ablation on the weighted combination demonstrates that integrating both signals outperforms either metric in isolation (e.g., by up to 5.4% at high pruning rates on CIFAR-100). Together, these analyses confirm that $\text{Score}_{\text{unified}}$ measures structural informativeness from complementary views: global rarity and local topological complexity.

3.4 Mislabel Filtering and Coreset Construction

Following CCS [108], we ensure our coreset is not corrupted by noisy or mislabeled data, which can receive high importance scores yet degrade model performance [90]. We incorporate a filtering step to create a clean dataset $\mathcal{D}_{clean} = \mathcal{D} \setminus \mathcal{I}_{mis}$ where $\mathcal{I}_{mis}$ are the mislabeled sample indices. Most prior methods identify such samples using training-dynamic metrics like Area Under the Margin (AUM) [79], which conflicts with our training-free objective. Therefore, we utilize a Neighborhood Label Purity Score (NLPS). Drawing on the training-free local voting introduced by Zhu et al. [110], NLPS calculates the fraction of a sample's k-nearest latent-space neighbors with a different class label. A high NLPS indicates a mixed-label neighborhood, effectively serving as a training-free analog to the "flip-flop" candidates identified by AUM. We validate NLPS against alternative proxies in Section C.4 and adopt the mislabel ratios from Zheng et al. [108] (Appendix Table 20).

From the resulting clean dataset $\mathcal{D}_{clean}$, we construct the coreset via stratified sampling on $\text{Score}_{\text{unified}}$, preserving the original class distribution as in Zheng et al. [108]. TopoPrune produces an unbalanced coreset, respecting the dataset's intrinsic class imbalance rather than enforcing uniform per-class counts. The full pipeline therefore comprises three phases: (1) dual-scale topological scoring, (2) NLPS-based mislabel filtering, and (3) stratified topological selection. Pseudocode appears in Section B.3 and an illustrative walkthrough in Section C.1.4.

4 Results

4.1 Experimental Setup

TopoPrune utilizes several tools and frameworks. Manifold projection is performed using `UMAP` [65], `multipers` [58] facilitates differential persistent homology which uses the `Gudhi` C++ library [62] as a backend, and `DeepCore` [42] is used to standardize coreset selection and training across different methods.

To ensure fair comparisons in our experiments, we evaluate two versions of our framework. When benchmarking against other training-free methods, we use *TopoPrune (NLPS)*, which incorporates NLPS for mislabeled samples. When comparing against methods that require training-time information, we use *TopoPrune*, which incorporates the original AUM score. We compare **TopoPrune (NLPS)** with several static geometry-based coreset selection methods: **Random** selection. **Moderate** [100] uses samples near the median distance to a class prototype (the barycenter of a point-mass distribution). **FDMat** [101] matches data distribution between dataset and coreset using optimal transport. We compare **TopoPrune** with several geometry, score, and optimization-based methods that require training-time information: **Moderate (AUM)** incorporating Moderate with AUM-based mislabeled removal. **Forgetting** [95] uses the number of times

Table 1: **Accuracy across coreset selection methods** on CIFAR-10, CIFAR-100 and ImageNet-1K. TopoPrune demonstrates a scaling advantage: while competitive on simpler datasets, performance improvements over baselines widen as dataset complexity and pruning severity increase.

	Pruning Rate (→)	30%	50%	70%	80%	90%
		CIFAR-10 (ResNet-18)				
No Training Dynamics	Random	94.5±0.1	93.5±0.1	90.8±0.2	86.6±0.3	76.7±0.9
	Moderate	94.2±0.1	93.1±0.1	89.9±0.2	87.2±0.2	76.9±1.0
	FDMat	94.7±0.1	93.6±0.2	**90.8±0.2**	87.3±0.4	74.4±0.7
	TopoPrune (NLPS)	**94.8±0.1**	**93.6±0.2**	90.3±0.2	**87.3±0.3**	**77.1±0.6**
With Training Dynamics	Moderate (AUM)	93.9±0.2	93.1±0.2	90.1±0.2	87.1±0.2	79.9±0.3
	Forgetting	94.5±0.2	92.6±0.1	89.8±0.2	85.6±0.3	67.6±0.4
	Glister	94.4±0.2	93.8±0.2	90.8±0.4	85.1±0.6	66.8±1.3
	LCMat-S	94.5±0.2	93.3±0.2	90.5±0.2	86.9±0.2	75.1±0.8
	ELFS	94.7±0.2	93.7±0.2	90.6±0.3	87.5±0.4	79.8±0.9
	CCS	95.5±0.1	94.8±0.2	93.0±0.2	**90.7±0.2**	81.9±0.7
	FAST	95.2±0.1	93.2±0.2	90.6±0.3	87.4±0.4	80.2±1.4
	D2	**95.6±0.1**	**94.8±0.1**	**93.1±0.1**	89.2±0.2	80.9±1.5
	TopoPrune	94.9±0.2	94.2±0.2	92.1±0.2	89.4±0.2	**82.0±0.2**
		CIFAR-100 (ResNet-18)				
No Training Dynamics	Random	75.3±0.2	71.6±0.1	63.7±0.5	55.9±1.0	34.0±1.1
	Moderate	74.9±0.3	70.1±0.3	63.7±0.2	56.1±0.5	34.9±2.1
	FDMat	75.4±0.2	71.9±0.3	64.0±0.6	56.1±1.5	37.5±1.6
	TopoPrune (NLPS)	**75.6±0.2**	**71.9±0.2**	**65.3±0.4**	**56.7±0.4**	**41.6±0.8**
With Training Dynamics	Moderate (AUM)	75.9±0.3	72.4±0.2	66.7±0.3	60.2±0.8	40.0±1.2
	Forgetting	74.8±0.2	67.2±0.9	50.6±0.7	32.3±0.9	24.3±1.4
	Glister	75.8±0.3	70.7±0.7	66.1±1.2	54.7±1.6	38.4±1.7
	LCMat-S	75.3±0.2	71.1±0.2	62.5±0.8	52.1±2.0	36.1±1.7
	ELFS	75.5±0.4	72.5±0.4	66.7±0.6	59.0±1.0	42.9±1.3
	CCS	**76.9±0.3**	73.8±0.3	67.8±0.7	60.7±0.6	45.2±2.4
	FAST	76.4±0.2	72.8±0.2	66.9±0.8	60.3±2.6	42.6±2.4
	D2	75.1±0.5	71.2±0.2	67.8±0.9	61.1±1.4	44.3±2.6
	TopoPrune	76.7±0.3	**73.8±0.3**	**68.1±0.3**	**62.3±0.4**	**45.7±0.5**
		ImageNet-1K (ResNet-50)				
No Training Dynamics	Random	69.8±0.5	68.4±0.5	65.1±0.4	61.9±0.5	52.5±0.6
	Moderate	69.5±0.2	65.8±0.4	60.5±0.1	57.7±0.2	50.0±0.4
	FDMat	**70.8±0.3**	68.7±0.5	65.5±0.7	62.0±0.3	51.9±0.3
	TopoPrune (NLPS)	70.7±0.4	**69.9±0.2**	**66.4±0.2**	**63.2±0.3**	**53.9±0.2**
With Training Dynamics	Moderate (AUM)	69.6±0.4	67.2±0.6	63.9±0.8	60.4±0.6	52.7±0.3
	Forgetting	69.9±0.2	66.8±0.6	60.2±0.5	59.1±0.4	50.0±0.5
	Glister	66.3±0.4	63.5±0.3	59.3±0.5	56.5±0.3	49.3±0.8
	LCMat-S	69.8±0.4	67.5±0.5	62.2±0.3	59.7±0.5	48.8±0.6
	ELFS	69.9±0.2	68.7±0.2	65.0±0.3	62.5±0.4	55.0±0.7
	CCS	70.1±0.5	69.1±0.3	65.7±0.3	62.6±0.6	55.2±0.7
	FAST	70.2±0.2	68.4±0.4	65.5±0.3	62.9±0.5	55.6±0.6
	D2	69.5±0.3	67.1±0.5	65.7±0.4	62.7±0.9	55.5±1.3
	TopoPrune	**70.8±0.2**	**69.5±0.2**	**66.2±0.1**	**63.1±0.3**	**56.1±0.2**

Table 2: **Text dataset pruning** (mean ± std over 3 seeds). Accuracy on ANLI, and held-out perplexity (PPL), ARC-Challenge (ArcC) and the average over MMLU and six other commonsense reasoning benchmarks (Avg.) after LoRA-tuning on Alpaca, with per-benchmark results in Appendix Table 13. TopoPrune is best on ANLI at every pruning rate and is most competitive on ARC-Challenge, the hardest commonsense reasoning benchmark.

		ANLI		**Alpaca (LoRA)**					
		RoBERTa-L	Qwen2.5-1.5B	Llama-3.2-3B			Pythia-2.8B		
Pruning Rate	Method	Acc.	Acc.	PPL ↓	ArcC	Avg.	PPL ↓	ArcC	Avg.
70%	Random	55.1±1.9	59.2±0.9	3.029±0.001	50.7±0.3	66.0±0.4	**3.953±0.015**	37.6±0.5	52.1±0.2
	Moderate	56.0±0.9	56.8±0.7	3.041±0.004	51.1±0.7	65.8±0.3	3.960±0.026	37.7±0.6	52.3±0.5
	D2	54.5±2.4	59.3±0.8	3.079±0.001	51.2±0.8	66.2±0.2	3.990±0.013	37.3±1.0	52.3±0.6
	TopoPrune	**56.4±0.3**	**60.3±0.5**	**3.029±0.001**	**53.8±0.5**	**66.2±0.1**	3.972±0.020	**37.9±1.1**	**52.3±0.4**
80%	Random	51.7±1.0	58.2±1.0	3.039±0.002	51.7±0.1	66.2±0.2	4.017±0.048	37.0±0.3	52.7±0.4
	Moderate	52.1±1.6	55.6±0.7	3.048±0.001	51.8±0.6	65.9±0.1	**3.977±0.017**	37.6±0.9	52.9±0.1
	D2	52.6±1.8	59.5±0.9	3.048±0.006	51.3±1.0	**66.6±0.2**	4.018±0.035	36.9±0.4	**53.1±0.1**
	TopoPrune	**52.9±0.8**	**60.2±0.5**	**3.036±0.001**	**52.4±0.4**	66.4±0.2	3.996±0.015	**38.1±0.5**	52.9±0.3
90%	Random	47.1±0.5	56.0±1.5	3.055±0.001	50.6±0.9	66.5±0.3	4.044±0.015	36.6±0.4	52.8±0.3
	Moderate	48.5±2.0	52.9±1.3	3.060±0.002	52.1±1.2	66.6±0.5	4.068±0.024	37.3±1.2	52.9±0.6
	D2	47.4±1.2	54.8±3.7	**3.051±0.003**	51.5±1.1	66.5±0.4	4.079±0.013	37.1±0.8	53.2±0.2
	TopoPrune	**49.1±1.1**	**58.5±0.6**	3.053±0.006	**52.6±0.5**	**66.6±0.2**	**4.038±0.022**	**37.9±0.7**	**53.3±0.7**

an example is incorrectly classified after being correctly classified earlier during training. **Glister** [51] uses bi-level optimization. **LCMat-S** [86] matches loss curvature between dataset and coreset. **ELFS** [109] computes training-dynamic difficulty scores from pseudo-labels obtained with the TEMI [1] deep clustering method. **CCS** [108] uses stratified sampling of difficulty scores (such as AUM or Forgetting) with intra-strata random sampling. **FAST** [25] performs frequency-domain distribution matching regularized by the graph Laplacian of a k-NN graph, a notion of topology based on graph connectivity rather than the persistent homology used by TopoPrune. **D2** [61] uses a message-passing graph network while also incorporating AUM for mislabeled samples. To isolate the selection criterion, ELFS and FAST use the same frozen encoder as the other geometric baselines (in place of the DINO or SwAV encoder used by ELFS) and receive the same mislabel filtering. Reported accuracies and standard deviations are computed over five independent training runs for the vision experiments and over three for the text experiments.

4.2 Performant and Stable Coresets with TopoPrune

Our vision experiments, detailed in Table 1, demonstrate that TopoPrune is competitive across CIFAR-10, CIFAR-100, and ImageNet-1K. Notably, this performance advantage scales with task difficulty. While competitive on simpler datasets, our method's dominance becomes most pronounced on the challenging ImageNet-1K benchmark and at extreme pruning rates (e.g., 90%). Beyond accuracy, TopoPrune exhibits notable precision gains. As evidenced by the standard deviation metrics in Table 1 and validated via statistical testing (Section D.2), our approach delivers better precision compared to extrinsic geometric baselines. For instance, on ImageNet-1K at 90% pruning, TopoPrune reduces variance by up to $6.5\times$ relative to the Euclidean graph-based D2. This precision ensures highly consistent data selection, demonstrating that our coresets faithfully represent the underlying distribution without requiring multiple costly trials. Finally, to explicitly confirm that these gains in accuracy and precision are driven fundamentally by our topological construction rather than the mislabel filtering, we provide a targeted component ablation in Section C.5.

To evaluate broader applicability, we compare TopoPrune against Random selection and geometry-based baselines, Moderate [100] and D2 [61], on two text settings at 70%, 80% and 90% pruning. (1) *Classifier fine-tuning:* A linear classification head is trained for 10 epochs on top of a RoBERTa-Large [56] or Qwen2.5-1.5B [104] backbone on ANLI [74] (3-way NLI), with embeddings for coreset construction from frozen *[CLS]* of an NLI-fine-tuned (RoBERTa-Large) model or the final-layer hidden state of the last token of a prompt-formatted premise–hypothesis pair under the base (Qwen) representations. (2) *Instruction tuning:* LoRA [48] fine-tuning of Llama-3.2-3B [40] and Pythia-2.8B [7] on Alpaca [94] (rank 8, scaling factor 16, dropout 0.05, 3 epochs) using *mean-pooled* representations. The scoring pipeline is mostly unchanged. The main difference is that Alpaca, unlike ANLI and our vision tasks, has no natural class labels, therefore we run k-means ($k = 15$, fixed a priori and not tuned) on the frozen embeddings and treat the clusters as pseudo-classes. Mislabel filtering also does not apply, so coresets are built on the full dataset. As shown in Table 2,

TopoPrune achieves the highest ANLI accuracy at every pruning rate for both backbones. The gains over the strongest baseline are modest for RoBERTa-Large (0.4 / 0.3 / 0.6 points, within one standard deviation) and larger for Qwen2.5-1.5B (1.0 / 0.7 / 2.5 points), and TopoPrune has the lowest standard deviation in five of the six settings. On Alpaca, reasoning benchmark averages are close across all methods, as most of the capability of a LoRA-tuned model at this scale is retained from pretraining. The clearest difference is on ARC-Challenge, the hardest benchmark, where TopoPrune scores highest at all three rates for both models, by 0.2 to 2.6 points. We report the full held-out perplexity, 5-shot MMLU [47], and six commonsense reasoning benchmarks via `lm-evaluation-harness` [37] in Appendix Table 13. Together, these results suggest that the topological signal remains informative on text embeddings.

4.3 Robustness to Noisy and Corrupted Representations

To evaluate the robustness of our method against representation degradation, we compare TopoPrune against geometry-based baselines, Moderate [100] and D2 [61] and evaluate three perturbation regimes on CIFAR-100. (1) *Isotropic latent noise:* Gaussian noise injected into the penultimate layer embeddings. For each sample's feature vector $\mathbf{z} \in \mathbb{R}^D$ with standard deviation $\sigma_{\mathbf{z}}$, we create a perturbed version $\mathbf{z}' = \mathbf{z} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_{\mathbf{z}})$. (2) *Input-level noise:* Pixel-level Gaussian perturbations applied to the images prior to feature extraction to observe how standard input noise propagates. (3) *Realistic image corruptions:* To test structured, non-isotropic latent distortions, we utilize the CIFAR-C benchmark [46]. We extract embeddings from images altered by four diverse corruption types (contrast, motion blur, frost, and JPEG compression) at severity level 3. Coresets are selected using these corrupted embeddings, but the resulting models are trained and evaluated on clean data. This serves to test whether topological sample importance is preserved under realistic degradation.

Table 3: **Robustness to noisy and corrupted representations at 90% pruning rate.** In this extreme compression regime, where selection sensitivity is most pronounced, TopoPrune maintains superior performance across all synthetic noise and realistic corruption types compared to geometric baselines.

Noise ($\rightarrow$)	$\mathcal{N}(0.25\sigma)$	$\mathcal{N}(\sigma)$	$\mathcal{N}(4\sigma)$	$\mathcal{N}(8\sigma)$
Moderate	33.2±0.9	33.9±0.2	32.0±1.2	32.1±1.4
D2	44.4±1.5	40.2±2.0	40.5±1.7	39.8±3.2
TopoPrune	**45.4**±0.8	**45.5**±0.6	**46.1**±0.7	**43.9**±0.4

(a) Isotropic Latent Noise

Noise ($\rightarrow$)	$\mathcal{N}(0.25\sigma)$	$\mathcal{N}(\sigma)$	$\mathcal{N}(4\sigma)$	$\mathcal{N}(8\sigma)$
Moderate	35.0±0.3	34.0±0.6	35.7±1.7	34.4±1.5
D2	40.4±1.5	43.4±0.9	41.3±2.2	42.6±2.5
TopoPrune	**40.9**±0.2	**43.5**±0.3	**41.9**±0.4	**43.3**±0.7

(b) Input-level Noise

Corr. ($\rightarrow$)	Contrast	Motion Blur	Frost	JPEG
Moderate	30.9±0.7	34.9±0.8	36.1±0.3	33.9±0.6
D2	43.2±2.2	41.3±1.2	43.2±1.4	40.9±1.1
TopoPrune	**45.0**±1.2	**42.7**±0.5	**43.2**±0.2	**43.5**±0.5

(c) Image Corruptions

As shown in Table 3, TopoPrune consistently demonstrates superior resilience across all three perturbation types at the challenging 90% pruning rate, confirming that TopoPrune identifies a high-fidelity coreset even when the underlying feature space is distorted. Detailed quantitative results for all pruning rates are provided in Table 16, Table 17, and Table 18 in the Appendix.

Implications. The preservation of topological importance under both synthetic noise and realistic, structured corruptions indicates our method is not overly dependent on a perfectly optimized source model. This suggests that effective coresets could be generated using embeddings from models that are partially trained, quantized for edge devices, or applied to slightly out-of-distribution data as similarly shown in [97].

4.4 Transferability & Unified Representations Across Architectures

We investigate the limitations of geometric metrics across a wide range of network architectures, finding that metric stability increases dramatically as we move toward topology-based metrics. (1) *Metric Stability*: First, we analyze the Euclidean distance of samples to their class prototype. This metric proves highly inconsistent across architectures, demonstrating "geometric brittleness" (Fig. 3a). The global Density Score (Fig. 3b) improves uniformity, while the local Persistence Score (Fig. 3c) aligns almost perfectly across all tested architectures. (2) *Diverse (larger) Embeddings $\rightarrow$ Fixed (smaller) Target*: Leveraging this metric stability, we evaluate transferability from diverse feature embeddings to a single target model. As detailed in Fig. 5, TopoPrune consistently yields higher accuracy and lower standard deviation across diverse proxies (ResNet, EfficientNet, Swin,

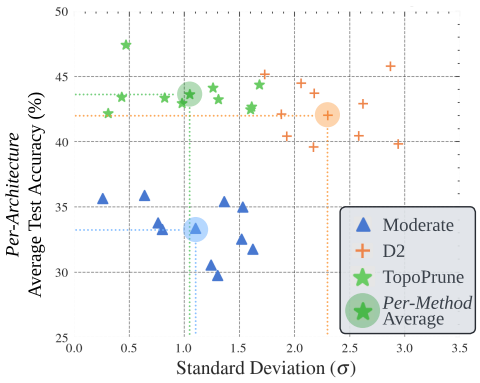


Figure 5: **Transferability of *Diverse (larger) Embeddings* → *Fixed (smaller) Target*.** TopoPrune achieves higher mean accuracy and lower standard deviation across 10 architectures at a high pruning rate of 90% for CIFAR-100, where top left (low standard deviation and high accuracy) is best.

Prun. Rate (→)	80%			90%		
CIFAR-100	Oracle	ResNet-18	Δ	Oracle	ResNet-18	Δ
ResNet-50	57.0±0.6	56.1±0.7	**-0.9**	38.7±1.4	39.6±1.6	**+0.9**
EffNet-B0	55.0±2.1	54.4±1.4	**-0.6**	39.8±1.9	39.7±3.3	**-0.1**
SwinV2-T	38.1±0.7	39.9±0.9	**+1.8**	27.6±1.2	29.6±1.5	**+2.0**
Prun. Rate (→)	80%			90%		
ImageNet-1K	Oracle	ResNet-50	Δ	Oracle	ResNet-50	Δ
EffNetV2-M	39.1±1.4	40.8±0.3	**+1.7**	35.9±0.3	37.1±1.3	**+1.2**
SwinV2-T	57.8±0.7	59.0±1.1	**+1.2**	38.3±3.4	38.0±4.1	**-0.3**
SwinV2-B	44.7±1.2	45.8±2.1	**+1.1**	44.4±2.1	45.1±1.1	**+0.7**

Table 4: **Transferability of *Fixed (smaller) Embedding* → *Diverse (larger) Targets***. We compare the "Oracle" performance (coreset selected from TopoPrune using the target model's own embeddings) against coreset selected from TopoPrune using smaller proxy model embeddings (ResNet-18 for CIFAR-100 and ResNet-50 for ImageNet-1k) to train a diverse range of models.

ViT, OpenCLIP) to train a fixed ResNet-18 target, significantly outperforming geometric baselines. Please see Appendix Table 19 for detailed values and Section A.2 for further justification. (3) *Fixed (smaller) Embedding → Diverse (larger) Targets*: Finally, we validate transferring from a static feature embedding to diverse target models. Using a standard ResNet proxy to select coresets for training EfficientNet and Swin Transformers, we find that TopoPrune achieves performance competitive with, and in some cases exceeding, "Oracle" selection where the target model selects its own coreset (see Table 4). This confirms that topological importance derived from standard proxy embeddings is highly generalizable, allowing selection of a single, high-quality coreset effective for a wide range of downstream architectures.

Implications. Recent literature spanning representational similarity [53], model stitching [54, 3], and topological divergence [4] establishes that as models scale, their latent spaces converge toward a shared statistical model of reality, formalized as the Platonic Representation Hypothesis [49]. The recent "Aristotelian" refinement [41] specifies that alignment is strictly anchored in *local neighborhood connectivity*, while global geometry remain highly variable across architectures.

TopoPrune builds on this insight. Rather than relying on brittle global geometry, we extract a topology-aware projection coupled with persistent homology (both grounded in topological connectivity rather than raw distances), the very signal the Aristotelian refinement identifies as preserved. Our metric stability analysis (Fig. 3) validates this: Euclidean-to-prototype distributions vary wildly, while our topology-aware density prior is more uniform and local persistence achieves near-perfect uniformity. This lets us access the aligned component of representations without requiring full geometric alignment, improving transferability for two regimes: (1) efficient scaling using small proxy embeddings to select data for larger targets (Table 4) and (2) transferable coresets from a single pretrained model (Fig. 5).

5 Conclusion

In this work, we address the critical challenge of instability in geometric coreset selection methods, which arises from their reliance on extrinsic metrics. We present TopoPrune, a novel framework that overcomes this "geometric brittleness" by leveraging topology to capture the data's intrinsic structure. Our dual-scale topological approach combines a global topology-aware manifold projection with a local importance score derived from differentiable persistent homology. TopoPrune exhibits several key advantages: (1) yields coresets with higher accuracy and precision, (2) is resilient to representation degradation, withstanding latent noise, input-space perturbations, realistic image corruptions and (3) is more stable across a wide range of network architectures and transfer directions. By grounding data selection in the stable invariants of manifold density and persistent homology, TopoPrune provides a principled, resilient foundation necessary for data-efficient learning.

Reproducibility Statement

We are committed to reproducible research and prioritize transparency in our work. The Methodology section (Section 3) provides a comprehensive description of our framework, including the global manifold projection, differentiable persistent homology optimization, and unified scoring formulation. The Experimental Setup subsection (Section 4.1) details the datasets, baselines, libraries (`UMAP`, `multipers`, `DeepCore`), and hardware environments used. Complete pseudocode for TopoPrune is provided in Section B.3, and all training, manifold projection, and persistent homology hyperparameters are documented in Appendix Table 20. Our code is also available at `https://github.com/ar-roy/topoprune`.

Limitations

While TopoPrune delivers strong accuracy, precision, and transferability gains, we highlight some limitations and suggest directions for future work. First, our topological backend currently lacks GPU acceleration and multi-processing support, resulting in higher wall-clock latency than non-topological baselines (Section B.2). We expect maturing topological software infrastructure to help close this gap. Also, while we provide background for the transferability of topological metrics (Section A.2), the relationship between latent-space topology and downstream training dynamics remains an open question facing the broader topological data analysis community.

Impact Statement

This work aims to reduce the cost of training deep networks by selecting smaller, representative training subsets. Its main positive impact is lower compute and energy use for training and fine-tuning, which can make experimentation more accessible to groups with limited resources. These savings are partly offset by the cost of selection itself, so the net benefit grows with the cost of the downstream training it replaces. TopoPrune produces highly consistent coresets across training runs and transfers across architectures which also makes training pipelines easier to reproduce and audit.

As with any data pruning method, removing data can change what a model learns about rare groups, as samples from underrepresented subpopulations often lie in sparse regions of the embedding. We do not test TopoPrune on subgroup populations or with fairness metrics, so practitioners applying TopoPrune to data involving people should check the composition of the selected coreset and the resulting model's performance across relevant subgroups.

Acknowledgments

This project was supported in part by the Purdue Center for Secure Microelectronics Ecosystem – CSME#210205 and the Center for the Co-Design of Cognitive Systems (CoCoSys), a DARPA-sponsored JUMP 2.0 center.

References

[1] N. Adaloglou, F. Michels, H. Kalisch, and M. Kollmann. Exploring the limits of deep image clustering using pretrained models. *British Machine Vision Conference*, 2023.

[2] À. Alonso, M. Kerber, T. Lam, and M. Lesnick. Delaunay bifiltrations of functions on point clouds. *ACM-SIAM Symposium on Discrete Algorithms*, 2024.

[3] Y. Bansal, P. Nakkiran, and B. Barak. Revisiting model stitching to compare neural representations. In *Advances in Neural Information Processing Systems*, 2021.

[4] S. Barannikov, I. Trofimov, N. Balabin, and E. Burnaev. Representation topology divergence: A method for comparing neural network representations. In *International Conference on Machine Learning*, 2022.

[5] U. Bauer, T. B. Masood, B. Giunti, G. Houry, M. Kerber, and A. Rathod. Keeping it sparse: Computing persistent homology revisited. *ArXiv*, 2022.

[6] E. Becht, L. McInnes, J. Healy, C.-A. Dutertre, I. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell. Dimensionality reduction for visualizing single-cell data using umap. *Nature Biotechnology*, 2018.

[7] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, and O. van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, 2023.

[8] T. Birdal, A. Lou, L. Guibas, and U. Simsekli. Intrinsic dimension, persistent homology and generalization in neural networks. In *Advances in Neural Information Processing Systems*, 2021.

[9] J.-D. Boissonnat and C. Maria. The simplex tree: An efficient data structure for general simplicial complexes. *Algorithmica*, 2014.

[10] Z. Borsos, M. Mutny, and A. Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in Neural Information Processing Systems*, 2020.

[11] M. B. Botnan and M. Lesnick. An introduction to multiparameter persistence. *ArXiv*, 2022.

[12] G. Carlsson and F. Mémoli. Characterization, stability and convergence of hierarchical clustering methods. *Journal of Machine Learning Research*, 2010.

[13] G. Carlsson and A. Zomorodian. The theory of multidimensional persistence. *Discrete and Computational Geometry*, 2007.

[14] M. Carrière and A. Blumberg. Multiparameter persistence image for topological machine learning. In *Advances in Neural Information Processing Systems*, 2020.

[15] M. Carrière, F. Chazal, M. Glisse, Y. Ike, H. Kannan, and Y. Umeda. Optimizing persistent homology based functions. In *International Conference on Machine Learning*, 2021.

[16] M. Carrière, M. Theveneau, and T. Lacombe. Diffeomorphic interpolation for efficient persistence-based topological optimization. In *Advances in Neural Information Processing Systems*, 2024.

[17] K. Chaudhuri and S. Dasgupta. Rates of convergence for the cluster tree. In *Advances in Neural Information Processing Systems*, 2010.

[18] F. Chazal, D. Cohen-Steiner, L. J. Guibas, F. Mémoli, and S. Y. Oudot. Gromov-hausdorff stable signatures for shapes using persistence. *Computer Graphics Forum*, 2009.

[19] F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba. Persistence-based clustering in riemannian manifolds. *Journal of the ACM*, 2013.

[20] H. Chen, W. Chen, H. Mei, Z. Liu, K. Zhou, W. Chen, W. Gu, and K.-L. Ma. Visual abstraction and exploration of multi-class scatterplots. *IEEE Transactions on Visualization & Computer Graphics*, 2014.

[21] Y. Cho, B. Shin, C. Kang, and C. Yun. Lightweight dataset pruning without full training via example difficulty and prediction uncertainty. In *International Conference on Machine Learning*, 2025.

[22] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. In *Symposium on Computational Geometry*, 2005.

[23] 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. In *International Conference on Learning Representations*, 2020.

[24] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

[25] J. Cui, B. Zhao, J. Xu, J. Guo, S. Guan, and P. Ren. Fast: Topology-aware frequency-domain distribution matching for coreset selection. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[26] C. de Bodt, A. Diaz-Papkovich, M. Bleher, K. Bunte, C. Coupette, S. Damrich, E. F. Sanmartin, F. A. Hamprecht, E.-Á. Horvát, D. Kohli, et al. Low-dimensional embeddings of high-dimensional data. *ArXiv*, 2025.

[27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[28] H. Edelsbrunner and J. L. Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.

[29] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 2002.

[30] M. Ehrgott. *Multicriteria Optimization (2. ed.)*. Springer, 2005.

[31] Y. Fan, Y. Wang, P. Zhu, D. Chen, and Q. Hu. Persistence homology distillation for semi-supervised continual learning. In *Advances in Neural Information Processing Systems*, 2024.

[32] A. Fay, I. García-Redondo, Q. Wang, H. Dubossarsky, and A. Monod. The shape of adversarial influence: Characterizing LLM latent spaces with persistent homology. In *International Conference on Learning Representations*, 2026.

[33] T. Fel, V. Boutin, L. Béthune, R. Cadene, M. Moayeri, L. Andéol, M. Chalvidal, and T. Serre. A holistic approach to unifying automatic concept extraction and concept importance estimation. In *Advances in Neural Information Processing Systems*, 2023.

[34] T. Fel, L. Béthune, A. K. Lampinen, T. Serre, and K. Hermann. Understanding visual feature reliance through the lens of complexity. In *Advances in Neural Information Processing Systems*, 2024.

[35] V. Feldman. Does learning require memorization? a short tale about a long tail. In *ACM SIGACT Symposium on Theory of Computing*, 2020.

[36] V. Feldman and C. Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 2020.

[37] L. Gao, J. Tow, B. Abbasi, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, A. Le Noac'h, H. Li, K. McDonell, N. Muennighoff, C. Ociepa, J. Phang, L. Reynolds, H. Schoelkopf, A. Skowron, L. Sutawika, E. Tang, A. Thite, B. Wang, K. Wang, and A. Zou. The language model evaluation harness. Zenodo, 2024.

[38] I. Garg and K. Roy. Samples with low loss curvature improve data efficiency. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[39] I. Garg, D. Ravikumar, and K. Roy. Memorization through the lens of curvature of loss function around samples. In *International Conference on Machine Learning*, 2024.

[40] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al. The Llama 3 herd of models. *ArXiv*, 2024.

[41] F. Gröger, S. Wen, and M. Brbić. Revisiting the platonic representation hypothesis: An aristotelian view. *International Conference on Machine Learning*, 2026.

[42] 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.

[43] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[44] M. He, S. Yang, T. Huang, and B. Zhao. Large-scale dataset pruning with dynamic uncertainty. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[45] Y. He, L. Xiao, and J. T. Zhou. You only condense once: Two rules for pruning condensed datasets. In *Advances in Neural Information Processing Systems*, 2023.

[46] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[47] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

[48] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[49] M. Huh, B. Cheung, T. Wang, and P. Isola. Position: The platonic representation hypothesis. In *International Conference on Machine Learning*, 2024.

[50] K. Killamsetty, 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*, 2021.

[51] K. Killamsetty, D. Sivasubramanian, G. Ramakrishnan, and R. Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *AAAI Conference on Artificial Intelligence*, 2021.

[52] J. Kim, J. You, D. Lee, H. Y. Kim, and J.-H. Jung. Do topological characteristics help in knowledge distillation? In *International Conference on Machine Learning*, 2024.

[53] N. Kriegeskorte, M. Mur, and P. Bandettini. Representational similarity analysis – connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2008.

[54] K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

[55] M. Lesnick and M. Wright. Computing minimal presentations and bigraded betti numbers of 2-parameter persistent homology. *SIAM Journal on Applied Algebra and Geometry*, 2022.

[56] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv*, 2019.

[57] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[58] D. Loiseaux and H. Schreiber. Multipers: Multiparameter Persistence for Machine Learning. *Journal of Open Source Software*, 9(103):6773, Nov. 2024. ISSN 2475-9066. doi: 10.21105/joss.06773.

[59] D. Loiseaux, M. Carrière, and A. Blumberg. A framework for fast and stable representations of multiparameter persistent homology decompositions. In *Advances in Neural Information Processing Systems*, 2023.

[60] D. Loiseaux, L. Scoccola, M. Carrière, M. B. Botnan, and S. Oudot. Stable vectorization of multiparameter persistent homology using signed barcodes as measures. *Advances in Neural Information Processing Systems*, 2023.

[61] A. Maharana, P. Yadav, and M. Bansal. D2 pruning: Message passing for balancing diversity & difficulty in data pruning. In *International Conference on Learning Representations*, 2024.

[62] C. Maria, P. Dlotko, V. Rouvreau, and M. Glisse. Rips complex. In *GUDHI User and Reference Manual*. GUDHI Editorial Board, 3.11.0 edition, 2025.

[63] R. T. Marler and J. Arora. The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, 2010.

[64] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *ArXiv*, 2018.

[65] L. McInnes, J. Healy, N. Saul, and L. Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.

[66] S. Mindermann, J. M. Brauner, M. T. Razzak, M. Sharma, A. Kirsch, W. Xu, B. Höltgen, A. N. Gomez, A. Morisot, S. Farquhar, et al. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, 2022.

[67] B. Mirzasoleiman, J. A. Bilmes, and J. Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, 2019.

[68] A. Mishra and F. C. Motta. Stability and machine learning applications of persistent homology using the delaunay-rips complex. *Frontiers in Applied Mathematics and Statistics*, 2023.

[69] M. Moor, M. Horn, B. Rieck, and K. Borgwardt. Topological autoencoders. In *International Conference on Machine Learning*, 2020.

[70] S. Mukherjee, S. N. Samaga, C. Xin, S. Oudot, and T. K. Dey. D-gril: End-to-end topological learning with 2-parameter persistence. *International Symposium on Computational Geometry*, 2026.

[71] M. Nagaraj, D. Ravikumar, and K. Roy. Coresets from trajectories: Selecting data via correlation of loss differences. *Transactions on Machine Learning Research*, 2025.

[72] G. Naitzat, A. Zhitnikov, and L.-H. Lim. Topology of deep neural networks. *Journal of Machine Learning Research*, 2020.

[73] A. Narayan, B. Berger, and H. Cho. Assessing single-cell transcriptomic variability through density-preserving data visualization. *Nature Biotechnology*, 2021.

[74] Y. Nie, A. Williams, E. Dinan, M. Bansal, J. Weston, and D. Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Annual Meeting of the Association for Computational Linguistics*, 2020.

[75] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 2017.

[76] M. Papillon, S. Sanborn, J. Mathe, L. Cornelis, A. Bertics, D. Buracas, H. J. Lillemark, C. Shewmake, F. Dinc, X. Pennec, and N. Miolane. Beyond euclid: an illustrated guide to modern machine learning with geometric, topological, and algebraic structures. *Machine Learning: Science and Technology*, 2025.

[77] 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*, 2021.

[78] K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1901.

[79] G. Pleiss, T. Zhang, E. Elenberg, and K. Q. Weinberger. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 2020.

[80] O. Pooladzandi, D. Davini, and B. Mirzasoleiman. Adaptive second order coresets for data-efficient machine learning. In *International Conference on Machine Learning*, 2022.

[81] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[82] C. Schuhmann, R. Beaumont, R. Vencu, C. W. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. R. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[83] L. Scoccola, S. Setlur, D. Loiseaux, M. Carrière, and S. Oudot. Differentiability and optimization of multiparameter persistent homology. In *International Conference on Machine Learning*, 2024.

[84] H. Seifert and W. Threlfall. *A textbook of topology*. Academic Press, 1980. ISBN 0-12-634850-2.

[85] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

[86] S. Shin, H. Bae, D. Shin, W. Joo, and I.-C. Moon. Loss-curvature matching for dataset selection and condensation. In *International Conference on Artificial Intelligence and Statistics*, 2023.

[87] B. Sorscher, R. Geirhos, S. Shekhar, S. Ganguli, and A. S. Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[88] S. Sun, D. Calandriello, H. Hu, A. Li, and M. K. Titsias. Information-theoretic online memory selection for continual learning. In *International Conference on Learning Representations (ICLR)*, 2022.

[89] S. Suresh, B. Das, V. Abrol, and S. D. Roy. On characterizing the evolution of embedding space of neural networks using algebraic topology. *Pattern Recognition Letters*, 2024.

[90] S. Swayamdipta, R. Schwartz, N. Lourie, Y. Wang, H. Hajishirzi, N. A. Smith, and Y. Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. *ArXiv*, 2020.

[91] H. Tan, S. Wu, F. Du, Y. Chen, Z. Wang, F. Wang, and X. Qi. Data pruning via moving-one-sample-out. In *Advances in Neural Information Processing Systems*, 2023.

[92] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019.

[93] M. Tan and Q. Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, 2021.

[94] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 2023.

[95] M. Toneva, A. Sordoni, R. T. des Combes, A. Trischler, Y. Bengio, and G. J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

[96] I. Trofimov, D. Cherniavskii, E. Tulchinskii, N. Balabin, E. Burnaev, and S. Barannikov. Learning topology-preserving data representations. In *International Conference on Learning Representations*, 2023.

[97] R. Turkes, G. F. Montufar, and N. Otter. On the effectiveness of persistent homology. *Advances in Neural Information Processing Systems*, 2022.

[98] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 2008.

[99] Y. Wang, H. Huang, C. Rudin, and Y. Shaposhnik. Understanding how dimension reduction tools work: an empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *Journal of Machine Learning Research*, 2021.

[100] X. Xia, J. Liu, J. Yu, X. Shen, B. Han, and T. Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *International Conference on Learning Representations*, 2023.

[101] W. Xiao, Y. Chen, Q. Shan, Y. Wang, and J. Su. Feature distribution matching by optimal transport for effective and robust coreset selection. *AAAI Conference on Artificial Intelligence*, 2024.

[102] T. Xie, J. Zhu, G. Ma, M. Lin, W. Chen, W. Yang, and S. Liu. Structural-entropy-based sample selection for efficient and effective learning. In *International Conference on Learning Representations*, 2025.

[103] Z. Xiong, N. Dalmasso, S. Sharma, F. Lecue, D. Magazzeni, V. K. Potluru, T. Balch, and M. Veloso. Fair wasserstein coresets. In *Advances in Neural Information Processing Systems*, 2024.

[104] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, et al. Qwen2.5 technical report. *ArXiv*, 2024.

[105] S. Yang, Z. Cao, S. Guo, R. Zhang, P. Luo, S. Zhang, and L. Nie. Mind the boundary: Coreset selection via reconstructing the decision boundary. In *International Conference on Machine Learning*, 2024.

[106] J. You, D. Kang, and J.-H. Jung. Topological alignment of shared vision-language embedding space. In *International Conference on Artificial Intelligence and Statistics*, 2026.

[107] H. Zhang, L. Zhang, Y. Zhang, and Z. Mao. Homology consistency constrained efficient tuning for vision-language models. In *Advances in Neural Information Processing Systems*, 2024.

[108] H. Zheng, R. Liu, F. Lai, and A. Prakash. Coverage-centric coreset selection for high pruning rates. In *International Conference on Learning Representations*, 2023.

[109] 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 *International Conference on Learning Representations*, 2025.

[110] Z. Zhu, Z. Dong, and Y. Liu. Detecting corrupted labels without training a model to predict. In *International Conference on Machine Learning*, 2022.

[111] A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete and Computational Geometry*, 2005.

Appendix

Table of Contents

A Conceptual Framework & Intuition **17**
- A.1 Overview of Simplicial Complexes and Persistent Homology . . . 17
- A.2 Background: On the Transferability of Topological vs. Euclidean Features . . . 19
  - A.2.1 Structural Fragility of Euclidean Centroids . . . 19
  - A.2.2 Stability Guarantees for Persistent Homology . . . 19
- A.3 Topological Connectivity and the Geometry of Sample Memorization . . . 20

B Implementation & Computational Analysis **21**
- B.1 Comparative Analysis of Manifold Projection Techniques . . . 21
  - B.1.1 Evaluation of Standard Linear and Non-linear Manifold Approximations 21
  - B.1.2 Evaluation of Deep Topological Autoencoders . . . 21
- B.2 Computational Complexity Analysis . . . 22
- B.3 TopoPrune Pseudocode . . . 24

C Ablations & Component Analyses **25**
- C.1 Interplay of Local Persistence (α) and Global Density (β) . . . 25
  - C.1.1 Multi-Objective Formulation and Metric Independence . . . 25
  - C.1.2 Qualitative Visualization across Networks . . . 25
  - C.1.3 Performance Impact of the Unified Score . . . 27
  - C.1.4 Illustrative Example of Coreset Construction . . . 27
- C.2 Differentiable Persistence Optimization Steps . . . 28
- C.3 Choice of Filtration and Homology Degree . . . 29
- C.4 Training-free Proxies of Area Under Margin (AUM) for Mislabel Detection . . . 29
- C.5 Component Isolation: The Impact of Mislabel Filtering . . . 30
- C.6 Sensitivity to UMAP Manifold Projection Hyperparameters . . . 30

D Extended Empirical Results **31**
- D.1 Extended Text Dataset Pruning Results . . . 31
- D.2 Statistical Significance of Precision Improvements . . . 31
- D.3 Roadmap of Detailed Experimental Results . . . 32
- D.4 Additional Persistence Optimization Visualizations . . . 35

A Conceptual Framework & Intuition

A.1 Overview of Simplicial Complexes and Persistent Homology

Persistent homology is used to characterize the topological variations in the shape of a finite metric space across multiple scales. At a high level, this can be described as the "birth" and "death" (persistence) of topological structures (defined by a homology group). The process begins by constructing a *simplicial complex*, a collection of points (0-simplices), edges (1-simplices), triangles (2-simplices), and their higher-dimensional counterparts that represents the data's structure [9]. To analyze how this structure changes with scale, a *filtration* is created (see Fig. 6a). This is a nested sequence of simplicial complexes, $K_{r_1} \subseteq K_{r_2} \subseteq \cdots \subseteq K_{r_n}$, indexed by a non-decreasing scale parameter r. For each complex K_r in the filtration, we can compute its *homology groups*, $H_k(K_r)$, which are vector spaces that algebraically capture its k-dimensional features.

The rank of this group, known as the k-th *Betti number* ($\beta_k = \text{rank}(H_k(K_r))$), provides a count of these features: β_0 counts connected components, β_1 counts loops or tunnels, β_2 counts voids, and so on. These values are central to understanding the distinction between an object's *extrinsic* geometry and its *intrinsic* topological properties. A classic example which illustrates this difference is that of a coffee mug and a torus (donut). These two objects are topologically equivalent because they share the same Betti numbers (see Fig. 6b). Although their extrinsic geometries (including their shape, curvatures, and distances as embedded in 3D space are very different), their intrinsic topology is

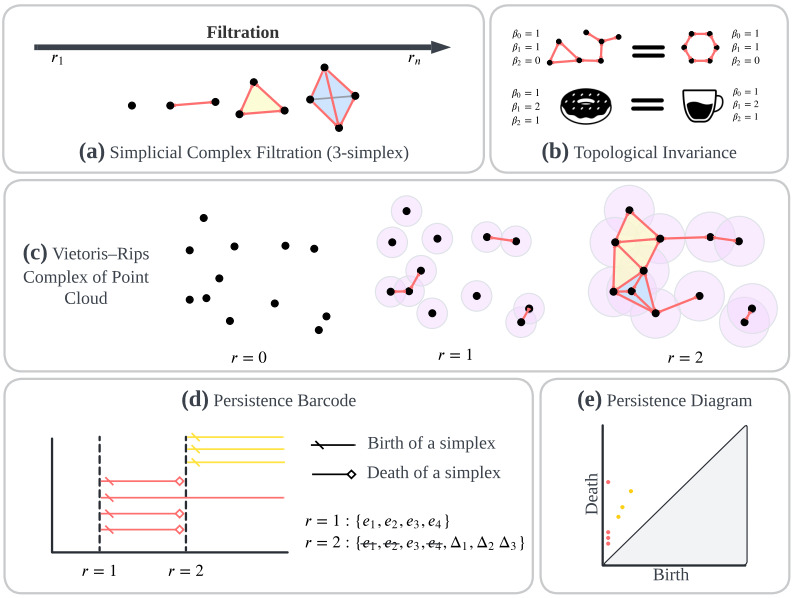


Figure 6: Overview of Simplicial Complexes and Persistent Homology

identical. This is because one can be continuously deformed into the other without tearing or gluing, preserving the single hole that defines them both.

Persistent homology is not wedded to any form of metric construction, and in fact you can do persistence on purely abstract simplicial complexes and any filtration on it. For conceptual clarity and alignment with foundational stability proofs, we illustrate this process on a point cloud $P = \{\mathbf{x}_i\}$ using the common *Vietoris-Rips (VR) complex* (see Fig. 6c). Note that while the VR complex serves as our pedagogical example in this section, the actual implementation of TopoPrune employs a Delaunay filtration to maximize computational scalability [68]. For a given scale $r \geq 0$, the complex $VR(P, r)$ contains all simplices $\sigma \subseteq P$ such that the Euclidean distance between any two points in σ is at most $2r$. As r increases, simplices are added to the complex, causing new components to merge with older components. *Persistent homology tracks the birth and death of these topological features throughout the filtration.* The inclusion map $K_{r_i} \hookrightarrow K_{r_j}$ for $r_i \leq r_j$ induces a homomorphism between the homology groups, $H_k(K_{r_i}) \to H_k(K_{r_j})$. A feature is said to be "born" at a scale r_{birth} when it first appears and "dies" at a scale r_{death} when it merges with an older feature visualized by the persistence barcode (Fig. 6d).

Definition A.1 (Vietoris-Rips Filtration)**.** *For a point cloud $P \subset \mathbb{R}^n$ and a scale parameter $r \geq 0$, the Vietoris-Rips complex $VR(P, r)$ is the simplicial complex whose vertices are the points in P and whose simplices are all finite subsets of P with a diameter of at most $2r$. A filtration is the nested sequence of complexes $\{VR(P, r)\}_{r \geq 0}$.*

The output of this process is summarized in a *persistence diagram* $\text{Dgm}(P)$, a multiset of points in the plane where each point corresponds to a single topological feature plotted at its $(\text{birth}, \text{death}) \to (b, d)$ coordinates (see Fig. 6e). The *persistence* of a feature is defined as its lifespan, $d - b$. *Points in the diagram that are further from the diagonal line $y = x$ represent robust, structurally significant features of the data*, while points close to the diagonal are interpreted as topological noise with short lifespans. This provides a stable, multi-scale signature of the data's underlying shape.

Definition A.2 (Persistence Diagram)**.** *Applying the homology functor $H_k(\cdot)$ (for a fixed dimension k, e.g., $k = 0$ for connected components) to a filtration yields a set of birth-death pairs (b, d) representing the scales at which topological features appear and disappear. This multiset of pairs is the persistence diagram, denoted $\text{Dgm}(P)$. The persistence of a feature (b, d) is defined as $d - b$.*

Please note that for clarity and ease of visualization in this overview section, we present the 1-parameter persistence analysis. It is important to note, however, that our method, TopoPrune, employs a multi-parameter persistence module, which is more complex to visualize but provides a richer description of the data's topology.

A.2 Background: On the Transferability of Topological vs. Euclidean Features

Previous literature formally demonstrates the superior transferability of topological features derived from persistent homology compared to conventional Euclidean measurements [76]. The performance gap across disparate network architectures stems from two distinct structural asymmetries: (1) Euclidean centroids are extrinsic vector-space constructions that do not commute with non-linear mappings, whereas persistent homology is computed purely from intrinsic pairwise distances; and (2) persistent homology enjoys a rigorous Lipschitz bound against metric distortion, whereas centroid-based distance rankings have no such guarantee.

Preliminaries and Notation. Let X be the input data space and $Y = \{1, \dots, K\}$ be the set of K class labels. A neural network architecture is a function $f : X \to \mathbb{R}^n$ mapping input data to an n-dimensional embedding space. Let f_A and f_B denote two distinct network architectures (e.g., ResNet18 and ViT-L-16). The outputs of these networks for the dataset X are the point clouds $X_A = f_A(X)$ and $X_B = f_B(X)$. We equip these spaces with the standard Euclidean metric, d_E.

Definition A.3 (Bottleneck Distance)**.** *The similarity between two persistence diagrams Dgm_1 and Dgm_2 is measured by the bottleneck distance $d_B(Dgm_1, Dgm_2)$, defined as the infimum over all bijections $\eta : Dgm_1 \to Dgm_2$ of the supremum of distances between matched points, where $p \in Dgm$ represents a birth-death pair $p = (b, d)$:*

$$d_B(Dgm_1, Dgm_2) = \inf_{\eta} \sup_{p \in Dgm_1} \|p - \eta(p)\|_\infty$$

Definition A.4 (Gromov-Hausdorff Distance)**.** *The distance between two metric spaces (M_1, d_1) and (M_2, d_2) is measured by the Gromov-Hausdorff distance $d_{GH}(M_1, M_2)$, which is the infimum of distances over all possible isometric embeddings into a common metric space. It bounds the maximum metric distortion required to map one space onto another.*

A.2.1 Structural Fragility of Euclidean Centroids

The fundamental limitation of Euclidean prototype (centroid) distances is that the centroid is a vector-space construction, not a metric one. For any non-linear transformation ϕ between network embedding spaces, the centroid of the mapped points is not equal to the mapped centroid of the original points: $\phi\left(\frac{1}{|X|}\sum x_i\right) \neq \frac{1}{|X|}\sum \phi(x_i)$. Consequently, under non-linear architectural shifts, the target centroid has no structural analog to the source centroid. Furthermore, even under simple affine transformations, distance rankings to the centroid are highly brittle.

Definition A.5 (Class Prototype and Distance Distribution)**.** *For an embedding $f(X)$ and a class $k \in Y$, the class prototype (centroid) is $c_k = \frac{1}{|X_k|}\sum_{x \in X_k} f(x)$. The set of distances to the prototype is $S_k(f) = \{d_E(f(x), c_k) \mid label(x) = k\}$.*

Proposition A.6 (Sensitivity to Anisotropic Distortion)**.** *Let f_A be a network embedding. Consider a new embedding f_B defined by a non-uniform anisotropic scaling transformation, $f_B(x) = \Lambda f_A(x)$, where Λ is a diagonal matrix with non-equal strictly positive entries. Under this transformation, the relative Euclidean distance rankings of samples to the class prototype can be inverted.*

Proof. Without loss of generality, let the centroid under f_A be at the origin, $c_k = (0, 0)$. Consider two samples $x_1 = (1, 0)$ and $x_2 = (0, 1.1)$. Under f_A, x_1 is strictly closer to the centroid than x_2 because $d_E(x_1, c_k) = 1 < d_E(x_2, c_k) = 1.1$.

Now apply an anisotropic scaling matrix $\Lambda = \text{diag}(2, 1)$. The new centroid remains at the origin, $c'_k = (0, 0)$. The mapped samples are $f_B(x_1) = (2, 0)$ and $f_B(x_2) = (0, 1.1)$. In the new embedding space, $d_E(f_B(x_1), c'_k) = 2$ and $d_E(f_B(x_2), c'_k) = 1.1$. Thus, $d_E(f_B(x_1), c'_k) > d_E(f_B(x_2), c'_k)$, completely inverting the relative importance ranking of the samples. □

Remark. *This illustrates that even arbitrarily small amounts of anisotropic distortion, which are unavoidable when mapping representations across different deep learning architectures, can invert sample distance rankings.*

A.2.2 Stability Guarantees for Persistent Homology

Unlike Euclidean centroids, persistent homology is constructed exclusively from intrinsic pairwise distances (e.g., via the Vietoris-Rips filtration). Because it does not rely on averaging coordinates, its transferability across embedding spaces is directly governed by established stability theorems.

Proposition A.7 (Stability of Persistent Homology [18, 22])**.** *Let X_A and X_B be two point clouds representing the same data manifold mapped into distinct embedding spaces. The bottleneck distance between their respective Vietoris-Rips persistence diagrams is bounded by the Gromov-Hausdorff distance between their metric spaces:*

$$d_B(Dgm(X_A), Dgm(X_B)) \leq 2 \cdot d_{GH}((X_A, d_E), (X_B, d_E))$$

Remark. *This provides a Lipschitz-type bound against metric distortion. The factor of 2 explicitly arises from the use of the Vietoris-Rips complex (which relies strictly on pairwise distances) as an approximation of the Čech complex. While structural distortions across varying model capacities (measured by d_{GH}) will inevitably perturb the persistence diagrams, this bounded behavior ensures that the fundamental topological features of the dataset (and thus their resulting importance scores) degrade gracefully.*

A.3 Topological Connectivity and the Geometry of Sample Memorization

Grounded in established coreset frameworks [87, 100], we utilize the intra-class manifold density as a global signal for sample prototypicality (where dense regions represent common samples and sparse regions represent rare, atypical samples). To qualitatively illustrate this, we compare our density metric against sample memorization [35, 36], measured via input curvature [39]. We observe a general correspondence: *high-density* samples consistently exhibit *low input curvature* (un-memorized), while *low-density* samples show *high input curvature* (memorized).

While intended as a qualitative illustration, both metrics successfully identify similar prototypical subsets (e.g., orange highlights in Fig. 8). However, divergent selections arise because input curvature relies on the Hessian of the loss, making it highly sensitive to training dynamics and intrinsic biases. As depicted in Fig. 7, it can capture spurious correlations such as "boys with red shirts" as the least memorized. In contrast, our density score utilizes frozen, pre-trained embeddings. This training-agnostic approach yields a robust structural prior for our fine-grained persistence optimization.

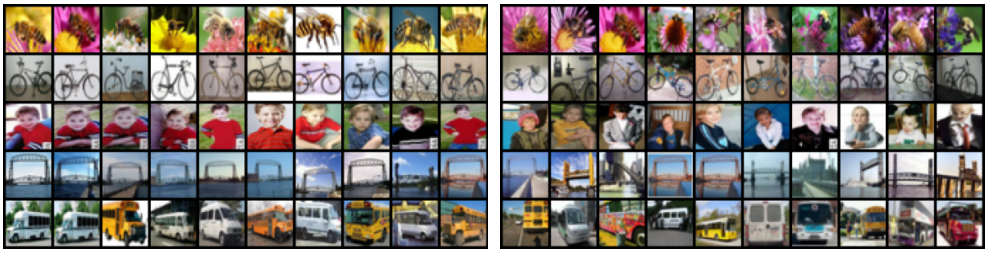


Figure 7: **Prototypical Samples:** Top-10 lowest curvature samples (left) vs. highest density samples (right) of the same class, for five CIFAR-100 classes.

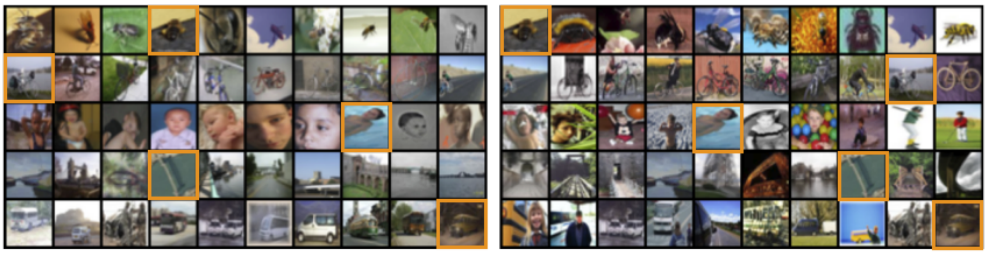


Figure 8: **Atypical Samples:** Top-10 highest curvature samples (left) vs. lowest density samples (right) of the same class, for five CIFAR-100 classes. We highlight a few examples of matched samples for each class in orange.

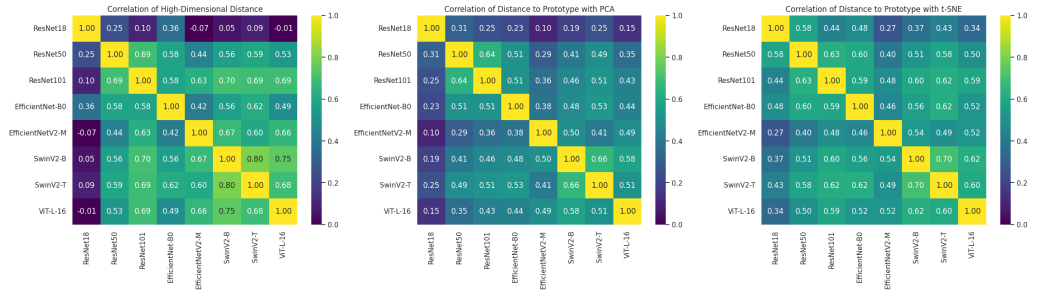
B Implementation & Computational Analysis

B.1 Comparative Analysis of Manifold Projection Techniques

Our global manifold projection is critical for achieving metric stability across diverse neural network architectures. While high-dimensional embeddings vary drastically in their extrinsic geometry and absolute cluster densities, they share a common intrinsic topology. UMAP leverages this shared structure to construct a low-dimensional manifold that explicitly normalizes the data distribution, effectively abstracting away architecture-specific scaling and artifacts. This standardization ensures that the global density score, a core component of our sample importance calculation, is a stable and reliable metric regardless of the source network.

B.1.1 Evaluation of Standard Linear and Non-linear Manifold Approximations

To further elaborate the standardization of topology-based manifold approximation and projection across perturbations in the embedding space we look at correlation (Fig. 9) of per-sample distance to prototypes across different manifold projection and feature reduction techniques (a) PCA [78], (b) t-SNE [98] (c) PaCMAP [99], (d) DensMAP [73], and (e) UMAP [64]. We see that the topology-based methods PaCMAP, DensMAP and UMAP demonstrate significantly higher correlation and thus better transferability across architectures compared to linear PCA or the more locally-focused t-SNE. Notably, UMAP exhibits strong transferability, outperforming PaCMAP. Although DensMAP yields a marginal +0.02 improvement in average correlation, UMAP provides more than sufficient structural fidelity for our framework. This high correlation between smaller (e.g., ResNet-18) and larger models is particularly valuable, as it confirms that computationally inexpensive networks can reliably generate embeddings to guide data selection for much larger models.



(a) Euclidean Distance to Prototype (b) PCA (c) t-SNE

(d) PaCMAP (e) DensMAP (f) UMAP

Figure 9: Correlation of per-sample distance to prototype across different architectures when applying different linear and non-linear manifold projection techniques.

B.1.2 Evaluation of Deep Topological Autoencoders

We also considered Topological Auto-Encoders (TopoAE) [69] and Representation Topological Divergence (RTD) [96] for our manifold embedding. While these methods are powerful for preserving global topology, we chose UMAP for computational efficiency, domain suitability, and alignment with coreset selection goals.

Table 5: **Throughput and accuracy when using Topological autoencoders. (a)** Topological autoencoders require costly training vs. UMAP's algorithmic projection. **(b)** Substituting UMAP with TopoAE or RTD for the global manifold embedding shows performance degradation.

Method	CIFAR-10	CIFAR-100
TopoAE [69]	14,847.79	15,248.11
RTD [96]	26,622.89	28,661.29
UMAP [64]	**22.42**	**22.74**

(a) Latency (s)

Method	CIFAR-10	CIFAR-100
TopoAE [69]	75.0±0.3	41.2±1.0
RTD [96]	78.0±1.7	**46.4±0.4**
UMAP [64]	**82.1±0.3**	45.8±0.7

(b) Accuracy at 90% Pruning Rate (Avg. over 3 runs)

TopoPrune aims to be a lightweight, training-free method applicable to frozen features, a requirement UMAP fits perfectly. In contrast, training a topological autoencoder is computationally prohibitive for preprocessing; training RTD on CIFAR-100 takes ∼8 hours compared to UMAP's ∼22 seconds, representing a speedup of $> 1000\times$ (see Table 5a). To empirically validate our choice, we trained both TopoAE and RTD models on CIFAR-10/100 and used their embeddings as a replacement for UMAP in our pipeline. As shown in Table 5b, UMAP consistently yields superior or comparable accuracy without this massive training overhead.

Domain suitability and structural alignment. The observed drop in coreset accuracy when using topological autoencoders likely stems from a fundamental misalignment of objectives. First, regarding latent space quality, TopoAE struggles to produce clean, separated representations for complex datasets. As noted in the TopoAE paper itself, CIFAR-10 is "challenging to embed... in a purely unsupervised manner" (Section 5.2.2 in Moor et al. [69]), often resulting in latent spaces where classes are homogeneously mixed rather than cleanly separated (see Figure 4 in Moor et al. [69]). This lack of separation severely hampers the effectiveness of our per-class density estimation, contrasting with the distinct cluster delineation achieved by UMAP. Furthermore, this issue is exacerbated because methods like TopoAE and RTD prioritize preserving global structural similarity (e.g., maintaining relative distances between distinct mammoth "head" and "foot" clusters as shown in Figure 1 in Trofimov et al. [96]). While this global constraint is valuable for visualization, it is less relevant for coreset selection, where we partition the data into class-based manifolds.

In summary, while topological autoencoders are robust tools for manifold learning, UMAP provides a more efficient, domain-appropriate, and higher-performing foundation for our specific multi-scale framework. We encourage future work exploring topological autoencoders that flexibly balance global and local structural priorities, specifically optimized for the task of point-cloud sparsification.

B.2 Computational Complexity Analysis

Theoretical Complexity Relative to Geometric Baselines. Our analysis follows the framework for evaluation of coreset selection complexity in Nagaraj et al. [71]. The core computational overhead of TopoPrune stems from the local topological optimization. However, we maintain tractability by leveraging efficient reductions of multi-parameter persistence:

- Crucially, we utilize the Hilbert decomposition signed measure, which *reduces the multi-parameter problem to one-parameter persistence slices along a grid.* As detailed in Appendix D.1 of Loiseaux et al. [60], for a 2-parameter filtration (Delaunay + Density) on a grid of size m, the algorithm performs m runs of a 1-parameter persistence optimization.
- Consequently, the persistent homology optimization cost is equivalent to that of a 1-parameter optimization on N_c points. While the theoretical worst-case for persistence is cubic [55], in the 1-parameter persistence case the computation is empirically linear [5].
- Furthermore, computing the gradient of the loss $\mathcal{L}_{\text{pers}}$ (Eq. (3)) simplifies to summing feature persistences (see Corollary E.2 in Scoccola et al. [83]). This operation is bounded by a constant K derived from the simplicial complex $\mathcal{K}$, making the backward pass $\mathcal{O}(K)$. Thus, a single local optimization cost is strictly $\mathcal{O}(m \cdot N_c \log N_c)$.

By applying this efficient reduction strategy per-class (where N_c is relatively small), TopoPrune scales far more efficiently in high-dimensional spaces than current geometric baselines. For instance, while D2 [61] reports a graph construction complexity of $\mathcal{O}(Nkd)$, this accounts only for edge weighting. Their official implementation relies on exact k-NN search without approximation (`sklearn.neighbors.kneighbors_graph`) and dense matrix allocation (`todense()`), which fundamentally scale unfeasibly at $\mathcal{O}(N^2 d)$ in time and $\mathcal{O}(N^2)$ in space for high-dimensional data. In

contrast, TopoPrune avoids this geometric memory bottleneck entirely by utilizing approximate nearest neighbor descent via UMAP, scaling efficiently at $\mathcal{O}(N \log N)$.

Table 6: **Complexity analysis of geometry-based methods.** N is the dataset size with C classes, $N_c \approx N/C$ samples per class, d dimension, and k neighbors. For TopoPrune, cost is dominated by T optimization steps and m grid resolution.

	Computational Complexity	**Explanation**
Moderate [100]	$\mathcal{O}(Nd) + C * \mathcal{O}(N_c \log N_c)$	Distance calc. (Nd) + Prototype sorting
D2 [61]	$\mathcal{O}(Nkd) + \mathcal{O}(T \cdot Nk)$	kNN graph (Nkd) + Message passing
TopoPrune	$\mathcal{O}(N \log N) + C * \mathcal{O}(Tm \cdot N_c \log N_c)$	Global UMAP + Local persistence

Empirical Wall-Clock Analysis. We benchmarked the latency and resource utilization of TopoPrune against baselines on an AMD EPYC 7502 (32-Core) CPU. Despite exhibiting higher absolute latency, our profiling reveals that TopoPrune significantly under-utilizes available hardware (16% CPU utilization vs. 82% for baselines). This indicates an implementation-specific bottleneck rather than a fundamental algorithmic flaw; our topological backend (`multipers`) currently lacks support for multi-processing and GPU acceleration. As topological software infrastructure matures to support parallelization, we expect this wall-clock gap to close significantly, offering the superior stability of topological methods with negligible latency trade-offs. Still, TopoPrune offers a positive cost-benefit for high-stakes data selection:

- TopoPrune is orders of magnitude faster than "training-dynamic" coreset methods (e.g., Glister, Forgetting), which require training a proxy model from scratch (hours of compute) compared to our topological probe of frozen embeddings (minutes).
- Unlike fast geometric heuristics (e.g., D2, Moderate) which suffer from high variance and lower precision (as shown in Table 1), TopoPrune accepts a marginally higher upfront computational cost to guarantee a precise, high-fidelity coreset. This structural reliability eliminates the need for repeated, redundant selection runs to mitigate randomness, ultimately yielding a highly efficient and trustworthy pipeline.

Table 7: **Latency and utilization** when performing selection on CIFAR-100. To ensure a fair comparison between single and multi-threaded implementations, latency is normalized by CPU utilization.

	Global (s)	**Local (s)**	**Total (s)**	**Max CPU Util.**
D2	-	-	86.05	82%
TopoPrune (Vietoris-Rips Complex)	22.74	280.78	303.52	16%
TopoPrune (Delaunay Complex)	22.74	146.16	168.90	16%

B.3 TopoPrune Pseudocode

Algorithm 1 Coreset Selection with `TopoPrune`

Input: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, Penultimate Layer Encoder h_θ, Pruning Rate p, Mislabel Ratio γ, Weights α, β, Persistence Optimization Steps T
Callables: `UMAP`, `KDE`, `KNN` ▷ see Hyperparameters in Table 20
Output: Selected Indices $\mathcal{S}$

 # Dual-Scale Topological Scoring

Extract embeddings: $Z \leftarrow h_\theta(X)$
Global manifold projection: $Y \leftarrow \texttt{UMAP}(Z)$
Initialize score vectors $S_{pers}, S_{dens} \leftarrow \mathbf{0}^N$
for each class $c \in \{1, \ldots, C\}$ **do**
  Get indices $\mathcal{I}_c \leftarrow \{i \mid y_i = c\}$ and subset $Y_c \leftarrow Y[\mathcal{I}_c]$
  $S_{dens}[\mathcal{I}_c] \leftarrow \texttt{KDE}(Y_c)$ ▷ Global Density Score
  Initialize optimizable inputs $Y_c' \leftarrow Y_c$
  for $t = 1$ **to** T **do**
    Compute Hilbert decomposition signed measure: $\mu^{Hil}_{H_1(Del_{Y_c'}, \hat{f})}$
    Compute topological loss: $\mathcal{L}_{\text{pers}}(Y_c') \leftarrow \text{OT}(\mu^{Hil}_{H_1(Del_{Y_c'}, \hat{f})}, \mathbf{0})$
    Update coordinates: $Y_c' \leftarrow Y_c' + \eta \nabla_{Y_c'} \mathcal{L}_{pers}$
  end for
  $S_{pers}[i] \leftarrow \|\mathbf{y}_i - \mathbf{y}_i'\|_2 \quad \forall i \in \mathcal{I}_c$ ▷ Local Persistence Score
end for

 # Mislabel Detection and Filtering

if method is NLPS **then**
  for each sample $i \in \{1, \ldots, N\}$ **do**
    Compute k nearest neighbors: $\mathcal{N}_k(z_i) \leftarrow \texttt{KNN}(Z)$
    Count mismatched neighbors: $m_i \leftarrow \sum_{j \in \mathcal{N}_k(z_i)} \mathbb{I}(y_j \neq y_i)$
    Compute label purity ratio: $S^{(i)}_{mis} \leftarrow m_i / k$ ▷ Training-Free
  end for
else if method is AUM **then**
  Grab precomputed score: $S_{mis} \leftarrow \text{AUM}(Z)$ ▷ with Training Dynamics
end if
Identify mislabeled sample indices: $\mathcal{I}_{mis} \leftarrow \text{TopK}(S_{mis}, \gamma \cdot N)$
Define clean candidate set: $\mathcal{D}_{clean} \leftarrow \mathcal{D} \setminus \mathcal{I}_{mis}$

 # Stratified Sampling on Unified Score

$S^{(i)}_{unified} \leftarrow \alpha \cdot \mathcal{N}(S^{(i)}_{pers}) + \beta \cdot \mathcal{N}(S^{(i)}_{dens})$ ▷ Min-max normalization $\mathcal{N}$
$\mathcal{S} \leftarrow \text{StratifiedSample}(\mathcal{D}_{clean}, S_{unified}, p)$ ▷ see Algorithm 1 in Zheng et al. [108]
return $\mathcal{S}$

C Ablations & Component Analyses

C.1 Interplay of Local Persistence (α) and Global Density (β)

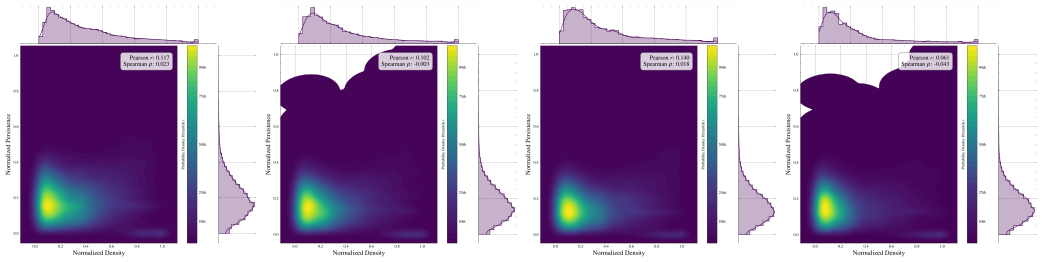
As discussed in Section 3.3, relying exclusively on either global density or local persistence is insufficient for optimal coreset selection at high compression rates. In this section, we provide the extended empirical analysis validating the orthogonal nature of these metrics and demonstrating why their unified formulation (Eq. (5)) is critical for model optimization.

C.1.1 Multi-Objective Formulation and Metric Independence

Formally, our unified metric functions as a weighted sum-scalarization [30, 63] of two complementary objectives: global representativeness (density) and local structural informativeness (persistence). For this scalarization to yield a non-trivial Pareto front, the underlying objectives must be non-redundant. To empirically verify this independence, we conducted a joint distribution analysis by extracting latent representations from 8 diverse network architectures evaluated on CIFAR-100. For each architecture, we computed both scores for all training samples and calculated the Pearson correlation coefficient (r). As detailed in Table 8 and Fig. 10, the linear correlation is consistently minimal across all architectures, yielding an average of $r = 0.102$. This near-zero correlation quantitatively confirms that global manifold density and local persistence do not redundantly encode the same feature space properties. Consequently, combining them with strictly positive weights ($\alpha, \beta > 0$) guarantees a Pareto optimal coreset selection that strictly outperforms the weakly Pareto optimal endpoints of using either metric in isolation.

Table 8: Pearson correlation (r) between density and persistence scores.

	RN-18	RN-50	RN-101	EffNet-B0	EffNetV2-M	SwinV2-T	SwinV2-B	ViT-L-16	**Avg.**
r	0.117	0.106	0.076	0.102	0.110	0.140	0.104	0.061	**0.102**



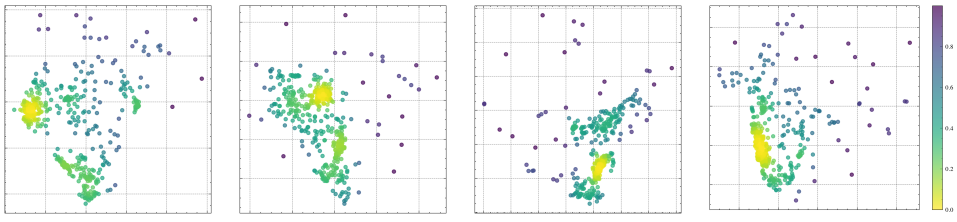
(a) ResNet-18 (b) EfficientNet-B0 (c) SwinV2-T (d) ViT-L-16

Figure 10: Joint probability distributions of persistence and density scores across networks. As the joint distributions are similar across architectures (all long tailed) we provide visualizations for just one architecture per network family.

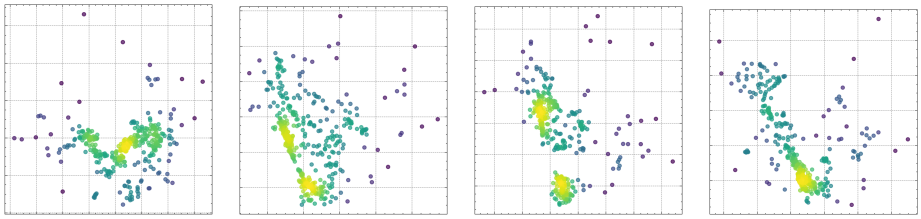
C.1.2 Qualitative Visualization across Networks

To illustrate the importance of fine-grained local neighborhood structure, we visualize the global density (KDE), local persistence, and our unified score of a single target class. This qualitative comparison demonstrates how combining these metrics effectively resolves the redundancy problem inherent among densely co-located samples.

- *Density (Fig. 11):* Exhibits smooth, continuous gradients. Samples situated tightly together in the dense core of the cluster share nearly identical scores. If a coreset algorithm samples purely based on high density, it will select highly redundant points, starving the model of boundary information.
- *Persistence (Fig. 12):* Exhibits a highly localized, non-continuous distribution. Within the high-density core, persistence highlights specific structural "anchors" while assigning low scores to immediately adjacent, redundant neighbors.
- *Unified Score (Fig. 13):* Effectively overlays the fine-grained structural map onto the global distributional map.

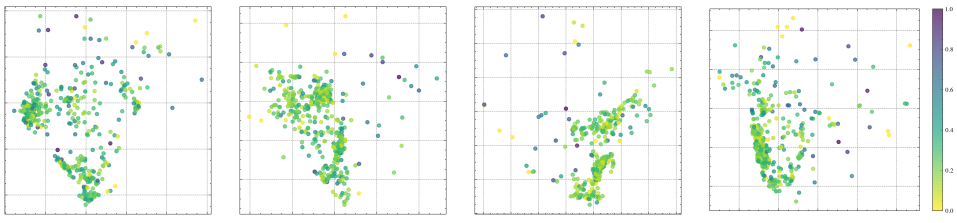


(a) ResNet-18 (b) ResNet-50 (c) ResNet-101 (d) EfficientNet-B0

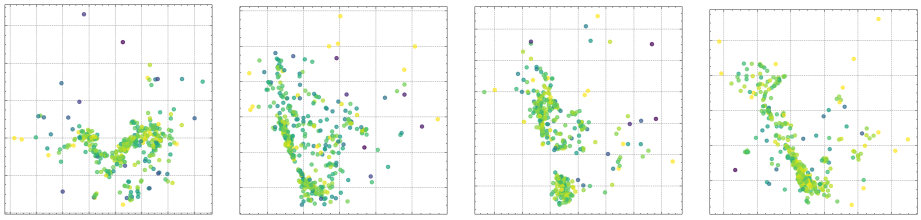


(e) EffNetV2-M (f) SwinV2-T (g) SwinV2-B (h) ViT-L-16

Figure 11: $\text{Score}_{\text{dens}}$ on Cifar-100 (butterfly class; label 14). Score [0, 1]: (0: Yellow indicated high density, 1: blue indicated low density).

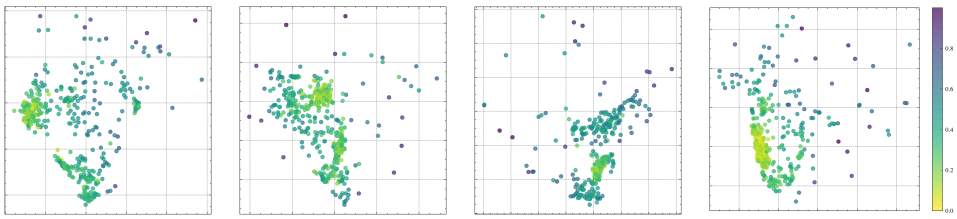


(a) ResNet-18 (b) ResNet-50 (c) ResNet-101 (d) EfficientNet-B0

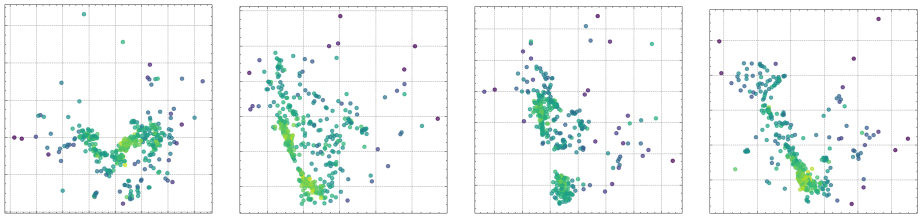


(e) EffNetV2-M (f) SwinV2-T (g) SwinV2-B (h) ViT-L-16

Figure 12: $\text{Score}_{\text{pers}}$ on Cifar-100 (butterfly class; label 14) with 6 optimization steps. Score [0, 1]: (0: Yellow = low persistence score (e.g., small optim. distance); 1: blue = higher persistence score (e.g., higher optim. distance).



(a) ResNet-18 (b) ResNet-50 (c) ResNet-101 (d) EfficientNet-B0

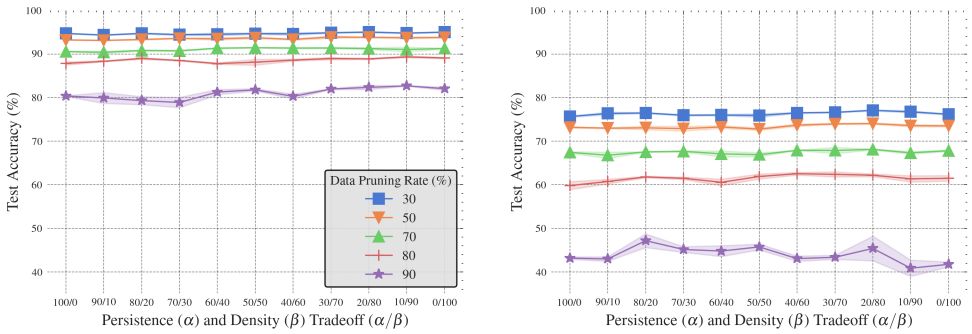


(e) EffNetV2-M (f) SwinV2-T (g) SwinV2-B (h) ViT-L-16

Figure 13: $\text{Score}_{\text{unified}}$ on Cifar-100 (butterfly class; label 14). Setting $\alpha = 0.5, \beta = 0.5$.

C.1.3 Performance Impact of the Unified Score

We investigate the performance impact of the hyperparameters α and β from Eq. (5), which balance the influence of our global density and local persistence scores (see Fig. 14). Our analysis reveals that while the coreset quality is generally stable across a range of (α, β) values, a combination of both metrics consistently yields the best performance. Although using either density or persistence alone provides a reasonable baseline, combining them is particularly crucial at high pruning rates (e.g., 90%), where a balanced score improves accuracy by up to 5.4% over using either metric in isolation. This demonstrates that both global and local topology are vital for optimal selection and justifies our use of a fixed and balanced configuration set at $(50/50)$ across all experiments, minimizing the need for extensive hyperparameter tuning.



(a) CIFAR-10 (b) CIFAR-100

Figure 14: α and β sweep across data pruning rates for both **(a)** CIFAR-10 and **(b)** CIFAR-100.

C.1.4 Illustrative Example of Coreset Construction

To provide a complete picture, we visualize TopoPrune at various rates (70%, 50%, 30%, 20% and 10%) for the "butterfly" class in CIFAR-100 (see Fig. 15). The visualization reveals a high variance in Persistence Scores within localized regions of the class manifold, demonstrating the method's sensitivity to fine-grained local structures and its ability to distinguish between nearby samples. Despite this focus on local complexity, the final coresets remain density-preserving, with their overall distribution closely matching that of the full dataset. This illustrates how TopoPrune successfully balances the selection of topologically critical local samples with the preservation of the global data structure.

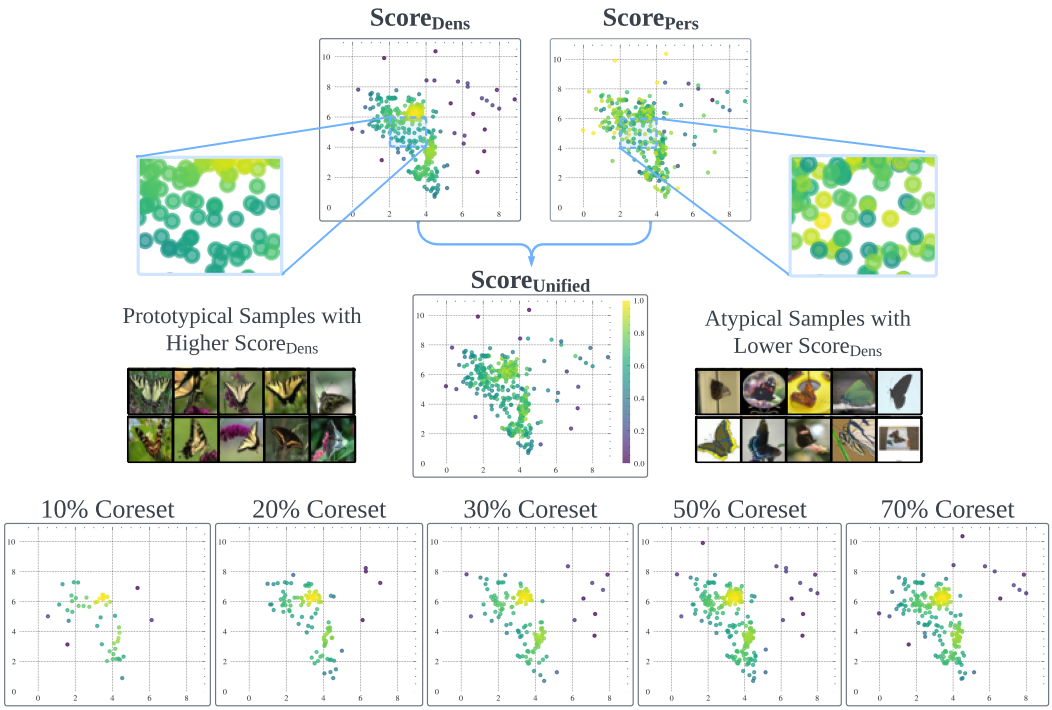
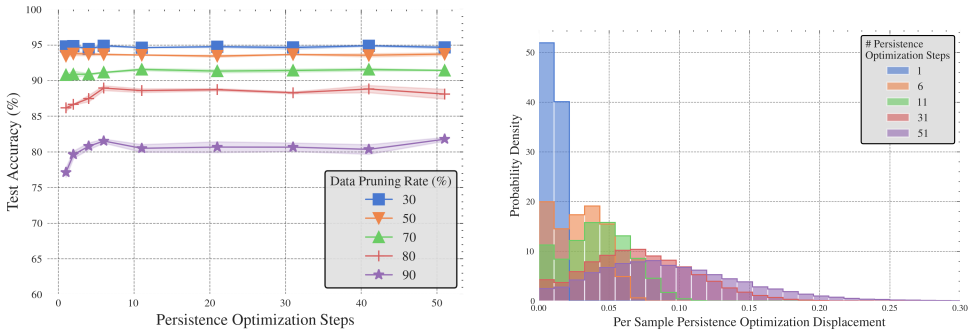


Figure 15: Qualitative analysis of TopoPrune on the CIFAR-100 "butterfly" class. **(Top)** A comparison of scoring components. In highly populated regions (zoomed in view), Kernel Density Estimation saturates uniformly; however, local Persistence effectively resolves the fine-grained topological structure within the local neighborhood to isolate structural anchors. **(Middle)** The unified TopoPrune score successfully separates prototypical samples (high score) from atypical variants (low score). **(Bottom)** The spatial distribution of selected samples across varying coreset sizes (10% to 70%) demonstrates how this dual-scale topological approach ensures comprehensive coverage of the underlying data manifold at extreme pruning rates.

C.2 Differentiable Persistence Optimization Steps

We investigate the impact of the number of optimization steps for multi-parameter persistent homology (see Fig. 16). The number of required persistence optimization steps is inversely correlated with the final coreset size. When selecting a large coreset (e.g., at a 30% pruning rate), the selection process is robust, and even a few optimization steps (1-2) suffice to identify a high-quality subset. However, at high pruning rates (e.g., 90%), the task of distinguishing the most crucial samples becomes more sensitive, necessitating a greater number of optimization steps ($\geq$6) to allow the point positions to converge and accurately reveal the most structurally important examples.



(a) Accuracy on CIFAR-10 (b) Distribution of Per-Sample Displacement

Figure 16: Higher compression rates magnify sample importance, necessitating more optimization steps for precise selection. Conversely, larger coresets are more forgiving and require fewer steps.

C.3 Choice of Filtration and Homology Degree

We ablate the density parameter of the bifiltration and the homology degree of our persistence objective (Eq. (3)) on CIFAR-100, holding every other component fixed (Table 9). We fix the global embedding to $d = 2$ (Appendix Table 20) because multi-parameter persistent homology scales poorly with dimension. The Delaunay complex is therefore planar, so only H_0 and H_1 exist.

The density parameter is crucial. Without it, the one-parameter H_1 variant (40.4±2.4 at 90% pruning) falls *below* density-only selection (41.8±0.6, Table 15), so unconstrained persistence maximization is actively harmful. Recomputing $\hat{f}$ at each epoch keeps Y_c' faithful to the original density landscape. Under a distance-only filtration, points drift across density levels, and displacement conflates structural importance with movement through the density landscape.

H_0 versus H_1. Degree-0 persistence of a point cloud is equivalent to single-linkage clustering [12, 28], and over a density parameter [2] it recovers the cluster tree of the density [19, 17], which our Density Score already estimates. Accordingly, H_0 outperforms H_1 only without explicit density (42.9 vs. 40.4 at 90%), and combining both degrees helps slightly (43.2). Once density is supplied, H_0 is redundant and trails H_1 (44.4 vs. 45.7). Variance follows the same ordering (±2.4, ±1.3, ±0.5), and TopoPrune is the best configuration at every pruning rate.

Table 9: **Ablation of filtration and homology degree** on CIFAR-100. Rows are ordered by accuracy at 90% pruning, and TopoPrune values are from Table 1. The explicit density parameter (KDE) is crucial, and H_1 is the best homology degree once density is present, with the tightest variance.

	Filtration	Homology	50%	70%	80%	90%
One-parameter PH	Del	H_1	72.1±0.2	66.4±0.7	61.1±1.0	40.4±2.4
One-parameter PH	Del	H_0	72.8±0.2	67.3±0.3	61.6±0.5	42.9±0.9
$H_0 + H_1$	$Del_{H_0} + Del_{H_1}$	H_0 & H_1	70.5±0.9	67.4±0.3	61.3±0.2	43.2±0.6
H_0 variant	$Del + KDE$	H_0	71.5±0.5	67.2±0.3	61.5±0.7	44.4±1.3
TopoPrune	$Del + KDE$	H_1	**73.8±0.3**	**68.1±0.3**	**62.3±0.4**	**45.7±0.5**

C.4 Training-free Proxies of Area Under Margin (AUM) for Mislabel Detection

We evaluate several training-free methods to serve as a proxy for Area Under the Margin (AUM) in Table 10. These proxies identify potentially noisy samples using different geometric criteria, ranging from culling samples based on their *Distance* to the class prototype, to using an *Adjacent Distance* ratio to remove points closer to another class's prototype. Other heuristics include culling samples with the lowest *Density* score, or using our proposed *Neighborhood Label Purity Score (NLPS)*, which identifies points in mixed-label regions by calculating the fraction of same-label nearest neighbors (from 20 nearest neighbors). Our results show that NLPS provides the highest coreset accuracy among all training-free proxies, with its advantage being most pronounced at high data pruning rates. While it does not fully match the performance of using the true AUM, NLPS serves as a simple and effective training-free proxy.

Table 10: **Training-free proxies for Area Under Margin (AUM)** [79] on CIFAR-100. We see that Neighborhood Label Purity Score (NLPS) performs closest to AUM.

Pruning Rate (→)	30%	50%	70%	80%	90%
Distance	75.4±0.4	71.3±0.3	63.2±0.1	56.1±0.6	37.8±0.3
Adjacent Distance	75.5±0.3	71.8±0.2	62.8±1.0	57.1±0.3	38.1±1.5
Density	75.6±0.1	71.4±0.2	64.3±0.3	52.4±0.3	38.0±0.7
NLPS	**75.6±0.2**	**71.9±0.2**	**65.3±0.4**	**56.7±0.4**	**41.6±1.0**
AUM	75.9±0.4	72.8±0.3	66.9±0.5	61.9±0.6	45.7±0.7

C.5 Component Isolation: The Impact of Mislabel Filtering

To confirm our performance gains are driven by topological scoring rather than initial data sanitization, we isolate the effect of filtering on TopoPrune and geometric baseline Moderate [100]. We evaluate both on CIFAR-100 (our noisiest benchmark) across three regimes: no filtering, training-free (NLPS), and training-dynamic (AUM). TopoPrune outperforms Moderate at every filtering stage (Table 11). This confirms the topological construction provides a fundamental improvement independent of mislabel filtering, remaining superior even without filtering at moderate pruning rates (50%–80%).

Notably, the performance gap widens under explicit filtering, particularly at extreme (90%) pruning. This exposes a structural limitation of Moderate: it restricts sampling to the interquartile range of the distance-to-prototype metric. This rigid cutoff acts as an implicit, naive filter that blindly discards outliers. Thus, applying explicit filters (NLPS/AUM) on top of Moderate yields diminishing returns, as it inherently ignores the regions containing most noisy samples. Conversely, TopoPrune scores the entire data manifold without arbitrary cutoffs. This allows it to fully capitalize on the cleaned distribution, driving the expanding performance gap.

Table 11: **Component isolation of mislabel filtering.** Evaluating TopoPrune and Moderate across un-filtered, NLPS-filtered, and AUM-filtered CIFAR-100 distributions. TopoPrune consistently outperforms Moderate, with the performance gap expanding significantly at extreme compression rates (90%) under explicit filtering.

	Pruning Rate (→)	50%	70%	80%	90%
No Filtering	Moderate [100]	70.1±0.3	63.7±0.2	56.1±0.5	34.9±2.1
	TopoPrune	71.4±0.2	64.6±0.3	56.4±0.3	35.5±1.0
	Acc. Δ	+1.3	+0.9	+0.3	+0.6
with NLPS	Moderate [100]	70.0±0.4	64.0±0.1	56.2±0.5	39.9±0.9
	TopoPrune	71.9±0.2	65.3±0.4	56.7±0.4	41.6±1.0
	Acc. Δ	+1.9	+1.3	+0.5	+1.7
with AUM	Moderate [100]	72.4±0.2	66.7±0.3	60.2±0.8	40.0±1.2
	TopoPrune	72.8±0.3	66.9±0.5	61.9±0.6	45.7±0.7
	Acc. Δ	+0.4	+0.2	+1.7	+5.7

C.6 Sensitivity to UMAP Manifold Projection Hyperparameters

To establish the robustness of the global manifold embedding stage of TopoPrune, we conducted a comprehensive hyperparameter sweep of UMAP. We specifically investigate two critical parameters: (1) *n_neighbors*, which controls the balance between local and global geometric preservation, and (2) *min_dist*, which governs how tightly samples are packed in the low-dimensional manifold. The results in Table 12 indicate that our selection of *n_neighbors=15* and *min_dist=0.1* provides a strong balance of high accuracy and low variance.

Table 12: **TopoPrune sensitivity to UMAP hyperparameters. (a)** accuracy on CIFAR-100 (90% pruning rate) and **(b)** relative change to default configuration when modulating *n_neighbors* and *min_dist* UMAP hyperparameters.

	n_neighbors				
min_dist	5	15	50	100	Row Avg.
0.05	43.9±2.2	43.0±0.9	44.4±1.7	44.9±1.6	44.1±1.6
0.1	43.1±1.8	**45.8±0.7**	45.7±0.8	43.6±0.7	44.6±0.9
0.5	46.2±1.0	43.8±0.7	44.3±2.2	43.9±0.4	44.6±1.0
1.0	44.3±0.9	45.4±0.6	45.0±1.1	43.1±1.3	44.4±1.0
Col Avg.	44.4±1.5	44.5±0.7	44.8±1.4	43.9±1.0	-

(a) Accuracy (over 5 runs)

	n_neighbors			
min_dist	5	15	50	100
0.05	-1.9	-2.8	-1.4	-0.9
0.1	-2.7	★	-0.1	-2.2
0.5	+0.4	-2.0	-1.5	-1.9
1.0	-1.5	-0.4	-0.8	-2.7

(b) Accuracy Δ

D Extended Empirical Results

D.1 Extended Text Dataset Pruning Results

We provide the per-benchmark breakdown for the instruction tuning setting of Section 4.2 in Table 13. Reasoning benchmark averages are close across all methods, as most of the capability is retained from pretraining when LoRA-tuning on 52K instructions, so data selection shifts instruction-following more than knowledge and these benchmarks largely saturate.

Table 13: **Per-benchmark results for instruction tuning on Alpaca** (mean ± std over 3 seeds). Held-out perplexity (PPL), 5-shot MMLU and commonsense reasoning (ArcE/ArcC: ARC-Easy/Challenge, BQ: OpenBookQA, WG: WinoGrande). We use 25-shot ArcC, 5-shot WG and 0-shot for all others. Avg. is the mean over all accuracy columns, with its std computed over the per-seed averages.

Pruning Rate	Method	PPL ↓	MMLU	ArcE	ArcC	PIQA	SciQ	BQ	WG	Avg.
70%	Random	3.029±0.001	**55.1±0.2**	72.3±1.1	50.7±0.3	77.3±0.9	91.1±0.3	42.9±2.0	**72.3±0.2**	66.0±0.4
	Moderate	3.041±0.004	54.7±0.5	72.6±1.0	51.1±0.7	77.6±0.1	91.4±0.4	42.6±0.2	70.6±1.1	65.8±0.3
	D2	3.079±0.001	53.2±0.8	**75.5±0.6**	51.2±0.8	**77.9±0.2**	**91.7±0.3**	**43.8±0.5**	70.1±0.7	66.2±0.2
	TopoPrune	**3.029±0.001**	53.4±0.5	73.8±0.3	**53.8±0.5**	77.7±0.1	91.2±0.6	42.1±1.0	71.7±0.3	**66.2±0.1**
80%	Random	3.039±0.002	54.9±0.6	74.0±0.6	51.7±0.1	77.5±0.3	91.9±0.6	42.3±0.1	71.2±1.0	66.2±0.2
	Moderate	3.048±0.001	**55.8±0.3**	71.6±0.6	51.8±0.6	78.2±0.3	90.9±0.1	41.9±0.4	70.9±0.5	65.9±0.1
	D2	3.048±0.006	54.4±0.6	**75.7±0.4**	51.3±1.0	77.9±0.2	**92.3±0.3**	42.7±0.9	**71.6±0.5**	**66.6±0.2**
	TopoPrune	**3.036±0.001**	54.7±0.8	73.9±0.8	**52.4±0.4**	**78.2±0.2**	91.5±0.3	**42.7±0.2**	71.5±0.7	66.4±0.2
90%	Random	3.055±0.001	55.4±0.2	73.9±0.7	50.6±0.9	78.3±0.4	92.3±0.5	43.1±1.0	71.8±0.8	66.5±0.3
	Moderate	3.060±0.002	**56.0±0.2**	73.7±1.1	52.1±1.2	78.0±0.3	**92.6±0.3**	42.2±1.5	71.9±0.5	66.6±0.5
	D2	**3.051±0.003**	55.4±0.3	**74.0±0.3**	51.5±1.1	77.6±0.1	92.1±0.4	42.9±0.4	71.9±1.5	66.5±0.4
	TopoPrune	3.053±0.006	55.7±0.4	72.7±0.6	**52.6±0.5**	**78.4±0.2**	91.5±0.3	**43.5±0.8**	**72.0±0.4**	**66.6±0.2**

(a) Llama-3.2-3B

Pruning Rate	Method	PPL ↓	MMLU	ArcE	ArcC	PIQA	SciQ	BQ	WG	Avg.
70%	Random	**3.953±0.015**	25.9±0.5	54.0±1.1	37.6±0.5	**74.3±0.5**	75.0±0.4	37.1±0.7	**60.9±1.1**	52.1±0.2
	Moderate	3.960±0.026	26.2±0.8	53.9±1.1	37.7±0.6	74.1±0.7	76.2±1.1	37.3±0.6	60.3±0.1	52.3±0.5
	D2	3.990±0.013	26.1±0.5	**54.5±0.8**	37.3±1.0	74.2±0.6	**76.7±1.1**	37.1±0.1	60.2±0.6	52.3±0.6
	TopoPrune	3.972±0.020	**26.5±0.6**	54.3±0.6	**37.9±1.1**	74.2±0.3	75.2±2.4	**37.3±0.5**	60.5±0.1	**52.3±0.4**
80%	Random	4.017±0.048	26.4±0.7	55.0±0.7	37.0±0.3	74.5±0.4	77.9±1.4	37.1±0.8	61.0±0.7	52.7±0.4
	Moderate	**3.977±0.017**	26.3±0.7	**56.0±1.1**	37.6±0.9	74.6±0.1	78.3±0.5	36.6±0.7	60.9±0.8	52.9±0.1
	D2	4.018±0.035	**26.5±1.1**	55.8±1.0	36.9±0.4	**74.9±0.4**	**78.5±1.3**	**37.7±0.6**	**61.1±0.4**	**53.1±0.1**
	TopoPrune	3.996±0.015	26.2±0.5	55.9±1.5	**38.1±0.5**	74.6±0.2	77.2±1.0	36.8±1.2	61.1±0.6	52.9±0.3
90%	Random	4.044±0.015	26.4±0.8	55.1±0.9	36.6±0.4	75.0±0.7	78.4±1.6	37.2±0.8	61.2±1.2	52.8±0.3
	Moderate	4.068±0.024	26.3±0.4	55.4±0.7	37.3±1.2	74.9±0.2	78.0±1.2	37.5±0.3	61.0±1.0	52.9±0.6
	D2	4.079±0.013	**26.9±0.7**	55.9±0.5	37.1±0.8	74.9±0.5	78.3±0.2	**37.7±0.4**	**61.2±0.8**	53.2±0.2
	TopoPrune	**4.038±0.022**	26.4±0.4	**56.3±1.7**	**37.9±0.7**	**75.3±0.2**	**78.9±1.8**	37.5±0.9	60.9±0.6	**53.3±0.7**

(b) Pythia-2.8B

D.2 Statistical Significance of Precision Improvements

To validate the stability benefits of TopoPrune (specifically, reduced variability in final model accuracy across independent runs), we conducted statistical tests focusing on high pruning rates (e.g., 90%), where coreset selection variance is typically most pronounced. We compare TopoPrune against top-performing baselines, D2 [61] and CCS [108]. Our analysis employed two complementary statistical approaches: (1) a one-tailed F-test to evaluate the hypothesis that TopoPrune exhibits lower variance than baselines ($\sigma^2_{\text{TopoPrune}} < \sigma^2_{\text{baseline}}$), and (2) Bootstrapped 95% Confidence Intervals (CI) for the standard deviation of final accuracies, computed using 10,000 resamples with replacement.

As detailed in Table 14, on simpler datasets the difference in variance between methods is not statistically significant. However, for more challenging benchmarks, TopoPrune demonstrates statistically significant p-values ($p < 0.05$, bolded) for lower variance compared to both D2 and CCS. Also, the bootstrapped confidence intervals for the standard deviation of TopoPrune are strictly lower, and in some cases disjoint, than those of the baselines. These findings confirm that our topological selection mechanism offers superior precision compared to prior geometric methods.

Table 14: **Statistical significance of variance** at high pruning rate (90%) using **(a, b, c)** F-test shows TopoPrune's lower variance is statistically significant for more difficult datasets. This is further validated by **(d)** TopoPrune's tighter 95% confidence intervals of standard deviation.

p_value	CCS	D2	TopoPrune
CCS	0.500	∼	∼
D2	0.128	0.500	∼
TopoPrune	0.422	0.094	0.500

(a) F-test on CIFAR-10

p_value	CCS	D2	TopoPrune
CCS	0.500	∼	∼
D2	0.437	0.500	∼
TopoPrune	**0.015**	**0.011**	0.500

(b) F-test on CIFAR-100

p_value	CCS	D2	TopoPrune
CCS	0.500	∼	∼
D2	0.059	0.500	∼
TopoPrune	**0.012**	**0.001**	0.500

(c) F-test on ImageNet-1K

95% CI	CIFAR-10	CIFAR-100	ImageNet-1K
CCS	[0.05, 0.91]	[0.85, 3.10]	[0.07, 0.75]
D2	[0.38, 1.77]	[0.26, 2.69]	[0.3, 2.12]
TopoPrune	[0.08, 0.35]	[0.1, 0.87]	[0.04, 0.23]

(d) Bootstrapped 95% confidence intervals of standard deviation

D.3 Roadmap of Detailed Experimental Results

Experiment / Ablation Topic	Main Section	Full Data Table
Overall Performance Across Baselines and Datasets	Section 4.2	Table 1
Text Dataset Pruning	Section 4.2	Table 2, Table 13
Transferability of *Diverse Embeddings* → *Fixed Target*	Section 4.4	Table 19
Transferability of *Fixed Embedding* → *Diverse Targets*	Section 4.4	Table 4
Robustness to Noisy and Corrupted Representations	Section 4.3	Table 16, Table 17, Table 18
Evaluation of Deep Topological Autoencoders	Section B.1.2	Table 5
Computational Complexity	Section B.2	Table 6, Table 7
Ablation: Performance Impact of (α vs. β)	Section C.1.3	Table 15
Ablation: Filtration and Homology Degree	Section C.3	Table 9
Ablation: Training-Free Proxies for Mislabel Detection	Section C.4	Table 10
Ablation: Sensitivity to UMAP Hyperparameters	Section C.6	Table 12
Implementation Hyperparameters	∼	Table 20

Table 15: **Hyperparameters for local persistence (α) and global density (β).** While our fixed 50/50 split provides strong, stable performance, the results indicate that further accuracy gains are possible with task-specific tuning. We observe a trend where the optimal balance increasingly relies on the persistence score (α) on more challenging datasets (e.g., CIFAR-100 vs. CIFAR-10) and at higher data pruning rates. Optimal results are in bold, ties are underlined.

CIFAR-10	Pruning Ratio (%)				
α/β	30%	50%	70%	80%	90%
100/0	94.7±0.1	93.2±0.1	90.6±0.1	87.9±0.4	80.4±0.1
90/10	94.3±0.1	93.1±0.1	90.4±0.3	88.3±0.1	79.9±1.2
80/20	94.7±0.1	93.3±0.1	90.8±0.2	89.0±0.2	79.3±0.9
70/30	94.4±0.2	93.6±0.1	90.7±0.2	88.5±0.1	78.9±1.2
60/40	94.5±0.3	93.5±0.3	91.3±0.1	87.8±0.2	81.3±0.6
50/50	94.7±0.2	93.7±0.2	**91.6±0.1**	88.7±0.7	82.1±0.3
40/60	94.6±0.3	93.3±0.1	91.4±0.2	88.6±0.2	80.3±0.5
30/70	94.9±0.1	**93.9±0.3**	91.4±0.2	89.0±0.3	82.0±0.1
20/80	**95.0±0.1**	93.8±0.1	91.3±0.2	88.9±0.1	82.3±0.5
10/90	94.8±0.1	93.7±0.2	91.0±0.5	**89.3±0.2**	**82.7±0.1**
0/100	94.9±0.1	93.8±0.1	91.3±0.1	89.1±0.2	82.2±0.4

CIFAR-100	Pruning Ratio (%)				
α/β	30%	50%	70%	80%	90%
100/0	75.7±0.2	73.2±0.2	67.4±0.2	59.8±0.9	43.2±0.3
90/10	76.4±0.3	73.0±0.1	66.8±0.7	60.8±0.5	43.0±0.5
80/20	76.5±0.1	73.1±0.4	67.5±0.1	61.8±0.2	**47.2±1.6**
70/30	76.0±0.2	72.9±0.6	67.7±0.2	61.5±0.3	45.2±0.7
60/40	76.0±0.2	73.3±0.3	67.1±0.7	60.6±0.8	44.8±1.2
50/50	75.9±0.4	72.8±0.3	66.9±0.5	61.9±0.6	45.8±0.7
40/60	76.5±0.1	73.7±0.3	67.9±0.1	**62.5±0.3**	43.1±0.6
30/70	76.6±0.1	**74.0±0.2**	67.9±0.7	62.4±0.6	43.4±0.5
20/80	**77.1±0.1**	74.0±0.2	**68.1±0.1**	62.2±0.3	45.4±2.8
10/90	76.7±0.2	73.6±0.3	67.3±0.3	61.4±0.7	40.9±1.9
0/100	76.2±0.1	73.5±0.2	67.8±0.2	61.5±0.7	41.8±0.6

Table 16: **Coreset performance under *latent Gaussian noise* on CIFAR-100.**

Noise (→)	$\epsilon \sim \mathcal{N}(0,\, 0.25\sigma)$				$\epsilon \sim \mathcal{N}(0,\, \sigma)$			
Pruning Rate (→)	50%	70%	80%	90%	50%	70%	80%	90%
Moderate [100]	71.1±0.2	63.7±0.4	56.0±0.6	33.2±0.9	70.4±0.1	62.5±0.4	54.6±0.6	33.9±0.2
D2 [61]	72.8±0.1	68.5±0.7	63.0±0.5	44.4±1.5	73.0±0.1	67.7±1.2	62.3±0.8	40.2±2.0
TopoPrune	73.5±0.3	67.8±0.1	60.5±0.2	**45.4±0.8**	73.3±0.2	68.0±0.5	62.7±0.3	**45.5±0.6**
Noise (→)	$\epsilon \sim \mathcal{N}(0,\, 4\sigma)$				$\epsilon \sim \mathcal{N}(0,\, 8\sigma)$			
Pruning Rate (→)	50%	70%	80%	90%	50%	70%	80%	90%
Moderate [100]	71.0±0.4	62.9±0.3	52.3±0.6	32.0±1.2	70.9±0.2	63.4±0.2	51.6±2.2	32.1±1.4
D2 [61]	72.3±0.1	67.8±1.1	62.1±1.0	40.5±1.7	71.6±0.5	67.2±0.8	60.3±2.4	39.8±3.2
TopoPrune	73.4±0.1	67.4±0.2	61.7±0.3	**46.1±0.7**	73.4±0.2	67.2±0.1	61.8±0.1	**43.9±0.4**

Table 17: **Coreset performance under *image-level Gaussian noise* on CIFAR-100.**

Noise (→)	$\epsilon \sim \mathcal{N}(0,\, 0.25\sigma)$				$\epsilon \sim \mathcal{N}(0,\, \sigma)$			
Pruning Rate (→)	50%	70%	80%	90%	50%	70%	80%	90%
Moderate [100]	70.4±0.3	62.8±0.2	53.3±0.4	35.0±0.3	70.8±0.2	62.8±0.3	53.1±0.3	34.0±0.6
D2 [61]	72.7±0.5	68.9±0.2	62.0±0.8	40.4±1.5	71.6±0.2	67.2±0.5	62.7±1.1	43.4±0.9
TopoPrune	73.0±0.3	67.8±0.2	61.1±0.4	**40.9±0.2**	72.7±0.2	67.0±0.4	62.3±0.1	**43.5±0.3**
Noise (→)	$\epsilon \sim \mathcal{N}(0,\, 4\sigma)$				$\epsilon \sim \mathcal{N}(0,\, 8\sigma)$			
Pruning Rate (→)	50%	70%	80%	90%	50%	70%	80%	90%
Moderate [100]	71.5±0.3	63.9±0.5	57.5±0.4	35.7±1.7	71.6±0.2	64.6±0.6	54.7±1.0	34.4±1.5
D2 [61]	72.5±0.2	67.2±0.9	60.3±1.2	41.3±2.2	72.5±0.2	67.7±1.0	59.9±1.0	42.6±2.5
TopoPrune	73.2±0.3	68.3±0.2	61.1±0.5	**41.9±0.4**	73.2±0.1	67.1±0.3	60.8±0.5	**43.3±0.7**

Table 18: **Coreset performance under *image-level corruptions* on CIFAR-100.** We apply four representative CIFAR-C corruptions [46] (contrast, motion blur, frost, and JPEG compression) at severity level 3.

Corruption (→)	Contrast				Motion Blur			
Pruning Rate (→)	50%	70%	80%	90%	50%	70%	80%	90%
Moderate [100]	70.9±0.3	63.4±0.4	53.7±0.3	30.9±0.7	71.2±0.1	64.1±0.2	56.4±0.5	34.9±0.8
D2 [61]	74.3±0.6	64.0±1.4	62.6±1.0	43.2±2.2	73.7±1.0	65.7±0.9	62.8±1.0	41.3±1.2
TopoPrune	73.1±0.3	67.9±0.6	62.0±0.5	**45.0±1.2**	73.1±0.1	67.7±0.1	62.4±0.8	**42.7±0.5**
Corruption (→)	Frost				JPEG Compression			
Pruning Rate (→)	50%	70%	80%	90%	50%	70%	80%	90%
Moderate [100]	70.4±0.4	63.5±0.4	55.1±0.8	36.1±0.3	70.8±0.2	63.8±0.2	55.9±0.9	33.9±0.6
D2 [61]	73.3±0.2	65.1±0.7	61.9±1.2	43.2±1.4	73.8±0.3	64.7±1.5	62.8±1.0	40.9±1.1
TopoPrune	73.0±0.3	65.9±0.1	61.7±0.3	**43.2±0.2**	73.0±0.1	67.9±0.5	62.8±0.2	**43.5±0.5**

Table 19: **Transferability of *Diverse (larger) Embeddings* → *Fixed (smaller) Target***. Features across many architectures to train a ResNet-18 model on CIFAR-100. Models taken from `torchvision` are finetuned on ImageNet-1K (all except OpenCLIP). We also look at the transferability of features from OpenCLIP foundational models trained on the LAION-2b dataset [82].

Pruning Rate (→)	50%			70%		
	Moderate	D2	**TopoPrune**	Moderate	D2	**TopoPrune**
ResNet-18 [43]	70.9±0.4	73.0±0.8	73.6±0.2	62.9±0.2	67.9±0.3	68.1±0.2
ResNet-50 [43]	71.1±0.1	73.0±0.2	73.7±0.2	63.3±0.4	67.7±0.4	68.0±0.1
ResNet-101 [43]	70.0±0.5	73.2±0.2	73.5±0.2	62.8±0.4	66.9±0.9	68.0±0.3
EfficientNet-B0 [92]	71.6±0.3	73.2±0.2	72.9±0.2	62.9±0.1	67.5±0.4	67.5±0.2
EfficientNetV2-M [93]	69.6±0.3	72.8±0.7	73.4±0.2	61.0±0.2	67.1±0.8	67.1±0.1
SwinV2-T [57]	70.5±0.1	73.6±0.1	73.6±0.2	60.4±0.6	66.9±1.0	67.6±0.1
SwinV2-B [57]	69.9±0.3	73.5±0.4	73.6±0.2	61.9±0.4	67.1±0.4	68.1±0.2
ViT-L-16 [27]	69.8±0.3	73.3±0.3	73.6±0.1	61.5±0.3	67.2±0.6	68.0±0.1
OpenCLIP ViT-L-14 [81, 82]	71.2±0.4	73.2±0.2	73.3±0.1	63.5±0.4	66.8±0.3	67.9±0.3
OpenCLIP ViT-H-14 [81, 82]	70.9±0.3	73.0±0.2	73.1±0.4	62.4±0.3	66.6±1.3	67.7±0.2
Overall Average	70.6±0.3	73.2±0.3	**73.4±0.2**	62.3±0.3	67.2±0.7	**67.8±0.2**

Pruning Rate (→)	80%			90%		
	Moderate	D2	**TopoPrune**	Moderate	D2	**TopoPrune**
ResNet-18 [43]	54.8±0.2	60.3±1.9	60.2±0.2	33.8±0.8	42.5±1.9	43.4±0.4
ResNet-50 [43]	55.9±0.6	60.8±1.0	61.3±0.7	31.8±1.6	44.5±1.7	47.4±0.5
ResNet-101 [43]	54.5±0.4	60.4±0.2	60.0±0.6	35.9±0.6	41.7±2.6	43.2±1.3
EfficientNet-B0 [92]	54.8±1.1	60.5±0.8	62.0±0.4	29.8±1.3	42.0±2.2	42.2±0.3
EfficientNetV2-M [93]	53.1±0.5	60.2±0.4	60.1±1.6	33.3±0.8	41.3±1.9	44.4±1.7
SwinV2-T [57]	53.3±1.2	59.3±2.0	61.8±0.4	32.5±1.5	41.4±2.2	43.4±0.8
SwinV2-B [57]	53.5±0.2	60.2±1.5	61.1±0.6	35.7±0.3	42.8±2.9	42.7±1.6
ViT-L-16 [27]	53.9±0.9	59.1±1.1	61.1±0.2	30.6±1.2	40.9±2.6	44.1±1.3
OpenCLIP ViT-L-14 [81, 82]	53.9±0.8	60.4±1.0	61.7±0.8	35.0±1.5	40.7±2.9	42.5±1.6
OpenCLIP ViT-H-14 [81, 82]	54.0±0.8	61.1±0.4	61.8±0.8	35.4±1.4	42.3±2.1	42.9±1.0
Overall Average	54.2±0.7	60.2±1.0	**61.1±0.6**	33.4±1.1	42.0±2.2	**43.6±1.1**

Table 20: **(a)** Training and topological hyperparameters. **(b)** Dataset mislabel ratios. Similar to those in Zheng et al. [108].

Section	Hyperparameter	CIFAR-10 CIFAR-100	ImageNet
Training (`DeepCore`)	Epochs	200	60
	Batch Size	256	128
	Optimizer	SGD	SGD
	Momentum	0.9	0.9
	Learning Rate	1e-1	1e-1
	Weight Decay	5e-4	5e-4
	Scheduler	CosineAnnealing	CosineAnnealing
Global Manifold Projection (`UMAP`)	Number Neighbors	15	15
	Minimum Distance	0.1	0.1
	Metric	Cosine	Cosine
	Dimensions	2	2
Kernel Density Estimation (`sklearn`)	Bandwidth	0.4	0.4
Local Persistent Homology (`multipers`)	Theta (Dens. Band.)	0.4	0.4
	Function/Kernel	Gaussian	Gaussian
	Complex	Weak-Delaunay	Weak-Delaunay
	Homology Degree	1	1
	Optimization Steps	6	6
Topology Score	Global Density (α)	0.5	0.5
	Local Persistence (β)	0.5	0.5
NLPS (KNN `sklearn`)	Number Neighbors	20	20

(a)

Pruning	Mislabel Ratio (%)		
	C-10	C-100	ImageNet
30%	0%	10%	0%
50%	0%	20%	10%
70%	10%	20%	20%
80%	10%	40%	20%
90%	30%	50%	30%

(b)

D.4 Additional Persistence Optimization Visualizations

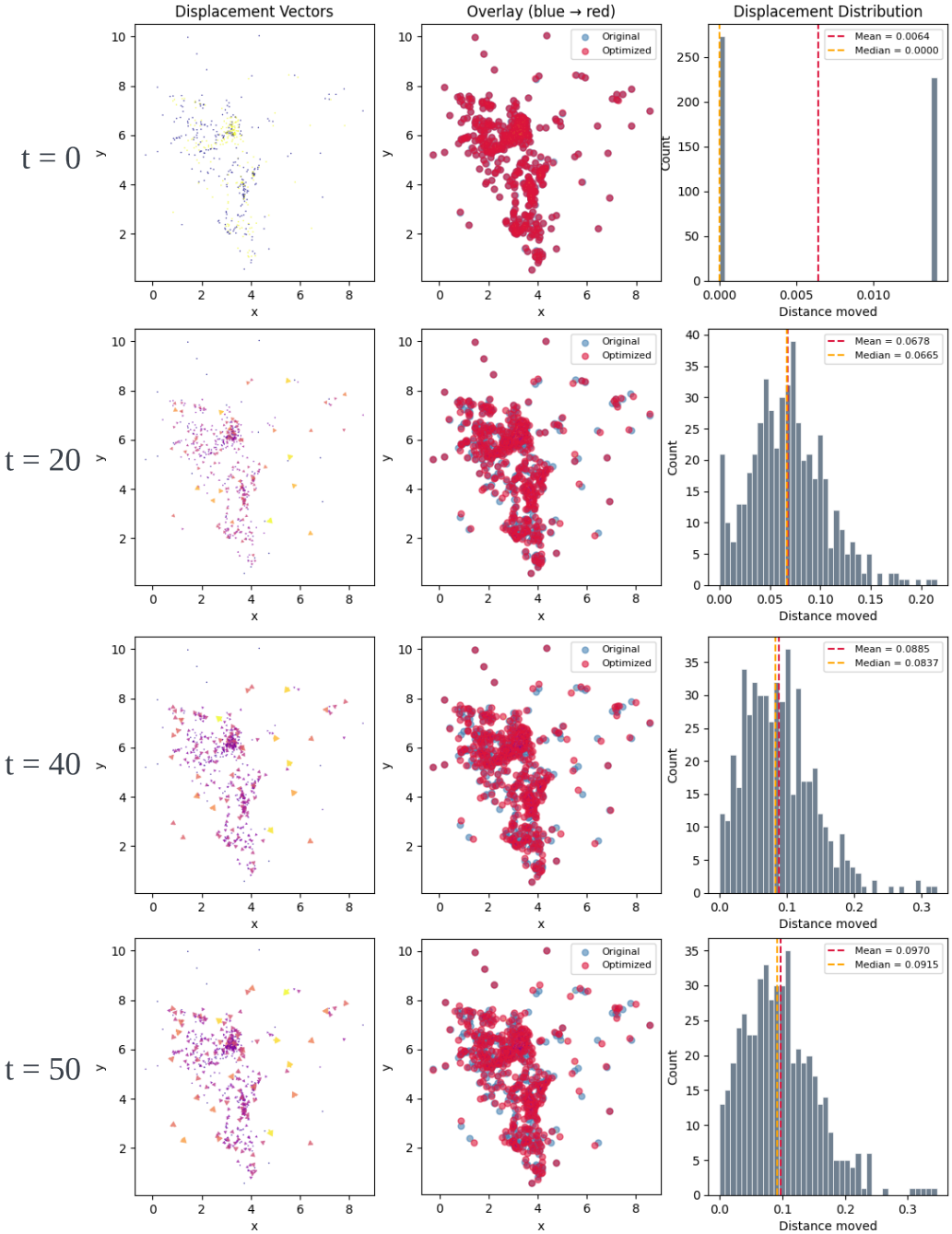


Figure 17: Persistence change across optimization (Label #14 C-100; ResNet-50)

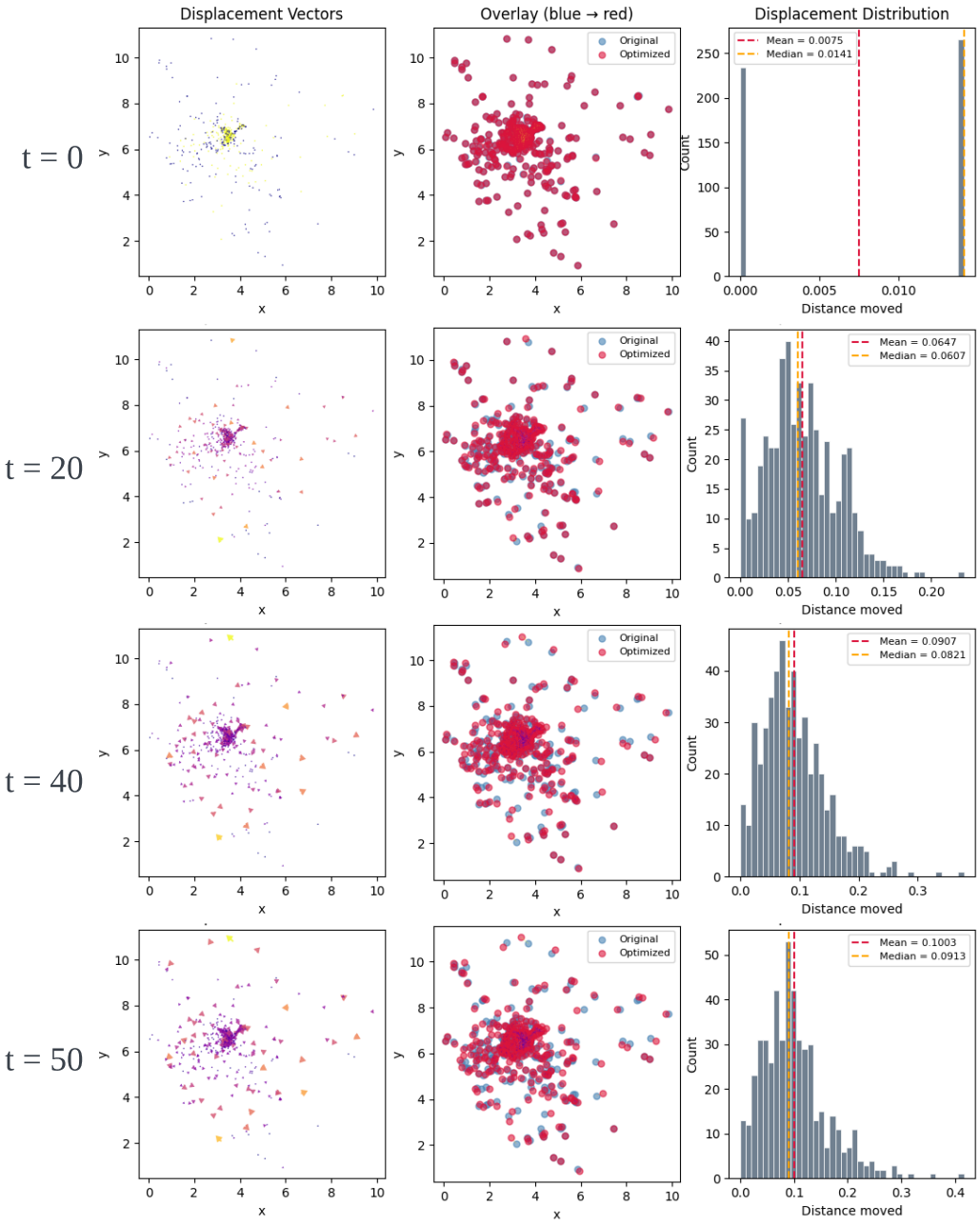


Figure 18: Persistence change across optimization (Label #18 C-100; ResNet-50)

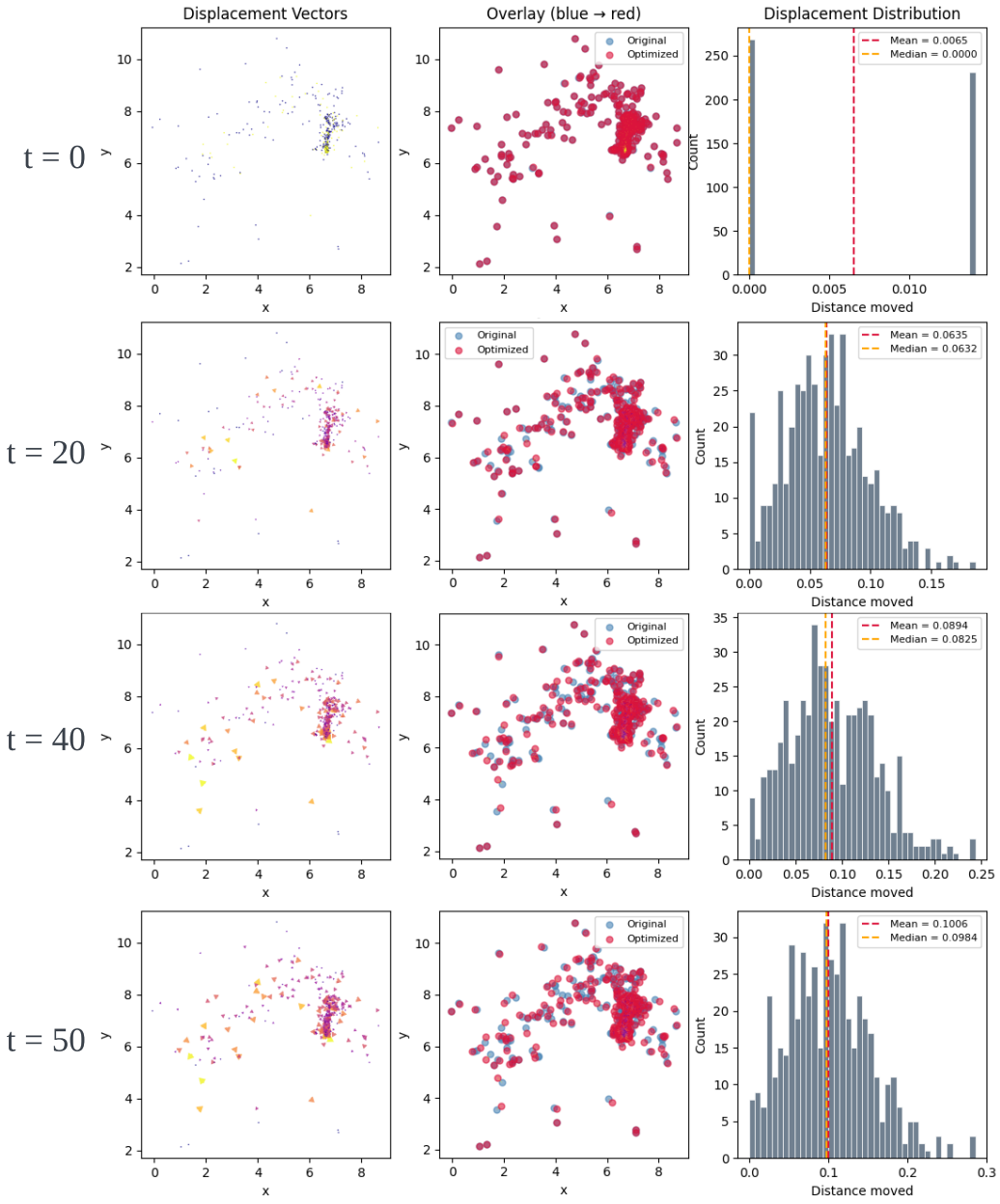


Figure 19: Persistence change across optimization (Label #8 C-100; ResNet-50)

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state three concrete claims: (1) higher accuracy and precision particularly at extreme pruning rates, (2) robustness to embedding perturbations, and (3) cross-architecture transferability. Each is supported by experiments in Sections 4.2, 4.3, and 4.4 respectively, with full quantitative results in the appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss computational limitations in Section B.2, where we transparently report that TopoPrune's wall-clock latency is higher than fast geometric heuristics due to the topological backend (`multipers`) currently lacking GPU acceleration and multi-processing. We note this is an implementation bottleneck rather than a fundamental algorithmic limitation. We also acknowledge that effectiveness on other modalities remains for future work. A full limitation section is provided in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theoretical background appear in Section A.2, including Proposition A.6 (Sensitivity to Anisotropic Distortion) with full proof, and Proposition A.7 (Stability of Persistent Homology) which we cite from Cohen-Steiner et al. [22] and Chazal et al. [18]. All assumptions, including the use of the Vietoris-Rips complex and the standard Euclidean metric, are stated in the Preliminaries (Section A.2).

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide complete experimental details in Section 4.1 (datasets, baselines, training protocol) and Section B.3 (full pseudocode for TopoPrune, Algorithm 1). All hyperparameters for training, UMAP projection, kernel density estimation, and persistent homology optimization are listed in Table 17. Mislabel ratios follow established conventions from Zheng et al. [108] and are reported in Table 17b.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce

the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets used (CIFAR-10, CIFAR-100, ImageNet-1K, CIFAR-C) are publicly available standard benchmarks. We will release anonymized code at submission time and a public repository upon acceptance. The codebase builds on standard open-source libraries (UMAP, multipers, GUDHI, DeepCore), all of which are cited.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training settings (epochs, batch size, optimizer, learning rate, scheduler, weight decay) are fully specified in Table 17a. Section 4.1 describes the experimental protocol, and individual experiments specify any deviations. Accuracies are averaged over five independent training runs for the vision experiments and three for the text experiments, as stated in Section 4.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: All reported accuracies include standard deviations computed over independent training runs (five for the vision experiments, three for the text experiments; Section 4.1). Section D.2 provides additional statistical validation via one-tailed F-tests and bootstrapped 95% confidence intervals (10,000 resamples), confirming the precision improvements of TopoPrune over baselines.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section B.2 reports compute resources for selection: experiments were benchmarked on an AMD EPYC 7502 (32-Core) CPU. Latency and CPU utilization are reported in Table 6. Training was conducted using the standardized DeepCore framework [42] with hyperparameters in Table 17a.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work uses publicly available image classification benchmarks (CIFAR-10/100, ImageNet-1K) and does not involve human subjects, scraped data, or applications with foreseeable misuse potential. We have reviewed the NeurIPS Code of Ethics and confirm conformance.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Data pruning methods like TopoPrune reduce the computational cost of training large models, with positive impacts on energy consumption and accessibility of deep learning research. Improved coreset stability could also benefit the reproducibility and auditability of training pipelines. As a potential negative impact, pruning may reduce the representation of underrepresented subgroups; we discuss this and a mitigation in the Impact Statement. We do not foresee other direct negative societal impacts, as the method is a generic data-efficiency technique rather than tied to a specific deployed application.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not release new datasets, pretrained models, or generative systems with high misuse risk. The proposed method operates on existing publicly available pretrained models and benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets (CIFAR-10/100, ImageNet-1K, CIFAR-C, LAION-2b for OpenCLIP) and code libraries (UMAP, multipers, GUDHI, DeepCore, torchvision) are cited in the manuscript with their respective publications. We use these assets under their standard research licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new datasets or pretrained models. Code release will follow standard documentation practices upon acceptance.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This research does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This research does not involve human subjects and does not require IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for writing assistance (editing and formatting), and did not contribute to the core methodology, experimental design, scientific analysis, or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.