DIVERSITY WITHOUT LABELS: TASK-AGNOSTIC CORESET SELECTION ACROSS PERCEPTION AND ACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

While data collection for model training is relatively easy, curating high-quality datasets from large collections is inherently challenging. Coreset estimation methods address this problem by identifying the best data subsets to train models on. However, existing approaches often require proxy model training, extensive annotations, or task-specific tailoring. Unsupervised alternatives typically exploit geometric properties in foundation model feature spaces, but are limited to pairwise image-to-image comparisons or comparisons against a fixed subset. We introduce CoreDNA, an unsupervised, task-agnostic coreset estimation approach that pools neuron activations from vision foundation models into per-image and per-dataset distributions. By comparing each image's distribution to the dataset-level distribution, CoreDNA can capture both low-level visual structure and high-level semantic information across layers. We find that high-quality samples concentrate in the tails of the last layer's activation statistics, which we validate by relating them to training-based coreset methods. Building on these findings, we develop a computationally efficient selection mechanism to construct task-agnostic, high-quality coresets for selective annotation and model training. Across image classification, semantic segmentation, depth estimation, and imitation learning for end-to-end driving and mobile manipulation, CoreDNA matches or outperforms the state of the art, demonstrating unprecedented generalization across perception and action tasks. These results highlight the effectiveness of compact activation-based representations for data selection.

1 INTRODUCTION

Vision models have become a core component of modern perception and decision-making systems, and their performance is critically shaped by the data they are trained on. What matters is not only how much data they see but which data, since often only a subset of a large dataset is the most informative for training. Restricting training to such subsets can substantially improve efficiency while preserving performance, but only if the right subset can be reliably identified. In robotics, for instance, subsets are still commonly selected based on fixed time intervals or geographic location, which neglects data diversity and might steer models toward oversampling similar scenarios (Hindel et al., 2025). As a result, training datasets are dominated by repetitive, easy-to-collect scenarios, while rare but safety-critical or informative edge cases remain underrepresented.

Coreset estimation tackles this problem by identifying a compact subset of a dataset that preserves the properties necessary for effective model training. In particular, *one-shot* coreset estimation performs this selection in a single pass, without repeatedly retraining a model to refine the coreset selection. However, existing one-shot coreset methods face two limitations. Some are limited to specific perception tasks, such as classification (Paul et al., 2021) or imitation learning (Dass et al., 2026), restricting their applicability beyond the settings for which they were designed. Others require training on the fully-labeled dataset to determine sample "hardness," i.e., how difficult a sample is to learn, measured by its loss or misclassifications during training. This presupposes full human annotation (Acharya et al., 2024; Zheng et al., 2023; Killamsetty et al., 2021). Nevertheless, full annotation is rarely available in practice, as human labeling is expensive and difficult to scale to large, continuously growing datasets. Unsupervised coreset estimation instead enables an a-priori

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

data selection without relying on labels. Since diversity matters more than hardness for downstream performance (Yao et al., 2026), coreset estimation methods exploit diversity in foundation model feature spaces to yield task-agnostic, universal datasets (Xie et al., 2023a; Griffin et al., 2026). However, these methods score images either relative to one another, or relative to a randomly sampled reference embedding distribution, rather than relative to the dataset's own distribution. Such pairwise image-to-image comparisons are computationally expensive and do not scale to larger coresets. In contrast, Visual DNA (Ramtoula et al., 2023) showed that distributions of neuron activations across all layers of a frozen foundation model can be used as general-purpose representations of datasets to measure similarity, capturing both low-level and semantic attributes. Both levels matter for task-agnostic coreset selection, as different downstream tasks rely on different cues, from appearance changes such as lighting or weather to semantic scene content. As curating a coreset amounts to shaping the data distribution a model will be trained on, such general-purpose representations are a natural fit, and we explore how to extend and apply them to coreset selection.

To this end, we propose CoreDNA (Coreset DNA), an unsupervised, task-agnostic coreset estimation method that evaluates each image relative to its source dataset (Fig. 1). We pool neuron activations across layers into compact per-image and dataset-level distributions and investigate the information they encode. Our analysis reveals that samples assigned high utility by training-based coreset methods tend to concentrate in the tails of final-layer activation statistics (Sec. 3.2). Motivated by this, we develop computationally efficient, density and coverage-based selection criteria for selective annotation and model training. We evaluate CoreDNA across image classification, semantic segmentation, depth estimation, and imitation learning for end-to-end driving and mobile manipulation (Sec. 4). Across these tasks, CoreDNA matches or outperforms state-of-the-art training-based and training-free methods, showing the value of dataset-relative activation statistics for data selection.

2 Related Work and Background

Training-based Coreset Methods curate coresets using signals such as losses or gradients, obtained by partially or fully training a model on the labeled dataset. A common strategy represents each sample by the gradient of the downstream task loss on that sample with respect to the model parameters, which captures how the sample would update the model during training (Mirzasoleiman et al., 2019; Yang et al., 2023; Chang & Khanna, 2025). Additional works cast coreset selection as a bilevel optimization problem, with an inner loop training the model on the selected subset and an outer loop optimizing the subset itself (Zhou et al., 2022b; Killamsetty et al., 2021; Xia et al., 2024b). A separate line of work focuses on loss-based selection. While forgetting events, i.e., switches from correct to incorrect classification between epochs, have been found to correlate with hard examples (Toneva et al., 2019), AUM (Area Under the Margin) instead measures the probability gap between the target class and the largest other class across training epochs, with low values indicating hard samples (Pleiss et al., 2020). On the other hand, EL2N focuses on samples with a large $L2$ distance between predicted class probabilities and labels (Paul et al., 2021). Follow-up work extends these methods with stratified sampling to improve data coverage at low retention rates (Zheng et al., 2023), or incorporates embedding similarity as a secondary criterion (Acharya et al., 2024; Yan et al., 2025; Xie et al., 2025; Maharana et al., 2024; Wu et al., 2026; Zhang et al., 2026b) to obtain diverse coresets. Moreover, many methods compute selection scores with a proxy model, a cheaper model trained in place of the target one. While denoising autoencoders select hard-to-reconstruct samples (Zhang et al., 2026a), smaller models from the same model family serve as cheap difficulty estimators (Zhang et al., 2025; Gu et al., 2025). Additionally, LoRA adapters attached to large pretrained models approximate gradients efficiently (Zhang et al., 2024; Xia et al., 2024a; He et al., 2026; Liu et al., 2024). Further methods relax the full-dataset labeling constraint. ELFS applies deep clustering for pseudo-labeling, then uses forgetting- or AUM-based scoring for coreset estimation (Zheng et al., 2025). OFDS extends this with object-level representations, selecting k-means centroids as the coreset (Popp et al., 2025). While most training-based coreset estimation methods target image or language understanding, other related works address video streaming (Mahdizadeh et al., 2026), or imitation learning (Dass et al., 2026).

Training-free Coreset Methods instead avoid training a model on the target data to yield the coreset, and commonly rely on geometric criteria in foundation-model feature spaces to identify the most diverse samples for optimal performance. SSL performs clustering in feature space and selects the sample closest to each cluster mean (Sorscher et al., 2022). Moderate (Xia et al., 2023)

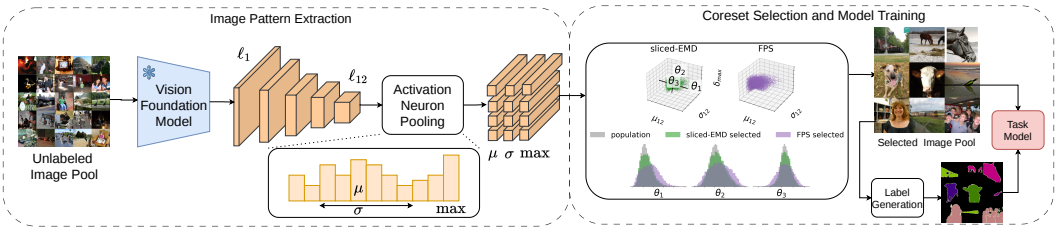


Figure 1: CoreDNA extracts neuron activation histograms from a vision foundation model, pooling them into per-layer mean μ, standard deviation σ, and maximum m. We use the last layer's μ, σ, and cross-layer dispersion δ to select a coreset via sliced Wasserstein minimization ("sliced-EMD") or farthest-point sampling ("FPS").

instead selects data points whose distance to the class center is closest to the median distance, later improved via classwise sampling (Jafari et al., 2024). Other methods select a subset whose geometric median matches that of the full dataset (Acharya et al., 2025), use optimal transport to find representatives of the full dataset (Zhu et al., 2026), score samples by their density in feature space after filtering those with mixed-label neighborhoods (Roy et al., 2026), or maximize coverage so that every sample is close to a selected one (Rustamov et al., 2026). ZCore uses a vision foundation model to extract features, then iteratively slices the full embedding space into numerous low-dimensional subspaces, sampled via a triangular distribution to estimate each example's value based on coverage and redundancy (Griffin et al., 2026). FreeSel extracts representations via attention patterns from pretrained models guided by clustering and then employs distance-based greedy selection to iteratively add images dissimilar to those already selected (Xie et al., 2023a). TFDP ranks images for instance segmentation by a complexity score computed from annotated shapes and classes, addressing instance-area variation and class imbalance (Dai et al., 2025), while ViSA selects images via foundation-model-based segmentation complexity and alignment scores (Liu et al., 2025). ActiveFT optimizes a parametric selection model via gradient descent on pretrained features to match the density of the unlabeled feature pool while maximizing diversity (Xie et al., 2023b). Further works rely on cosine similarity within a vision-language model (Qi et al., 2026; Bi et al., 2026; Maini et al., 2024; Abbas et al., 2024; 2023) or on foundation-model agreement (Meding et al., 2022). However, all outlined methods score images through pairwise image-to-image comparisons rather than capturing information about the full dataset's distribution. In this work, we examine multi-layer activation patterns relative to the full dataset, as a better proxy for the diversity of a training sample.

3 Technical Approach

We first formalize the problem of coreset estimation and our desiderata for a suitable solution (Sec. 3.1). We then introduce per-image descriptors derived from pooled foundation-model activations and analyze their properties (Sec. 3.2). We relate them to training-based coreset methods to uncover patterns behind good coresets (Sec. 3.3). Finally, we present CoreDNA (Fig. 1), our selection algorithm that leverages our descriptors to construct high-quality, task-agnostic coresets (Sec. 3.4).

3.1 Problem Definition

Coreset estimation selects the most informative subset of a dataset in a one-shot manner. This differs from active learning (Ash et al., 2020), which iteratively queries an oracle for labels as training progresses. Formally, consider a dataset $\mathcal{P}$, $|\mathcal{P}| = P$, and a fixed, pre-defined retention ratio $\alpha \in (0, 1]$[1]. The goal is to find a coreset $\mathcal{C} \subset \mathcal{P}$ with $|\mathcal{C}| = C = \alpha P$ that maximizes the downstream performance of a model trained on it. Our coreset estimation method should satisfy six requirements: it should be training-free, task-agnostic, encompassing information about the full dataset's distribution, agnostic to the number and type of visual inputs, able to incorporate data recording information, and computationally scalable to larger datasets. Additionally, we aim for a compact, low-dimensional representation that allows the underlying properties to be studied in detail. To achieve this, we divide the problem into pattern extraction and selection. Pattern extraction maps the P images to a compact feature matrix $F \in \mathbb{R}^{P \times D}$, representing each image relative to the full dataset. A computationally scalable selection algorithm then selects a coreset of C samples from F.

[1]Determining the optimal retention ratio is an active research direction in dataset pruning, not addressed here.



Figure 2: Example Cityscapes-DVPS images ranked by μ_{12} into low (L), medium (M), and high (H) buckets. High-μ_{12} images tend to contain rare classes, while near-duplicates appear have similar scores in both L and H.

3.2 Per-Image Pattern Extraction

Image Patterns Considered: We first explore potential ways to produce image features that might serve as a good base for our design of a coreset curation method. To do so, we first create image-wise embeddings using VDNA (Ramtoula et al., 2023), which was originally designed to measure dataset similarity. VDNA computes, for every neuron of a frozen ViT-B backbone, a histogram of that neuron's activations over an image x. We use the Mugs ViT-B/16 (Zhou et al., 2022a), as in VDNA. The backbone has $L = 12$ layers ($\ell = 1, \dots, L$), each with $N = 768$ neurons ($n = 1, \dots, N$). Neuron (ℓ, n) yields raw bin counts $r_{\ell n}(x)$ over B bins, which we normalize into a histogram $h_{\ell n}(x) \in \Delta^{B-1}$. The dataset-level VDNA $H_{\ell n}$ is the histogram of the same neuron accumulated over the full training set. We then compute the per-neuron EMD between an image's histogram and the dataset-level histogram:

$$h_{\ell n}(x) = \frac{r_{\ell n}(x)}{\|r_{\ell n}(x)\|_1}, \qquad H_{\ell n} = \frac{\sum_{i=1}^{P} r_{\ell n}(x_i)}{\|\sum_{i=1}^{P} r_{\ell n}(x_i)\|_1}, \qquad d_{\ell n}(x) = \mathrm{EMD}\big(h_{\ell n}(x),\, H_{\ell n}\big). \quad (1)$$

The resulting distances $d_{\ell n}(x)$ across all neurons reflect how typical or atypical image x is relative to the full dataset, at the granularity of individual neurons. To reduce dimensionality and build a compact representation, we explore multiple candidate statistics over these distances separately for each layer ℓ. Over all the neurons n of that layer ℓ, we compute $d_{\ell n}(x)$ and record their mean $\mu_\ell(x)$, standard deviation $\sigma_\ell(x)$, and maximum $m_\ell(x)$. $\mu_\ell(x)$ captures a layer's overall atypicality, while $\sigma_\ell(x)$ and $m_\ell(x)$ reveal whether that atypicality is broadly distributed or concentrated in a few strongly reacting neurons. We omit the minimum, as it is dominated by the many near-zero, uninformative neurons and adds no additional signal. Further, we also compute the same three statistics computed over all layers jointly, $\mu_{1:L}(x)$, $\sigma_{1:L}(x)$, and $m_{1:L}(x)$. Together, these $3L + 3$ statistics form the representation $\phi(x) \in \mathbb{R}^{3L+3}$ of image x, yielding a $P \times (3L + 3)$ representation matrix $\Phi = [\phi(x_1), \dots, \phi(x_P)]^\top$ across the full dataset, containing statistics describing diverse types of properties and visual features, some of which we hope can help identify good coresets.

Beyond these per-layer statistics, we propose to explore two further candidate statistics that measure how consistently the entries of ϕ rank an image, to identify images where the foundation model's representation shifts across layers, hypothesizing these may add an informative axis for selection. For each of the $3L + 3$ entries, indexed by k, we sort all P training images in ascending order of that entry and let $\rho_k(x) \in \{0, \dots, P - 1\}$ denote the position of image x in this ordering. We measure how much the individual representations in $\phi(x)$ disagree about the image rank. Let $\tilde{\rho}_k(x)$ denote the absolute deviation of entry k's rank from the per-image median rank,

$$\tilde{\rho}_k(x) = \big|\, \rho_k(x) - \mathrm{med}_{k'}\, \rho_{k'}(x) \big|. \quad (2)$$

We then define $\delta_{\max}(x) = \max_k \tilde{\rho}_k(x)$ and $\delta_{\mathrm{avg}}(x) = \frac{1}{3L+3} \sum_k \tilde{\rho}_k(x)$ as the maximum and average deviation across all $3L + 3$ entries of $\phi(x)$, respectively. Small $\delta_{\max}(x)$ or $\delta_{\mathrm{avg}}(x)$ means all entries rank the image alike, whereas large values mean one or more entries rank it differently from the rest. We add both to $\phi(x)$ as further candidate statistics, which thus contains $3L + 5$ entries.

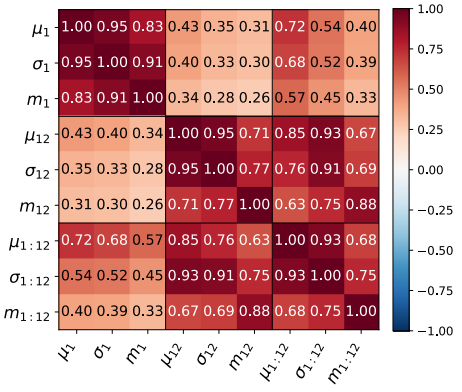


Figure 3: Correlation analysis of image patterns averaged across all datasets. Same-layer statistics correlate strongly; cross-layer correlation decays.

Analysis of Image Patterns: We analyze the layerwise representations with the Pearson correlation coefficient r of μ_ℓ, σ_ℓ, and m_ℓ ($\ell = 1, 12, 1{:}12$) in Fig. 3, with qualitative examples of μ_{12} shown in Fig. 2. μ_ℓ and σ_ℓ are strongly correlated ($r = 0.93$–0.95) across same layers, whereas μ_ℓ and m_ℓ

show more variable correlation across datasets ($r = 0.68$–0.83). Further, the correlation between the means of different layers decays with the distance between them and the two most distant blocks, ℓ_1 and ℓ_{12}, share only $r = 0.43$, the weakest pair in every dataset. Even the all-layers aggregate $\mu_{1:12}$ is only moderately tied to the mean of the first and last block ($r = 0.72$ for ℓ_1, 0.85 for ℓ_{12}). Layers are therefore only moderately correlated on raw values, not independent, and markedly less correlated across layers than μ_ℓ is with its own σ_ℓ within a layer. However, different layers' μ_ℓ largely agree on which images are extreme. We compare if the top or bottom 10% of μ_1 and μ_{12} overlap and compute a Jaccard index of 0.12–0.27, which is 2.3–$5.1\times$ the chance level (0.053), whereas a same-size band of middle-ranked images overlaps at chance (0.054–0.073). The same images thus recur in the tails of the lowest and the highest layer, even though the layers' raw values are only moderately correlated.

Further, we compute the Pearson correlation of $\delta_{\max}(x)$ and $\delta_{\text{avg}}(x)$ with (μ_{12}, σ_{12}). We find that they are largely orthogonal: mean absolute correlation is 0.04–0.08 on CIFAR10, PascalVOC, Cityscapes, Cityscapes-DVPS, and NAVSIM. There are two exceptions, SemanticKITTI-DVPS (correlation of $0.36 - 0.38$) and LIBERO-10 ($0.20 - 0.25$). In both exceptions, the correlation is negative: images with high μ_{12} (strongly atypical) tend to be ranked as atypical consistently across all layers, resulting in low δ, which shows that they are semantic outliers.

3.3 Analysis on Coreset Methods

Choice of the Selection Axes: We introduce our dataset representation Φ, which contains embeddings reflecting diverse levels of visual features and relative rarity within the full dataset. We perform coreset selection in a low-dimensional space formed by a few of these $3L + 5$ axes to facilitate sampling, so we first aim to identify which axes form a space in which coreset quality best predicts downstream performance. We consider candidate spaces formed by two or three axes. Since our earlier analysis showed that the top and bottom tails overlap between ℓ_1 and ℓ_{12} despite their raw values being only moderately correlated, we restrict the search to combinations of statistics within a single layer, keeping the axis search space minimal while still capturing cross-layer tail agreement. To do so, we look for patterns in 5,265 previous training runs from random, training-based, and unsupervised baselines. These span seven dataset-task combinations across five datasets and three task types (classification, segmentation, depth; 246–1,111 runs per combination). Since downstream performance is a property of an entire coreset rather than of individual samples, relating it to our sample-wise features requires summarizing each coreset, within a given space, into a single coreset-level score. We propose that a coreset might benefit from both covering the full dataset's support and matching its density, and we therefore consider two kinds of scores measuring each of these goals, which we call coverage and fidelity, respectively. These two goals can conflict as matching density under-samples the tails, while maximizing coverage over-represents outliers. We measure a coverage family (e.g., the mean distance from each sample to its nearest selected sample) and a fidelity family (e.g., the Wasserstein distance between the coreset and the full dataset), listed in Appx. A.6 and oriented so that higher values indicate a better coreset. For each candidate space, we then compute the Pearson correlation across runs between each score and downstream performance, only comparing runs with the same coreset ratio so that differences in coreset size do not drive the correlation. Then we average correlations of each family. A high correlation indicates that coresets with high coverage (or fidelity) in that space also lead to high downstream performance, which makes it a suitable space in which to select coresets.

Table 1: Axes of the selection space (base: μ_{12}, the mean of layer ℓ_{12}). Correlation of run performance with the coverage and fidelity metrics of the selected set, averaged over the metrics of each family and over seven dataset-task combinations (5,265 historical runs). Best in bold.

Selection space	Coverage ↑	Fidelity ↑
μ_{12}	0.273	0.300
μ_{12} + $\delta_{\max}$	0.276	0.340
μ_{12} + σ_{12}	0.280	0.341
μ_{12} + $\delta_{\max}$ + σ_{12}	**0.308**	**0.357**

Source Layer Selection for Features: We find that the last layer's mean μ_{12}, standard deviation σ_{12}, and dispersion $\delta_{\max}$ correlate best with performance on fidelity, while all candidate axes perform similarly on coverage (Tab. 1 and per-dataset results in Tab. 10 in the appendix). A correlation of 0.357 may seem modest at first glance, but it holds consistently across $5{,}265$ runs, spanning five datasets across three task types, each evaluated at various retention rates and with multiple selection methods. Despite training noise entirely unrelated to sample selection, this signal persists, making it a substantial finding. The two are also complementary since combining them into a three-axis space gives the highest correlations for both coverage and fidelity (0.308 and 0.357), improving over either

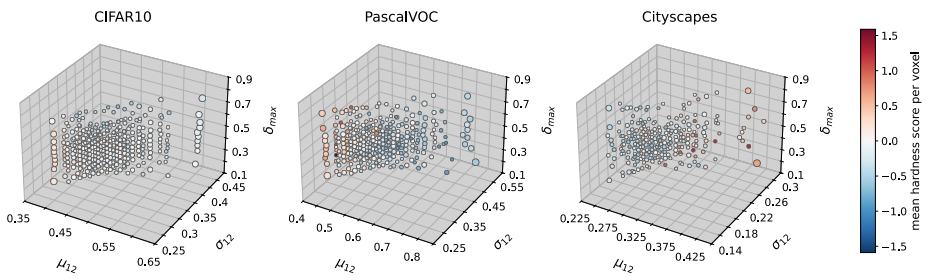


Figure 4: 3D scatter of $z(x)$, colored by EL2N-measured hardness, for CIFAR10, PascalVOC, and Cityscapes.

two-axis space on most datasets (Tab. 11). We note, however, that Cityscapes shows no correlation with any fidelity metric, suggesting that coverage may be the more informative metric for this dataset. Given this analysis, we represent each image x by $z(x) = \big(\mu_{12}(x),\, \sigma_{12}(x),\, \delta_{\max}(x)\big)$, yielding a feature matrix $F \in \mathbb{R}^{P\times 3}$ on which we perform coreset selection.

Additional Axis: For tasks with multiple visual inputs, we combine them analogously, $z(x) = \big(z(x_a), z(x_b)\big)$, with x_a and x_b denoting the different visual inputs. Beyond images, many datasets also carry auxiliary recording information, such as timing or proprioceptive sensor readings, and we evaluate the effect of adding such non-visual information to $z(x)$. For video sequences, we add a frame index to encourage temporal diversity, since consecutive frames at high frame rates are highly redundant. For autonomous agents, we suggest incorporating pose, velocity, and acceleration. We compress this numerical information into one dimension with PCA before adding it to $z(x)$.

Comparison to Training-Based Metrics: We compare our selection features $z(x)$ to difficulty scores from a full training run such as AUM, EL2N, and Forgetting to understand where the hard samples sit along the μ_{12}, σ_{12}, and $\delta_{\max}$ axes. We report two statistics. The first is the Pearson correlation r between $\mu_{12}(x)$ and difficulty over all samples. The second is the tail enrichment, which measures how much harder the samples in a tail of μ_{12} are than average. We take the 10% of samples with the highest (or lowest) $\mu_{12}(x)$ and express their mean difficulty as a z-score relative to the whole training set, with difficulty oriented so that positive means harder. For example, +0.63 SD means that the average sample in that tail is 0.63 standard deviations harder than the average training sample. The relation differs across datasets (Fig. 4). On Cityscapes, hard samples concentrate in the high-μ_{12} tail (+0.63 SD), and Cityscapes-DVPS (Depth) shows the same trend more weakly. On PascalVOC, the relation reverses, with hard samples in the low tail (up to +0.80 SD). On CIFAR10 and SemanticKITTI-DVPS, the relation is weak or absent. We omit Cityscapes-DVPS (Segm.), where the difficulty scores disagree with each other. This dataset dependence motivates offering both a coverage-based and a fidelity-based selection procedure.

3.4 CoreDNA Coreset Selection Procedure

Given our prior analysis of where the hardest samples lie and whether fidelity or coverage can predict it, we experiment with two formulations: a fidelity-based one that matches the data distribution via the sliced Wasserstein distance, and a coverage-based one that spreads the coreset over the data via farthest-point sampling (FPS). Each axis of $z(x)$ is first standardized to unit variance over the dataset so that no axis dominates the distance. We apply a redundancy heuristic such that an axis is removed when its correlation with another exceeds 0.95, which holds for σ_{12} and μ_{12} on Cityscapes (all variants), NAVSIM, SemanticKITTI-DVPS, and often on LIBERO-10, ensuring that the remaining axes are decorrelated and each contributes independent information to selection. For fidelity, we select coresets by minimizing a sliced Wasserstein distance (Kolouri et al., 2019) between the coreset and the full dataset across all D dimensions. In 1D, the Wasserstein distance between equal-weight empirical measures equals the mean absolute quantile difference, minimized by picking the center of each of the P/C quantile bins. To extend this to $\mathbb{R}^D$, we use the sliced Wasserstein distance, drawing E unit directions $\theta_1, \ldots, \theta_E$, with θ_1 fixed to the primary axis and the rest random. For each direction, we score each point by its rank distance to the nearest quantile center, and sum these scores across directions. Since near-duplicates can occur, we scan points in increasing score order and skip any point within distance g of an already selected one, where $g = 0.01 \cdot (0.1/\alpha)^{1/D}$ is a unit-cube

spacing heuristic. If fewer than C points are selected, we relax the constraint and add skipped points in score order until C is reached. This selection strategy has a compute cost of $O(EP \log P)$; we show pseudocode in Appx. A.2.

Since fidelity does not correlate with performance for some datasets (Sec. 3.3), we introduce farthest-point sampling (FPS) as an independent, coverage-based selection method. Starting from an arbitrary point, we repeatedly add the point farthest from the current coreset, at $O(PC)$ total cost via a running minimum-distance array. FPS is a 2-approximation for the k-center objective and the standard heuristic for spread-out coverage. Unlike sliced Wasserstein selection, it maximizes diversity and is therefore biased toward sparse and extreme regions.

4 Experimental Evaluation

To assess the generality of CoreDNA, we evaluate it across a broad range of tasks (classification, segmentation, depth estimation, driving, manipulation), multiple data domains (real, synthetic), both curated and non-curated dataset types, and varying coreset sizes. We classify our benchmarks into three main categories: curated perception datasets, non-curated perception datasets, and imitation learning. Curated datasets are off-the-shelf and already a-priori filtered without access to the raw data, consisting of single images for which all labels are available (e.g., CIFAR10). Non-curated datasets contain raw sensor observations from a sequence, such as a video with framewise labels (e.g., Cityscapes-DVPS). Imitation learning tasks are a variant of non-curated datasets with an underlying Markov decision process, which we analyze independently (e.g., NAVSIM). We evaluate CoreDNA against nine baselines (*Random*, *Forgetting*, *EL2N*, *AUM*, *CCS*, *BOSS*, *ELFS*, *FreeSel*, and *ZCore*) across 41 dataset-ratio settings spanning seven benchmarks (CIFAR10 (Krizhevsky, 2009), PascalVOC (Everingham et al., 2010), Cityscapes (Cordts et al., 2016), Cityscapes-DVPS (Qiao et al., 2021), SemanticKITTI-DVPS (Qiao et al., 2021), NAVSIM (Dauner et al., 2024), and LIBERO-10 (Liu et al., 2023)). All experiments are run over three seeds, and we report averages.

4.1 Performance On Perception Datasets

We compare performance on CIFAR10, Cityscapes, and PascalVOC across various retention ratios α in Tab. 2. BOSS cannot be trained at $\alpha = 0.01$ (24 images) since it does not function with such a small number of images. At higher retention ratios (0.5–0.7), results across all methods and datasets converge into a tight band, consistent with the intuition that keeping more data makes the choice of selection matter less. On CIFAR10 at low retention (0.01), our sliced-Wasserstein-based sampling outperforms all label-free baselines which we attribute to our representation's semantic sensitivity combined with density-based sampling. Purely hardness-based training methods are unsuitable at such low ratios, as their focus on difficult samples is not favorable for this task at this retention level. At higher ratios, training-dynamics-based methods perform better, while our method remains on par with other unsupervised methods while being cheaper and not requiring any additional training. For Cityscapes, we note that FPS sampling dominates. We observe that the hardest samples lie in the high tail (Sec. 3.3), and oversampling these tails appears important for this task, as reflected by the high scores of training-based methods which concentrate on high samples (EL2N, AUM, and Forgetting). Only at high ratios does fidelity-based sampling slightly outperform FPS. Our method outperforms both unsupervised and training-based methods at $\alpha = 0.7$ by at least 0.64 pp, showing that diversity-based sampling can even outperform hardness-based sampling at high retention rates. On PascalVOC, CoreDNA outperforms both training-based and unsupervised methods by a margin of 2 pp at $\alpha = 0.1$, demonstrating superior selection. For this task as well, fidelity-based sampling is clearly better than farthest-point sampling, which we attribute to spurious outliers hurting performance under FPS. More broadly, we observe that neither FPS nor sliced Wasserstein sampling dominates universally, with the better choice depending on the specific dataset and task. However, CoreDNA with either selection consistently performs well across all settings, unlike several baselines whose performance varies substantially. In many cases, CoreDNA outperforms considerably stronger and more cumbersome baselines, and remains competitive even in the settings where it does not lead.

We now turn to non-curated, sequence-based data, where redundancy between consecutive frames introduces properties absent from curated benchmarks. We compare Cityscapes-DVPS and SemanticKITTI-DVPS for monocular metric depth estimation and semantic segmentation in Tab. 3. Depth estimation is a regression problem, for which the classification-specific methods *AUM*, *CCS*,

Table 2: Downstream accuracy (%) across coreset ratios for CIFAR10, Cityscapes, and PascalVOC. Full: training on all data. WS: Sliced Wasserstein sampling. FPS: farthest-point-sampling.

		CIFAR10 ↑ (Full: 95.05)							Cityscapes ↑ (Full: 76.22)					PascalVOC ↑ (Full: 79.01)				
	Method	0.01	0.05	0.1	0.2	0.3	0.5	0.7	0.01	0.1	0.3	0.5	0.7	0.01	0.1	0.3	0.5	0.7
Ref.	Random	41.41	71.21	80.90	88.17	90.69	93.11	94.43	48.81	65.33	71.87	73.68	74.87	40.02	55.56	74.81	76.65	78.06
Training w/ labels	AUM	16.10	22.80	31.93	60.54	**92.53**	**95.24**	95.28	50.96	69.38	71.41	**75.42**	74.00	26.51	29.03	71.73	76.84	78.14
	Forgetting	25.88	30.95	39.50	68.78	90.35	95.08	95.32	50.42	69.33	70.61	74.75	74.68	25.44	31.22	75.21	**77.30**	**78.26**
	EL2N	13.94	18.16	26.31	68.37	90.34	94.90	**95.34**	**51.51**	**69.88**	71.33	75.21	73.89	16.38	29.12	72.93	77.12	**78.26**
	CCS	41.68	74.88	86.49	**88.85**	91.03	91.16	91.40	47.58	61.78	72.04	71.50	74.63	44.01	43.15	73.94	75.47	78.05
	BOSS	**45.75**	**78.02**	84.28	87.63	91.43	94.32	95.33	-	64.01	66.82	69.93	70.11	46.19	56.44	75.24	76.75	77.77
Training w/o labels	ELFS AUM	35.21	74.82	**86.66**	87.74	89.21	89.25	91.13	46.38	64.77	71.55	74.68	74.28	45.05	53.00	**75.56**	76.93	77.90
	ELFS Forget	34.72	76.09	86.60	87.52	89.12	89.01	90.95	46.27	64.96	71.92	74.37	73.92	44.92	49.94	75.45	76.94	78.00
Training- & label-free	ZCore	40.60	66.25	83.56	87.95	91.43	93.39	94.45	48.19	67.07	72.01	74.03	74.56	39.98	57.52	75.31	76.76	78.01
	FreeSel	40.62	70.57	82.66	88.26	91.64	93.38	94.35	47.98	64.27	71.90	74.48	74.68	**47.54**	55.79	74.94	76.70	77.95
	CoreDNA (WS)	41.95	68.61	82.40	87.95	91.24	93.49	94.40	49.92	66.98	72.42	74.40	**75.32**	44.19	**59.21**	75.36	77.09	78.16
	CoreDNA (FPS)	41.65	67.42	80.62	86.79	90.98	93.30	94.39	50.20	67.35	**72.76**	74.46	75.08	37.21	57.60	73.99	76.46	77.81

Table 3: Downstream accuracy across coreset ratios for Cityscapes-DVPS and SemanticKITTI-DVPS. Segmentation is evaluated in mIoU [%] and monocular metric depth estimation in RMSE. Full: training on all data. WS: Sliced Wasserstein sampling. FPS: farthest-point-sampling.

		Cityscapes-DVPS Segm. ↑ (Full: 66.43)					Cityscapes-DVPS Depth ↓ (Full: 3.91)			SemanticKITTI-DVPS Segm. ↑ (Full: 55.58)					SemanticKITTI-DVPS Depth ↓ (Full: 3.39)		
	Method	0.01	0.1	0.3	0.5	0.7	0.01	0.1	0.3	0.01	0.1	0.3	0.5	0.7	0.01	0.1	0.3
Ref.	Temporal Stride	38.28	**56.71**	**63.83**	64.47	65.02	6.17	**4.28**	**3.97**	38.74	**53.14**	**55.28**	55.80	56.32	3.67	**3.34**	3.35
	Random	37.88	55.77	61.23	64.03	64.83	5.70	4.36	4.02	36.81	51.17	55.08	55.98	55.90	3.70	3.36	3.36
Training w/ labels	AUM	35.54	51.28	57.52	60.56	63.21	-	-	-	27.76	42.85	51.03	53.45	55.06	-	-	-
	Forgetting	32.86	45.70	56.63	60.15	62.63	5.56	4.44	4.12	26.44	40.35	46.26	52.84	54.46	3.95	3.45	3.34
	EL2N	32.84	52.01	58.99	62.00	64.65	7.40	5.74	5.15	28.07	43.58	51.12	53.55	55.04	3.74	3.52	3.40
	CCS	37.98	50.43	58.15	60.53	62.16	-	-	-	27.76	50.63	54.69	55.18	55.37	-	-	-
	BOSS	34.47	51.13	57.53	57.70	58.39	-	-	-	38.07	51.04	52.93	52.30	54.35	-	-	-
Training w/o labels	ELFS AUM	36.53	52.77	59.60	62.28	64.89	6.28	4.36	4.07	38.23	46.96	50.82	52.63	54.27	3.77	3.42	3.37
	ELFS Forget	36.65	52.64	59.79	62.76	64.18	6.29	4.36	4.07	38.31	47.31	54.05	55.65	55.74	3.74	3.43	3.39
Training- & label-free	ZCore	38.07	55.05	61.54	63.33	64.30	5.81	4.43	4.08	38.69	52.47	54.89	54.90	56.24	**3.60**	3.39	3.37
	FreeSel	38.12	56.30	63.18	63.36	64.40	**5.02**	4.31	**3.97**	**39.33**	51.74	55.16	**56.22**	55.90	3.74	3.35	**3.33**
	CoreDNA (WS)	38.18	55.66	61.70	64.00	64.52	6.05	4.35	4.05	37.56	50.02	54.90	55.50	55.62	3.73	3.39	3.35
	CoreDNA (WS+T)	**38.51**	55.38	62.04	**64.77**	**65.22**	6.10	4.39	4.08	37.73	51.37	55.16	55.99	**56.39**	3.77	3.36	3.35
	CoreDNA (FPS)	37.36	53.79	61.09	63.26	64.90	6.20	4.40	4.06	38.89	49.50	54.05	55.65	55.74	3.74	3.41	3.40

and *BOSS* cannot be applied. SemanticKITTI-DVPS reaches within ± 1 pp of full-dataset performance at $\alpha = 0.3$ for segmentation, and already at $\alpha = 0.1$ for depth, showing that many samples do not meaningfully contribute to training. For all four non-curated datasets, training-free methods outperform training-based methods in all settings, showing that diversity-based selection is more effective than hardness-based selection when redundant data is present. We further observe that random selection outperforms training-based methods for depth estimation, indicating that these methods are limited to classification-style problems. Further, temporal stride (i.e., selecting every P/C-th frame) is a strong baseline which is consistent with consecutive frames being highly uniform while sufficient diversity still emerges over time. We therefore evaluate adding a frame index as a fourth dimension (WS+T), which spreads selected frames further apart in time and outperforms pure WS in 9 of 10 segmentation settings and 2 of 6 depth settings. In general, temporal stride is difficult to beat in the mid-range ($\alpha = 0.1$–0.3) across all four tasks, while unsupervised methods perform better at the extremes ($\alpha = 0.01$ and $\alpha \geq 0.5$). CoreDNA performs consistently well across this entire range, and particularly excels at high retention ratios on Cityscapes-DVPS segmentation, outperforming baselines by at least 0.82 pp at $\alpha \geq 0.5$.

4.2 Performance on Imitation Learning

To demonstrate CoreDNA's applicability, we apply it to end-to-end driving and visuo-motor policy learning. Since these policies process frames independently, we apply CoreDNA at the frame level.

Autonomous Driving: For end-to-end driving, we complement the features from images with the ego *state*, describing how the vehicle moves (velocity, acceleration, and heading), and its *position*, describing where the vehicle is in global map coordinates. We report results using the common Extended Predictive Driver Model Score (EPDMS) of Latent TransFuser (Chitta et al., 2023) on the NAVSIM framework in Tab. 4. With farthest-point sampling, selecting 50% of the data nearly matches training on the full dataset. Combining the image representation with both state and position performs best at $\alpha = 0.1$, whereas at $\alpha = 0.5$, excluding position from the selection is preferable.

We attribute this to $\alpha = 0.1$ benefiting from explicitly selecting for diverse locations at such a low sample count, whereas at $\alpha = 0.5$, sufficient location diversity is already achieved naturally, making the position axis redundant. In general, FPS outperforms WS and the random baseline at both ratios, indicating that picking rare samples is helpful for this task, and CoreDNA can significantly improve autonomous driving policy training efficiency with minimal performance drop.

Visuo-Motor Policies: For visuo-motor policy learning, we demonstrate the capabilities of CoreDNA on all 10 available tasks of the LIBERO-10 benchmark, which comprises long-horizon manipulation tasks. We perform two experiments aligned with DataMIL's (Dass et al., 2026) 5-demonstration setup, i.e. the base for both experiments includes a set of 5 out of the 50 available demonstrations for each LIBERO-10 task. We use a retention rate $\alpha = 0.9$ for all experiments. We show major results in Tab. 5 and full results in Sec. A.5. In the first experiment, we select frames from the pooled dataset of all selected 5 demonstrations of the LIBERO-10 task and extend and ablate CoreDNA's formulation against combinations of the external camera image and the agent's wrist camera image. Additionally, we show results when performing frame selection on all 50 LIBERO-10 demonstrations. We observe that CoreDNA improves over or is on-par with full frame sampling for both the 5 and 50 demonstration variant. We note that the retention rate itself may be suboptimal, and other values could yield even better results. Nonetheless, these results demonstrate that CoreDNA can improve performance over naively training using all available frames.

Table 4: Downstream driving performance on NAVSIM. Img, State, and Pos. indicate the coreset selection axes.

Method	Img	State	Pos.	EPDMS ↑ 0.1	0.5
Full				80.51	
Random				65.44	78.88
CoreDNA (WS)	✓			66.31	78.17
CoreDNA (WS)	✓	✓		65.29	78.90
CoreDNA (WS)	✓	✓	✓	<u>67.03</u>	78.62
CoreDNA (FPS)	✓			66.98	78.60
CoreDNA (FPS)	✓	✓		65.42	**79.98**
CoreDNA (FPS)	✓	✓	✓	**67.43**	<u>79.76</u>

In our second experiment, we follow DataMIL's experimental setup and keep all 5 demonstrations fixed and rather select an auxiliary subset of 10% demonstrations from the full LIBERO-90 demonstration corpus to mix with the 5 LIBERO-10 demonstrations during training. Thus, the focus shifts as the goal is no longer to select the most diverse frames from a training set but rather the best auxiliary mix-in demos. As a result, we alter our original formulations (WS and FPS) and select demonstrations from the LIBERO-90 dataset which rank closest to the five LIBERO-10 demonstrations, rather than matching LIBERO-90's own distribution. We denote this variant CoreDNA (CR). In this experiment, we observe a stark increase in the success rate, aligned with current observations regarding data composition for imitation learning (Hejna et al., 2024). This shows that pooling activations across a demonstration's frames also yields a good representation of sample quality for imitation learning, which even outperforms the training-based method DataMIL (Dass et al., 2026).

Table 5: Mean success rate across all tasks in LIBERO-10.

Method	External Img	Wrist Img	Success Rate ↑
LIBERO-10 Frame-level sampling			
Full (5 demos)			16.9
CoreDNA (WS)	✓		**17.3**
CoreDNA (WS)	✓	✓	16.8
CoreDNA (FPS)	✓	✓	<u>17.1</u>
LIBERO-10 Frame-level sampling (50 demos)			
Full (50 demos)			76.7
CoreDNA (WS)	✓	✓	**77.8**
LIBERO-90 Demonstration-level sampling (mix-in)			
All			18.3
DataMIL	✓	✓	<u>25.3</u>
CoreDNA (WS)	✓	✓	19.0
CoreDNA (CR)	✓	✓	**27.3**

5 Conclusion

In this work, we investigated pooling neuron activations into per-image and per-dataset distributions for one-shot, task-agnostic coreset estimation. We find that the last layer's mean, standard deviation, and cross-layer rank agreement correlate best with downstream performance, which we attribute to hard samples concentrating in the tails of their distribution. Further, we introduced recording priors, such as frame index or position, as an additional axis to address redundancy in non-curated datasets. We employed farthest-point and sliced Wasserstein sampling to construct high-quality coresets. We demonstrated that our selection is task- and data-type-agnostic on image classification, semantic segmentation, depth estimation, and imitation learning, outperforming most baselines while remaining computationally efficient. We believe these findings can help extend coreset estimation beyond curated datasets and toward real-world action domains.

AI USE STATEMENT

In this work, we used generative AI tools for writing assistance and implementation of methods. We have not used generative AI tools for related work discovery or for generating research ideas. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We include implementation details in Section A.2 and A.3 in the Appendix. We make the code publicly available upon acceptance.

REFERENCES

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023.

Amro Abbas, Evgenia Rusak, Kushal Tirumala, Wieland Brendel, Kamalika Chaudhuri, and Ari S. Morcos. Effective pruning of web-scale datasets based on complexity of concept clusters. In *International Conference on Learning Representations*, 2024.

Abhinab Acharya, Dayou Yu, Qi Yu, and Xumin Liu. Balancing feature similarity and label variability for optimal size-aware one-shot subset selection. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 96–116, 21–27 Jul 2024.

Anish Acharya, Sujay Sanghavi, Alex Dimakis, and Inderjit S Dhillon. Geometric median (GM) matching for robust k-subset selection from noisy data. In *Forty-second International Conference on Machine Learning*, 2025.

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. 2020.

Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

Jinhe Bi, Yifan Wang, Danqi Yan, Xun Xiao, Artur Hecker, Volker Tresp, and Yunpu Ma. Prism: Self-pruning intrinsic selection method for training-free multimodal data selection. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2026.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving. 2025.

Wei-Kai Chang and Rajiv Khanna. Stable coresets via posterior sampling: Aligning induced and full loss landscapes. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. 45(11), 2023.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Yalun Dai, Lingao Xiao, Ivor Tsang, and Yang He. Training-free dataset pruning for instance segmentation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Shivin Dass, Alaa Khaddaj, Logan Engstrom, Aleksander Madry, Andrew Ilyas, and Roberto Martín-Martín. DataMIL: Selecting data for robot imitation learning with datamodels. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=AcTsKglDdh.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: data-driven non-reactive autonomous vehicle simulation and benchmarking. 2024.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

Brent A. Griffin, Jacob Marks, and Jason J. Corso. Zero-shot coreset selection via iterative subspace sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2026.

Yuxian Gu, Li Dong, Hongning Wang, Yaru Hao, Qingxiu Dong, Furu Wei, and Minlie Huang. Data selection via optimal control for language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Qiang He, Yaozong Yang, Kaibin Wang, Ziteng Wei, Feifei Chen, Caslon Chua, and Yun Yang. Loprune: Efficient data pruning for lora-based fine-tuning of vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5242–5252, June 2026.

Joey Hejna, Chethan Anand Bhateja, Yichen Jiang, Karl Pertsch, and Dorsa Sadigh. Remix: Optimizing data mixtures for large scale imitation learning. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=fIj88Tn3fc.

Julia Hindel, Rohit Mohan, Jelena Bratulić, Daniele Cattaneo, Thomas Brox, and Abhinav Valada. Label-efficient lidar semantic segmentation with 2d-3d vision transformer adapters. pp. 4115–4122, 2025.

Mohammad Jafari, Yimeng Zhang, Yihua Zhang, and Sijia Liu. The power of few: Accelerating and enhancing data reweighting with coreset selection. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7100–7104, 2024.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):8110–8118, 2021.

Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo Kunde Rohde. Generalized sliced wasserstein distances. In *Neural Information Processing Systems*, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36: 44776–44791, 2023.

Zhenyu Liu, Yunxin Li, Baotian Hu, Wenhan Luo, Yaowei Wang, and Min Zhang. Picking the cream of the crop: Visual-centric data selection with collaborative agents, 2025.

Zifan Liu, Amin Karbasi, and Theodoros Rekatsinas. TSDS: Data selection for task-specific model finetuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Adyasha Maharana, Prateek Yadav, and Mohit Bansal. $\mathbb{D}^2$ pruning: Message passing for balancing diversity & difficulty in data pruning. In *The Twelfth International Conference on Learning Representations*, 2024.

Ailar Mahdizadeh, Puria Azadi, Muchen Li, Xiangteng He, and Leonid Sigal. Cords: Coreset-based representative and diverse selection for streaming video understanding, 2026. URL https://arxiv.org/abs/2605.14310.

Pratyush Maini, Sachin Goyal, Zachary Chase Lipton, J Zico Kolter, and Aditi Raghunathan. T-MARS: Improving visual representations by circumventing text feature learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Kristof Meding, Luca M. Schulze Buschoff, Robert Geirhos, and Felix A. Wichmann. Trivial or impossible—dichotomous data difficulty masks model differences (on imagenet and beyond). In *International Conference on Learning Representations*, 2022.

Baharan Mirzasoleiman, Jeff A. Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, 2019.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

Geoff Pleiss, Tianyi Zhang, Ethan R. Elenberg, and Kilian Q. Weinberger. Identifying mislabeled data using the area under the margin ranking. In *Neural Information Processing Systems*, 2020.

Niclas Popp, Dan Zhang, Jan Hendrik Metzen, Matthias Hein, and Lukas Schott. Single-pass object-focused data selection, 2025.

Xuan Qi, Luxi He, Dan Roth, and Xingyu Fu. Demystifying supervision data generalization in multimodal LMs. In *The Fourteenth International Conference on Learning Representations*, 2026.

Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. VIP-DeepLab: Learning Visual Perception with Depth-Aware Video Panoptic Segmentation. pp. 3997–4008, 2021.

Benjamin Ramtoula, Matthew Gadd, Paul Newman, and Daniele De Martini. Visual DNA: Representing and comparing images using distributions of neuron activations. pp. 11113–11123, 2023.

Arjun Roy, Prajna G. Malettira, Manish Nagaraj, and Kaushik Roy. Topoprune: Robust data pruning via unified latent space topology, 2026.

Zahiriddin Rustamov, Nadia Badawi, Rafat Damseh, and Nazar Zaki. One-shot data selection for medical image classification via graph coverage. In *Medical Image Computing and Computer Assisted Intervention*, 2026.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 19523–19536, 2022.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJlxm30cKm.

Dongyue Wu, Zilin Guo, Xiaoyu Li, Jiajia Liu, Jingdong Chen, Nong Sang, and Changxin Gao. Selecting samples on graphs: A unified dataset pruning framework for lossless training acceleration, 2026. URL https://arxiv.org/abs/2606.12913.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning (ICML)*, 2024a.

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

Xiaobo Xia, Jiale Liu, Shaokun Zhang, Qingyun Wu, Hongxin Wei, and Tongliang Liu. Refined coreset selection: Towards minimal coreset size under model performance constraints. In *Forty-first International Conference on Machine Learning*, 2024b.

Tianchi Xie, Jiangning Zhu, Guozu Ma, Minzhi Lin, Wei Chen, Weikai Yang, and Shixia Liu. Structural-entropy-based sample selection for efficient and effective learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yichen Xie, Mingyu Ding, Masayoshi Tomizuka, and Wei Zhan. Towards free data selection with general-purpose models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a.

Yichen Xie, Han Lu, Junchi Yan, Xiaokang Yang, Masayoshi Tomizuka, and Wei Zhan. Active finetuning: Exploiting annotation budget in the pretraining-finetuning paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23715–23724, June 2023b.

Yichen Yan, Ming Zhong, Qi Zhu, Xiaoling Gu, Jinpeng Chen, and Huan Li. Coido: Efficient data selection for visual instruction tuning via coupled importance-diversity optimization. In *Advances in Neural Information Processing Systems*, 2025.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *Advances in Neural Information Processing Systems*, volume 37, pp. 21875–21911. Curran Associates, Inc., 2024.

Yu Yang, Hao Kang, and Baharan Mirzasoleiman. Towards sustainable learning: Coresets for data-efficient deep learning. In *International Conference on Machine Learning*, 2023.

Jihan Yao, Gantavya Bhatt, Arnav Das, Peter Jin, Ke Bao, Qiaolin Yu, Khushi Bhardwaj, Chang Su, Jialei Wang, Yikai Zhu, Sugam Devare, Damon Mosk-Aoyama, Zhen Dong, Venkat Krishna Srinivasan, Yineng Zhang, Oleksii Kuchaiev, Jiantao Jiao, Banghua Zhu, and Jeff Bilmes. Coresets before score sets: Evaluation-unsupervised prompt subset selection for llm benchmarks, 2026.

Jin Zhang, Zhe Cao, Biwen Yang, and Ruiheng Zhang. Annotation-efficient coreset selection for context-dependent segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20410–20420, June 2026a.

Jipeng Zhang, Yaxuan Qin, Renjie Pi, Weizhong Zhang, Rui Pan, and Tong Zhang. Tagcos: Task-agnostic gradient clustered coreset selection for instruction tuning data. *ArXiv*, abs/2407.15235, 2024.

Xiaoyu Zhang, Juan Zhai, Shiqing Ma, Chao Shen, Tianlin Li, Weipeng Jiang, and Yang Liu. STAFF: Speculative coreset selection for task-specific fine-tuning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yuan Zhang, Lifeng Guo, Junwen Pan, Wenzhao Zheng, Wen Zhou, Kuan Cheng, Kurt Keutzer, and Shanghang Zhang. Seed: Targeted data selection by weighted independent set, 2026b. URL https://arxiv.org/abs/2605.15691.

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

Haizhong Zheng, Elisa Tsai, Yifu Lu, Jiachen Sun, Brian R. Bartoldson, Bhavya Kailkhura, and Atul Prakash. ELFS: Label-free coreset selection with proxy training dynamics. In *The Thirteenth International Conference on Learning Representations*, 2025.

Pan Zhou, Yichen Zhou, Chenyang Si, Weihao Yu, Teck Khim Ng, and Shuicheng Yan. Mugs: A multi-granular self-supervised learning framework. In *arXiv preprint arXiv:2203.14415*, 2022a.

Xiao Zhou, Renjie Pi, Weizhong Zhang, Yong Lin, and Tong Zhang. Probabilistic bilevel coreset selection. In *International Conference on Machine Learning*. PMLR, 2022b.

Ning Zhu, Xiaochuan Ma, Juntao Xu, Jingze Liang, Mengfei Zhao, An Chen, and Liang-Jian Deng. One knob to rule them all: A unified optimal transport view of cold-start active learning, 2026.

A Appendix

A.1 Overview

We describe additional CoreDNA selection settings in Sec. A.2, and the experimental setup for our tasks in Sec. A.3. We then show extended experimental results for end-to-end driving in Sec. A.4 and visuomotor policy learning in Sec. A.5. We also expand on our performance correlation experiments from the paper in Sec. A.6, detailing selection criteria and per-dataset results. We then compare the rankings of image patterns produced by different backbones and justify our choice of histogram bin size in Sec. A.7. Sec. A.8 shows ablations of our axis choices and selection mechanism. Finally, we show that training-based methods struggle with redundancy, using an artificial setup of duplicated CIFAR10 in Sec. A.9.

Require: Points $F = \{z_i\}_{i=1}^{P} \subset [0,1]^D$, budget C, projections E, minimum gap g
Ensure: Selected index set $S \subseteq \{1, \dots, P\}$ with $|S| = C$
Sample $E-1$ random unit directions $\theta_2, \dots, \theta_E$; set $\theta_1 \leftarrow e_1$
$w \leftarrow P/C$; $\quad s_i \leftarrow 0$ for all i
for $e = 1, \dots, E$ **do**
  Rank all points by $\theta_e^\top z_i$, giving ranks $\rho_i \in \{0, \dots, P-1\}$
  $s_i \leftarrow s_i + \min_j \left| \rho_i - (j + \frac{1}{2})w \right|$
end for
$\pi \leftarrow$ indices sorted by s_i ascending; $S \leftarrow \emptyset$
for $i \in \pi$ **while** $|S| < C$ **do**
  if z_i is at least g from every point already in S **then**
    $S \leftarrow S \cup \{i\}$
  end if
end for
return S, filling any remaining slots by score if g prevented reaching C

Algorithm 1: Sliced Wasserstein coreset selection

A.2 Additional Hyperparameters and Pseudo-Code

We use $B = 16$ bins to produce the per-image histograms $h_{\ell n}$ and per-dataset histograms $H_{\ell n}$, and ablate this choice in Sec. A.7. For the sliced Wasserstein distance, we set the number of projections $E = 10$ and ablate additional settings in Sec. A.8. We additionally show the pseudocode of our sliced Wasserstein selection algorithm in Algo. 1.

A.3 Detailed Experimental Setting

We evaluate our method on image classification, semantic segmentation, metric depth estimation, end-to-end autonomous driving and imitation learning. We use the authors' published code for all baselines. We train all experiments over 3 seeds to remove the effect of training-specific variations. We train all coresets for a dataset with consistent numbers of iterations. We detail the set-up for each below.

Image classification. Following prior work (Zheng et al., 2023; Xie et al., 2023a; Zheng et al., 2025), we use a ResNet18 (He et al., 2016) architecture to train a network on CIFAR10 (Krizhevsky, 2009) with Cross Entropy loss. We train for 40,000 iterations with a learning rate of 0.1 using an SGD optimizer with momentum 0.9, weight decay of 5e−4, and cosine annealing learning rate. We apply a random 32×32 crop with 4-pixel reflect padding and a random horizontal flip, followed by mean and standard-deviation normalization. We report results in terms of accuracy.

Semantic Segmentation. We perform semantic segmentation on PascalVOC (Everingham et al., 2010), Cityscapes (Cordts et al., 2016), Cityscapes-DVPS (Qiao et al., 2021), and SemanticKITTI-DVPS (Qiao et al., 2021). We train a DeepLabv3 (Chen et al., 2017) with a ResNet-101 backbone and a batch size of 16 with Cross Entropy loss. We use a learning rate of 0.03 for PascalVOC, 0.1 for Cityscapes and SemanticKITTI-DVPS, and 0.3 for Cityscapes-DVPS. We set the backbone's learning rate to $0.1\times$ the head's. We train for 39,720 iterations on PascalVOC, 11,160 on Cityscapes, 9,000 on Cityscapes-DVPS, and 23,920 on SemanticKITTI-DVPS, corresponding to 60 epochs on the full training set for PascalVOC, Cityscapes, and Cityscapes-DVPS, and 30 epochs for SemanticKITTI-DVPS. All four segmentation datasets share a random horizontal flip ($p = 0.5$) and ImageNet-statistics normalization as base augmentations. PascalVOC and Cityscapes-DVPS (Segm.) use a random resized crop (PascalVOC: 513×513, scale $[0.5, 2.0]$; Cityscapes-DVPS (Segm.): 512×512, scale $[0.5, 2.0]$) with no color augmentation. Cityscapes additionally applies color jitter (brightness, contrast, saturation $= 0.5$) on top of a 768×768 random resized crop. SemanticKITTI-DVPS (Segm.) instead resizes to 370×1226 and takes a random 370×784 crop, combined with milder color jitter (brightness, contrast, saturation, hue $= 0.1$). We use an SGD optimizer with a polynomial learning rate schedule. We report mean Intersection over Union (mIoU). To extend *Forgetting*, *EL2N*, and *AUM* beyond classification, we define correctness as at least 80% correctly classified pixels for segmentation.

Metric Depth estimation. We perform metric depth estimation on SemanticKITTI-DVPS and Cityscapes-DVPS. We train a ZoeDepth-Small network (Bhat et al., 2023) with a Depth Anything V2 backbone (Yang et al., 2024), using a scale-invariant log loss with a weight of 10. We train for 23,920 iterations on SemanticKITTI-DVPS and 4,500 on Cityscapes-DVPS (amounting to 30 epochs on the full dataset), using a batch size of 16, a learning rate of 2e−5 for the head, and 4e−7 for the encoder. Both depth datasets share a heavier augmentation

pipeline than their segmentation counterparts: a random horizontal flip ($p = 0.5$), a random affine transform (rotation $\pm 2.5°$, translation 5%, scale $[0.95, 1.05]$), color jitter (brightness, contrast, saturation, hue $= 0.1$), and random gamma correction (range $[0.8, 1.2]$), followed by ImageNet-statistics normalization. Cityscapes-DVPS uses a random resized crop to 518×518 (scale $[0.5, 2.0]$), while SemanticKITTI-DVPS resizes to 350×1204 and takes a random 350×518 crop. We use an Adam optimizer with a cosine annealing learning rate scheduler. We report Root Mean Squared Error (RMSE) as the primary metric. For depth estimation, we define correctness as an absolute relative error below 0.1 for at least 10% of the image, a threshold determined empirically.

Imitation Learning – Driving. We perform end-to-end driving on the NAVSIM benchmark. We train on the `navtrain` split, which includes both training and validation scenes, and evaluate on the held-out `navtest` split. For the baseline model, we use Latent TransFuser (Chitta et al., 2023), which uses only camera inputs without LiDAR. We train for 50 epochs with a batch size of 128 and a learning rate of $1e-4$. We refer the reader to the original work for further training details (Cao et al., 2025; Chitta et al., 2023). We report the Extended Predictive Driver Model Score (EPDMS) as the primary metric (Dauner et al., 2024; Cao et al., 2025).

Imitation Learning – Manipulation. We perform end-to-end closed-loop visuo-motor policy training on the LIBERO-10 benchmark (Liu et al., 2023) using the OCTO model (Octo Model Team et al., 2024). We fine-tune OCTO using off-the-shelf parameters following the DataMIL benchmark (Dass et al., 2026) configuration, thus we use an action chunking head of size eight, a batch size of 128 and an annealing cosine learning rate schedule with an initial learning rate of $3e-4$. We evaluate all trained policies on the LIBERO-10 benchmark for 50 evaluation roll-outs and report the average success rate over them.

For selecting the five demonstration subset of all 50 demonstrations for each LIBERO-10 task, we could not rely on DataMIL's (Dass et al., 2026) selection as it was unavailable to us. Instead, we first filter all failed demonstrations by replaying each action sequence from the initial simulator state and only keep the ones that are successful. Second, we manually go through the remaining demonstrations and select the most diverse and relevant ones, as a human would if they had to collect them, knowing one has a budget of 5 demonstrations. For future reproducibility we report all selected demonstration indices in Tab. 6.

We further include the robot's proprioception (joint states) through PCA projection in the axis selection in Sec. A.5.

A.4 Detailed Performance on End-to-End Driving

EPDMS (Dauner et al., 2024; Cao et al., 2025) consists of No at-fault Collisions (NC), Drivable Area Compliance (DAC), Driving Direction Compliance (DDC), Traffic Light Compliance (TLC), Ego Progress (EP), Time to Collision (TTC), Lane Keeping (LK), History Comfort (HC), and Extended Comfort (EC). For each scenario, the score is computed as

$$\text{EPDMS} = \prod_{m \in \mathcal{M}_{\text{mult}}} \tilde{m} \cdot \frac{\sum_{m \in \mathcal{M}_{\text{wt}}} w_m \tilde{m}}{\sum_{m \in \mathcal{M}_{\text{wt}}} w_m}, \tag{3}$$

where $\mathcal{M}_{\text{mult}} = \{\text{NC}, \text{DAC}, \text{DDC}, \text{TLC}\}$ are multiplier metrics and $\mathcal{M}_{\text{wt}} = \{\text{EP}, \text{TTC}, \text{LK}, \text{HC}, \text{EC}\}$ are weighted metrics with $w = \{5, 5, 2, 2, 2\}$. $\tilde{m}$ denotes the score of the driving model after false-positive filtering, which sets the score to 1 whenever the human driver also violates the same metric in that scenario. The final EPDMS is averaged over all scenarios.

Table 6: Selected demonstration indices for each LIBERO-10 task.

Task	Demos
Book–Caddy	{7, 8, 19, 27, 41}
Bowl–Cabinet	{11, 26, 32, 38, 43}
Mug–Microwave	{23, 27, 37, 44, 48}
Stove–Moka	{19, 27, 30, 40, 43}
Soup–Sauce	{3, 11, 25, 44, 45}
Cream–Butter	{8, 13, 29, 36, 42}
Mug–Mug	{9, 12, 16, 17, 23}
Mug–Pudding	{4, 5, 11, 12, 31}
Soup–Cheese	{1, 2, 22, 23, 44}
Moka–Moka	{7, 9, 13, 20, 44}

Table 7 reports all sub-metrics. Reducing the training data primarily affects DAC and EC, whereas it has little effect on TLC and HC. TLC and HC are nearly saturated even with 10% of the data. In contrast, DAC, which requires precise perception of road boundaries, and EC, which is sensitive to noisy trajectory predictions, degrade markedly with less data. Compared to random selection, at $\alpha = 0.1$, the best configuration, CoreDNA (FPS) with image, state, and position, improves on the safety-related metrics NC ($+1.1$), DAC ($+1.5$), and TTC ($+1.4$), as well as on EC ($+3.6$), while EP is slightly lower (-0.4). At $\alpha = 0.5$, the best configuration, CoreDNA (FPS) with image and state, again gains mainly on DAC ($+1.1$), with smaller improvements on NC, DDC, and TTC, while EP is again slightly lower (-0.7). In both cases, CoreDNA thus trades a small amount of progress for safer driving behavior.

A.5 Detailed Performance for End-to-End Manipulation Imitation Learning

We show full task results in Tab. 8. In addition to the findings in the main section of the paper, we observe that the full method only wins in four out of ten tasks for 5 demonstrations and in three out of ten tasks for

Table 7: Per-metric end-to-end driving performance on NAVSIM.

Method	Img	State	Pos.	Ratio	NC ↑	DAC ↑	DDC ↑	TLC ↑	EP ↑	TTC ↑	LK ↑	HC ↑	EC ↑	EPDMS ↑
Full				–	96.99	89.20	99.07	99.74	87.81	95.86	95.50	98.34	85.95	80.51
Random				0.1	93.23	78.27	94.31	99.65	86.27	91.22	88.61	98.27	72.67	65.44
CoreDNA (WS)	✓			0.1	93.64	78.88	94.69	99.64	86.20	92.01	89.95	98.28	67.53	66.31
CoreDNA (WS)	✓	✓		0.1	93.03	77.97	94.01	99.60	**86.81**	91.38	88.58	98.28	73.32	65.29
CoreDNA (WS)	✓	✓	✓	0.1	93.69	79.31	94.14	99.65	86.01	92.10	89.24	98.27	**77.09**	67.03
CoreDNA (FPS)	✓			0.1	93.60	79.09	**94.91**	99.65	86.22	92.03	90.35	98.28	74.04	66.98
CoreDNA (FPS)	✓	✓		0.1	93.18	77.53	94.24	99.64	85.97	91.26	**90.53**	98.29	73.80	65.42
CoreDNA (FPS)	✓	✓	✓	0.1	**94.29**	**79.77**	94.20	**99.66**	85.87	**92.61**	89.28	**98.31**	76.31	**67.43**
Random				0.5	96.73	88.02	98.17	99.72	**87.71**	95.43	94.41	98.31	86.83	78.88
CoreDNA (WS)	✓			0.5	96.62	87.67	97.97	99.69	87.61	95.16	92.83	98.33	86.14	78.17
CoreDNA (WS)	✓	✓		0.5	96.67	88.02	98.29	99.73	87.25	95.54	93.79	98.33	86.64	78.90
CoreDNA (WS)	✓	✓	✓	0.5	96.50	87.96	98.37	99.73	87.51	95.21	94.21	98.32	86.21	78.62
CoreDNA (FPS)	✓			0.5	96.58	87.85	98.04	99.71	87.69	95.44	94.04	98.31	85.93	78.60
CoreDNA (FPS)	✓	✓		0.5	**96.93**	**89.11**	**98.57**	99.73	87.05	95.65	**94.51**	**98.34**	**86.98**	**79.98**
CoreDNA (FPS)	✓	✓	✓	0.5	96.85	88.90	98.23	**99.74**	87.58	**95.72**	93.86	98.31	86.28	79.76

Table 8: Mean success rate across all tasks in LIBERO-10, with per-task results, split across all experiments. We report different variations of CoreDNA's modality input, the external camera image, the agent's wrist camera image and its joint state.

Method	Ext Img	Wrist Img	Joint State	Soup–Sauce	Cream–Butter	Stove–Moka	Bowl–Cabinet	Mug–Mug	Book–Caddy	Mug–Pudd.	Soup–Cheese	Moka–Moka	Mug–Micr.	Success Rate ↑
LIBERO-10 Frame-level sampling														
Full (5 demos)				9.3±5.0	**25.3**±11.6	30.7±9.0	36.7±16.7	**6.0**±7.1	17.3±3.4	1.3±1.9	**14.0**±2.8	**10.0**±6.5	18.7±6.2	16.9±7.0
CoreDNA (WS)	✓			16.0±11.8	21.3±5.2	32.0±2.8	40.0±5.9	**2.7**±3.8	19.3±4.1	2.7±0.9	10.7±0.9	6.0±5.7	**22.0**±11.8	**17.3**±5.3
CoreDNA (WS)	✓	✓		**19.3**±7.7	17.3±10.5	**39.3**±1.9	44.7±7.4	1.3±1.9	22.0±8.6	0.7±0.9	7.3±2.5	5.3±0.9	10.7±0.9	16.8±4.3
CoreDNA (FPS)	✓	✓		18.0±2.8	23.3±2.5	34.0±4.9	**51.3**±12.7	0.7±0.9	20.0±1.6	0.0±0.0	11.3±4.1	4.7±1.9	8.0±4.3	17.1±3.6
CoreDNA (WS)	✓	✓	✓	15.3±7.5	24.0±7.1	30.0±7.5	39.3±5.0	0.0±0.0	18.7±8.4	0.0±0.0	11.3±6.6	8.0±3.3	9.3±2.5	15.6±4.8
CoreDNA (FPS)	✓	✓	✓	15.3±7.7	21.3±6.6	36.7±5.7	34.7±10.6	0.7±0.9	**25.3**±11.1	0.0±0.0	7.3±2.5	4.7±2.5	14.0±5.9	16.0±5.4
LIBERO-90 Demonstration-level sampling (mix-in)														
All				24.7±9.6	8.0±3.3	14.0±4.3	51.3±9.0	**3.3**±2.5	40.7±13.7	6.0±1.6	10.7±2.5	6.7±0.9	18.0±7.5	18.3±5.5
DataMIL	✓	✓	✓	31.3±3.4	9.3±2.5	**46.0**±9.1	59.3±12.3	**2.7**±3.8	**56.0**±12.8	4.0±2.8	12.7±2.5	**9.0**±1.0	22.7±4.7	25.3±5.5
CoreDNA (WS)	✓	✓		27.3±8.2	**23.5**±9.4	36.7±8.4	28.0±5.9	1.3±0.9	33.3±9.3	0.7±0.9	9.0±3.0	6.5±3.8	**24.0**±8.5	19.0±5.8
CoreDNA (CR)	✓	✓		**40.0**±9.8	18.0±13.4	40.7±15.2	**69.3**±6.6	2.0±1.4	44.7±15.7	**12.0**±10.7	**29.3**±13.2	6.0±2.8	10.7±1.9	**27.3**±9.1
LIBERO-10 Frame-level sampling (50 demos)														
Full (50 demos)				69.3±6.6	**88.7**±3.4	86.0±5.9	92.7±5.0	67.3±3.4	**85.3**±7.7	**50.0**±4.3	82.7±0.9	66.0±8.6	79.3±0.9	76.7±4.7
CoreDNA (WS)	✓	✓		**80.7**±7.5	83.3±12.7	**91.3**±0.9	**96.0**±1.6	64.7±7.4	76.0±4.3	38.7±3.4	**88.7**±6.8	**79.3**±5.0	79.5±1.7	**77.8**±5.1
CoreDNA (WS)	✓	✓	✓	73.3±4.1	83.3±6.2	88.7±3.8	83.3±5.0	**72.0**±9.9	81.3±6.6	34.0±8.2	87.3±4.1	70.0±7.1	**82.0**±4.3	75.5±5.9

50 demonstrations, following the trend of the mean success rate across all tasks. The same observation can also be made for the LIBERO-90 mix-in experiment, where only on one task the full set is winning. Further, Data-MIL (Dass et al., 2026) shows the best performance on three out of ten tasks.

We attribute the better/on-par performance to the fact that continuously recording during task demonstration will produce irrelevant frames that can be pruned. We would expect an even higher success rate, when extending CoreDNA to also include different modalities like force as it would provide additional information for frames in which images are mostly uninformative which is during the most crucial part of manipulation: dexterous interaction with the environment. We are excited about this future research direction to extend coreset estimation methods to robot manipulation.

Unlike the autonomous driving scenario, where adding low-dimensional states (i.e., joint states) improved performance, we do not observe a similar increase here.

A.6 Detailed Performance Correlation Experiments

We first define the individual scores used to compare performance in terms of coverage and fidelity, and then discuss dataset-wise results. Coverage and fidelity scores are computed on the same D-dimensional point cloud used for coreset selection ($D = 2$ for the main two-axis sampler space, $D = 3$ for the (mean, std, third-axis) space): each axis $a = 1, \ldots, d$ is independently min–max normalized to $[0, 1]$ over the full dataset P, and $C \subset P$ denotes the selected coreset. All metrics are oriented so that *higher is better*. For each candidate space, we then compute the Pearson correlation across runs between each score and downstream performance, pooled within coreset ratio, and average it over the scores of each family.

Coverage family.

- **Mean nearest-selected distance:** $\sqrt{d} - \frac{1}{N}\sum_{p\in P}\min_{c\in C}\|p-c\|_2$ (the additive constant $\sqrt{d}$ is the diagonal length of the unit hypercube $[0,1]^d$, so the score stays positive), computed via a k-d tree nearest-neighbor query.
- **Coverage@k** ($k \in \{3, 10\}$): for every population point, precompute its k nearest neighbors in P (a k-d tree query, self included at distance 0). Coverage@k is the fraction of population points that have *at least one* of their k nearest neighbors in C.
- **Recall@k** ($k \in \{3, 10\}$): using the same precomputed k-NN table, recall@k is the fraction of the full population that appears as a k-nearest-neighbor of *at least one* selected point $c \in C$ — how much of P is "reachable" from C within k hops.
- **Tail coverage** (percentiles $\{5, 95\}$): per axis a, checks whether C contains at least one point at or below P's 5th percentile and at least one at or above its 95th percentile on that axis; averaged over both indicators and all D axes.
- **Quantile coverage** (5 quantile bins per axis, i.e. a 5^D-cell grid): the fraction of population-occupied grid cells that contain at least one selected point.
- **Extreme coverage**: after normalizing P and C to $[0,1]^D$, for each of the 2^d corners of the unit hypercube a distance threshold is set to the population's own 10th-percentile distance to that corner; extreme coverage is the fraction of corners for which C has at least one point within that threshold.

Fidelity family.

- **Wasserstein distance** (reported as a similarity): the 1D Wasserstein-1 (earth mover's) distance is computed independently on each axis a, averaged over all D axes, and converted to $1 - \overline{W_1}$.
- **Jensen–Shannon similarity**: Gaussian KDEs (SciPy's `gaussian_kde`, Scott's-rule bandwidth) are fit to P and to C (subsampled to at most 1500 points for tractability) and evaluated on a shared grid spanning the population's bounding box (100 points/axis for $D = 2$, 15 points/axis for $D = 3$); both are normalized to sum to 1 over the grid, and the Jensen–Shannon divergence $\mathrm{JS}(p\|q) = \frac{1}{2}D_{\mathrm{KL}}(p\|m) + \frac{1}{2}D_{\mathrm{KL}}(q\|m)$, $m = \frac{1}{2}(p+q)$, is rescaled to a similarity via $1 - \mathrm{JS}/\ln 2$ (natural-log JS divergence is bounded by $\ln 2$).
- **Density correlation**: the Pearson correlation between the two KDEs' density values on the same grid used for the JS similarity above.
- **Bhattacharyya coefficient**: the discretized Bhattacharyya coefficient $\sum_{\text{grid}}\sqrt{\hat{p}\cdot\hat{q}}h^d$ (h = grid spacing) between the same two KDEs, numerically integrated over the shared grid.

When an axis is degenerate (e.g. a purely 1D selection space) several metrics fall back to a 1D-only comparison of the primary axis, or default to 0 when no meaningful density estimate can be formed.

We show layerwise results in Tab. 9. Paired with $\delta_{\max}$, the mean of layer ℓ_{12} gives the highest fidelity correlation (0.340, against 0.310 for ℓ_1, 0.336 for ℓ_6, and 0.336 for the all-layer mean $\mu_{1:12}$), while ℓ_6 and $\mu_{1:12}$ are slightly ahead on coverage (0.297 and 0.293 against 0.276 for ℓ_{12} and 0.267 for ℓ_1). We further note that Cityscapes exhibits no correlation to any fidelity metric, highlighting that coverage might be a better metric for this dataset. We continue our analysis with ℓ_{12} as main layer since it outperforms on fidelity and because the difficulty signal in the scores grows with network depth.

Among the candidates for a second axis next to μ_{12}, $\delta_{\max}$ and σ_{12} give the highest fidelity correlation (0.340 and 0.341; Tab. 1, per-dataset results in Tab. 10), followed by m_{12} (0.322) and δ_{avg} (0.318). All five are within 0.007 on coverage. δ_{avg} is ahead only on the two depth datasets, so we default to $\delta_{\max}$.

Inside the sampler space, each of the two axes adds to the other (Tab. 11): adding $\delta_{\max}$ to (μ_{12}, σ_{12}) raises the correlations by 0.028 (coverage) and 0.016 (fidelity), and adding σ_{12} to $(\mu_{12}, \delta_{\max})$ raises them by 0.032 and 0.017, on six of seven datasets in both cases. The three-axis space therefore correlates best with performance on average (0.308 and 0.357, against 0.276–0.280 and 0.340–0.341 for the two-axis subspaces). The differences between options are small, so we treat these axes as a sensible default, not an optimum.

A.7 VDNA BINS AND BACKBONE

To evaluate how many histogram bins are needed for the VDNA density score to reliably order samples, we compute the score at bin counts of 2, 16, 128, and 1000 for the Mug-SSL backbone on CIFAR10, Cityscapes, and PascalVOC, and measure the Spearman rank correlation of the resulting per-sample ordering against the 1000-bin reference at each smaller bin count (Fig. 5a). The ordering of samples is confirmed to be identical (Spearman $\rho = 1.000$) from 16 bins up to the 1000-bin reference, across all three datasets. Even 2 bins already

Table 9: Layer ℓ of the mean axis (second axis fixed to $\delta_{\max}$); $\mu_{1:12}$ is the mean over all layers. Pearson correlation between run performance and the metrics of its selected set averaged over the metrics of each family.

	Coverage ↑				Fidelity ↑			
Dataset	μ_1	μ_6	μ_{12}	$\mu_{1:12}$	μ_1	μ_6	μ_{12}	$\mu_{1:12}$
CIFAR10	0.17	0.19	0.18	**0.20**	0.26	0.27	**0.29**	0.28
PascalVOC	0.28	0.33	0.32	**0.34**	0.32	0.38	**0.40**	**0.40**
Cityscapes	**0.26**	**0.26**	**0.26**	0.23	0.01	**0.05**	0.03	0.04
Cityscapes-DVPS (Depth)	0.27	0.27	**0.29**	**0.29**	0.48	**0.53**	0.52	0.52
Cityscapes-DVPS (Segm.)	**0.39**	0.37	0.36	0.36	0.53	**0.54**	**0.54**	0.51
SemanticKITTI-DVPS (Depth)	0.30	**0.45**	0.33	0.43	0.34	**0.36**	0.35	**0.36**
SemanticKITTI-DVPS (Segm.)	**0.21**	**0.21**	0.20	0.20	0.23	0.23	**0.26**	0.24
Mean	0.267	**0.297**	0.276	0.293	0.310	0.336	**0.340**	0.336

Table 10: Second axis (mean axis fixed to layer ℓ_{12}; *none* = the mean axis alone). Pearson correlation between run performance and the metrics of its selected set averaged over the metrics of each family.

	Coverage ↑					Fidelity ↑				
Dataset	none	m_{12}	σ_{12}	δ_{avg}	$\delta_{\max}$	none	m_{12}	σ_{12}	δ_{avg}	$\delta_{\max}$
CIFAR10	0.18	**0.19**	**0.19**	0.16	0.18	0.27	0.28	**0.29**	0.24	**0.29**
PascalVOC	**0.32**	**0.32**	**0.32**	0.28	**0.32**	0.36	0.39	**0.41**	0.33	0.40
Cityscapes	**0.38**	0.29	0.27	0.19	0.26	0.02	**0.04**	**0.04**	0.01	0.03
Cityscapes-DVPS (Depth)	0.19	0.24	0.23	**0.30**	0.29	0.47	0.47	0.45	**0.53**	0.52
Cityscapes-DVPS (Segm.)	**0.40**	0.36	0.36	0.37	0.36	0.47	0.51	**0.54**	0.52	**0.54**
SemanticKITTI-DVPS (Depth)	0.28	0.33	0.31	**0.39**	0.33	0.30	0.31	0.34	**0.35**	**0.35**
SemanticKITTI-DVPS (Segm.)	0.16	0.23	**0.27**	0.22	0.20	0.23	0.25	**0.31**	0.24	0.26
Mean	0.273	0.278	**0.280**	0.275	0.276	0.300	0.322	**0.341**	0.318	0.340

preserves the ordering closely ($\rho = 0.976$ to 0.990). We argue that with the original 1000 bins, a large fraction are empty and do not affect the result, so a substantially smaller bin count can be used with no measurable loss in ordering fidelity. Concretely, on CIFAR10's 50,000 training images the materialized representation occupies 1.4GB at 2 bins, 3.1GB at 16 bins, and 11GB at 128 bins. Extrapolating this trend to 1000 bins gives an estimated 28GB for no additional gain in ordering agreement over the 16-bin representation.

We further compare three backbones, Mug-SSL (our default), Dinov2, and Dinov3, normalized in the same way at 16 bins. We show the Spearman correlation of their full orderings (Fig. 5b) and the Jaccard overlap of the samples each backbone places in the top and bottom 10% of the distribution, i.e. the dataset's outliers and inliers (Fig. 5c). The backbones' orderings only moderately agree, with Spearman ρ ranging from 0.814 to 0.877 across datasets. However, the tails of the distribution overlap much more strongly between backbones: a Jaccard overlap of 0.410 to 0.511 in the tails, versus only 0.087 to 0.163 for a same-sized band around the median, against a chance level of 0.053 for two independent random subsets of that size.

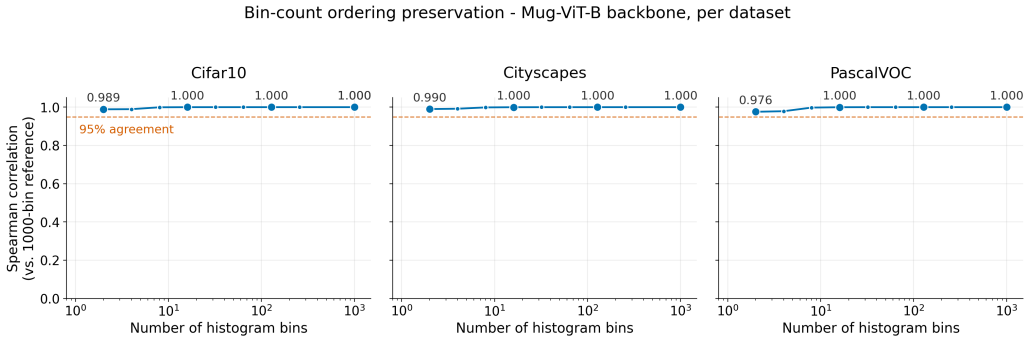
A.8 Ablation over axis choices and Selection Mechanism

We analyze the effect of our axis space in Tab. 12. We note that for CIFAR10, choosing δ_{avg} instead of $\delta_{\max}$ yields the best performance at $\alpha = 0.1$. However, since we seek a universal solution, we stick with $\delta_{\max}$, which we found empirically to correlate best across datasets. For this dataset, using only μ_{12} already achieves a comparable score, showing that even a single axis captures image-dataset similarity well, whereas using only σ_{12} or $\delta_{\max}$ decreases performance by at least 1.23 pp. We also test whether using all per-layer means and standard deviations, $\mu_1, \ldots, \mu_{12}$ and $\sigma_1, \ldots, \sigma_{12}$, provides a good solution. We observe that this does not achieve the best performance at $\alpha = 0.1$. We reason that with small retention rates, the influence of important axes is diluted by redundant axes that do not help. For larger coresets, however, additional axes may be worth exploring, as they enable finer-grained differentiation between near-duplicate samples, consistent with this configuration being only on par with our chosen space at $\alpha = 0.5$.

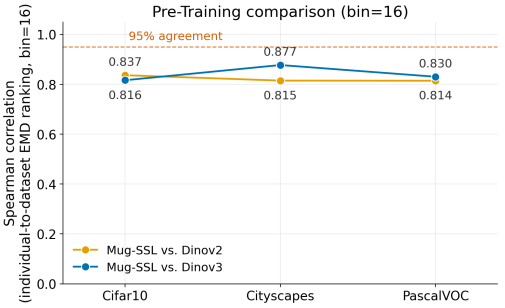
Next, we ablate the sensitivity to the sliced Wasserstein hyperparameters in Tab. 13. We show that a smaller g only drops performance by 0.34 pp at $\alpha = 0.1$, whereas a larger g hurts performance more substantially, as it disrupts quantile sampling. Further, $E = 10$ is optimal in both settings, with smaller or larger E resulting in drops of at least 0.43 pp.

Table 11: Correlation between run performance and the metric of its selected set for the three-axis space and for each two-axis subspace. Coverage and fidelity are the same metric families as in the other tables.

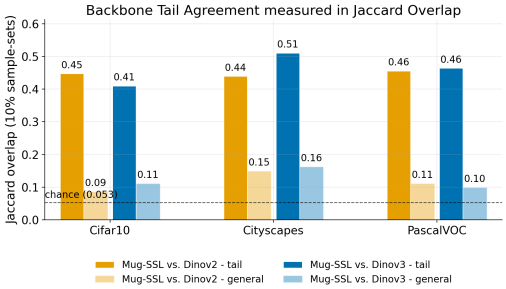
	Coverage ↑			Fidelity ↑		
Dataset	$\mu_{12}+\sigma_{12}$	$\mu_{12}+\delta_{\max}$	$\mu_{12}+\sigma_{12}$ + $\delta_{\max}$	$\mu_{12}+\sigma_{12}$	$\mu_{12}+\delta_{\max}$	$\mu_{12}+\sigma_{12}$ + $\delta_{\max}$
CIFAR10	0.19	0.18	**0.20**	0.29	0.29	**0.30**
PascalVOC	0.32	0.32	**0.35**	0.41	0.40	**0.42**
Cityscapes	0.27	0.26	**0.28**	**0.04**	0.03	**0.04**
Cityscapes-DVPS (Depth)	0.23	**0.29**	**0.29**	0.45	**0.52**	0.50
Cityscapes-DVPS (Segm.)	0.36	0.36	**0.42**	0.54	0.54	**0.57**
SemanticKITTI-DVPS (Depth)	0.31	0.33	**0.37**	0.34	0.35	**0.37**
SemanticKITTI-DVPS (Segm.)	**0.27**	0.20	0.26	**0.31**	0.26	0.29
Mean	0.280	0.276	**0.308**	0.341	0.340	**0.357**



(a) Bin-count ordering preservation, per dataset



(b) Pre-training comparison (Spearman)



(c) Backbone tail agreement (Jaccard overlap)

Figure 5: We compare different histogram bin sizes B in (a) and different backbones for neuron-pooling feature extraction in (b, c).

A.9 CIFAR10 Duplication Experiment

To test robustness to redundant data, we construct CIFAR10-Dupl by duplicating all training images ten times and re-evaluate all methods on this dataset under a constant selection amount ($C = 5000$ and $C = 25000$) in Tab. 14. Training-dynamics-based methods collapse dramatically at $C = 25000$: AUM, Forgetting, and EL2N drop by 63.36, 32.50, and 63.65pp, respectively, since duplicated samples distort their difficulty estimates and cause them to concentrate selection on a small, redundant subset of hard examples. In contrast, CCS and BOSS, which augment a training-based difficulty signal with a diversity or stratification component, remain stable, as do the fully unsupervised methods (CoreDNA, ZCore, FreeSel, ELFS), all staying within a few points of their performance on the original dataset. This demonstrates that pure training-dynamics-based selection is far more fragile to duplicated data than approaches that account for diversity, whether unsupervised or training-based. This supports our claim that diversity is a more robust criterion for coreset selection than hardness alone. This robustness is particularly relevant for real-world data collection. Unlike curated benchmarks, video recordings

Table 12: Ablation over axis choices for coreset selection (CIFAR10, sliced Wasserstein selection). Performance is downstream accuracy (%).

			Accuracy ↑	
Mean	Std	Third axis	0.1	0.5
$\mu_{1:12}$	σ_{12}	$\delta_{\max}$	82.06	92.88
μ_1	σ_{12}	$\delta_{\max}$	80.67	93.05
$\mu_{1:12}$	$\sigma_{1:12}$	$\delta_{\max}$	79.21	93.22
μ_1	σ_1	$\delta_{\max}$	79.39	93.05
μ_{12}	σ_{12}	δ_{avg}	**82.63**	93.14
μ_{12}	-	$\delta_{\max}$	80.14	93.21
μ_{12}	σ_{12}	-	80.95	93.30
-	σ_{12}	-	81.17	93.24
-	-	$\delta_{\max}$	80.39	93.52
μ_{12}	-	-	82.38	93.46
$\mu_1, \ldots, \mu_{12}$	$\sigma_1, \ldots, \sigma_{12}$	-	81.67	**93.58**
μ_{12}	σ_{12}	$\delta_{\max}$ (main)	82.40	93.49

Table 13: Sensitivity to the sliced Wasserstein hyperparameters E (number of projections) and g (near-duplicate gap) on CIFAR10. Performance is downstream accuracy (%).

		Accuracy ↑	
E	g	0.1	0.5
3	0.01	81.35	93.34
5	0.01	80.16	93.40
10	0.001	82.06	93.56
10	0.01	82.40	93.49
10	0.1	80.88	93.51
20	0.01	81.57	93.25

from robots or vehicles naturally contain many near-duplicate frames, making resilience to redundancy a practical necessity rather than a synthetic edge case.

Table 14: Downstream accuracy (%) on CIFAR10, comparing the normal dataset to a version with duplicated samples CIFAR10-Dupl. Values in parentheses show the difference (duplicated − normal).

	Normal ↑		With Duplicates ↑	
Method	50000 ($\alpha = 0.1$)	250000 ($\alpha = 0.5$)	50000 ($\alpha = 0.01$)	250000 ($\alpha = 0.05$)
Full	95.05	95.05	95.33 (+0.28)	95.33 (+0.28)
Random	80.90	93.11	81.46 (+0.56)	92.57 (−0.54)
AUM	31.93	95.24	18.87 (−13.06)	31.88 (−63.36)
Forgetting	39.50	95.08	36.42 (−3.08)	62.58 (−32.50)
EL2N	26.31	94.90	17.64 (−8.67)	31.25 (−63.65)
CCS	86.49	91.16	85.90 (−0.59)	91.28 (+0.12)
BOSS	84.28	94.32	84.54 (+0.26)	92.03 (−2.29)
ZCore	83.56	93.39	79.63 (−3.93)	93.27 (−0.12)
ELFS AUM	86.66	89.25	86.97 (+0.31)	90.98 (+1.73)
ELFS Forget	86.60	89.01	86.83 (+0.23)	91.25 (+2.24)
FreeSel	82.66	93.38	81.45 (−1.21)	92.58 (−0.80)
CoreDNA (WS)	82.40	93.36	81.73 (−0.67)	93.07 (−0.29)