Zero-Shot Coreset Selection via Iterative Subspace Sampling

Supplementary Material

Reproducibility Statement

We provide detailed method and experiment descriptions in Secs. 4 and 5 of the main paper. We generate all ZCore experimental results from a single attempt of five consecutive trials with the exception of ImageNet, which is from a single attempt of one trial. Our paper's code is publicly available at `https://github.com/voxel51/zcore`.

Experiment Datasets & External Baselines

To confirm that ZCore selects high-performing coresets, we perform experiments with ten state-of-the-art methods to select coresets, train downstream models, and evaluate performance on four datasets. Our experiments expand the recent work of Zhang et al. [54], which evaluated their coreset method with seven external baselines and three datasets.

Here, we provide detailed method descriptions for external baselines. **Entropy** selects examples with high entropy of predicted probabilities at the end of fully-supervised training on all candidate data [10]. **Forgetting** selects examples that change to being misclassified after correct classification the most times during training [46]. **EL2N** selects examples with high gradient magnitude using the L2 norm of error vectors [37]. **AUM** selects examples with high area under the margin, i.e., the probability gap between the target class and the next largest class across all epochs [38]. **Moderate** selects examples closest to the median class value in the full dataset trained model embedding space [50]. **Dyn-Unc** selects examples with high target class probability variance during training [18]. **TDDS** selects examples with high projected gradient variance across many epochs [54]. **Prototypes$_{\text{Sup.}}$** selects examples based on difficulty, which is determined using k-means embedding clustering and the distance of each example to cluster centroids or *prototypes* [44]. Notably, Prototypes$_{\text{Sup.}}$ uses a supervised model trained on candidate data to generate embeddings while **Prototypes$_{\text{SS}}$** uses a self-supervised model [44]. **ELFS** uses an extensive framework of deep clustering on pretrained model embeddings to generate psuedo labels, subsequent full model training on pseudo labels, and finally selecting examples based on pseudo training dynamic-based scores without ground truth labels [59].

ZCore is the only method to make selections without ground truth labels *or* dataset training (see Fig. 2 of the main paper) aside from **Random**, which selects examples with uniform random sampling. Furthermore we evaluate ZCore under constant algorithmic settings across a dataset and coreset selection scale greater than three orders of magnitude. Specifically, full dataset sizes span from 1.3 M to 2,700 examples and coreset sizes span from 896,817 to 270 examples (see Tab. 4.). To our knowledge, there is no precedent for this selection range of experiments and generalization in the coreset selection literature.

We also considered several other selection methods as baselines for our experiments. Unfortunately, as we will explain, these selection methods and many others are out of scope. Notably, our experiments use consistent prune rates for fair comparisons across coreset selection methods. On the other hand, SemDeDup [1], which selects coresets to remove semantic duplicates, uses configuration parameters that result in a prune rate that is unknown until *after* coreset selection experiments end. Perhaps due to this variable prune rate, the SemDeDup paper includes only a single data point as an external methodological comparison. GIGA constructs Bayesian coresets that scale log-likelihood optimality [6], but is originally applied to regression and would need to be modified for image classification. Other selection baselines originally applied to natural language processing would similarly require modification for our experiments [8, 21], which would materially change their functionality.

Expanded Coreset Selection Efficiency Details

We provide a detailed comparison of coreset selection efficiency in Tab. 5. In terms of runtime efficiency, ZCore selects data using embeddings generated from a single forward pass of off-the-shelf models, while TDDS and ELFS, which are representative of current SOTA methods, train on the entire dataset prior to coreset selection. However, training on data prior to selection takes substantially *more* time than any other process, whether using a laptop or GPU (see TDDS "Train Dynamics," 52.3 hours and 1,540s respectively). Furthermore, ELFS requires repeat full data deep clustering *and* model training on pseudo labels to calculate scores for each prune rate. On the other hand, generating embeddings prior to selection is a one-time and parallelizable operation that takes substantially less time than deep clustering, pretraining, or downstream model training, whether

Table 4. Comparison of **full training and coreset size** across all datasets. ZCore uses constant algorithmic settings across all experiments.

Dataset	Scale	Number of Classes	Full Dataset Training Size	Percent of Dataset Selected for Coreset				
				70%	50%	30%	20%	10%
ImageNet	Large	1,000	1,281,167	896,817	640,584	384,350	256,233	128,117
CIFAR100	Medium	100	50,000	35,000	25,000	15,000	10,000	5,000
CIFAR10	Medium	10	50,000	35,000	25,000	15,000	10,000	5,000
EuroSAT80	Medium	10	21,600	15,120	10,800	6,480	4,320	2,160
EuroSAT40	Small	10	10,800	7,560	5,400	3,240	2,160	1,080
EuroSAT20	Small	10	5,400	3,780	2,700	1,620	1,080	540
EuroSAT10	Small	10	2,700	1,890	1,350	810	540	270

Table 5. **Comparison of Coreset Selection Efficiency** on CIFAR100 for 70%, 30%, and 10% data rates. Selection "Total" is the runtime to generate all three coresets. Notably, ZCore uses the same score across all settings, while TDDS uses a unique validation-tuned setting for each prune rate (see Fig. 2). ELFS also uses a unique hyperparameter-tuned setting for each prune rate and additionally requires full-data deep clustering, pseudo training dynamics generation, and score calculation for each prune rate. Runtimes use a standard M3 Max-equipped "Laptop" or a single L40S "GPU" on a Lambda Scalar. ELFS hardware requirements prohibit laptop experiments. Following the protocol of TDDS [54], downstream model training at a 10% data rate on a GPU takes longer than 30% due to a reduced batch size.

Compute Hardware	Selection Runtime (s)					Model Train Time (s)			Selection + Train Time (s)			Total Labels		
	Components				Total	70%	30%	10%	70%	30%	10%	70%	30%	10%
ZCore	Embedding Generation ResNet18	CLIP	Parallel	ZCore Score										
Laptop	635	6,471	6,471	382	**6,853**	130,503	64,877	41,711	**137,356**	**71,730**	**48,564**	**35K**	**15K**	**5K**
GPU	145	97.0	145	54.7	**199.8**	990	589	643	**1,190**	**789**	**843**	**35K**	**15K**	**5K**
TDDS	Train Dynamics	TDDS Score 70%	30%	10%										
Laptop	188,295	27.0	25.5	0.9	188,296	130,503	64,877	41,711	318,825	253,198	230,008	50K	50K	50K
GPU	1,540	22.6	20.9	1.3	1,541	990	589	643	2,553	2,150	2,184	50K	50K	50K
ELFS	DINO Embed	Deep Cluster	Pseudo Train	ELFS Score										
GPU	111	1,093	1,476	30.3	7,908	990	589	643	3,701	3,300	3,353	**35K**	**15K**	**5K**
TDDS / ZCore Runtime & Label Requirement Ratio														
Laptop	—	—	—	—	**27.5**	—	—	—	2.32	3.53	4.74	**1.43**	**3.33**	**10.0**
GPU	—	—	—	—	**7.7**	—	—	—	2.14	2.72	2.59	**1.43**	**3.33**	**10.0**
ELFS / ZCore Runtime Requirement Ratio														
GPU	—	—	—	—	**39.6**	—	—	—	3.11	4.18	3.98	1.00	1.00	1.00

on the laptop or GPU (see ZCore "Embedding Generation," 1.8 hours and 145s respectively).

From the results in Tab. 5, ZCore selects the three coresets 27.5× faster than TDDS on a laptop and 39.6× faster than ELFS on a GPU. Importantly, as discussed in Sec. 5.3, the relative time to implement ZCore is even faster. Reason being, ZCore uses the same settings across all datasets and the same score across all prune rates. On the other hand, both comparison methods perform parameter searches on *each* dataset *and* prune rate to determine their final dataset- and prune-specific settings. It is cost-prohibitive to replicate the hyperparameter search of each comparison method for our experiments in Tab. 5, but ELFS's developers report that it takes approximately 17 hours using four A6000 GPUs to find a single prune rate setting on ImageNet [59].

In terms of label efficiency, TDDS, like many SOTA methods, uses ground truth labels on all candidate data prior to coreset selection. On the other hand, ZCore and ELFS select data without any ground truth labels. Accordingly, labels are only required during downstream model training for the selected coreset if the model is fully-superivsed. Thus, TDDS requires 1.43, 3.33, and 10× more labels than ZCore to train downstream models at the respective 70%, 30%, and 10% data selection rates in Tab. 5.

Class Recall after ZCore Selection

We provide the class recall of ZCore selections across all datasets and data selection rates of the main paper in Tab. 6. Notably, ZCore achieves 100% recall across all datasets and original data selection rates (10-70%). In addition to the main paper's quantitative and qualitative results (Secs. 5.2 and 5.4), ZCore's reliable class recall further establishes the viability of using iterative subspace sampling (Sec. 4) to achieve coreset data coverage without labels or training.

As an additional challenge, we also provide the class re-

Table 6. **ZCore Coreset Class Recall** after selection on all datasets. ZCore achieves 100% class recall in all the main paper's experiments (**10-70%**, middle), so we include a few minimal selection rates as an added challenge (**0.5-5%**, right). Recall % is mean across all trials.

			Class Recall (%) at Original Data Selection Rates (**10-70%**)					Class Recall (%) at Minimal Data Rates (**0.5-5%**)			
Dataset	Number of Classes	Full Dataset Training Size	**70%**	**50%**	**30%**	**20%**	**10%**	**5%**	**2%**	**1%**	**0.5%**
ImageNet	1,000	1,281,167	100%	100%	100%	100%	100%	100%	100%	100%	100%
CIFAR100	100	50,000	100%	100%	100%	100%	100%	100%	99%	92%	75%
CIFAR10	10	50,000	100%	100%	100%	100%	100%	100%	100%	100%	100%
EuroSAT80	10	21,600	100%	100%	100%	100%	100%	100%	100%	96%	88%
EuroSAT40	10	10,800	100%	100%	100%	100%	100%	100%	90%	86%	78%
EuroSAT20	10	5,400	100%	100%	100%	100%	100%	98%	90%	76%	66%
EuroSAT10	10	2,700	100%	100%	100%	100%	100%	92%	82%	64%	58%

Table 7. Comparison of coreset selection methods using downstream model validation on **ImageNet**. Full dataset training on the ResNet32 model training achieves 73.54% accuracy. A corresponding results plot is provided in Fig. 5 of the main paper.

% of Data Selected	30%	20%	10%	Mean / Rel. Rand.
Unlabeled Coreset Selection without Training				
ZCore$_{\text{CLIP}}$	**64.42**	**61.39**	**53.54**	**59.78** +0.59
Random	64.19	60.76	52.63	59.19 +0.00
Unlabeled Coreset Selection with Self-Supervised Training				
Prototypes$_{\text{SS}}$ [44]	60.42	53.73	38.06	50.74 -8.45
Labeled Coreset Selection with Training-based Pruning				
TDDS [54]	**64.69**	**62.56**	53.91	**60.39** +1.19
ZCore$_{\text{ResNet18, CLIP}}$	64.43	61.31	**53.99**	59.91 +0.72
Forgetting [46]	64.29	62.01	52.14	59.48 +0.29
Moderate [50]	64.04	61.35	52.45	59.28 +0.09
ZCore$_{\text{ResNet18}}$	63.37	60.57	52.97	58.97 -0.22
Entropy [10]	62.34	56.80	43.39	54.18 -5.02
Prototypes$_{\text{Sup.}}$ [44]	53.59	42.25	22.41	39.42 -19.77
EL2N [37]	46.92	32.68	15.90	31.83 -27.36
AUM [38]	39.34	23.64	11.70	24.89 -34.30

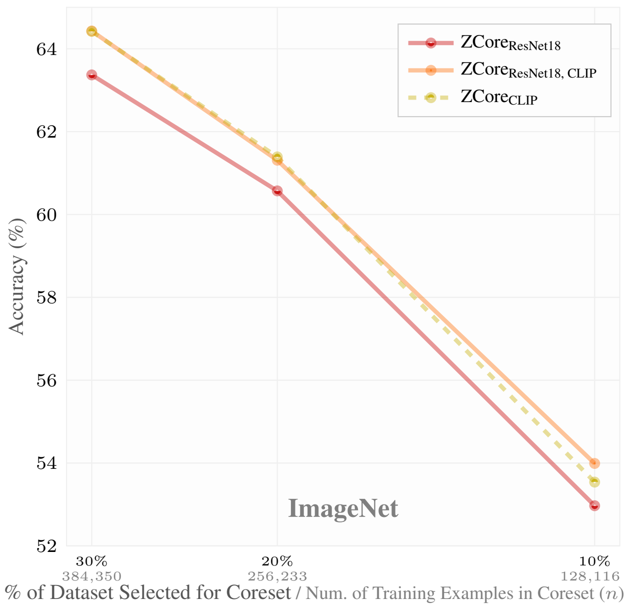

Figure 10. Comparison of coreset selection using downstream model validation on **ImageNet** for ZCore embedding ablations.

call of ZCore selections on a few minimal data selection rates beyond those of the main paper (0.5-5%). Notably, ZCore maintains 100% class recall all the way down to a 0.5% data selection rate on CIFAR10 and even ImageNet, which has 1,000 classes. On the other hand, class recall drops below 100% for datasets with fewer initial images per class (e.g., CIFAR100 recall is 99% at a 2% data rate) or small datasets with much smaller corresponding coresets (EuroSAT10, EuroSAT20 at 5% data rates). Predictably, the lowest recall of 58% occurs on EuroSAT10 at a 0.5% data rate, which is selecting for a coreset of only 14 images.

ImageNet Embedding Ablations

ZCore selects coreset data using embeddings from "off-the-shelf" foundation models that were trained before this work and remain static. Notably, a component of the default embeddings (ResNet18) is previously trained on ImageNet. On the other hand, we made a methodological commitment to use constant ZCore settings across all datasets and selection rates in our experiments, which includes consistent foundation models for embeddings.

To understand the relative contribution of ResNet18 embeddings for coreset selection on ImageNet, we perform an additional ablative experiment with ResNet18- and CLIP-only embeddings (see ZCore$_{\text{ResNet18}}$ and ZCore$_{\text{CLIP}}$ in Tab. 7 and Fig. 10). Notably, ZCore selections using CLIP-only embeddings are outperformed by *all* other embedding settings on CIFAR100 (see Tab. 3 in the main paper). On the other hand, ZCore$_{\text{CLIP}}$ selections on ImageNet outperform ZCore$_{\text{ResNet18}}$ across all data selection rates, despite ResNet18 pre-training on ImageNet. Furthermore, ZCore$_{\text{CLIP}}$ performs comparably to the concatenated ZCore$_{\text{ResNet18, CLIP}}$ setting on ImageNet, validating that ZCore is state-of-the-art for coreset selection without labels or training on candidate data across all experiment settings.

Table 8. Comparison of **ZCore ablations** on CIFAR 100 with ResNet18. Accuracy is the mean across 70%, 50%, 30%, 20%, and 10% data rates over five repeat trials. "ResNet18, CLIP" is the default concatenated embedding space using both models off-the-shelf.

	Zero-shot	Dimension	Sampling		Distance	Redundancy		Use Full	CIFAR100
Ablation	Embedding Model	Reduction	Distribution	Dimensions	Metric	Neighbors	Exponent	Score	Accuracy
Full Method (**ZCore**)	**ResNet18, CLIP**	$\boldsymbol{ZD}$	**Triangular**	**2**	$\boldsymbol{L_1}$	**1,000**	**4**	**Yes**	**65.77**
Embedding Space (Z)	**ResNet18 [17]**	ZD	Triangular	2	L_1	1,000	4	Yes	65.52
	CLIP [39]	ZD	Triangular	2	L_1	1,000	4	Yes	64.84
	DINOv2 [33]	ZD	Triangular	2	L_1	1,000	4	Yes	65.61
Embedding Dimension Reduction ($\hat{Z}$)	ResNet18, CLIP	**PCA**	Triangular	2	L_1	1,000	4	Yes	65.55
	ResNet18, CLIP	**t-SNE [27]**	Triangular	2	L_1	1,000	4	Yes	65.62
	ResNet18, CLIP	**UMAP [30]**	Triangular	2	L_1	1,000	4	Yes	62.43
Sampling Distribution ($\mathbf{z}$)	ResNet18, CLIP	ZD	**Gaussian**	2	L_1	1,000	4	Yes	65.75
	ResNet18, CLIP	ZD	**Uniform**	2	L_1	1,000	4	Yes	64.84
Subspace Sampling Dimensions (m)	ResNet18, CLIP	ZD	Triangular	**1**	L_1	1,000	4	Yes	64.59
	ResNet18, CLIP	ZD	Triangular	**3**	L_1	1,000	4	Yes	65.10
	ResNet18, CLIP	ZD	Triangular	**10**	L_1	1,000	4	Yes	63.27
	ResNet18, CLIP	ZD	Triangular	**100**	L_1	1,000	4	Yes	62.20
Distance Metric (Eq. (5), Eq. (7))	ResNet18, CLIP	ZD	Triangular	2	$\boldsymbol{L_2}$	1,000	4	Yes	65.60
	ResNet18, CLIP	ZD	Triangular	2	**cos**	1,000	4	Yes	64.48
Number of Nearest Neighbors (α)	ResNet18, CLIP	ZD	Triangular	2	L_1	**100**	4	Yes	65.68
	ResNet18, CLIP	ZD	Triangular	2	L_1	**10,000**	4	Yes	65.72
Distance Penalty Exponent (β)	ResNet18, CLIP	ZD	Triangular	2	L_1	1,000	**3**	Yes	65.64
	ResNet18, CLIP	ZD	Triangular	2	L_1	1,000	**5**	Yes	65.74
Random Coverage Sample (k)	**NA**	**NA**	**NA**	**NA**	L_1	1,000	4	**No**	64.74
No Redundancy Score (s^{R})	ResNet18, CLIP	ZD	Triangular	2	L_1	**NA**	**NA**	**No**	61.71
No Random Initialization ($\mathbf{s}$)	ResNet18, CLIP	ZD	Triangular	2	L_1	1,000	4	**No**	65.66
No Score Loss Weight (w)	ResNet18, CLIP	ZD	Triangular	2	L_1	1,000	4	**No**	63.53

Embedding Dimension Reduction Ablations

For dimension reduction, ZCore iteratively slices random subsets of the full embedding space to generate numerous subspace distributions, which are then sampled to find valuable examples based on coverage and redundancy (see Fig. 1 and Sec. 4 of the main paper for details). However, there are several alternative dimension reduction methods that are applicable to this work, including Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE) [27], and Uniform Manifold Approximation and Projection (UMAP) [30]. Notably, PCA maximizes variance to preserve pairwise global embedding distances while t-SNE and UMAP better preserve local embedding distances (see McInnes *et al.* for additional details and experiment comparisons [30]).

To validate the design decision to use ZCore's iterative embedding dimension reduction technique, we provide experiment ablations using PCA, t-SNE, and UMAP as alternative embedding reduction processes for subsequent sampling and ZCore score generation in Tab. 8. Notably, the proposed ZCore dimension reduction technique ($\hat{\boldsymbol{Z}} = \boldsymbol{ZD}$ in Eq. (4)) outperforms all three dimension reduction alternatives. Nonetheless, the t-SNE and PCA ablations exhibit good performance, indicating that the subsequent ZCore processes for coverage and redundancy scoring are relatively robust to changes in the underlying dimension reduction technique used. From the results, we do not recommend UMAP as an alternative dimension reduction technique for ZCore, but we still find UMAP to be an excellent tool for embedding space visualization (see Fig. 11).

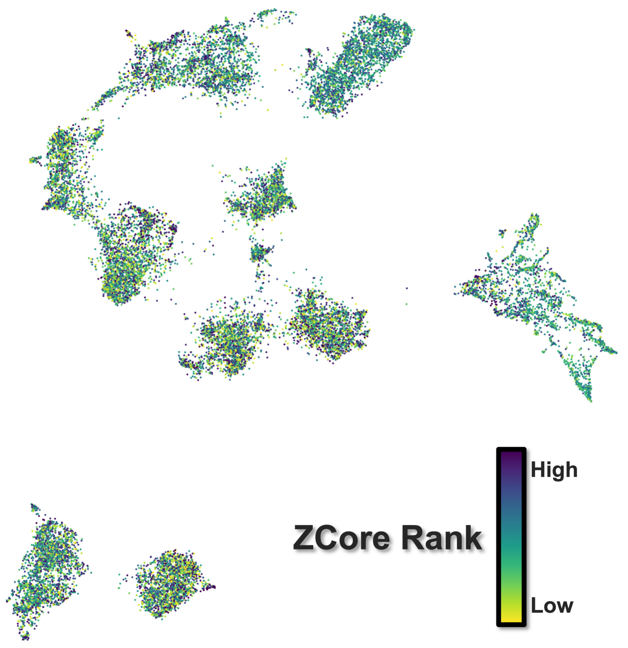

Figure 11. **ZCore Coreset Rank Visualization** for EuroSAT80. Model embeddings and 2D approximation of the full embedding space generated using the FiftyOne Library and UMAP [30, 32].

Supplementary Figures & Tables

To supplement the evaluation in Sec. 5 of the main paper, we provide several additional figures and tables.

First, we show the entire ZCore rankings for EuroSAT80 within a 2D manifold approximation of the full embedding space in Fig. 11. Second, we plot self-supervised DINOv2 model embeddings and corresponding sample distributions in Fig. 12. Third, we plot the runtime and accuracy performance of ZCore over a range of subspace sampling dimensions in Fig. 13. Fourth, we show the highest 30 ranked ZCore images for EuroSAT80 in Fig. 14, CIFAR100 in Fig. 15, and ImageNet in Fig. 16. We also show the lowest 30 ranked ZCore images for ImageNet in Fig. 17. Finally, we tabulate coreset selection results for CIFAR10 and CIFAR100 in Tab. 9 (evaluating selection for two medium-sized datasets), ImageNet in Tab. 7 (evaluating selection at a large scale), and all EuroSAT splits in Tab. 10 (evaluating selection of satellite images at a decreasing scale).

Future Work

For downstream model training experiments in Sec. 5.2, ZCore outperforms all prior coreset selection methods except for the state-of-the-art TDDS method. However, TDDS performance relies on several requirements that, although useful in experiment settings, limit applicability and efficiency at scale. Specifically, TDDS assumes all candidate data are labeled prior to selection (which is cost prohibitive), performs fully-supervised model training on all candidate data prior to selection (which is runtime prohibitive), and uses a hyperparameter search on the validation set to determine the final configuration used on a per-dataset and per-prune rate basis (which limits generalization). On the other hand, ZCore uses no labels or training on candidate data prior to selection (which is label *and* runtime efficient) and uses constant algorithmic settings across all experiments (which is generalizable). Despite these procedural differences, ZCore maintains competitive performance and even outperforms TDDS at a 10% selection rate on EuroSAT40 and ImageNet. Nonetheless, there remain several promising methodological avenues to improve ZCore and further advance the viability of coreset selection without labels, training, or parameter tuning.

First, although ZCore performance degrades relative to TDDS on the small EuroSAT20 and EuroSAT10 datasets at low selection rates, we find that ZCore performance improves with alternative settings (e.g., reducing the number of nearest neighbors for redundancy in Eq. (7)). Thus, one area of future work is developing adaptive iterative subspace sampling so that ZCore automatically adjusts with dataset size and data selection rate to improve performance in challenge settings.

Second, in addition to the coverage and redundancy scores developed in this paper, we postulate that there are many more label- and training-free features and metrics that can be developed to quantify coreset value for individual candidate examples. Expanding the set of metrics ZCore uses to consider each data example will likely make ZCore coreset selection more robust and performant across a wide range of datasets and data selection rates.

Finally, there is no domain-specific limitation to our method, so we can apply ZCore in other domains like point cloud and natural language and other problems like object detection and segmentation. However, each new domain and problem space will present new challenges *and* opportunities to improve ZCore. Taking object detection as one example, ZCore currently selects data using image-level embeddings but can also operate on object-level embeddings that are more representative of what downstream models will use for object detection. Furthermore, guiding data selection and annotation at an object level can lead to massive improvements in terms of overall human annotation time and downstream model performance [26]. Thus, we find that there are many exciting opportunities to expand and improve upon ZCore coreset selection in future work.

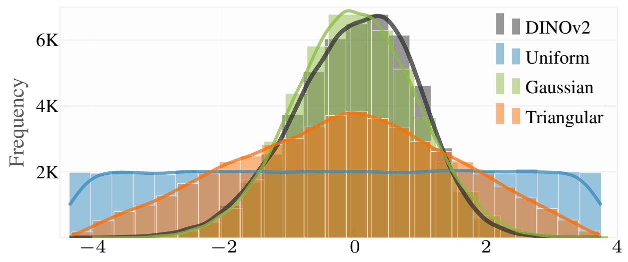

Figure 12. **Comparison of embeddings and sampling techniques**. DINOv2 is the first dimension embeddings for 50,000 CIFAR100 train set examples, while each corresponding distribution type is sampled 50,000 times. Corresponding plots for ResNet18 and CLIP are provided in Fig. 3 of the main paper.

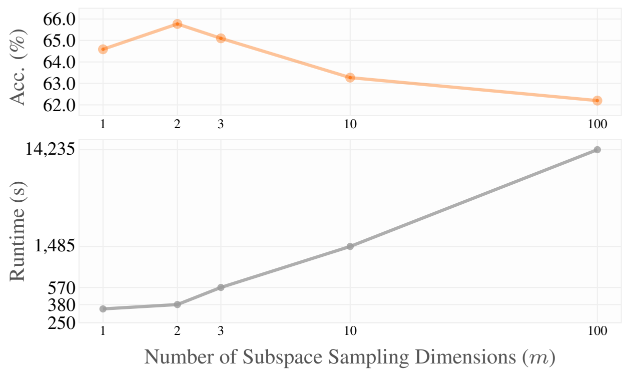

Figure 13. Comparison of number of **embedding subspace sample dimensions** (m) vs. accuracy (top) and **runtime** (bottom) on CIFAR100 with ResNet18. Accuracy is the mean across 70%, 50%, 30%, 20%, and 10% data selection rates over five repeat trials. Runtime is selection time using a M3 Max-equipped laptop.

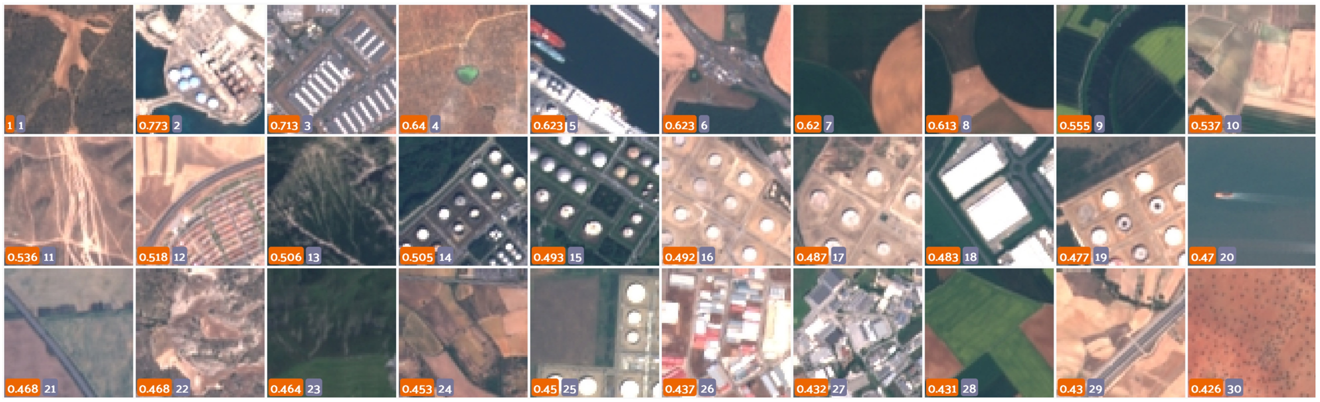

Figure 14. **Top-30 ZCore Selection Scores** (orange) for EuroSAT80.

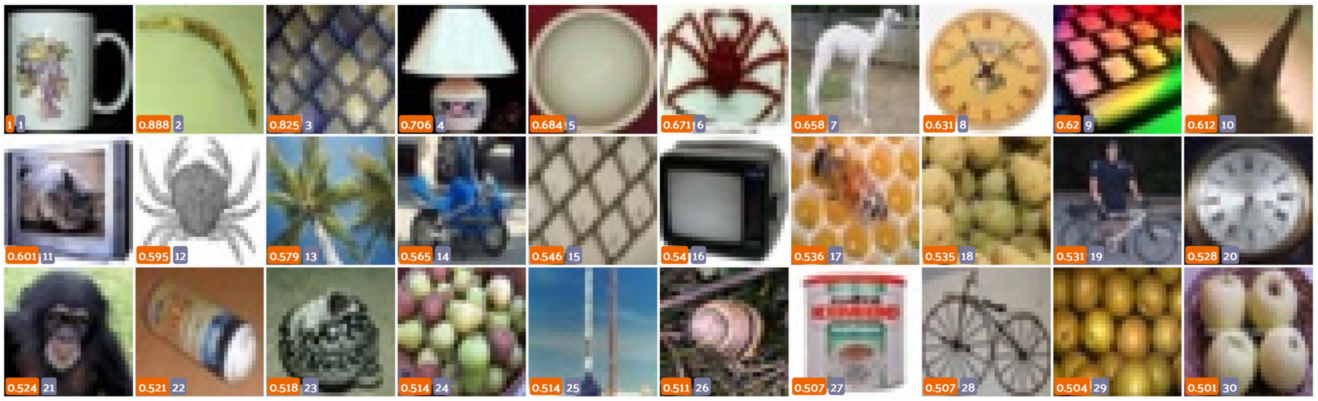

Figure 15. **Top-30 ZCore Selection Scores** (orange) for CIFAR100.

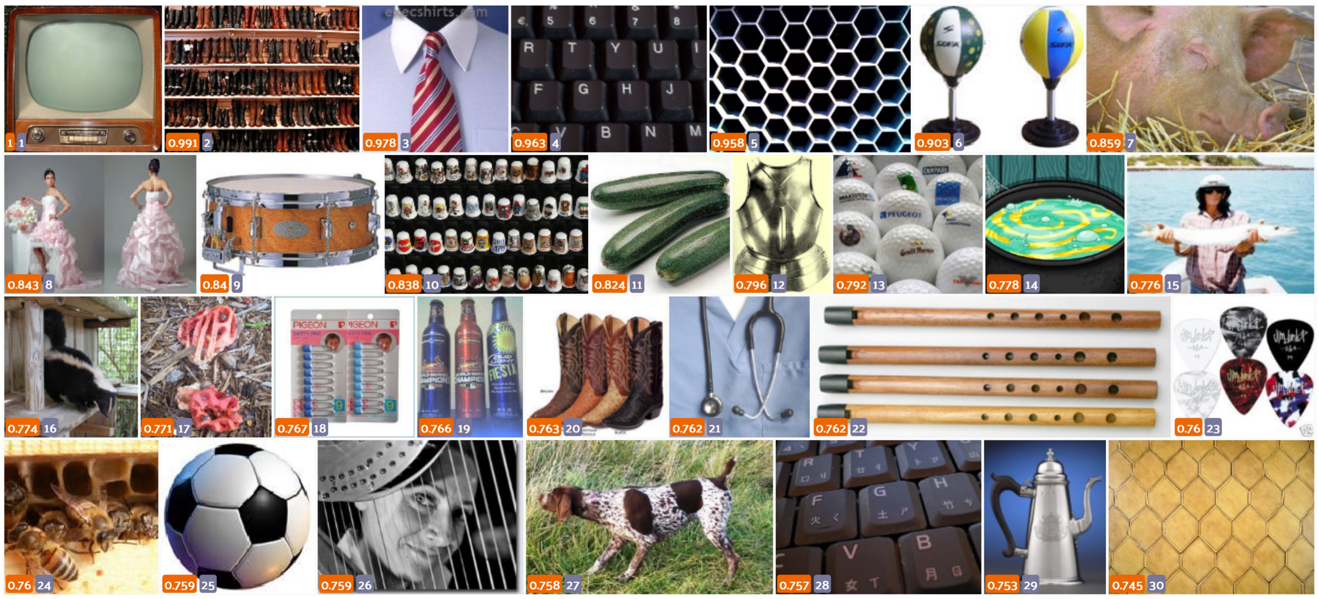

Figure 16. **Top-30 ZCore Selection Scores** (orange) for ImageNet.

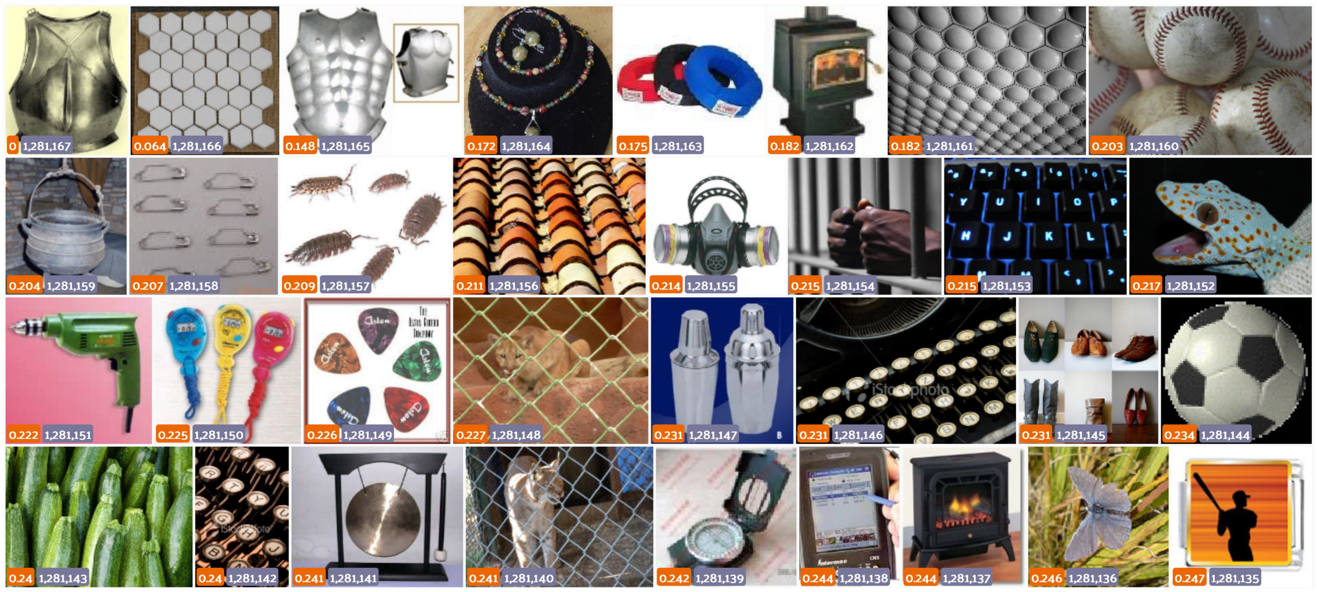

Figure 17. **Lowest-33 ZCore Selection Scores** (orange) for ImageNet.

Table 9. Comparison of coreset selection methods using downstream model validation on **CIFAR10** and **CIFAR100**. Full dataset training on the ResNet18 model achieves 95.23% (CIFAR10) and 78.21% (CIFAR100) accuracy. "Rel. Rand." is Mean accuracy across all data selection rates on both datasets relative to Random. A corresponding results plot is provided in Fig. 5 of the main paper.

	CIFAR10					**CIFAR100**					
% Data Selected	70%	50%	30%	20%	10%	70%	50%	30%	20%	10%	Mean Rel. Rand.
Unlabeled Coreset Selection without Training											
ZCore	**94.58** ±0.09	**93.46** ±0.16	**90.97** ±0.17	**89.06** ±0.33	**84.18** ±0.21	**76.04** ±0.15	**72.87** ±0.18	**65.92** ±0.15	**61.92** ±0.39	**52.11** ±0.66	**78.11** **+1.34**
Random	**94.58** ±0.04	93.38 ±0.17	90.61 ±0.44	88.87 ±0.47	83.77 ±0.26	75.53 ±0.04	71.95 ±0.16	64.59 ±0.32	57.79 ±0.24	46.68 ±1.07	76.78 +0.00
Unlabeled Coreset Selection with Self-Supervised Training											
ELFS [59] ICLR 2025	94.04 ±0.14	92.61 ±0.26	90.37 ±0.07	88.04 ±0.38	82.39 ±0.35	74.10 ±0.15	70.36 ±0.18	63.83 ±0.56	58.51 ±0.20	47.42 ±1.32	76.17 -0.61
Labeled Coreset Selection with Training-based Pruning											
TDDS [54] CVPR 2024	**95.47** ±0.06	**95.21** ±0.04	**93.03** ±0.25	**91.30** ±0.25	**85.46** ±0.21	**77.56** ±0.06	**74.04** ±0.34	**67.78** ±0.44	**63.01** ±0.12	**54.51** ±0.22	**79.74** **+2.96**
Moderate [50] ICLR 2023	93.96 ±0.06	92.34 ±0.09	89.71 ±0.14	87.75 ±0.27	83.61 ±0.24	74.60 ±0.10	70.29 ±0.31	62.81 ±0.08	56.52 ±0.37	41.82 ±1.12	75.34 -1.43
Entropy [10] ICLR 2020	94.45 ±0.07	91.90 ±0.16	86.24 ±0.26	83.49 ±0.21	72.06 ±0.81	72.39 ±0.20	64.44 ±0.36	50.73 ±0.86	42.86 ±0.25	29.56 ±0.54	68.81 -7.96
Forgetting [46] ICLR 2019	95.45 ±0.24	95.05 ±0.05	89.14 ±2.04	76.18 ±3.18	45.87 ±1.87	77.38 ±0.09	70.76 ±0.40	49.92 ±0.28	38.42 ±1.13	25.82 ±0.52	66.40 -10.38
Dyn-Unc [18] CVPR WS '24	95.08 ±0.02	94.03 ±0.14	89.40 ±0.13	79.76 ±1.09	37.12 ±1.12	73.36 ±0.10	65.90 ±0.25	50.16 ±0.47	39.19 ±0.27	15.20 ±0.41	63.92 -12.86
AUM [38] NeurIPS 2020	95.44 ±0.09	95.19 ±0.09	91.19 ±0.63	69.60 ±3.11	34.74 ±0.11	77.35 ±0.18	68.17 ±0.52	31.69 ±0.34	18.43 ±0.47	9.29 ±0.27	59.11 -17.67
EL2N [37] NeurIPS 2021	95.43 ±0.10	95.06 ±0.04	86.69 ±1.71	68.64 ±3.70	31.89 ±1.51	76.89 ±0.31	67.57 ±0.15	36.45 ±1.36	17.31 ±0.33	9.10 ±0.69	58.50 -18.27

Table 10. Comparison of coreset selection methods using downstream model validation on decreasing sized splits of **EuroSAT**. Full dataset training on the ResNet18 model achieves 98.59% (EuroSAT80), 98.20% (EuroSAT40), 97.36% (EuroSAT20), and 93.64% (EuroSAT10) accuracy. "Rel. Rand." is Mean accuracy across all data selection rates relative to Random. "EuroSAT All" is Mean accuracy for all EuroSAT splits. EuroSAT10 10% selection has only 270 examples. A corresponding results plot is provided in Fig. 6 of the main paper.

Selection		Percent of Dataset Selected / Number of Examples					
Method	Coreset Selection Requirements	70%	50%	30%	20%	10%	Mean Rel. Rand.
EuroSAT All							
ZCore	Unlabeled Data	96.53	95.74	93.21	91.74	83.27	**92.10** **+1.14**
Random	Unlabeled Data	94.56	92.91	89.80	87.88	83.61	90.96 +0.00
TDDS CVPR 2024	Full Training on Labeled Data	96.93	96.35	94.55	93.56	87.80	**93.96** **+ 3.01**
EuroSAT80 (21,600)		15,120	10,800	6,480	4,320	2,160	
ZCore	Unlabeled Data	98.32 ±0.08	98.15 ±0.13	97.72 ±0.13	97.31 ±0.17	95.80 ±0.18	97.46 **+0.56**
Random	Unlabeled Data	98.20 ±0.11	97.94 ±0.10	96.98 ±0.17	96.65 ±0.29	94.72 ±0.49	96.90 +0.00
TDDS CVPR 2024	Full Training on Labeled Data	98.62 ±0.05	98.58 ±0.11	98.43 ±0.03	98.09 ±0.10	96.28 ±0.11	98.00 **+1.10**
EuroSAT40 (10,800)		7,560	5,400	3,240	2,160	1,080	
ZCore	Unlabeled Data	97.59 ±0.05	97.53 ±0.16	96.45 ±0.12	96.06 ±0.19	92.94 ±0.55	96.11 **+1.54**
Random	Unlabeled Data	97.04 ±0.07	96.43 ±0.37	95.00 ±0.67	93.73 ±0.58	90.69 ±0.53	94.58 +0.00
TDDS CVPR 2024	Full Training on Labeled Data	97.97 ±0.09	98.06 ±0.06	97.55 ±0.08	96.79 ±0.16	92.78 ±0.27	96.63 **+2.05**
EuroSAT20 (5,400)		3,780	2,700	1,620	1,080	540	
ZCore	Unlabeled Data	96.49 ±0.16	95.45 ±0.22	92.60 ±0.29	91.80 ±0.70	80.39 ±3.91	91.35 **+1.53**
Random	Unlabeled Data	95.14 ±0.32	93.25 ±0.59	89.36 ±0.54	88.01 ±0.22	83.30 ±0.73	89.81 +0.00
TDDS CVPR 2024	Full Training on Labeled Data	96.94 ±0.10	96.45 ±0.07	93.86 ±0.56	94.70 ±0.35	86.94 ±0.55	93.78 **+3.97**
EuroSAT10 (2,700)		1,890	1,350	810	540	270	
ZCore	Unlabeled Data	93.71 ±0.23	91.82 ±0.23	86.08 ±1.16	81.77 ±2.68	63.96 ±2.76	83.47 **+0.92**
Random	Unlabeled Data	90.35 ±0.64	87.55 ±0.67	83.06 ±1.61	78.97 ±1.88	72.81 ±2.25	82.55 +0.00
TDDS CVPR 2024	Full Training on Labeled Data	94.62 ±0.09	92.92 ±0.33	89.41 ±0.52	85.56 ±0.67	74.74 ±2.02	87.45 **+4.90**