Rethinking Coreset Selection: The Surprising Effectiveness of Soft Labels

Anonymous authors
Paper under double-blind review

Abstract

Data-efficient deep learning is an emerging and powerful branch of deep learning that focuses on minimizing the amount of labeled data required for training. Coreset selection is one such method, where the goal is to select a representative subset from the original dataset, which can achieve comparable generalization performance at a much lower computation and disk space overhead. Dataset Distillation (DD), another branch of data-efficient deep learning, achieves this goal through distilling a small synthetic dataset from the original dataset. While DD works exploit soft labels (probabilistic target labels instead of traditional one-hot labels), which have yielded significant improvement over hard labels, to the best of our knowledge, no such study exists for coreset selection. In this work, for the first time, we study the impact of soft labels on generalization accuracy for the image classification task for various coreset selection algorithms. While soft labels improve the performance of all the methods, surprisingly, random selection with soft labels performs on par or better than existing coreset selection approaches. Our findings suggest that future coreset algorithms should benchmark against random selection with soft labels as an important baseline.

1 Introduction

The escalating scale and complexity of contemporary datasets and deep learning models have necessitated increasingly intensive computational resources. While large-scale datasets have driven the stunning success of deep learning (Alzubaidi et al., 2023; Aach et al., 2023; Zhou et al., 2025), increasing computing power has become central to performance improvements, which in turn has led to an alarming rise in the environmental cost of deep learning (Thompson et al., 2022; Patterson et al., 2021). Moreover, obtaining State-Of-The-Art (SOTA) results on complex architectures requires extensive hyperparameter tuning, which in turn results in large computational resource consumption (Zoph & Le, 2017; Li et al., 2018).

Coreset selection is one of the data-efficient deep learning mechanisms, which aims to select a smaller but representative subset of the original large-scale dataset (Mirzasoleiman et al., 2020) without sacrificing generalization performance much. Approximating the learning characteristics of the full dataset, such as gradient information, data redundancy, etc., enables learning a representative subset, significantly reducing training time and computational requirements (Feldman, 2020). Various coreset selection methods have been developed based on these principles. These include geometry-based (Chen et al., 2010; Sener & Savarese, 2018; Zheng et al., 2023), loss-based (Toneva et al., 2019; Paul et al., 2021), decision boundary based (Ducoffe & Precioso, 2018; Margatina et al., 2021), gradient based (Killamsetty et al., 2021; Mirzasoleiman et al., 2020; Mohanty et al., 2025).

Most image classification tasks utilize hard labels, where each sample is assigned a particular class. Szegedy et al. (2016) demonstrated that label smoothing using soft target distributions instead of one-hot labels improves model generalization and calibration. In their work on knowledge distillation, Hinton et al. (2015) used soft labels generated through a teacher model (a sophisticated neural network) to transfer its knowledge to a student model (a relatively simpler neural network), improving the performance of the student model substantially.

In the case of a hard label, the target class has a probability of 1, and the rest of the classes have a probability of 0. While with soft labels, all the classes assume non-zero probabilities, with the highest probability assigned to the target class. In knowledge distillation, soft labels encode the teacher model's uncertainty arising from class similarities, guiding the student network to learn nuanced inter-class relationships.

Recent SOTA works on Dataset Distillation (DD) methods (Yin et al., 2023; Du et al., 2024; Sun et al., 2024) have utilized soft labels to achieve significant improvements over methods using hard labels. Qin et al. (2024) explored the contribution of soft labels in improving generalization performance by these DD works compared to the contribution of the methods. They have highlighted soft labels' crucial role in distillation, proving that baseline random selection methods with soft labels perform on par or higher than existing distillation methods. Their study raises a natural question, ***whether a similar behavior is expected in the case of coreset selection?*** Crucially, dataset distillation and coreset selection reduce training cost by very different means: distillation synthesizes new (often optimized) images and labels, while coreset selection chooses a subset of existing images. Unlike dataset distillation, where sophisticated optimization may encode teacher priors on the synthetic samples, coreset selection operates under the stricter constraint of choosing real examples, making it a fundamentally different and underexplored setting for evaluating the role of soft labels.

In this paper, we explore soft labels' impact on the performance of coreset selection methods and ask *when training on real image subsets, do improvements come mainly from enriched label information or from smarter subset selection?*

We have conducted extensive ablation studies on benchmark datasets for image classification with various CNN and transformer-based architectures. Our major contributions in this paper are:

- Comprehensive study on the impact of utilizing soft labels with coreset selection methods in the context of image classification.
- Comparative analysis of the performance gains with soft labels on random selection and coreset selection methods.
- Empirical demonstration that soft labels with random selection perform on par or better than coreset selection methods.

2 Related Works

2.1 Coreset selection

Geometry-based methods (Agarwal et al., 2020; Chen et al., 2010; Sener & Savarese, 2018; Zheng et al., 2023; Xia et al., 2023) aim to select the coreset that preserves the geometric structure of the original dataset, while removing data points that are close to each other in the feature space. Decision boundary-based methods (Ducoffe & Precioso, 2018; Margatina et al., 2021) exploit the fact that data points nearer the decision boundary generally carry more information value for the model than points far away from it.

Uncertainty-based methodologies posit that samples assigned lower confidence scores by a model exert greater influence on enhancing generalization compared to those with higher confidence. The Selection via Proxy framework (Coleman et al., 2020) employs a reduced-scale proxy model in lieu of the full target model. These computationally efficient models serve as effective surrogates, offering reliable signals for coreset selection grounded in both uncertainty and representativeness. Furthermore, commonly adopted criteria for quantifying sample uncertainty include least-confidence measures (Shen et al., 2018) and entropy-based metrics (Settles, 2012). Loss/Error-based methods (Toneva et al., 2019; Paul et al., 2021) select coresets based on each sample's contribution towards the loss function during model training. Nagaraj et al. (2025) proposed to construct the coreset by considering samples which have a higher correlation to the loss difference between the training and validation trajectory.

Zheng et al. (2023) proposed Coverage-centric Coreset Selection (CCS), which selects a coreset by utilizing both data coverage and sample importance. D^2 Pruning method (Maharana et al., 2024) combines both diversity and difficulty score through message passing on a dataset graph to select the most representative subset. Cho et al. (2025) introduced *Difficulty and Uncertainty Aware Lightweight* (DUAL) score to combine example difficulty and prediction uncertainty to select a coreset during the early training stage.

Gradient Matching-based methods (Mirzasoleiman et al., 2020; Killamsetty et al., 2021) utilize gradients produced by the full training dataset and select a coreset whose weighted gradients would result in the minimal difference. An intuitive coreset selection method termed 'Noise-free Loss Gradients' based on the similarity of loss gradients was proposed by Mohanty et al. (2025), which composes a coreset of samples that have the maximum number of neighbours with higher cosine similarity among their gradients.

2.2 Use of soft labels

In their seminal work on knowledge distillation, Hinton et al. (2015) showed that, by reflecting output distributions of a well-trained model, soft-labels can encode usable information in a way that one-hot labels can not. Szegedy et al. (2016) introduced soft labels through a technique called label smoothing regularization. This work showed that blending true one-hot labels with a soft label scheme of uniform distribution is an effective regularizer of the model.

Soft labels help reduce overfitting by encouraging the network to be less confident in any class, stabilizing training, and improving calibration of predicted probabilities. Lukasik et al. (2020) have shown that, under noisy labels conditions, label smoothing is quite effective.

3 Methodology

In this section, we detail the proposed study. We first train a teacher network on the given large dataset with hard labels. Then, obtain the coreset for the large dataset. Later, we get the corresponding soft labels using one of the two methods (sec. 3.1 and sec. 3.2) described below. Finally, we use the coreset and soft labels to train a new classifier accordingly (sec. 3.3). We perform this analysis for various coreset selection methods and compare their performance with that of their hard label versions.

3.1 Generating soft labels via Teacher Network

The Knowledge Distillation (KD) (Hinton et al., 2015) process needs soft labels to transfer the learned knowledge from a sophisticated teacher onto a simpler student network. For this purpose, the softmax output of the trained teacher is used as the soft labels to train the student network. The rich information in these soft labels enables the simpler student to achieve a better generalization performance compared to training it with hard labels.

In our study, one way of generating the soft labels is via a trained teacher network. Let $\mathbf{z}^{\text{teacher}}$ be the logits predicted by the teacher for a sample in the coreset. $\mathbf{z}^{\text{teacher}} \in \mathbf{R}^K$, where K is the number of classes. Logits refer to the unnormalized output values produced by the final layer of a neural network before applying the softmax (Goodfellow et al., 2016) operation. Let $\tau > 0$ be the temperature parameter. The teacher-based soft labels are given by:

$$\mathbf{y}_i^{\text{teacher}} = \frac{\exp(\mathbf{z}_i^{\text{teacher}}/\tau)}{\sum_{k=1}^{K} \exp(\mathbf{z}_k^{\text{teacher}}/\tau)} \tag{1}$$

3.2 Generating soft labels via label smoothing

We explored Label smoothing, introduced by Szegedy et al. (2016) as another way to obtain soft labels. The standard one-hot vector is replaced by a softened version, where the target becomes a mixture of the original ground truth and a uniform distribution. It can be represented as:

$$\mathbf{y}_i^{\text{smooth}} = \begin{cases} 1-\varepsilon & \text{if } i = \text{ground truth label}, \\ \frac{\varepsilon}{K-1} & \text{otherwise.} \end{cases} \tag{2}$$

where $\varepsilon \in [0, 1]$ is the label smoothing parameter. It may be noted that, the label smoothing does not add any priors unlike Knowledge Distillation and it is primarily used as a regularizer to address overfitting and overconfidence while training deep neural networks.

3.3 Training the classifier on the coreset

The next step after obtaining the soft labels is training the classifier freshly on the coreset in order to assess its effectiveness in representing the original dataset.

3.3.1 With teacher-based soft labels ($\mathbf{y}^{\textbf{teacher}}$)

Similar to the teacher, logit output of the student ($\mathbf{z}^S$) is also processed with temperature (τ) to obtain its prediction ($\hat{\mathbf{y}}$):

$$\hat{\mathbf{y}}_i = \frac{\exp(\mathbf{z}_i^S/\tau)}{\sum_{k=1}^{K} \exp(\mathbf{z}_k^S/\tau)} \tag{3}$$

The classifier training objective over the soft labels is the KL-divergence loss function, which is given by:

$$L_{KD} = \tau^2 KL(\mathbf{y}^{\text{teacher}} \| \hat{\mathbf{y}}) = \tau^2 \sum_{k=1}^{K} \mathbf{y}_k^{\text{teacher}} log \frac{\mathbf{y}_k^{\text{teacher}}}{\hat{\mathbf{y}}_k} \tag{4}$$

In the case of hard labels, the standard cross-entropy loss is used to train the network.

$$L_{CE} = -\sum_{k=1}^{K} \mathbf{y}_k \cdot log(\hat{\mathbf{y}}_k) \tag{5}$$

Where, $\mathbf{y} \in \{0, 1\}^K$ is the ground truth one-hot label for the sample. Finally, a weighted combination of soft and hard label losses trains the student network.

$$L = (1 - \lambda) * L_{CE} + \lambda * L_{KD} \tag{6}$$

Where λ is the weight assigned to the soft label loss. Figure 1 provides an overview of the classifier training on coreset with teacher-based soft labels.

3.3.2 With label-smoothing based soft labels

If we obtain the soft labels for the coreset via label smoothing (Eq. 2), we train the target classifier optimizing the cross-entropy loss (Eq. 5). In our experiments, we have used the PyTorch implementation of cross-entropy loss with label smoothing [1].

3.4 Analysis of soft vs hard labels from Information Theory viewpoint

From an information-theoretic viewpoint, soft labels increase entropy, where each label carries more uncertainty and more mutual information about class structure. Soft labels act as a regularizer by replacing the empirical one-hot targets with a smoother distribution that better approximates the population distribution [2]. This reduces the KL divergence between the model's predictive distribution and the population distribution, compared to training with hard labels.

A hard label for an N-class classification problem is a one-hot vector, where each output $y \in \{0, 1\}$. Entropy of y is given by $H(y) = 0$, as all probability mass is concentrated on the target class.

[1] https://docs.pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html

[2] It refers to the underlying, true distribution of classifications for a given sample, often aggregated from a population of human annotators.

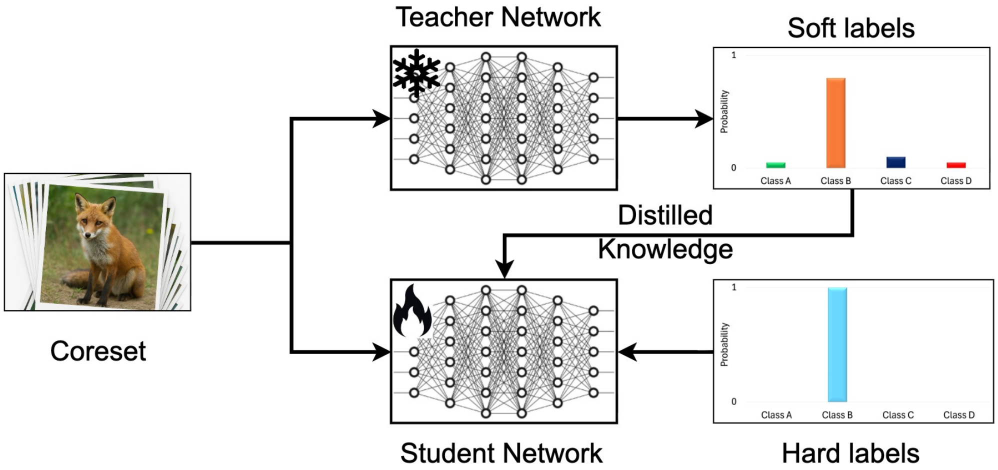

Figure 1: Classifier training with teacher-based soft labels over the coreset. Note that the teacher is trained on the large dataset with hard labels. It supplies the soft labels and is frozen during the training of the student network.

A soft label is a probability distribution, and its entropy is given by:

$$H(y) = -\sum_{i=1}^{N} y_i log(y_i) > 0 \tag{7}$$

The richer supervision due to strictly positive entropy encodes teacher uncertainty and inter-class similarity structure. This additional entropy per sample acts as an informational regularizer, reducing overfitting (Pereyra et al., 2017). In low-data regimes typical of coreset selection, each example must carry more learning signal. Hard labels collapse all probability mass onto a single class, discarding inter-class similarity. By contrast, soft labels increase per-sample entropy and encode richer structure, effectively raising the information budget per example.

4 Experimentation

Applications. We have evaluated the impact of soft labels with coreset selection for image classification with various benchmark datasets.

Datasets. We worked with benchmark datasets for image classification, such as CIFAR-100 (Krizhevsky, 2009), Tiny ImageNet (Le & Yang, 2015), and ImageNet-1K (Deng et al., 2009). To study the impact of soft labels when a coreset is selected from a dataset with significant label noise, we have also considered CIFAR-100N (Wei et al., 2022).

Classifier. We trained randomly initialized ResNet-18 (He et al., 2016), ResNet-50 (He et al., 2016), VGG-16 (Simonyan & Zisserman, 2015) and ViT (Liu et al., 2021) architectures as the target classifiers.

Teacher networks. Unless otherwise specified, a well-trained ResNet-18 model (trained on the full dataset) is used as the teacher network for the teacher-based soft label generation. **Coreset selection methods**. We have considered a variety of popular coreset selection methods, such as CRAIG (Mirzasoleiman et al., 2020), Forgetting (Toneva et al., 2019), GradMatch (Killamsetty et al., 2021), GraNd (Paul et al., 2021), GraphCut (Wei et al., 2015), Moderate (Xia et al., 2023), CCS (Zheng et al., 2023), D^2 pruning (Maharana et al., 2024), DUAL (Cho et al., 2025), CLD (Nagaraj et al., 2025), and Noise-free gradients (NFG) (Mohanty et al., 2025) on which the proposed study is carried out. Importantly, we have also studied the impact of soft labels on a randomly selected subset as the coreset.

We have utilized the DeepCore (Guo et al., 2022) library [3] for the implementation of CRAIG, Forgetting, GradMatch, GraNd, and GraphCut methods. We have used publicly available codebase released by the authors of respective papers for the implementation of Moderate [4], CCS [5], D^2 pruning [6], DUAL [7], NFG [8], and CLD [9]. The CCS and D^2 pruning methods use iterations instead of epochs for training over a selected coreset. The number of epochs varies based on the size of the coreset chosen. Thus, we have presented the results obtained on CCS and D^2 pruning separately in subsection 4.2.

4.1 Results

We have observed that soft labels generated through the teacher network outperform those generated using label smoothing. This is expected as teacher-generated soft labels encode reliable information than randomly assigned probabilities through label smoothing. Hence, all our experimental results reported in the main paper are based on teacher-generated soft labels. We have carried out each experiment 5 times with different random seeds and have reported the mean and standard deviation of the accuracy values obtained. For completeness, the results obtained with the label smoothing-based soft labels are provided in the appendix. The best performances are highlighted in bold, while the second-best performances are underlined.

4.1.1 Result on CIFAR-100

Table 1 presents the test set accuracies obtained with hard (HL) and soft labels (SL) for the CIFAR-100 dataset with ResNet-18 architecture. Two noteworthy observations are (i) Soft labels improve generalization accuracy for all the coreset selection methods, (ii) random selection with soft labels performs on par and sometimes outperforms the existing coreset selection methods.

4.1.2 Result on Tiny ImageNet

Figure 2 shows test set accuracy values obtained on ResNet-18 and ViT architectures for the Tiny ImageNet dataset. The coreset selection methodologies we have considered are Moderate (Xia et al., 2023), Noise Free Gradients (NFG) (Mohanty et al., 2025). It can be noted that while random selection with hard labels performs less than the coreset selection methods, random selection with soft labels performs on par with NFG and better than the moderate selection method.

4.1.3 Result on ImageNet-1K

Table 2 compares the results obtained with hard labels and soft labels on the ImageNet-1K dataset with the ResNet-18 architecture. Results for ViT architecture are shown in Figure 3. Soft labels enhance the performance of the student network quite significantly. However, similar to other experiments, random selection with soft labels outperforms other coreset selection methods at 5% and 10% selection ratios, underscoring the importance of soft labels when training the student network on a limited dataset.

4.2 Result for CCS and D^2 pruning

Table 3 compares the performance of CSS and D^2 method in hard and soft label settings for the CIFAR-100 dataset with the ResNet-18 architecture. Comparing with Random selection with soft labels, we can observe that random selection with soft labels is performing almost on par with these two coreset selection methods, beating them narrowly for 50% selection ratio. Figure 4 compares the performance of CCS and D^2 methods with hard and soft labels and random selection with soft labels for ImageNet-1K dataset with ResNet-34 architecture. Random selection with soft labels performs at par with CCS with soft labels while outperforming D^2 with soft labels for lower selection fractions.

[3] https://github.com/PatrickZH/DeepCore

[4] https://github.com/tmllab/2023_ICLR_Moderate-DS

[5] https://github.com/haizhongzheng/Coverage-centric-coreset-selection

[6] https://github.com/adymaharana/d2pruning

[7] https://github.com/behaapyy/dual-pruning

[8] https://github.com/ai23resch04001/Noise_free_gradient

[9] https://github.com/manishnagaraj/CLD_Coresets_from_Trajectories

Table 1: Performance of the coreset selection methods with hard and soft labels for the CIFAR-100 dataset with the ResNet-18 architecture. Various Coreset sizes are considered from 0.5% to 20% of the full dataset. The best performance in each column is presented in bold, and the second best is underlined. While soft labels can improve test set accuracy on all the methods, random selection with soft labels performs on par with other coreset selection methods. Test accuracy with a model trained on full dataset is 77.41%.

Methods	0.5%	1%	5%	10%	20%
GradMatch (HL)	4.15 ± 0.25	11.07 ± 0.03	28.88 ± 1.45	40.04 ± 1.39	56.18 ± 0.82
GradMatch (SL)	10.84 ± 0.07	23.02 ± 0.49	57.11 ± 0.02	59.24 ± 0.08	60.80 ± 0.12
Craig (HL)	6.44 ± 0.38	10.19 ± 0.15	19.64 ± 0.99	26.45 ± 0.59	45.14 ± 0.45
Craig (SL)	11.41 ± 0.44	19.77 ± 0.10	<u>57.34 ± 0.11</u>	59.26 ± 0.07	60.34 ± 0.08
GraphCut (HL)	8.18 ± 0.41	12.52 ± 0.07	30.30 ± 0.62	41.05 ± 0.80	56.78 ± 0.32
GraphCut (SL)	13.69 ± 0.19	24.08 ± 0.15	57.22 ± 0.32	<u>59.41 ± 0.22</u>	60.70 ± 0.04
GraNd (HL)	2.38 ± 0.23	3.71 ± 0.05	7.92 ± 0.33	12.83 ± 0.10	32.02 ± 1.02
GraNd (SL)	9.68 ± 0.22	18.87 ± 0.53	56.37 ± 0.08	58.12 ± 0.05	60.06 ± 0.08
Forgetting (HL)	6.10 ± 0.43	9.45 ± 0.07	24.08 ± 0.22	37.63 ± 0.09	56.09 ± 0.31
Forgetting (SL)	11.33 ± 0.35	<u>27.13 ± 0.09</u>	55.26 ± 0.26	56.83 ± 0.15	58.07 ± 0.13
Moderate (HL)	5.36 ± 0.12	7.78 ± 0.20	18.62 ± 0.33	32.57 ± 1.40	54.40 ± 0.19
Moderate (SL)	9.26 ± 0.09	14.67 ± 0.21	51.18 ± 0.31	57.61 ± 0.27	58.35 ± 0.32
DUAL (HL)	11.13 ± 0.35	15.10 ± 0.71	38.43 ± 0.52	53.37 ± 0.23	57.63 ± 0.45
DUAL (SL)	<u>14.87 ± 0.43</u>	20.37 ± 0.22	49.19 ± 0.15	**61.95 ± 0.23**	<u>63.12 ± 0.69</u>
CLD (HL)	7.83 ± 0.17	8.73 ± 0.51	23.06 ± 0.39	36.04 ± 0.47	52.32 ± 0.28
CLD (SL)	12.23 ± 0.21	14.15 ± 0.19	40.04 ± 0.46	56.97 ± 0.51	**66.22 ± 0.12**
NFG (HL)	12.00 ± 0.11	18.95 ± 0.08	37.67 ± 0.10	46.48 ± 0.28	58.06 ± 0.34
NFG (SL)	11.98 ± 0.80	26.34 ± 0.20	56.88 ± 0.50	58.51 ± 0.30	59.38 ± 0.10
Random (HL)	7.97 ± 0.03	12.82 ± 0.16	31.93 ± 0.15	41.00 ± 0.04	52.75 ± 0.20
Random (SL)	**16.26 ± 0.91**	**36.07 ± 0.15**	**57.35 ± 0.27**	58.45 ± 0.08	59.24 ± 0.08

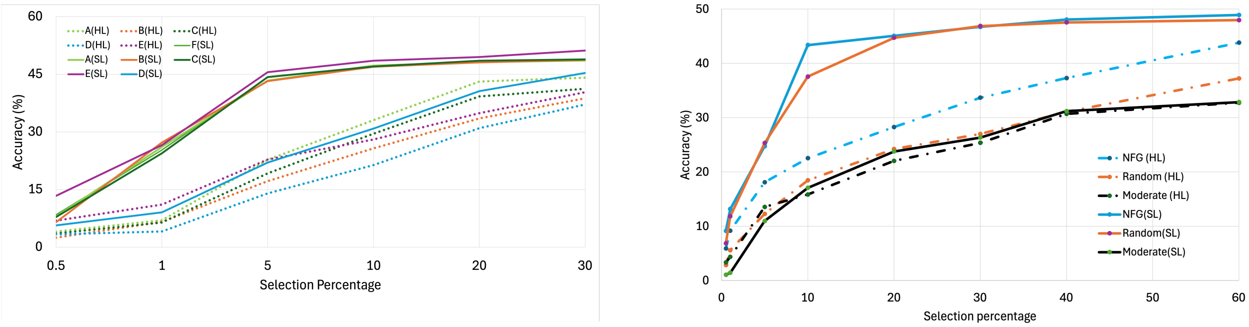

Figure 2: Test set accuracy obtained on Tiny ImageNet Dataset with hard label and soft label with knowledge distillation. The left panel shows results obtained on the ResNet-18 architecture (Methods used are, A: GraphCut, B: GradMatch, C: Facility Location, D: Moderate, E: NFG, F: Random), and the right panel shows results obtained on the ViT architecture. HL refers to training with hard labels, while SL refers to training with soft labels.

4.3 Ablation Studies

In this section, we perform ablation studies to analyze the impact of various hyperparameters on generalization accuracy.

Table 2: Top-1 accuracies for ImageNet-1K dataset with ResNet-18 architecture. Soft labels are generated through the teacher network. As can be seen, soft labels can enhance the performance of the student network quite significantly. However, similar to other experiments, the random selection with soft labels is performing almost on par with other methods, underlying the significance of soft labels during the training of the student network with a limited dataset. Test accuracy with a model trained on full dataset is 69.76%.

Methods	0.1%	0.5%	1%	5%	10%
GradMatch (HL)	0.93 ± 0.04	5.20 ± 0.22	12.28 ± 0.49	40.16 ± 2.28	45.91 ± 1.73
GradMatch (SL)	2.13 ± 0.12	13.85 ± 0.31	28.54 ± 0.24	52.56 ± 0.23	58.14 ± 0.47
Craig (HL)	1.13 ± 0.08	5.44 ± 0.52	9.40 ± 1.69	32.30 ± 1.24	38.77 ± 0.56
Craig (SL)	2.23 ± 0.12	16.52 ± 0.37	32.67 ± 0.21	56.92 ± 0.34	59.38 ± 0.17
GraphCut (HL)	1.21 ± 0.09	7.66 ± 0.43	16.43 ± 0.53	42.23 ± 0.60	50.53 ± 0.42
GraphCut (SL)	2.24 ± 0.23	14.84 ± 0.37	31.08 ± 0.17	56.56 ± 0.21	59.48 ± 0.17
Forgetting (HL)	0.76 ± 0.01	4.69 ± 0.17	14.02 ± 0.13	47.64 ± 0.03	55.12 ± 0.13
Forgetting (SL)	2.02 ± 0.13	15.87 ± 0.15	32.12 ± 0.31	56.65 ± 0.24	59.50 ± 0.18
NFG (HL)	1.88 ± 0.10	11.58 ± 0.15	22.82 ± 0.07	43.50 ± 0.41	49.51 ± 0.26
NFG (SL)	**4.70 ± 0.17**	**21.22 ± 0.14**	**37.77 ± 0.23**	55.50 ± 0.27	58.60 ± 0.41
Random (HL)	0.76 ± 0.01	3.78 ± 0.14	8.85 ± 0.46	40.09 ± 0.21	52.10 ± 0.22
Random (SL)	1.97 ± 0.20	15.78 ± 0.14	32.26 ± 0.23	**56.97 ± 0.18**	**59.61 ± 0.31**

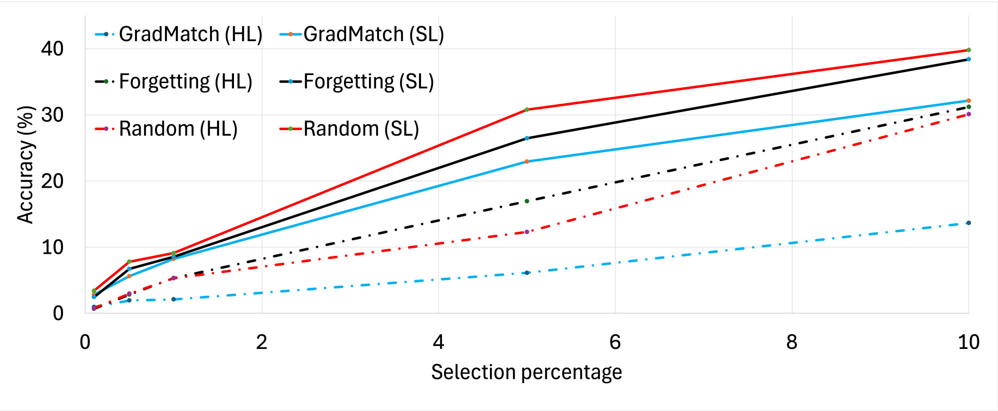

Figure 3: Test set accuracy comparison for various coreset selection methods with hard and soft labels, on ViT architecture and ImageNet-1K dataset.

Table 3: Comparison of test set accuracies on CIFAR-100 dataset with ResNet-18 architecture with hard and soft labels with CSS and D^2 pruning methods. Random selection with soft labels is also provided. As can be seen, random selection with soft labels is performing on par with coreset selected through these two methods for majority of selection ratios. Test accuracy with a model trained on full dataset is 77.41%.

Selection ratio	Hard Label		Soft Label		Random + SL
	CCS	D^2	CCS	D^2	
0.01	12.0 ± 0.7	11.1 ± 0.1	17.1 ± 0.6	**17.2 ± 0.8**	14.8 ± 0.5
0.05	40.6 ± 0.3	31.1 ± 1.2	**56.4 ± 0.3**	50.1 ± 1.4	49.0 ± 0.6
0.1	55.0 ± 0.7	56.8 ± 0.2	65.4 ± 0.6	**66.2 ± 0.4**	63.5 ± 0.3
0.2	63.2 ± 0.5	64.9 ± 0.7	71.4 ± 0.8	**72.4 ± 0.2**	71.3 ± 0.2
0.3	68.8 ± 0.1	70.2 ± 0.5	74.5 ± 0.2	**74.6 ± 0.7**	74.2 ± 0.2
0.5	73.8 ± 0.2	75.3 ± 0.4	76.4 ± 0.7	75.4 ± 0.1	**76.6 ± 0.3**

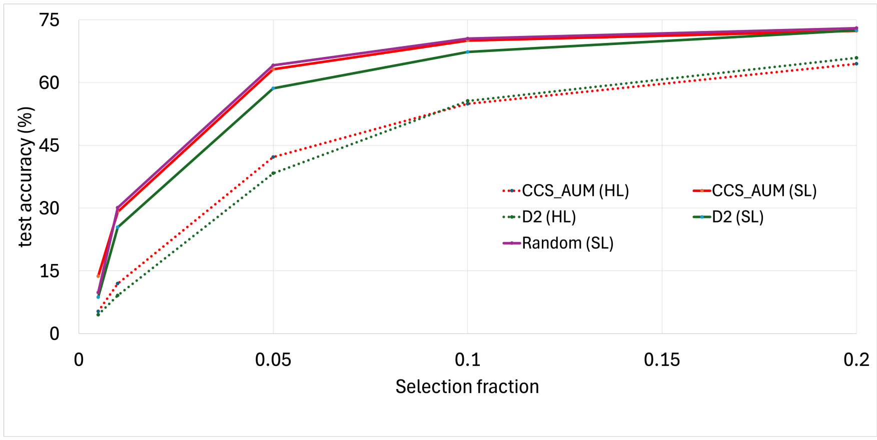

Figure 4: Comparison of test set accuracies on ImageNet dataset with ResNet-34 architecture with hard and soft labels with CSS and D^2 pruning methods. Random selection with soft labels performs at par with CCS + soft labels and outperforms the D^2 + soft labels.

4.3.1 Impact of the weight on the soft labels towards generalization

Eq. 6 describes how both hard label loss (cross entropy loss) and soft label loss (KL-divergence) are weighted by the λ to constitute the total loss function. Figure 5 shows the impact of the weight on the soft label loss on the test set accuracy. As the soft label loss weightage increases, accuracy also increases.

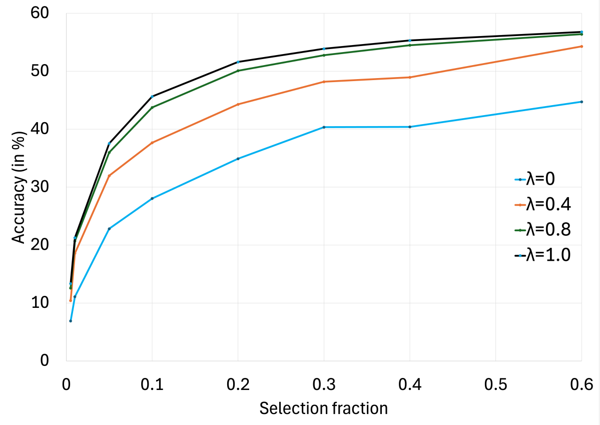

Figure 5: Impact of soft label loss weightage on test set accuracy. The graph shows that using pure soft labels (corresponding to $\lambda = 1.0$) gives the best results, while pure hard label loss gives the poorest results. Results are obtained for the Tiny ImageNet dataset on the ResNet-18 architecture with Noise Free Gradients (Mohanty et al., 2025) as the coreset selection method.

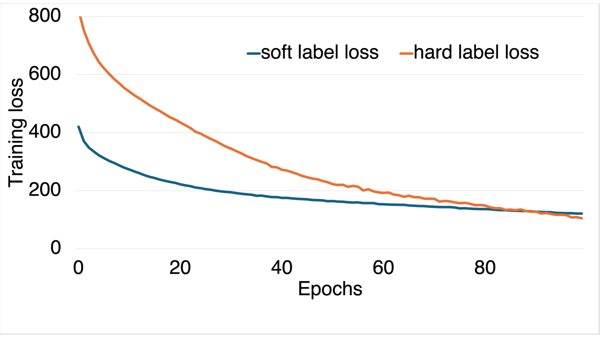

Figure 6: Comparison of training loss progression with soft label and hard label. Results are shown for the Tiny ImageNet dataset, with a random selection baseline for 20% coreset selection. Training with soft labels converges much faster than training with hard labels.

4.3.2 Training dynamics comparison

Figure 6 shows the progression of the training loss while training with hard and soft labels. Keeping all the hyperparameters equal, training with soft labels converges faster than training with hard labels. It signifies a potential for training for a lesser number of epochs to obtain better results. It also shows that soft labels possess more learnable information than hard labels.

4.3.3 Impact of temperature

The temperature parameter τ is vital in smoothing the soft labels (Müller et al., 2019). Table 4 tabulates test set accuracy vs the value of τ.

Table 4: Impact of temperature parameter on test-set accuracy. Results are obtained on ImageNet-1K dataset at various selection percentages with ResNet-18 architecture. Test set accuracy first increases with temperature, then decreases further.

Temperature	Accuracy (0.5%)	Accuracy (1%)
1	19.22 ± 0.91	28.76 ± 0.11
5	20.77 ± 0.21	30.79 ± 0.12
10	**21.52 ± 0.13**	**31.10 ± 0.27**
20	20.75 ± 0.22	28.49 ± 0.15
30	20.67 ± 0.16	27.45 ± 0.33

4.4 Cross architecture performance study

In this section, we study the cross-architecture performance of various coreset selection methods and compare against a random selection baseline, under training with soft labels. The coreset selection is carried out by the ResNet-18 architecture for Forgetting, GradMatch and NFG methods, while ResNet-34 architecture is used for CCS and D^2 coreset selection method. Cross-architectural performance is tested on VGG-16 and ViT architectures, on ImageNet-1K dataset. The results are tabulated in Tables 5 and 6. We can observe that random selection with soft labels is performing on par with other coreset selection methods.

Table 5: Cross architecture evaluation for various coreset selection methods with hard labels and soft labels. The coresets are generated through the ResNet-18 architecture and evaluated on the VGG-16 and ViT architectures. In the cases of soft labels, ResNet-18 is used as the teacher model.

Target Architecture→	VGG-16		ViT	
Selection % →	0.5%	1%	0.5%	1%
Forgetting (HL)	4.34 ± 0.21	9.12 ± 0.17	2.68 ± 0.08	4.86 ± 0.31
Forgetting (SL)	24.01 ± 0.13	40.36 ± 0.27	3.64 ± 0.18	5.88 ± 0.19
GradMatch (HL)	0.71 ± 0.21	8.71 ± 0.23	1.08 ± 0.14	1.53 ± 0.17
GradMatch (SL)	22.66 ± 0.23	37.61 ± 0.17	2.85 ± 0.25	3.99 ± 0.41
NFG (HL)	10.89 ± 0.37	18.03 ± 0.41	3.64 ± 0.23	6.09 ± 0.65
NFG (SL)	**25.45 ± 0.29**	**42.54 ± 0.69**	**5.12± 0.33**	**7.13 ± 0.43**
Random (HL)	1.92 ± 0.16	8.83 ± 0.23	2.46 ± 0.31	4.82 ± 0.32
Random (SL)	23.63 ± 0.31	41.04 ± 0.13	3.73 ± 0.17	5.91 ± 0.17

Table 6: Cross architecture evaluation for various coreset selection methods with hard labels and soft labels. The coresets are generated through the ResNet-34 architecture and evaluated on the VGG-16 and ViT architectures. In the cases of soft labels, ResNet-34 is used as the teacher model.

Target Architecture→	VGG-16		ViT	
Selection % →	0.5%	1%	0.5%	1%
CCS (HL)	5.46 ± 0.67	11.04 ± 0.26	1.11 ± 0.08	2.54 ± 0.17
CCS (SL)	**21.91 ± 0.18**	38.31 ± 0.47	1.47 ± 0.71	3.12 ± 0.81
D^2 (HL)	4.54 ± 0.83	9.02 ± 0.07	1.02 ± 0.05	2.43 ± 0.19
D^2 (SL)	16.84 ± 0.29	33.61 ± 0.52	1.21 ± 0.16	**3.43 ± 0.28**
Random (HL)	4.03 ± 0.17	8.56 ± 0.49	1.07 ± 0.15	2.63 ± 0.14
Random (SL)	18.03 ± 0.25	**38.34 ± 0.46**	**1.55 ± 0.19**	3.31 ± 0.82

4.5 Impact on dataset with label noise

We have studied the impact of soft labels with a coreset selected from the CIFAR-100N dataset, which is a dataset specifically generated to study the impact of label noise introduced by human annotators. Table 7 provides results for the CIFAR-100N dataset with the ResNet-18 architecture. Soft labels are generated through the teacher network. Random selection with soft labels outperforms other coreset selection methods for lower selection percentages, while performs second-best for 10% and 20% selection.

Table 7: Results for CIFAR-100N dataset. ResNet-18 architecture is used. While soft labels are able to improve test set accuracy on all the methods, random selection with soft labels performs on par with other coreset selection methods.

Methods	0.5%	1%	5%	10%	20%
GradMatch (HL)	4.96 ± 0.13	8.89 ± 0.38	19.91 ± 0.41	21.15 ± 0.11	32.21 ± 0.52
GradMatch (SL)	9.53 ± 0.39	14.61 ± 0.59	30.98 ± 0.05	38.22 ± 0.23	40.99 ± 0.12
Craig (HL)	5.83 ± 0.13	7.98 ± 0.40	11.12 ± 0.11	19.00 ± 0.25	21.29 ± 0.12
Craig (SL)	9.92 ± 0.02	12.95 ± 0.24	27.60 ± 0.71	37.15 ± 0.12	40.02 ± 0.02
GraphCut (HL)	6.84 ± 0.11	9.84 ± 0.33	20.07 ± 0.07	22.27 ± 0.04	33.61 ± 1.22
GraphCut (SL)	10.22 ± 0.26	12.45 ± 0.12	27.88 ± 0.55	37.18 ± 0.29	40.55 ± 0.03
NFG (HL)	7.71 ± 0.27	11.34 ± 0.30	30.87 ± 0.92	40.88 ± 0.54	49.21 ± 0.16
NFG (SL)	9.15 ± 0.43	22.91 ± 0.36	41.72 ± 0.21	**42.91 ± 0.17**	**50.01 ± 0.24**
random (HL)	7.06 ± 0.52	8.59 ± 0.28	22.97 ± 1.15	29.65 ± 0.31	42.81 ± 0.21
random (SL)	**16.9 ± 0.40**	**29.77 ± 0.21**	**42.31 ± 0.23**	42.86 ± 0.26	43.04 ± 0.15

4.6 Impact of teacher accuracy on students' performance

In this experiment, we use multiple checkpoints of the teacher model to train the student model for various percentages of coreset selection. Figure 7 shows that the student network performs better on the same coreset when the teacher network accuracy is higher. The results are shown for ImageNet-1K for 0.5%, 1%, and 5% of coreset selection. A more accurate teacher generates soft labels that better capture true inter-class relationships and uncertainty, rather than noisy or misleading distributions. These high-quality soft labels act as a stronger supervisory signal, guiding the student toward more robust decision boundaries even when trained on small coresets. Consequently, the student benefits from the teacher's superior knowledge, leading to improved generalization performance.

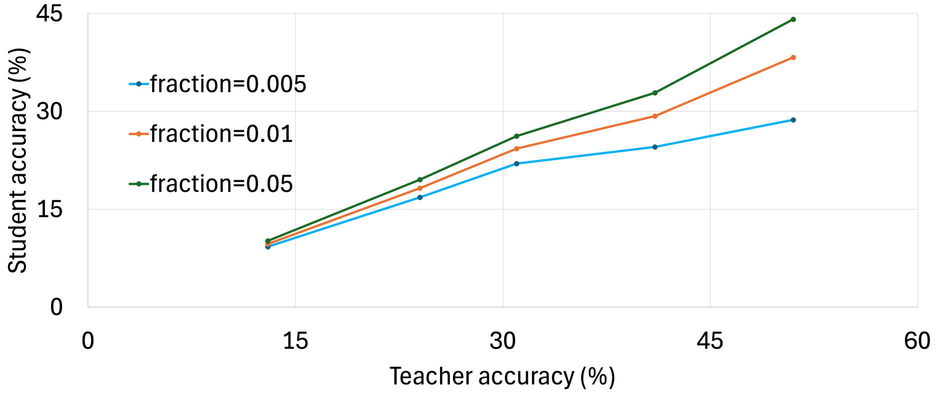

Figure 7: Impact of teacher model accuracy on performance of the student network for various percentages of coreset selection on ImageNet-1K dataset. As we can observe, the student performs better when the teacher's accuracy is improved.

5 Discussion

Our results reveal a striking and consistent trend: introducing soft labels substantially improves the performance of models trained on coresets, to the extent that random selection with soft labels often matches or surpasses sophisticated coreset selection algorithms with hard labels. This finding carries several important implications.

Rethinking evaluation baselines. Traditionally, coreset selection methods are benchmarked against random selection with hard labels. Our results show that this baseline underestimates (at least on existing image recognition benchmarks) what can be achieved with minimal selection bias once label information is enriched. Furthermore, it may be noted that the teacher model required to produce informative soft labels is trained on the full dataset. In other words, a well-trained teacher improves students' performance better than a poorly trained teacher. Consequently, we argue that random selection with soft labels should be adopted as a new standard baseline when evaluating future coreset methods.

Interpreting the role of soft labels. Soft labels encode richer information about inter-class similarity and teacher uncertainty. When coresets drastically reduce the available training data, this additional structure mitigates overfitting and improves calibration, providing robustness especially in noisy-label regimes such as CIFAR-100N. Our ablations on λ (soft label weight) and τ (temperature) confirm that carefully tuned soft labels can drive consistent generalization gains.

Beyond image classification. Although our experiments focus on standard image classification benchmarks, the principle is not domain-specific. We expect similar dynamics to emerge in text and audio tasks where inter-class structure is informative. Extending this investigation to other modalities and downstream applications is a promising direction for future work.

Limitations. Our findings do not imply that subset selection is obsolete. Subset choice continues to matter in scenarios where teacher models are unavailable or too costly (need to train on the entire dataset) and storage/memory constraints prohibit dense soft-label distributions.

In summary, this study demonstrates that the perceived advantage of coreset methods can be largely neutralized once soft labels are introduced. Any future evaluation of coreset selection should include random selection with soft labels as a baseline.

6 Conclusion

In this work, for the first time, we systematically examined the role of soft labels in coreset selection for data-efficient learning. We found that soft labels consistently enhance performance across multiple datasets, architectures, and coreset algorithms. Remarkably, even simple random selection combined with soft labels often achieves accuracy comparable to or better than the state-of-the-art coreset selection methods using hard labels.

Our study also highlights a broader insight: investing in improved teacher models or better soft-label generation methodology may provide greater benefits than devising increasingly complex selection methods. At the same time, coresets remain relevant in scenarios where teacher models are unavailable or computational/storage budgets prohibit them. Overall, our study suggests that the benchmark for coreset methods should be redefined to include random selection with soft labels as a stronger baseline.

Looking ahead, we believe our study opens two promising avenues: (1) extending the analysis of soft labels with coresets beyond image classification into object detection, text, audio, and multimodal domains, and (2) exploring hybrid methods that jointly optimize subset selection and label softening.

References

Marcel Aach, Eray Inanc, Rakesh Sarma, Morris Riedel, and Andreas Lintermann. Large scale performance analysis of distributed deep learning frameworks for convolutional neural networks. *Journal of Big Data*, 10(1), June 2023. ISSN 2196-1115. doi: 10.1186/s40537-023-00765-w. URL http://dx.doi.org/10.1186/s40537-023-00765-w.

Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI*, pp. 137–153, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58516-7. doi: 10.1007/978-3-030-58517-4_9. URL https://doi.org/10.1007/978-3-030-58517-4_9.

Laith Alzubaidi, Jinshuai Bai, Aiman Al-Sabaawi, Jose Santamaría, A. S. Albahri, Bashar Sami Nayyef Al-dabbagh, Mohammed A. Fadhel, Mohamed Manoufali, Jinglan Zhang, Ali H. Al-Timemy, Ye Duan, Amjed Abdullah, Laith Farhan, Yi Lu, Ashish Gupta, Felix Albu, Amin Abbosh, and Yuantong Gu. A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications. *Journal of Big Data*, 10(1):46, Apr 2023. ISSN 2196-1115. doi: 10.1186/s40537-023-00727-2. URL https://doi.org/10.1186/s40537-023-00727-2.

Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, pp. 109–116, Arlington, Virginia, USA, 2010. AUAI Press. ISBN 9780974903965.

Yeseul Cho, Baekrok Shin, Changmin Kang, and Chulhee Yun. Lightweight dataset pruning without full training via example difficulty and prediction uncertainty. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=9rLxi2cnZC.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJg2b0VYDr.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009.

Jiawei Du, Xin Zhang, Juncheng Hu, Wenxin Huang, and Joey Tianyi Zhou. Diversity-driven synthesis: Enhancing dataset distillation through directed weight adjustment. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=uwSaDHLlYc.

Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach, 2018. URL https://arxiv.org/abs/1802.09841.

Dan Feldman. Introduction to core-sets: an updated survey, 2020. URL https://arxiv.org/abs/2011.09384.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, 2016. URL http://www.deeplearningbook.org.

Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. In *Database and Expert Systems Applications: 33rd International Conference, DEXA 2022, Vienna, Austria, August 22–24, 2022, Proceedings, Part I*, pp. 181–195, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-031-12422-8. doi: 10.1007/978-3-031-12423-5_14. URL https://doi.org/10.1007/978-3-031-12423-5_14.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. URL https://api.semanticscholar.org/CorpusID:7200347.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training, 2021.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical Report TR-2009, University of Toronto, Toronto, ON, Canada, 2009.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. https://tinyimagenet.herokuapp.com/, 2015. Accessed: 2025-08-13.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18 (185):1–52, 2018. URL http://jmlr.org/papers/v18/16-558.html.

Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco De Nadai. Efficient training of visual transformers with small datasets. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Krishna Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

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. URL https://openreview.net/forum?id=thbtoAkCe9.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 650–663, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-main.51.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Saumyaranjan Mohanty, Chimata Anudeep, and Konda Reddy Mopuri. Noise-free loss gradients: A surprisingly effective baseline for coreset selection. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=OE4P1tW8iQ.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. *When does label smoothing help?* Curran Associates Inc., Red Hook, NY, USA, 2019.

Manish Nagaraj, Deepak Ravikumar, and Kaushik Roy. Coresets from trajectories: Selecting data via correlation of loss differences. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=QYOpbZTWJ9.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training, 2021. URL https://arxiv.org/abs/2104.10350.

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. URL https://openreview.net/forum?id=Uj7pF-D-YvT.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions, 2017. URL https://openreview.net/forum?id=HkCjNI5ex.

Tian Qin, Zhiwei Deng, and David Alvarez-Melis. A label is worth a thousand images in dataset distillation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=oNMnR0NJ2e.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1aIuk-RW.

Burr Settles. *Active Learning*. Springer International Publishing, 2012. ISBN 9783031015601. doi: 10.1007/978-3-031-01560-1. URL http://dx.doi.org/10.1007/978-3-031-01560-1.

Yanyao Shen, Hyokun Yun, Zachary C. Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1409.1556.

Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.

Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. The computational limits of deep learning, 2022. URL https://arxiv.org/abs/2007.05558.

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.

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TBWA6PLJZQm.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Data subset selection via graph cuts. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 1077–1085. PMLR, 2015.

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. URL https://openreview.net/forum?id=7D5EECbOaf9.

Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=5Fgdk3hZpb.

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. URL https://openreview.net/forum?id=QwKvL6wC8Yi.

Qing Zhou, Junyu Gao, and Qi Wang. Scale efficient training for large datasets. In *CVPR*, 2025.

Barret Zoph and Quoc Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=r1Ue8Hcxg.

A Appendix

In this appendix, we provide some additional insights and results to aid the analysis provided in the main paper.

A.1 Analysis of soft vs hard labels from Information Theory viewpoint

From an information-theoretic viewpoint, soft labels increase entropy, where each label carries more uncertainty and more mutual information about class structure. Hence, soft labels act as a regularizer, reducing the KL divergence between empirical and population distributions.

A hard label for an N-class classification problem is a one-hot vector, where each output $y \in \{0, 1\}$. Entropy of y is given by $H(y) = 0$, as all probability mass is concentrated on the target class.

A soft label is a probability distribution, and its entropy is given by:

$$H(y) = \sum_{i=1}^{N} y_i log(y_i) > 0 \tag{8}$$

The richer supervision due to strictly positive entropy encodes teacher uncertainty and inter-class similarity structure. This additional entropy per sample acts as an informational regularizer, reducing overfitting (Pereyra et al., 2017).

A.2 Results for VGG16 architecture on CIFAR-100 dataset

Table 8: Results for CIFAR-100 dataset. VGG16 architecture is used. While soft labels are able to improve test set accuracy on all the methods, random selection with soft labels performs on par with other coreset selection methods.

Methods	0.5%	1%	5%	10%	20%
GradMatch (HL)	1.87 ± 0.28	4.32 ± 1.29	18.52 ± 0.43	31.90 ± 0.38	45.8 ± 0.32
GradMatch (SL)	**6.12 ± 0.11**	12.15 ± 0.04	41.73 ± 0.46	49.80 ± 0.37	55.12 ± 0.01
Craig (HL)	1.94 ± 0.33	3.39 ± 0.26	18.91 ± 0.93	30.37 ± 0.09	50.2 ± 0.25
Craig (SL)	6.08 ± 0.10	**12.89 ± 0.08**	46.14 ± 0.58	52.16 ± 0.01	56.76 ± 0.01
GraphCut (HL)	2.42 ± 0.21	7.13 ± 0.03	27.49 ± 1.07	40.10 ± 0.26	52.72 ± 0.15
GraphCut (SL)	5.75 ± 0.33	14.91 ± 1.08	45.72 ± 0.33	51.75 ± 0.17	55.77 ± 0.05
GraNd (HL)	1.73 ± 0.29	1.94 ± 0.42	8.75 ± 1.16	18.68 ± 1.37	39.07 ± 0.27
GraNd (SL)	5.54 ± 0.51	12.24 ± 0.09	40.12 ± 0.47	49.49 ± 0.37	54.92 ± 0.02
Moderate (HL)	2.00 ± 0.10	2.10 ± 0.23	13.70 ± 0.91	28.10 ± 1.60	49.90 ± 0.30
Moderate (SL)	5.15 ± 0.41	8.02 ± 0.28	41.64 ± 0.18	52.33 ± 0.42	56.58 ± 0.45
random (HL)	2.73 ± 0.17	4.75 ± 0.2	20.18 ± 0.18	24.54 ± 1.99	40.59 ± 1.05
random (SL)	5.88 ± 0.02	12.64 ± 0.49	**46.45 ± 0.08**	**52.94 ± 0.19**	**56.98 ± 0.16**

A.3 Comparison between soft labels generated through label smoothing and teacher network

Table 9 compares results obtained by soft labels generated from label smoothing and through a teacher network on the Tiny ImageNet dataset with the ResNet-18 architecture. Soft labels generated through the teacher network perform much better compared to soft labels generated through label smoothing, which is expected, as teacher network-generated soft labels encode more features than randomly assigned probabilities through label smoothing.

A.4 Impact of ε in label smoothing

The hyperparameter ε controls the label smoothing (Eq. 2). Table 10 shows the impact of ε on test set accuracy for 10% and 20% coreset selection on the Tiny ImageNet dataset with the ResNet-18 architecture.

Table 9: Comparative analysis of performance for soft labels generated through label smoothing (LS) and teacher network (TN). The results are obtained for training the ResNet-18 network on the Tiny ImageNet dataset with the Noise Free Gradient coreset selection method.

Selection %	LS	TN
0.5	6.84 ± 0.02	**13.37 ± 0.14**
1.0	12.60 ± 0.14	**21.28 ± 0.08**
5.0	26.16 ± 0.01	**37.55 ± 0.17**
10.0	31.80 ± 0.20	**45.55 ± 0.23**
20.0	38.50 ± 0.15	**51.61 ± 0.21**
30.0	41.93 ± 0.11	**53.89 ± 0.34**
40.0	45.49 ± 0.23	**55.32 ± 0.21**

Similar to the temperature hyperparameter for teacher-generated soft label generation, test set accuracy first increases with ε, but then it decreases.

Table 10: Impact of label smoothing parameter ε on test set accuracy for 10% and 20% coreset selection. Results are obtained on the Tiny ImageNet dataset with the ResNet-18 architecture.

$\varepsilon \downarrow$ Selection Ratio $\rightarrow$	10%	20%
0	31.3 ± 0.2	37.6 ± 0.2
0.05	31.5 ± 0.1	37.8 ± 0.3
0.1	31.4 ± 0.1	37.8 ± 0.3
0.15	**31.8 ± 0.2**	**38.5 ± 0.1**
0.2	31.4 ± 0.1	38.3 ± 0.3
0.3	31.5 ± 0.1	37.8 ± 0.4

A.5 Impact of τ and λ

In this experiment, we study how various combinations of temperature parameter τ and soft label loss weightage λ impact test set accuracy. Figure 8 shows the impact of various combinations of τ and λ on test set accuracy. The combination of higher temperature (τ) and soft label weight (λ) yields optimal accuracy because it maximizes the entropy and inter-class similarity encoded in soft labels, providing richer supervision than hard labels. In low-data coreset regimes, this enhanced information per sample reduces overfitting and improves generalization. The combination of flatter soft label distribution and higher emphasis on this during training results in better performance.

A.6 Compute requirement analysis

Table 11 provides a comparative analysis of the memory requirements for training with hard vs. soft labels.

Table 11: Memory requirement comparison for training random selection with hard labels and soft labels. Training is carried out on ImageNet-1K dataset for 0.5% coreset selection, with ResNet-18 architecture. A batch size of 128 is used. The additional memory requirement resulted in an increase in accuracy from 3.78% to 15.79%.

Method	Memory requirement
Random (HL)	4.7 GB
Random (SL)	6.5 GB

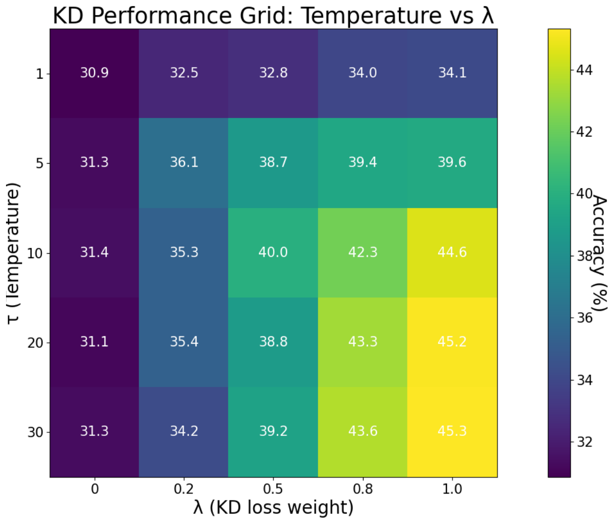

Figure 8: Heatmap of accuracy values obtained on test dataset of Tiny ImageNet with ResNet-18 architecture by varying temperature τ and soft label loss weight λ. We can observe that a combination of higher temperature τ and 100% soft labels provides the optimal result.

A.7 Computation Experiment Information

Information regarding computing infrastructure for carrying out the experiments is provided in Table 12.

Table 12: Details of the computing infrastructure.

Description	Value
GPU	32 GB Tesla V100
OS	Ubuntu 18.04.1 LTS
Python	3.11.7
Pytorch	2.4.1
NumPy	1.24.3