Spanning Training Progress: Temporal Dual-Depth Scoring (TDDS) for Enhanced Dataset Pruning

Supplementary Material

A. Derivation of Equation 5

The objective in Equation 4 is to minimize the Mean Squared Error (MSE) between $\boldsymbol{\mathcal{G}}_{t,\mathbb{U}} \in \mathbb{R}^{1\times N}$ and $\tilde{\boldsymbol{\mathcal{G}}}_{t,\mathbb{S}} \in \mathbb{R}^{1\times N}$,

$$\mathcal{J} = \frac{1}{T}\sum_{t=1}^{T} \|\boldsymbol{\mathcal{G}}_{t,\mathbb{U}} - \tilde{\boldsymbol{\mathcal{G}}}_{t,\mathbb{S}}\|^2. \tag{14}$$

Assuming we have a complete N-dimension orthonormal basis,

$$\boldsymbol{w}_n^T \boldsymbol{w}_m = \delta_{nm} = \begin{cases} 1, & n = m \\ 0, & \text{else} \end{cases}, \tag{15}$$

where δ_{nm} is the kronecker delta and $n, m = 1, 2, \cdots N$. Given that any vector can be represented as a linear combination of the basis vectors,

$$\boldsymbol{\mathcal{G}}_{t,\mathbb{U}} = \sum_{n=1}^{N} \alpha_{tn} \boldsymbol{w}_n. \tag{16}$$

According to the property of orthonormal basis, we have,

$$\begin{aligned} \alpha_{tn} &= \boldsymbol{\mathcal{G}}_{t,\mathbb{U}}^{\top} \cdot \boldsymbol{w}_n \\ \boldsymbol{\mathcal{G}}_{t,\mathbb{U}} &= \sum_{n=1}^{N} \left(\boldsymbol{\mathcal{G}}_{t,\mathbb{U}}^{T} \cdot \boldsymbol{w}_n\right) \boldsymbol{w}_n. \end{aligned} \tag{17}$$

Our goal is to find an M-dimension representation $\boldsymbol{\mathcal{G}}_{t,\mathbb{S}}$,

$$\tilde{\boldsymbol{\mathcal{G}}}_{t,\mathbb{S}} = \sum_{n=1}^{M} \alpha_{tn} \boldsymbol{w}_n + \sum_{n=M+1}^{N} b_n \boldsymbol{w}_n. \tag{18}$$

The second term indicates bias. With Equation 17 and Equation 18, we can calculate the difference between $\boldsymbol{\mathcal{G}}_{t,\mathbb{U}}$ and $\tilde{\boldsymbol{\mathcal{G}}}_{t,\mathbb{S}}$,

$$\begin{aligned} \boldsymbol{\mathcal{G}}_{t,\mathbb{U}} - \tilde{\boldsymbol{\mathcal{G}}}_{t,\mathbb{S}} &= \sum_{n=1}^{N} \alpha_{tn} \boldsymbol{w}_n - \sum_{n=1}^{M} \alpha_{tn} \boldsymbol{w}_n - \sum_{n=M+1}^{N} b_n \boldsymbol{w}_n \\ &= \sum_{n=M+1}^{N} \alpha_{tn} \boldsymbol{w}_n - b_n \boldsymbol{w}_n. \end{aligned} \tag{19}$$

After substituting Equation 19 in Equation 14, we have

$$\mathcal{J} = \frac{1}{T}\sum_{t=1}^{T} \| \sum_{n=M+1}^{N} \alpha_{tn} \boldsymbol{w}_n - b_n \boldsymbol{w}_n\|^2. \tag{20}$$

Taking derivative *w.r.t* α_{tn} and b_n and setting to zero, we have,

$$b_n = \overline{\boldsymbol{\mathcal{G}}}_{\mathbb{U}}^{\top} \cdot \boldsymbol{w}_n, \tag{21}$$

where $\overline{\boldsymbol{\mathcal{G}}}_{\mathbb{U}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\mathcal{G}}_{t,\mathbb{U}}$. Thus, Equation 20 can be reformed as

$$\begin{aligned} \mathcal{J} =& \frac{1}{T}\sum_{t=1}^{T} \left\| \sum_{n=M+1}^{N} \left(\boldsymbol{\mathcal{G}}_{t,\mathbb{U}}^{\top} \cdot \boldsymbol{w}_n - \overline{\boldsymbol{\mathcal{G}}}_{\mathbb{U}}^{\top} \cdot \boldsymbol{w}_n\right) \boldsymbol{w}_n \right\|^2 \\ =& \frac{1}{T}\sum_{t=1}^{T} \left\| \sum_{n=M+1}^{N} \left(\left(\boldsymbol{\mathcal{G}}_{t,\mathbb{U}} - \overline{\boldsymbol{\mathcal{G}}}_{\mathbb{U}}\right)^{\top} \cdot \boldsymbol{w}_n\right) \boldsymbol{w}_n \right\|^2 \\ =& \frac{1}{T}\sum_{t=1}^{T} \Big(\sum_{n=M+1}^{N} \left(\left(\boldsymbol{\mathcal{G}}_{t,\mathbb{U}} - \overline{\boldsymbol{\mathcal{G}}}_{\mathbb{U}}\right)^{\top} \cdot \boldsymbol{w}_n\right) \boldsymbol{w}_n \Big)^{\top} \cdot \\ & \Big(\sum_{n=M+1}^{N} \left(\left(\boldsymbol{\mathcal{G}}_{t,\mathbb{U}} - \overline{\boldsymbol{\mathcal{G}}}_{\mathbb{U}}\right)^{\top} \cdot \boldsymbol{w}_n\right) \boldsymbol{w}_n \Big) \\ =& \frac{1}{T}\sum_{t=1}^{T} \sum_{n=M+1}^{N} \sum_{m=M+1}^{N} \left(\left(\boldsymbol{\mathcal{G}}_{t,\mathbb{U}} - \overline{\boldsymbol{\mathcal{G}}}_{\mathbb{U}}\right)^{\top} \cdot \boldsymbol{w}_n\right) \boldsymbol{w}_n^{\top} \cdot \\ & \boldsymbol{w}_m \left(\left(\boldsymbol{\mathcal{G}}_{t,\mathbb{U}} - \overline{\boldsymbol{\mathcal{G}}}_{\mathbb{U}}\right)^{\top} \cdot \boldsymbol{w}_m\right) \\ =& \frac{1}{T}\sum_{t=1}^{T} \sum_{n=M+1}^{N} \left(\boldsymbol{\mathcal{G}}_{t,\mathbb{U}} - \overline{\boldsymbol{\mathcal{G}}}_{\mathbb{U}}\right)^{T} \cdot \boldsymbol{w}_n \left(\boldsymbol{\mathcal{G}}_{t,\mathbb{U}} - \overline{\boldsymbol{\mathcal{G}}}_{\mathbb{U}}\right)^{\top} \cdot \boldsymbol{w}_n \\ =& \frac{1}{T}\sum_{t=1}^{T} \sum_{n=M+1}^{N} \boldsymbol{w}_n^{\top} \cdot \left(\boldsymbol{\mathcal{G}}_{t,\mathbb{U}} - \overline{\boldsymbol{\mathcal{G}}}_{\mathbb{U}}\right) \left(\boldsymbol{\mathcal{G}}_{t,\mathbb{U}} - \overline{\boldsymbol{\mathcal{G}}}_{\mathbb{U}}\right)^{\top} \cdot \boldsymbol{w}_n. \end{aligned} \tag{22}$$

Minimizing $\mathcal{J}$ is equivalent to reducing the variance of the pruned samples. Consequently, the goal outlined in Equation 4 effectively becomes maximizing the variance of coreset shown in Equation 5.

B. Comparision Methods

Random randomly selects partial data from the full dataset to form a coreset.

Entropy [8] is a metric of sample uncertainty. Samples with higher entropy are considered to have a greater impact on model optimization. The entropy is calculated with the predicted probabilities at the end of training.

Forgetting [44] counts how many times the forgetting happens during the training. The unforgettable samples can be removed with minimal performance drop.

EL2N [32] selects samples with larger gradient magnitudes which can be approximated by error vector scores. Only the first 10-epoch error vector scores are averaged to evaluate samples.

AUM [34] selects samples with the highest area under the margin, which measures the probability gap between the

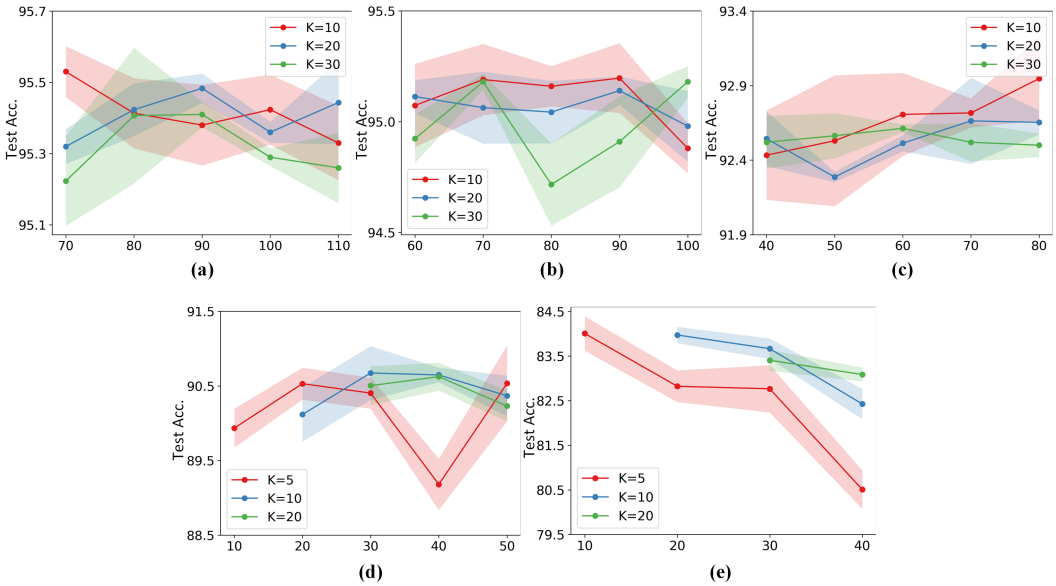

Figure 7. Parameter analysis of range T and window size K on CIFAR-10 with ResNet-18. From left to right, the corresponding pruning rates are 0.3, 0.5, 0.7, 0.8, and 0.9.

Table 6. Accuracy results on CIFAR-10 and 100 with smaller batch size. With a smaller batch size, all compared methods are enhanced under aggressive pruning, while our superiority remains consistent.

	CIFAR-10				**CIFAR-100**			
p	80%		90%		80%		90%	
batch size	128	64	128	32	128	64	128	32
Random	86.92 ±0.28	88.87 ±0.47	76.71 ±0.15	83.77 ±0.27	56.19 ±1.09	57.79 ±0.24	34.88 ±1.74	46.68 ±1.07
Entropy [8]	80.77 ±0.26	83.49 ±0.21	63.65 ±0.62	72.06 ±0.81	38.55 ±1.49	42.86 ±0.25	24.09 ±0.47	29.56 ±0.54
Forgetting [44]	61.94 ±1.33	76.18 ±3.18	38.95 ±0.28	45.87 ±1.87	38.11 ±0.55	38.42 ±1.13	19.88 ±0.69	25.82 ±0.52
EL2N [32]	59.28 ±3.62	68.64 ±3.70	23.54 ±0.69	31.89 ±1.51	14.67 ±0.94	17.31 ±0.33	5.54 ±0.08	9.10 ±0.69
AUM [34]	59.11 ±3.46	69.60 ±3.11	30.62 ±0.29	34.74 ±0.11	16.85 ±0.49	18.43 ±0.47	7.99 ±0.17	9.29 ±0.27
Moderate [47]	86.45 ±0.31	87.76 ±0.28	76.11 ±2.25	83.61 ±0.24	54.22 ±0.58	56.52 ±0.37	30.50 ±1.21	41.82 ±1.11
Dyn-Unc [15]	73.28 ±0.50	79.76 ±1.09	31.99 ±0.74	37.12 ±1.12	36.21 ±0.18	39.19 ±0.27	11.68 ±0.08	15.20 ±0.41
TDDS	**89.82 ±0.15**	**91.30 ±0.25**	**77.96 ±0.29**	**85.46 ±0.21**	**59.56 ±0.42**	**63.01 ±0.12**	**51.32 ±0.16**	**54.51 ±0.22**

target class and the next largest class across all training epochs. A larger AUM suggests higher importance.

Moderate [47] calculates sample-wise distance in feature space. Samples near the median are considered more important. Here, the features are generated by a pretrained model.

CCS [52] uses a variation of stratified sampling across importance scores to improve the coverage of coreset, which can be combined with other criteria. In our experiments, AUM [34] is used as the importance measurement in CCS [52].

Dyn-Unc [15] calculates the dynamic uncertainty defined as the variance of target class predicted probabilities during the training progress. The samples with larger uncertainties are more important than those with smaller uncertainties.

Note that, we use the same experimental hyperparameter settings to ensure equity for all the compared methods.

C. Parameter Settings

The grid search of CIFAR-10 is shown in Figure 7. The optimal (p, T, K) setting is $(0.3, 70, 10)$, $(0.5, 90, 10)$, $(0.7, 80, 10)$, $(0.8, 30, 10)$, and $(0.9, 10, 5)$. For ImageNet-

Table 7. Cross-architecture generalization performance on CIFAR-10 with ResNet-18.

	ResNet-50			VGG-16			MobileNet-v2			ShuffleNet		
p	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%
Random	94.33	93.40	90.94	92.93	91.52	88.55	91.83	91.69	89.66	90.86	89.08	85.87
Entropy [8]	94.44	92.11	85.67	93.20	90.05	85.42	91.69	86.29	89.92	90.46	87.56	82.03
Forgetting [44]	95.36	95.29	90.56	94.03	93.71	90.14	93.29	93.54	91.11	92.08	90.69	80.37
EL2N [32]	95.44	94.61	87.48	93.86	93.19	87.23	92.96	92.99	88.38	92.12	90.73	79.63
AUM [34]	95.07	95.26	91.36	94.14	93.73	88.44	93.43	93.37	90.97	92.23	91.72	79.41
Moderate [47]	93.86	92.58	90.56	92.57	90.80	87.94	91.86	90.82	89.06	90.03	89.05	84.66
Dyn-Unc [15]	94.80	94.21	87.28	92.98	92.1	86.99	92.16	92.08	89.93	90.29	88.80	80.70
CCS [52]	95.40	95.04	93.00	94.01	93.34	91.18	93.30	93.15	91.88	91.61	90.84	88.38
TDDS	**95.50**	**95.66**	**93.92**	**94.45**	**93.74**	**91.34**	**94.52**	**94.05**	**92.31**	**92.79**	**92.07**	**88.96**

Table 8. Cross-architecture generalization performance on CIFAR-100 with ResNet-18.

	ResNet-50			VGG-16			MobileNet-v2			ShuffleNet		
p	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%
Random	72.09	68.27	61.75	71.75	67.57	61.03	70.27	67.76	63.02	68.31	65.17	58.29
Entropy [8]	73.09	63.12	47.61	69.52	61.16	48.42	67.91	61.69	51.74	64.12	56.28	44.68
Forgetting [44]	78.17	70.60	48.74	73.29	66.01	47.85	72.37	68.05	54.06	66.94	60.64	40.65
EL2N [32]	76.27	65.83	23.35	72.42	63.07	36.47	71.96	63.81	42.47	69.21	56.82	29.22
AUM [34]	77.38	64.2	32.36	73.60	62.01	30.88	72.29	64.33	36.35	66.98	54.31	29.24
Moderate [47]	72.67	68.75	57.61	70.1	65.56	57.80	70.01	67.03	60.78	66.53	62.53	50.33
Dyn-Unc [15]	73.00	63.7	46.26	68.48	61.27	47.24	68.01	62.04	49.40	64.57	56.14	42.35
CCS [52]	76.96	72.43	64.74	74.02	70.14	**64.40**	73.04	70.63	66.31	69.80	66.71	**61.31**
TDDS	**79.53**	**76.24**	**66.56**	**74.23**	**70.66**	64.08	**74.23**	**71.14**	**66.36**	**70.14**	**67.14**	61.07

1K with ResNet-34, we set $(0.3, 20, 10)$, $(0.5, 20, 10)$, and $(0.7, 30, 20)$. For ImageNet-1K with Swin-T, we set $(0.25, 200, 10)$, $(0.3, 180, 10)$, $(0.4, 150, 10)$, and $(0.5, 100, 10)$.

D. Small Batch Size Boosts Aggressive Pruning

In the experiments, we reveal that smaller batch size boosts coreset training, especially under aggressive pruning rates. This phenomenon is attributed to so-called *Generalization Gap* [18], which suggests that when the available data is extremely scare, smaller batch size can prevent overfitting by allowing more random explorations in the optimization space before converging to an optimal minimum. As reported in Table 6, smaller batch size improves the accuracy of high pruning rates for all the compared methods. Note that, regardless of the batch size, our method consistently demonstrates a significant advantage.

E. Generalization Across Architectures

We conduct cross-architecture experiments to examine whether coresets perform well when being selected on one architecture and then tested on other architectures. Four representative architectures including ResNet-50, VGG-16, MobileNet-v2, and ShuffleNet are used to assess the cross-architecture generlization. Table 7 lists the results on CIFAR-10, while Table 8 reports the results on CIFAR-100. We can see the coresets constructed by the proposed TDDS achieves stably good testing results, regardless of which model architecture is used to test. Experiments on CIFAR-100 are reported in Supplementary.

F. Robustness to Complex Realistic Scenarios

We also investigate the robustness of corsets in complex and realistic scenarios, including image corruption and label noise. Following the settings stated in [47], we consider five types of realistic noise, namely Gaussian noise,

Figure 8. Illustration of the different types of noise used for image corruption. Here we consider Gaussian noise, random occlusion, resolution, fog, and motion blur.

Table 9. Robustness to image corruption on CIFAR-100 with ResNet-18. 20% training images are corrupted. The model trained with the full dataset achieves 75.30% accuracy.

	Image Corruption				
p	30%	50%	70%	80%	90%
Random	71.34 ±0.29	67.17 ±0.43	58.56 ±0.87	51.85 ±0.48	37.30 ±0.60
Entropy [8]	68.83 ±0.17	62.19 ±0.26	49.25 ±0.20	41.26 ±0.24	28.03 ±0.44
Forgetting [44]	73.77 ±0.07	65.38 ±1.87	47.41 ±1.11	36.07 ±1.44	22.02 ±0.40
EL2N [32]	70.58 ±0.30	48.17 ±3.26	14.47 ±0.73	12.21 ±0.35	8.62 ±0.29
AUM [34]	71.14 ±0.63	44.06 ±1.80	13.89 ±0.43	8.06 ±0.15	4.93 ±0.15
Moderate [47]	72.20 ±0.11	67.52 ±0.18	59.84 ±0.17	52.89 ±0.12	36.16 ±1.23
Dyn-Unc [15]	67.74 ±0.38	59.40 ±0.15	45.39 ±0.37	34.11 ±0.47	13.55 ±0.29
CCS [52]	70.14 ±0.14	64.77 ±0.31	54.95 ±1.08	44.95 ±0.69	30.16 ±1.13
TDDS	**75.40 ±0.12**	**72.49 ±0.22**	**65.84 ±0.30**	**60.85 ±0.07**	**49.35 ±0.12**

random occlusion, resolution, fog, and motion blur (shown in Figure 8). Here, the ratio for each type of corruption is 4%, resulting in a total 20% of training images being corrupted. Besides, we also consider label noise by replacing the original label with labels from other classes. The mislabel ratio is also set to 20%. The results reported in Table 9 and Table 10 verify the robustness of our proposed TDDS in complex and realistic scenarios.

Table 10. Robustness to label noise on CIFAR-100 with ResNet-18. 20% training samples are mislabeled. The model trained with the full dataset achieves 65.48% accuracy.

	Lable Noise				
p	30%	50%	70%	80%	90%
Random	62.17 ±0.42	55.3 ±0.25	40.8 ±0.49	34.41 ±0.69	22.74 ±0.27
Entropy [8]	60.01 ±0.59	54.27 ±0.90	42.75 ±0.80	35.18 ±0.28	24.34 ±1.01
Forgetting [44]	58.75 ±0.28	47.90 ±0.79	29.34 ±0.51	21.38 ±0.34	13.31 ±0.35
EL2N [32]	63.76 ±0.07	50.39 ±0.89	20.89 ±1.79	10.20 ±0.95	5.97 ±0.19
AUM [34]	50.49 ±0.81	22.86 ±0.11	5.79 ±0.36	2.31 ±0.39	1.25 ±0.04
Moderate [47]	61.58 ±0.29	57.23 ±0.05	49.28 ±0.25	43.25 ±1.02	32.07 ±0.25
Dyn-Unc [15]	52.99 ±0.34	38.83 ±0.17	19.17 ±0.15	3.41 ±0.04	1.64 ±0.08
CCS [52]	53.38 ±0.86	40.59 ±0.21	25.30 ±0.17	20.49 ±0.43	15.49 ±0.61
TDDS	**65.15 ±0.06**	**62.72 ±0.37**	**54.97 ±0.20**	**50.14 ±0.20**	**39.32 ±0.19**