Reinforcement Learning-Guided Data Selection via Redundancy Assessment

Supplementary Material

1. Proof of Proposition 1

Proof 1 *For simplicity, we approximate the fully connected layer as a linear model, i.e.,* $\hat{\boldsymbol{y}}_i = \boldsymbol{w}\boldsymbol{x}_i + \boldsymbol{b}$*.*

$$\begin{aligned} \left\|\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{y}}_j\right\| &= \|\boldsymbol{w}(\boldsymbol{x}_i - \boldsymbol{x}_j)\| && (12) \\ &\leq \|\boldsymbol{w}\| \, \|\boldsymbol{x}_i - \boldsymbol{x}_j\| && (13) \\ &\leq \epsilon \, \|\boldsymbol{w}\| && (14) \\ & && (15) \end{aligned}$$

In the above, the Inequality (13) *follows from Hölder's inequality. For a given model with corresponding* $\boldsymbol{w}$*,* $\left\|\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{y}}_j\right\| \leq \mathcal{O}(\epsilon)$ *holds.*

2. Proof of Proposition 2

Proof 2 *Let* $\boldsymbol{x}_i$ *and* $\boldsymbol{x}_j$ *are two samples, and* $\boldsymbol{x}_i$ *is* ϵ*-covered by* $\boldsymbol{x}_j$*. The cross-entropy loss of* $\boldsymbol{x}_i$ *is given as*

$$loss(\boldsymbol{x}_i) = L(\boldsymbol{w}\boldsymbol{x}_i + b) = -\sum_{c=1}^{K} y_{ic} \log(\boldsymbol{w}\boldsymbol{x}_i + b)_c. \quad (16)$$

Let $\hat{y}_{ic} = (\boldsymbol{w}\boldsymbol{x}_i + b)_c$*, then* $loss(\boldsymbol{x}_i) - loss(x_j)$ *can be computed as*

$$\begin{aligned} loss(\boldsymbol{x}_i) - loss(x_j) &= L(\boldsymbol{w}\boldsymbol{x}_i + b) - L(\boldsymbol{w}\boldsymbol{x}_j + b) \\ &= -\sum_{c=1}^{K} y_{ic}(\log(\hat{y}_{ic}) - \log(\hat{y}_{jc})) \\ &= -\sum_{c=1}^{K} y_{ic} \log \frac{\hat{y}_{ic}}{\hat{y}_{jc}} \\ &= -log \frac{\hat{y}_{ik}}{\hat{y}_{jk}}. \end{aligned} \quad (17)$$

According to Definition 1, $\hat{y}_{ik}/\hat{y}_{jk}$ *can be re-written as follows,*

$$\frac{\hat{y}_{ik}}{\hat{y}_{jk}} = \frac{(\boldsymbol{w}\boldsymbol{x}_i + b)_k}{(\boldsymbol{w}\boldsymbol{x}_j + b)_k} \leq \frac{(\boldsymbol{w}(\boldsymbol{x}_i + \epsilon) + b)_k}{(\boldsymbol{w}\boldsymbol{x}_j + b)_k}. \quad (18)$$

Substituting Eq. (18) *into Eq.* (17)*, the gap in losses can be derived as follows.*

$$\lim_{\epsilon \to 0} \frac{\hat{y}_{ik}}{\hat{y}_{jk}} \leq \lim_{\epsilon \to 0} \frac{(\boldsymbol{w}(\boldsymbol{x}_i + \epsilon) + b)_k}{(\boldsymbol{w}\boldsymbol{x}_j + b)_k} = \lim_{\epsilon \to 0} \frac{(\boldsymbol{w}\boldsymbol{x}_j + b)_k + \epsilon \boldsymbol{w}_k}{(\boldsymbol{w}\boldsymbol{x}_j + b)_k} = 1. \quad (19)$$

Similarly,

$$\lim_{\epsilon \to 0} \frac{\hat{y}_{ik}}{\hat{y}_{jk}} \geq \lim_{\epsilon \to 0} \frac{(\boldsymbol{w}(\boldsymbol{x}_i - \epsilon) + b)_k}{(\boldsymbol{w}\boldsymbol{x}_j + b)_k} = 1. \quad (20)$$

Thus, according to Eq. (19) *and Eq.* (20)*,* $\lim_{\epsilon \to 0} \frac{\hat{y}_{ik}}{\hat{y}_{jk}} = 1$*, and we have*

$$\lim_{\epsilon \to 0} loss(\boldsymbol{x}_i) - loss(\boldsymbol{x}_j) = 0. \quad (21)$$

3. Proof of Lemma 1

Proof 3 *Without loss of generality, the gradient of loss for* $\boldsymbol{x}$ *w.r.t. the weight* W_h *of the* h*-th intermediate layer is given as*

$$g_{W_h} = (\frac{\partial}{\partial W_h^T} L(f_\theta(\boldsymbol{x}), y))^T. \quad (22)$$

According to the chain rule, the gradient of the i*-th element of the weight* W_h *can be written as*

$$\begin{aligned} g_{W_{hi}^T} &= \frac{\partial L(f_\theta(\boldsymbol{x}), y)}{\partial \tilde{\boldsymbol{x}}_{h+1}^T} \frac{\partial \tilde{\boldsymbol{x}}_{h+1}^T}{\partial W_{hi}^T} \\ &= \frac{\partial L(f_\theta(\boldsymbol{x}), y)}{\partial \tilde{\boldsymbol{x}}_{h+1}^T} \tilde{\boldsymbol{x}}_h^T, \end{aligned} \quad (23)$$

where $\tilde{\boldsymbol{x}}_{h+1}$ *is the output obtained from the* h*-th intermediate layer, i.e.,* $\tilde{\boldsymbol{x}}_{h+1} = W_h^T \tilde{\boldsymbol{x}}_h + b$*, where* $\tilde{\boldsymbol{x}}_h$ *can be understood as the feature map from the intermediate layer. Then, we have*

$$\begin{aligned} g_{W_h} &= [g_{W_{h1}^T}, g_{W_{h2}^T}, ..., g_{W_{hD}^T}]^T = [\frac{\partial L(f_\theta(\boldsymbol{x}), y)}{\partial \tilde{\boldsymbol{x}}_{h+1}^T} \tilde{\boldsymbol{x}}_h^T]^T \\ &= \tilde{\boldsymbol{x}}_h \frac{\partial L(f_\theta(\boldsymbol{x}), y)}{\partial \tilde{\boldsymbol{x}}_{h+1}^T} = \tilde{\boldsymbol{x}}_h g_{\tilde{\boldsymbol{x}}}^T \end{aligned} \quad (24)$$

According to Definition 1, we have $\|\tilde{\boldsymbol{x}}_{i,h} - \tilde{\boldsymbol{x}}_{j,h}\| \leq \epsilon$*. Let* $\Delta\boldsymbol{x} = \tilde{\boldsymbol{x}}_{i,h} - \tilde{\boldsymbol{x}}_{j,h}$*, i.e.,* $\Delta\boldsymbol{x} \in \mathbb{R}^D$ *and* $\|\Delta\boldsymbol{x}\| \leq \epsilon$*. Meanwhile, we can derive the following equation,*

$$\tilde{\boldsymbol{x}}_{i,h+1} = W_h^T \tilde{\boldsymbol{x}}_{i,h} + b = \tilde{\boldsymbol{x}}_{j,h+1} + W_h^T \Delta\boldsymbol{x} \quad (25)$$

According to Eq. (24) *and Eq.* (25)*, for* $\boldsymbol{x}_i$*, the gradient of loss w.r.t. the weight* W_h *is given as*

$$g_{W_h}^{\boldsymbol{x}_i} = (\tilde{\boldsymbol{x}}_{j,h} + \Delta\boldsymbol{x}) g_{\tilde{\boldsymbol{x}}+\Delta\boldsymbol{x}}^T. \quad (26)$$

We further use the first-order Taylor expansion to decompose the gradient $g_{\tilde{\boldsymbol{x}}+\Delta\boldsymbol{x}}$

$$g_{\tilde{\boldsymbol{x}}+\Delta\boldsymbol{x}} = g_{\tilde{\boldsymbol{x}}} + H_{\tilde{\boldsymbol{x}}} \Delta\boldsymbol{x} + \mathcal{R}_2(\Delta\boldsymbol{x}), \quad (27)$$

where $\mathcal{R}_2(\Delta\boldsymbol{x})$ *denotes the terms no less than the second order. Substituting Eq.* (27) *back to Eq.* (26)*, the gradient for parameter update is*

$$g_{W_h}^{\boldsymbol{x}_i} = (\tilde{\boldsymbol{x}}_{j,h} + \Delta\boldsymbol{x})(g_{\tilde{\boldsymbol{x}}} + H_{\tilde{\boldsymbol{x}}} \Delta\boldsymbol{x} + \mathcal{R}_2(\Delta\boldsymbol{x}))^T \quad (28)$$

In this way,

$$\begin{aligned}\Delta g_{W_h} &= g_{W_h}^{\boldsymbol{x}_i} - g_{W_h}^{\boldsymbol{x}_j} \\ &= \tilde{\boldsymbol{x}}_{j,h}(H_{\tilde{\boldsymbol{x}}}\Delta\boldsymbol{x})^T + \Delta\boldsymbol{x}(g_{\tilde{\boldsymbol{x}}} + H_{\tilde{\boldsymbol{x}}}\Delta\boldsymbol{x})^T \\ &\quad + (\tilde{\boldsymbol{x}}_{j,h} + \Delta\boldsymbol{x})\mathcal{R}_2(\Delta\boldsymbol{x})^T \\ &\approx \tilde{\boldsymbol{x}}_{j,h}(H_{\tilde{\boldsymbol{x}}}\Delta\boldsymbol{x})^T + \Delta\boldsymbol{x}(g_{\tilde{\boldsymbol{x}}} + H_{\tilde{\boldsymbol{x}}}\Delta\boldsymbol{x})^T \\ &= (\tilde{\boldsymbol{x}}_{j,h} + \Delta\boldsymbol{x})(H_{\tilde{\boldsymbol{x}}}\Delta\boldsymbol{x})^T + \Delta\boldsymbol{x}g_{\tilde{\boldsymbol{x}}}^T \\ &= \tilde{\boldsymbol{x}}_{i,h}\Delta\boldsymbol{x}^T H_{\tilde{\boldsymbol{x}}}^T + \Delta\boldsymbol{x}g_{\tilde{\boldsymbol{x}}}^T\end{aligned} \tag{29}$$

4. Proof of Proposition 3

Proof 4 *According to Eq.* (29)*, we have*

$$\begin{aligned}\|\Delta g_{W_h}\| &= \left\|\tilde{\boldsymbol{x}}_{i,h}\Delta\boldsymbol{x}^T H_{\tilde{\boldsymbol{x}}}^T + \Delta\boldsymbol{x}g_{\tilde{\boldsymbol{x}}}^T\right\| && (30)\\ &\le \left\|\tilde{\boldsymbol{x}}_{i,h}\Delta\boldsymbol{x}^T H_{\tilde{\boldsymbol{x}}}^T\right\| + \left\|\Delta\boldsymbol{x}g_{\tilde{\boldsymbol{x}}}^T\right\| && (31)\\ &\le \|\Delta\boldsymbol{x}\|\left(\|\tilde{\boldsymbol{x}}_{i,h}\|\,\|H_{\tilde{\boldsymbol{x}}}\| + \|g_{\tilde{\boldsymbol{x}}}\|\right). && (32)\end{aligned}$$

Given a model f_θ, since $\|\Delta\boldsymbol{x}\| \le \epsilon$, the change of the gradient can be formulated as follows,

$$\lim_{\epsilon\to 0}\|\Delta g_{W_h}\| = 0. \tag{33}$$

5. Proof of Proposition 4

Proof 5 *According to Lemma 2, $\forall \boldsymbol{x'} \in -\hat{\mathcal{D}}$, we can find a $\boldsymbol{x} \in \mathcal{D}$ such that $\boldsymbol{x'}$ is ϵ-covered by $\boldsymbol{x}$. The gap in the parameter changes between $\boldsymbol{x'}$ and $\boldsymbol{x}$ can be estimated as follows*

$$\begin{aligned}\Delta\mathcal{I}_{\mathit{up,params}} &= \mathcal{I}_{\mathit{up,params}}(\boldsymbol{x}) - \mathcal{I}_{\mathit{up,params}}(\boldsymbol{x'}) && (34)\\ &= -(H_{\hat{\theta}}^{-1}g_\theta^{\boldsymbol{x}} - H_{\hat{\theta}}^{-1}g_\theta^{\boldsymbol{x'}}) && (35)\\ &= H_{\hat{\theta}}^{-1}(g_\theta^{\boldsymbol{x'}} - g_\theta^{\boldsymbol{x}}). && (36)\end{aligned}$$

Based on Eq. (36)*, the significance of this gap can be estimated by*

$$\begin{aligned}\|\Delta\mathcal{I}_{\mathit{up,params}}\| &= \left\|H_{\hat{\theta}}^{-1}(g_\theta^{\boldsymbol{x'}} - g_\theta^{\boldsymbol{x}})\right\| && (37)\\ &\le \left\|H_{\hat{\theta}}^{-1}\right\|\left\|g_\theta^{\boldsymbol{x'}} - g_\theta^{\boldsymbol{x}}\right\| && (38)\\ &= \left\|H_{\hat{\theta}}^{-1}\right\|\|\Delta g_\theta\|. && (39)\end{aligned}$$

According to Propostion 3 and the definition of $H_{\hat{\theta}}$, we can derive that

$$\lim_{\epsilon\to 0}\|\Delta\mathcal{I}_{\mathit{up,params}}\| \le 0. \tag{40}$$

Finally, $\lim_{\epsilon\to 0}\|\Delta\mathcal{I}_{\mathit{up,params}}\| = 0$.

6. The General Workflow of Our Proposed Method

To better understand the workflow of our proposed method, we summarize the detailed algorithm in Algorithm 1. The total epoch is 200, and no warm-up schedule is used. The Adam optimizer is used with a weight decay of $1e-4$ and an initial learning rate of $3e-4$.

Algorithm 1 The general workflow.

```
Require: a dataset $\mathcal{D}$ of images and label pairs $(\boldsymbol{x}, y)$, the expected selection ratio $s_r$, total epoch number $T$, classification model $f$, batch size $B$
Ensure: Importance scores $\mathcal{IS}$ for sample selection
for $t = 0{:}T-1$ do
    Calculate the distance matrix $\boldsymbol{D}_k$ for each class in current feature space obtained by $f_t$;
    Calculate the degree of $\epsilon$-cover for each sample according to Eq. (6);
    Sample a mini-batch $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^B$ from $\mathcal{D}$;
    Determine the corresponding importance scores $\mathcal{IS}$ for each sample $\boldsymbol{x}_i$ using $\theta_{at}$;
    Calculate the reward signal $r_1$ according to Eq. (5);
    Calculate the reward signal $r_2$ according to Eq. (7);
    Calculate the general reward value $r = r_1 + r_2$;
    Update $\theta_a$ and $\theta_c$ according to Eq. (8) and Eq. (9), respectively;
    Update model $f_t$ on the current mini-batch using vanilla cross-entropy loss;
end for
```

7. Choice of A2C Network

Table 6. The A2C network architecture details.

	index	Layer	Dimension
Actor	1	Linear	(512, 512)
	2	Linear	(512, 256)
	3	Linear	(256, 1)
Critic	1	Linear	(512, 512)
	2	Linear	(512, 256)
	3	Linear	(256, 1)

In Table 6, we provide the details of the A2C network. Both the actor and critic in A2C consist of three linear layers. Therefore, forwarding and updating the A2C network can be very efficient.

8. Complexity Analysis

According to the algorithm pipeline in Algorithm 1, the main computational costs can be divided into three components: 1) distance metric calculation, 2) degree of ϵ-cover calculation, and 3) the RL network forward passes and updates. The complexities of the first two steps are $O\left(N_k^2 d\right)$ and $O\left(N_k^2\right)$, respectively, where N_k is the number of samples in class k and d is the feature dimension (e.g., 512 for ResNet-18). Since a precise theoretical analysis of the computational complexity of the A2C algorithm is challenging due to its inherent nature, we provide some insights into its computational cost. The structure of A2C algorithm is

Table 7. The reductions in training costs with ImageNet-1k selected datasets compared to vanilla training. The reported results are the average ± std across five independent runs.

Selection Ratio	60%	70%	80%	90%	100%
ResNet-50	-35.45±0.3	-26.15±0.3	-18.23±0.3	-10.16±0.3	-0

very simple (in Table 6): both the policy and critic networks consist of only a few linear layers, making it more computationally efficient compared to previous methods. Thus, our method can achieve competitive training efficiency compared to other baselines in Figure 3. As a result, our proposed method achieves a better trade-off between computational costs and performance, making it the best-performing method with competitive computational costs.

9. Implementation Details

Our RL module follows the design and parameter settings from [45], without introducing any additional hyperparameters. For experiments on CIFAR-10 and CIFAR-100, following [72, 77, 88], we train ResNet-50 models for 200 epochs with a batch size of 256 and a 0.1 learning rate with a cosine annealing learning rate decay strategy, an SGD optimizer with a momentum of 0.9, and weight decay of 5e-4. Data augmentation of random crop and random horizontal flip is added. For experiments on Tiny-ImageNet, following [72], we adopt a batch size of 256, an SGD optimizer with a momentum of 0.9, weight decay of 1e-4, and an initial learning rate of 0.1. The learning rate is divided by 10 after the 30th and the 60th epoch. The total number of epochs is 90. In each experiment, we perform three independent random trials. For experiments on ImageNet-1k, following [61, 72, 88], the VISSL library [14] is exploited. We adopt a base learning rate of 0.01, a batch size of 256, an SGD optimizer with a momentum of 0.9, and a weight decay of 1e-3. Because of the huge computational cost, the experiment in each case is performed once. Note that the methods Glister and CG-Score incur high computational and memory costs due to the iterative solving of the bi-level optimization problem [28] and the calculation of large Gram matrix inversions [49] for subset selection, respectively. Thus, they are not compared on ImageNet-1k.

10. Data Selection Improves Training Efficiency

Data selection substantially enhances the efficiency of model training efficiency for subsequent tasks. Specifically, once the selected datasets are obtained, only a subset needs to be stored as a replacement for the full dataset, leading to savings in memory costs. Moreover, the reduction in training data volume translates to diminished training costs. In Table 7, we show the reductions in training deep models on ImageNet-1k. It can be seen that the reductions in training costs are proportional to the number of selected data. Meanwhile, it is important to note that with high data selection ratios, the generalization performance is nearly lossless or even enhanced, as shown in Section 5. Therefore, the selected datasets offer practical benefits.

11. Experiment Results on CIFAR-10

Due to space constraints, we present experiment results on CIFAR-10. As shown in Table 8, our method can achieve superior results. Notably, with relatively high selection ratios (e.g., $\geq$ 90%), our method exhibits performance surpassing that of the full dataset. Moreover, when the data selection ratio is very low, e.g., 20% and 30%, our method substantially improves over existing baselines. Therefore, through extensive experiment results alongside those obtained from other benchmark datasets as delineated in Section 5, we demonstrate the promising efficacy of our proposed method.

12. More Experimental Results on Vision Transformer

To further demonstrate the superiority of our proposed method, we employ Vision Transformer [10] to train with selected datasets. Following [72], the implementation is based on the public Github repository [1], where ViT small is used. We systematically evaluate our approach across different selection ratios using the CIFAR-10 dataset. Experimental results in Table 9 demonstrate the notable performance gains achieved by our method with ViT compared to other baselines.

13. Limitation and Future Work

First, the proposed method focuses on optimizing sample-wise importance scores for selection. This may limit its applicability to extremely large-scale datasets, which is also a primary challenge for all score-based data selection methods. Future work should emphasize extending existing data selection approaches to such large-scale datasets. Second, the proposed method drops samples that are more likely to be ϵ-covered by others. This is based on an important assumption that the amount of noise in datasets is limited. However, if the amount of noise is dominant, the usefulness of our method is not guaranteed, as noisy samples are more likely to be outliers and less likely to be covered by normal samples. Therefore, it is very necessary to develop a variant that adapts to high-noise conditions with theoretical guarantees in future work. Lastly, this paper evaluates the performance of the proposed method on classification tasks. Future work should, therefore, extend its application

[1] https://github.com/kentaroy47/vision-transformers-cifar10

Table 8. Test accuracy (%) on CIFAR-10 with ResNet-50.

Method / Selection ratio	20%	30%	40%	60%	70%	80%	90%	100%
Random	84.12±1.53	90.34±0.39	92.71±0.38	94.43±0.37	95.02±0.29	95.55±0.14	95.89±0.11	96.12±0.12
EL2N	70.32±0.74	87.48±0.80	89.23±0.61	94.43±0.27	95.17±0.27	95.55±0.18	96.01±0.20	96.12±0.12
MoSo	83.33±0.47	89.17±0.14	92.47±0.14	94.69±0.20	95.50±0.00	95.93±0.01	96.26±0.02	96.12±0.12
GraNd	79.23±0.84	87.88±0.90	92.17±0.73	94.14±0.47	95.19±0.12	95.35±0.38	95.96±0.05	96.12±0.12
Glister	79.23±0.55	87.88±0.49	92.17±0.34	95.03±0.13	95.61±0.05	95.98±0.17	96.34±0.02	96.12±0.12
Herding	78.42±0.78	87.77±0.66	89.40±0.54	89.12±0.35	92.11±0.13	93.92±0.36	95.50±0.13	96.12±0.12
CG-Score	80.50±1.23	89.35±0.87	92.73±0.37	95.19±0.23	95.87±0.17	**95.99**±0.16	96.16±0.15	96.12±0.12
Forgetting	67.58±1.05	88.12±1.40	**93.61**±0.87	95.17±0.25	95.85±0.20	95.46±0.27	95.85±0.37	96.12±0.12
Moderate-DS	81.75±0.38	90.94±0.27	92.79±0.31	94.69±0.24	95.26±0.30	95.73±0.19	96.17±0.15	96.12±0.12
Self-sup. prototypes	84.60±1.01	90.07±1.14	92.64±0.93	94.42±0.72	94.98±0.61	95.87±0.53	95.95±0.44	96.12±0.12
Ours	**88.47**±0.79	**91.27**±0.28	93.02±0.32	**95.39**±0.29	**96.08**±0.19	95.84±0.18	**96.38**±0.10	96.12±0.12

Table 9. The test accuracy (%) on CIFAR-10 with ViT-small. ResNet → ViT.

Method/Selection Ratio	20%	30%	40%	60%	70%	80%	90%	100%
Random	67.98±0.29	71.99±0.12	74.69±0.26	78.98±0.28	80.30±0.36	81.33±0.10	82.63±0.18	84.00±0.32
EL2N	68.34±0.18	72.03±0.52	74.85±0.24	79.35±0.09	80.73±0.08	81.62±0.08	82.90±0.09	84.00±0.32
MoSo	67.15±0.19	71.80±0.18	74.88±0.19	79.45±0.11	80.27±0.23	81.82±0.15	82.92±0.34	84.00±0.32
GraNd	67.74±0.25	71.99±0.32	75.24±0.12	79.22±0.06	80.59±0.19	81.53±0.18	82.72±0.08	84.00±0.32
Glister	61.16±0.07	67.36±0.05	71.77±0.14	78.33±0.01	79.84±0.27	81.33±0.05	82.65±0.09	84.00±0.32
Herding	64.97±0.66	70.18±0.13	73.27±0.19	76.08±0.19	78.53±0.45	80.31±0.01	82.08±0.02	84.00±0.32
CG-Score	57.21±0.11	66.38±0.09	71.89±0.07	79.09±0.29	80.96±0.05	82.02±0.23	82.91±0.05	84.00±0.32
Forgetting	49.50±0.14	58.83±0.53	67.43±0.18	77.87±0.32	80.86±0.08	81.90±0.31	82.69±0.20	84.00±0.32
Moderate-DS	68.69±0.40	72.36±0.11	75.44±0.53	79.54±0.19	81.28±0.13	81.98±0.16	82.61±0.27	84.00±0.32
Self-sup. prototypes	67.97±0.17	72.08±0.32	75.38±0.05	79.24±0.16	80.34±0.21	81.66±0.25	82.86±0.19	84.00±0.32
Ours	**69.21±0.21**	**73.28±0.21**	**75.48±0.11**	**80.52**±0.21	**81.73**±0.19	**82.15**±0.26	**82.96**±0.06	84.00±0.32

to a broader range of tasks, such as fine-grained classification, semantic segmentation, and object detection, and further explore its application on multimodal data.