CS772 Project Report

Learning What Matters: Prioritized Data Selection for Improving Model Generalization

Edula Vinay Kumar Reddy
Department of CSE
IIT Kanpur
241110024
vinaykr24@iitk.ac.in

Telugu Sudhakar
Department of CSE
IIT Kanpur
241110077
tsudhakar24@iitk.ac.in

Pokala Dattatreya
Department of CSE
IIT Kanpur
241110048
pokala24@iitk.ac.in

Source code: https://github.com/Sudhakar6324/CS772_PROJECT/

1 Motivation

Deep learning models require extensive training on large-scale datasets, often leading to prolonged training times.The proposed method in the seed paper-**Prioritized Training on Points that are learnable, Worth Learning, and Not Yet Learnt** (6), optimizes data selection by prioritizing data points that most effectively reduce generalization loss. Unlike traditional data selection approaches that focus on hard examples, RHO-LOSS is based on a probabilistic formulation that approximates the reduction in holdout loss a data point would provide if included in training.However, several key limitations remain underexplored, such as the use of a fixed batch size across all training stages, redundancy among already selected samples, the absence of uncertainty modeling, and limited adaptivity throughout the training process. In this project, we implemented and evaluated enhancements to the RHO-LOSS framework. Each of these methods is compared against the original RHO-LOSS baseline on the CIFAR-10 dataset to assess their impact. Notably, some of our proposed methods outperformed the RHO framework in terms of data usage and generalization performance.

2 Literature review

Loshchilov and Hutter (1) proposed an online batch selection method that accelerates neural network training by prioritizing harder examples. Each data point is ranked based on its most recent loss value, and the probability of selecting a sample decreases exponentially with its rank—meaning higher-loss samples are more likely to be chosen. This rank-based selection helps the model focus on informative samples early in training, while the bias is gradually reduced to maintain generalization. Bengio et al. (2) introduced Curriculum Learning, a method where models are trained by starting with easier examples and gradually moving to more complex ones. This approach reflects the way humans typically learn—beginning with simple tasks and gradually moving to more complex ones. Adopting this learning process allows models to train more efficiently, often leading to faster convergence and enhanced overall performance.Bayesian Active Learning by Disagreement (BALD), proposed by Houlsby et al. (3), introduces an acquisition function that selects data points by maximizing the mutual information between model predictions and model parameters. This approach effectively captures *epistemic uncertainty* by targeting samples where the model exhibits high uncertainty due to variability in its parameters.

In contrast, to avoid selecting similar samples during active learning, Hong et al. (4) introduced a method called Diversified Batch Selection for Training Acceleration (DivBS), which addresses a common limitation in active learning—redundancy in sample selection. DivBS mitigates this by using a greedy orthogonalization strategy that selects samples with maximally diverse gradient directions. By analyzing gradient-feature interactions and selecting examples whose gradients are as orthogonal as possible, DivBS ensures broader coverage of the parameter space, thereby enhancing learning efficiency even with smaller batches. In some active learning methods small models are used to evalute the usefulness of the point to train on target model one such method is Selection via Proxy (SVP) proposed by Coleman et al. (5),a data selection framework that uses a lightweight proxy model to estimate the utility of training samples. Instead of evaluating informativeness using the full model, SVP trains a smaller proxy model and ranks data points based on uncertainty measures like entropy or margin. The top-scoring samples are selected for training the target model, significantly reducing computation. This method offers an efficient alternative to traditional active learning and sample prioritization techniques without sacrificing accuracy.

2.1 Original RHO-LOSS Approach

RHO-LOSS is introduced by Mindermann et al. (6). In the RHO-LOSS method, two models are used: the *irreducible loss (IL) model* and the *target model*. The training data is split into a hold-out set and a training set. The IL model is trained on the hold-out set, and it is used to compute the irreducible loss of the remaining training points. A random batch of N_b samples is drawn from the training data, and from this batch, the top n_b points are selected based on their RHO-LOSS values. RHO-LOSS of a point can be interpreted as the *reducible loss* of a point.

$$\rho\text{-loss}(x_i) = \mathcal{L}[y_i \mid x_i; D_t] - \mathcal{L}[y_i \mid x_i; D_{ho}]$$

Where:

- $\mathcal{L}[y \mid x; D_t]$ is the loss of sample (x, y) evaluated on the *target model* trained using the current labeled dataset D_t.
- $\mathcal{L}[y \mid x; D_{ho}]$ is the irreducible loss of the same sample, computed on a reference or *ideal (IL) model* trained with a small hold-out dataset D_{ho}.
- $\rho\text{-loss}(x_i)$ is the reducible loss of the data point x_i.

The selection function is defined as: $(x_1, x_2, \ldots, x_k) = \arg\max_{(x_1,y_1),\ldots,(x_k,y_k)\in B_t} (\rho\text{-loss}(x_i))$.

For noisy points, both the training loss and IL loss are high, so the reducible loss is low—these points are not selected. For redundant points, the training loss is low, and since the IL loss is also low, the reducible loss remains low—these are also not selected. In the end, RHO-LOSS prioritizes only those samples that are *learnable*, *worth learning*, and *not yet learned*.

3 Novelty of our work

In this work, we proposed several enhancements over the original RHO-LOSS framework to improve sample selection efficiency and training convergence. Our methods addressed key limitations such as redundancy in selected samples, static batch sizes, lack of uncertainty modeling, class imbalance, and training instability in early epochs. To improve diversity, we incorporated gradient-based and similarity-based penalties. We introduced adaptive strategies that adjust selection intensity based on training dynamics. Additionally, we experimented with entropy and Bayesian-based uncertainty measures, and proposed class-aware sampling to handle imbalanced learning progress. A curriculum warm-up strategy was also tested but found to hinder convergence on clean datasets. In section 5, we present these enhancements in detail and compare them against the baseline to assess their effectiveness in improving convergence speed and data efficiency—some of which outperformed the traditional RHO-LOSS, while others were found to offer limited or no improvement in this setting.

4 Tools and Software Used

This project was implemented using Python 3.10 and the PyTorch deep learning framework for all model training and experimentation. We conducted our experiments on Google Colab, utilizing the free-tier NVIDIA T4 GPU (16GB VRAM) for accelerated training. For datasets, we used the `torchvision.datasets` module to load the CIFAR-10 dataset, along with standard data augmentation and normalization techniques.Visualization of accuracy trends was done using `Matplotlib`.

5 Proposed Enhancements and Experimental results

5.1 Replicating Results of Seed Paper

Before starting our experiments, we first replicated the results of the RHO-Loss paper using the exact same model setup. Specifically, we used the ResNet-18 architecture adapted for small images, as provided in the official codebase by the authors (7). Both the target model and the irreducible-loss (IL) model use the same ResNet-18 architecture. The RHO-Loss method works by first reserving half of the CIFAR-10 training set as a hold-out split. This hold-out is used to train the IL model. The IL model is trained using the AdamW optimizer, and training continues until it reaches approximately 60% accuracy on its own validation set. Once trained, this model is fixed. Then, for every sample in the other half of the training set, the model's cross-entropy loss is calculated and stored as the irreducible loss (IL_i).

During training of target model, batch of 320 samples is drawn at each step. For each of these, the current training loss L_i and the cached irreducible loss IL_i are used to compute the reducible loss $\rho_i = L_i - IL_i$. Out of the 320 samples, only the top 10% (i.e., 32 samples) with the highest ρ_i values are selected for backpropagation; the remaining samples in the batch are ignored. According to the original paper, the model achieves 80% accuracy by the 39th epoch. In our replication:

- At epoch 39, we observed 79.47% accuracy.
- At epoch 41, we reached exactly 80% accuracy.

We carried out multiple improvements to this baseline and compared each of our proposed methods against the original baseline on the CIFAR-10 dataset throughout all our experimentation.The following sections present all of our proposed implementations.

5.2 DivBS + RHO-LOSS Approach

The RHO-LOSS method selects training samples those with high residual loss (i.e., difference between the model's current loss and an irreducible baseline loss). While this method ensures the model focuses on "not-yet-learned" examples, it does not account for redundancy in already selected samples which may be similar and contribute overlapping gradient information. This leads to inefficient learning and slower convergence.

To address this, we integrated DivBS (Diverse Batch Selection) (4) at batch selection step, it introduces gradient-based diversity criterion. It selects samples whose gradients point in different directions, ensuring each one contributes unique learning signals. Instead of just choosing the hardest examples, DivBS finds a subset whose influence on the model is as diverse as possible—maximizing gradient coverage and reducing wasted computation.

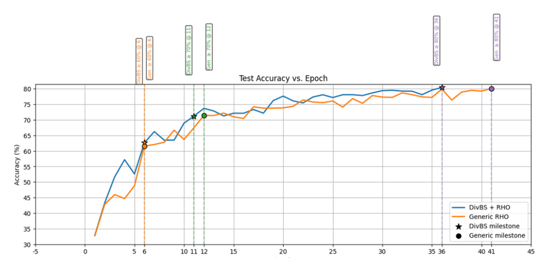

Figure 1: Comparison between RHO method and Improved method

5.2.1 Implementation and Results on CIFAR-10

In each training step, the RHO-LOSS method first identifies the top 50 hardest samples and from a minibatch using the difference between current loss and irreducible loss, same as RHO Loss approach. Now these 50 samples may contain samples which are redundant.DivBS then filters these 50 to 32 diverse examples based on their gradient directions. These 32 go through backpropagation, resulting in broader and more efficient parameter updates. Experiments on CIFAR-10 show that DivBS + RHO outperforms plain RHO-LOSS across the board. Notably, it reaches 70% test accuracy by epoch 11 (vs. 14 in RHO) and hits 80% at epoch 36 (vs. 41 in RHO), showing faster and smoother convergence. This proves that adding diversity to hard-sample selection can enhance learning efficiency and robustness.

5.3 EMA-Adaptive RHO

Traditional RHO-LOSS methods fix the number of samples selected per batch (e.g., top-32). However, this static strategy does not adapt to the model's training phase. In early epochs, many samples may carry high reducible loss, reflecting significant room for improvement. Conversely, as the model converges, most samples become redundant repeatedly updating on them will not be any use to us. Generic RHO Loss model ignores this dynamic nature, treating all phases equally in terms of training intensity. To mitigate above issues, We propose EMA-Adaptive RHO, an enhancement to the RHO-LOSS mechanism that dynamically adjusts the number of samples selected per batch based on recent learning signal trends using Exponential moving average of RHO loss.

First, the average reducible loss in the current batch is computed as $\bar{\rho} = \frac{1}{B} \sum_{i=1}^{B} \rho_i$, where $\rho_i = L_i - IL_i$ and B is the batch size. An EMA of this mean loss is maintained using $\text{EMA}_t = \beta \cdot \text{EMA}_{t-1} + (1 - \beta) \cdot \bar{\rho}$, with choice of $\beta = 0.9$ to smooth over fluctuations and capture long-term informativeness.

To determine training intensity, we compute a scaling ratio: $r_t = \exp(\bar{\rho} - \text{EMA}_t)$. This ensures the ratio remains strictly positive even when losses are negative. Finally, the number of samples selected in the current step is scaled as $k_t = \text{clip}(\alpha \cdot r_t \cdot k_{\text{base}}, k_{\min}, k_{\max})$, where α is a scaling hyperparameter (e.g., 0.1), and clip restricts k_t to lie within predefined bounds, such as $[16, 96]$. This method allows the model to selecting more examples when data has high learnable samples, and fewer when redundancy is detected thereby making training both efficient and adaptive.

5.3.1 Implementation and Results on CIFAR-10

EMA-Adaptive RHO achieves same accuracy as traditional RHO-LOSS (both 81% at epoch 41) but with improved data usage. By epoch 50, it uses approximately 3,532 fewer samples than traditional RHO (122,868 vs. 126,400), reflecting a 2.8% reduction in forward-backward passes without sacrificing performance. Unlike the baseline RHO, which selects a fixed number of samples per step, EMA-Adaptive RHO dynamically adjusts the number of selected examples based on an EMA-scaled reducible loss ρ. This adaptivity leads to slightly lower computational cost benefiting training time by focusing updates on high-information samples.

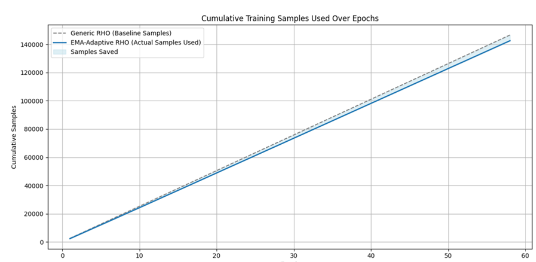

Figure 2: Comparison between RHO method and EMA method

5.4 Entropy-Regularized RHO-LOSS

While RHO-LOSS didnt considered prediction uncertainty during training. This creates an implicit bias toward high-loss examples, potentially ignoring ambiguous samples that are still valuable but uncertain. To address this, we introduce Entropy-Regularized RHO-LOSS, a hybrid scoring function that augments the traditional RHO-LOSS score $\rho_i = L_i - IL_i$ by incorporating entropy-based uncertainty. The goal is to balance between reducing loss and exploring uncertain regions of the decision boundary. Entropy encourages the inclusion of points where the model exhibits high predictive uncertainty, ensuring exploration of ambiguous but potentially learnable examples.

Entropy-Augmented RHO Method: The training pipeline closely follows the RHO-LOSS procedure, with the key difference being the modified RHO Loss scoring mechanism that includes entropy:

1. Compute the model's predicted class probabilities p_i using the softmax function.
2. Calculate the entropy of each prediction: $H(p_i) = -\sum_j p_i^j \log(p_i^j)$
3. Define the new entropy-regularized scoring function: $\rho_i' = L_i - IL_i + \beta \cdot H(p_i)$
4. Select the top-32 examples from each batch (batch size = 320) with the highest ρ_i' values.

5.4.1 Implementation and Results on CIFAR-10

We evaluate this method using three different values of the hyperparameter $\beta \in \{0.5,\ 0.6,\ 0.7\}$ to control the trade-off between reducible loss and uncertainty.

Variant	Epoch to 60%	Epoch to 70%	Epoch to 80%
Generic RHO-LOSS	6	12	41
Entropy-RHO ($\beta = 0.5$)	6	13	39
Entropy-RHO ($\beta = 0.6$)	8	16	39
Entropy-RHO ($\beta = 0.7$)	7	15	**37**

Table 1: Epoch milestones to reach accuracy thresholds for different RHO variants

Entropy-RHO consistently matches or outperforms the Generic RHO-LOSS baseline across early and mid-stage milestones. The $\beta = 0.7$ configuration achieved the fastest convergence to 80% test accuracy (epoch 37).

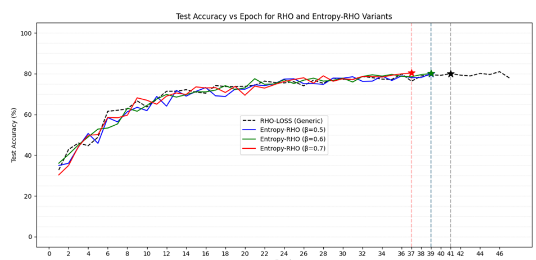

Figure 3: Comparison between RHO method and Entropy regularized RHO

5.5 RHO-LOSS + BALD (Bayesian Active Learning by Disagreement)

Traditional RHO-LOSS prioritizes samples with high reducible loss but overlooks predictive uncertainty, potentially leading to redundant updates on confidently incorrect examples. To address this, we integrate **Bayesian Active Learning by Disagreement (BALD)** into the RHO-LOSS framework to account for epistemic uncertainty. BALD quantifies model uncertainty due to limited knowledge (epistemic), making it useful for identifying unexplored or ambiguous regions of the input space. This complements RHO-LOSS, which focuses purely on learnability.

We adopt a two-stage selection mechanism: (1) select samples with high reducible loss $\rho_i = L_i - IL_i$; and (2) among these, select the most uncertain examples based on BALD scores.

5.5.1 Implementation and Results on CIFAR-10

Each training batch follows a two-stage filtering process:

RHO-LOSS Stage: For each batch of 320 samples, we compute per-sample cross-entropy loss L_i and look up irreducible loss IL_i. Next we, Compute reducible loss $\rho_i = L_i - IL_i$. Next we, Select the top 100 samples with the highest ρ_i.

BALD Stage: We enable dropout, and perform 10 stochastic forward passes over the 100 selected samples. Compute the BALD score for each sample using

$$\text{BALD}(x) = H\left[\mathbb{E}_{\theta}\left[P(y \mid x, \theta)\right]\right] - \mathbb{E}_{\theta}\left[H\left[P(y \mid x, \theta)\right]\right]$$

Next, we Select the top 32 most uncertain samples for the final backward pass.

Accuracy Threshold	RHO-LOSS Epoch	RHO + BALD Epoch	Improvement
60%	6	6	No difference
70%	12	11	1 epoch faster
80%	41	35	6 epochs faster

Table 2: Empirical comparison of accuracy milestones for RHO-LOSS and RHO + BALD

RHO-LOSS + BALD approach demonstrates that combining learnability (via reducible loss) with uncertainty (via BALD) leads to faster convergence and more effective training. By focusing only on the most impactful and uncertain samples, the model reaches target accuracies earlier with fewer redundant updates.

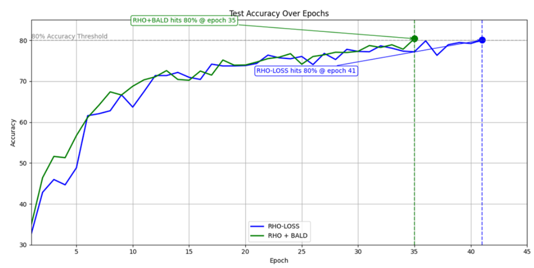

Figure 4: Comparison between RHO method and Entropy regularized RHO

5.6 RHO + Margin Sampling(Two-Stage Data Selection for Target Model Training)

Traditional RHO-LOSS, While effective at focusing on difficult examples, RHO-LOSS can still suffer from redundancy. Some of the selected high- examples may be confidently misclassified. To address this, we introduce a two-stage filtering mechanism that refines RHO-based selection using margin-based uncertainty sampling.

We implement a two-stage selection strategy in each batch of training:

1. **RHO Filtering:** From a mini-batch of 320 examples, we compute the per-example score $\rho_i = L_i - IL_i$, where IL_i is the irreducible loss estimated by a fixed proxy ResNet-18. We select the top-100 examples with the highest ρ_i values.
2. **Margin Sampling:** Among these top-100 examples, we compute the *softmax margin*—defined as the difference between the top-2 class probabilities—and select the 32 most ambiguous examples (i.e., those with the lowest margins). These 32 samples are then used in the backward pass.

5.6.1 Implementation and Results on CIFAR-10

Threshold Accuracy	Generic RHO (Baseline)	RHO + Margin Sampling
60% Accuracy	Epoch 6	Epoch 10
70% Accuracy	Epoch 12	Epoch 17
80% Accuracy	Epoch 41	Epoch 44

Table 3: Epoch milestones comparison between Generic RHO and RHO + Margin Sampling

Both methods ultimately cross 80% accuracy, but the margin-sampling strategy consistently lags behind by 3–4 epochs at each milestone. The results of this experiment demonstrate that the standard RHO-LOSS method is more effective than the proposed RHO + Margin Sampling strategy. RHO-LOSS consistently reaches key accuracy thresholds (60%, 70%, 80%) in fewer epochs, indicating faster and more efficient convergence. The additional ambiguity-based filtering step in the RHO + Margin approach, while conceptually appealing, introduces unnecessary overhead. It slows down learning by potentially excluding high-impact samples that are not ambiguous but still highly informative.In contrast, RHO-LOSS directly targets examples with the highest remaining learnability—those that the current model performs poorly on relative to their irreducible loss. This focused prioritization enables the model to improve more rapidly without being distracted by overly complex or uncertain samples that may not provide the best gradient signal.

5.7 Greedy Diversity-Aware Selection

RHO-LOSS may select multiple highly similar samples, reducing training efficiency due to redundant gradient updates. *Greedy Diversity-Aware Selection* discourages this redundancy by penalizing the selection of similar examples, ensuring that the chosen batch covers a broader range of information from the input space.

To mitigate this issue, we experimented with a modified selection score, computed as:

$$\text{Score}_i = \rho_i - \lambda \sum_{j \in S} \text{Similarity}(x_i, x_j)$$

where ρ_i is the RHO-LOSS of sample i, S is the set of previously selected samples in the current batch, and $\text{Similarity}(x_i, x_j)$ is measured via cosine similarity. The penalty term discourages selecting samples that are too similar to those already chosen.

5.7.1 Implementation and Results on CIFAR-10

Although this method aimed to encourage diversity, it required more epochs to reach 80% accuracy—taking 48 epochs compared to 41 for standard RHO-LOSS. This suggests that the greedy penalty, while helpful in reducing redundancy, may cause the model to exclude highly informative samples that happen to be similar to others. As a result, it may prioritize dissimilar but less valuable examples, slowing down learning.

Key Insight: Penalizing total similarity across all previously selected samples can lead to overly cautious behavior and locally suboptimal selections, especially in high-density regions of the data distribution.

5.8 Hybrid RHO-Diversity Selection

To address the limitations of Greedy Diversity, we introduce *Hybrid RHO-Diversity Selection*, which instead penalizes only the **maximum similarity** to any already selected sample. This formulation allows for greater flexibility by tolerating mild similarity but actively avoiding the selection of near-duplicates.

The selection score is computed as: $\text{Score}_i = \rho_i - \lambda \cdot \max_j \text{Similarity}(x_i, x_j)$.

Here, the penalty term targets only the most similar prior sample, enabling the model to retain informative examples while still encouraging diversity.

5.8.1 Implementation and Results on CIFAR-10

We evaluated Hybrid RHO-Diversity with different penalty weights λ. The results show that this method achieves faster convergence than both Greedy Diversity and standard RHO-LOSS, especially at $\lambda = 0.3$, where it reached 80% accuracy in just 38 epochs.

By penalizing only the most similar sample, Hybrid RHO-Diversity strikes a better balance between informativeness and diversity, resulting in more efficient learning than the Greedy approach. This targeted regularization leads to faster convergence and better overall selection quality.

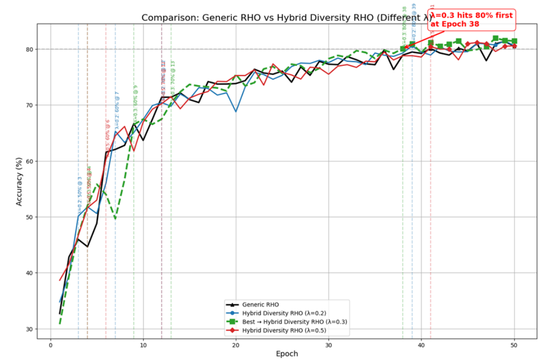

Figure 5: Comparison between RHO method and Proposed Improvement

For the Hybrid Diversity RHO method, the number of epochs required to reach 80% test accuracy varies slightly with the choice of the class-balancing coefficient λ. When $\lambda = 0.2$, the model reaches 80% accuracy at epoch 39. With $\lambda = 0.3$, the model achieves this accuracy slightly earlier, at epoch 38. Increasing λ to 0.5 results in reaching the 80% threshold at epoch 41.

5.9 RHO+ Diversity aware, Class Aware Subset Training (Class-Balanced RHO)

Generic RHO does not adapt to class difficulty or imbalance, potentially causing some classes to dominate early training. RHO-DCAST adjusts the selection size per class, dynamically balancing the selection based on class difficulty and dynamics during training.

The selection per class is defined as: $k_t = \text{clip}(\alpha \cdot r_t \cdot k_{\text{base}}, k_{\min}, k_{\max}), \quad r_t = \exp(\bar{\rho} - \text{EMA}_t)$.

where k_t is the number of samples selected from class t, α is a scaling factor, k_{base} is the base number of samples, and $k_{\min}$, $k_{\max}$ are the minimum and maximum limits on class selection size. The term r_t adjusts the class selection size based on the dynamic progress of each class, measured by an exponential moving average (EMA) of the relevance score $\bar{\rho}$.

5.9.1 Implementation and Results on CIFAR-10

RHO-DCAST significantly improves learning efficiency, especially at $\lambda = 0.2$, where 80% accuracy is reached in just 34 epochs(Traditional RHO took 41 epochs). It adapts to class-level learning progress and maximizes the gain from challenging classes, leading to more stable and efficient training dynamics.

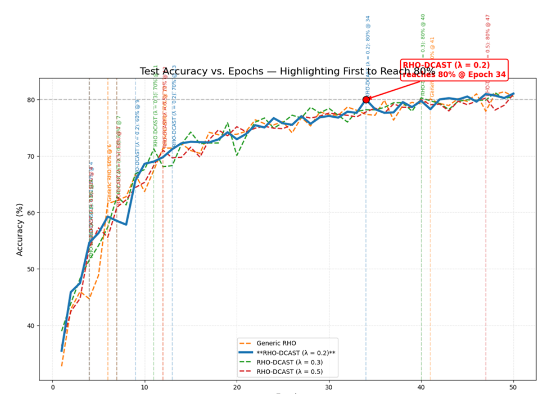

Figure 6: Comparison between RHO method and Proposed Improvement

We observed that the RHO-DCAST (Class-Balanced RHO) method reaches 80% test accuracy at different epochs depending on the value of the class-balancing coefficient λ. Specifically, with

$\lambda = 0.2$, the model achieves 80% accuracy by epoch 34. Increasing the coefficient to $\lambda = 0.3$ delays this milestone to epoch 40, while a further increase to $\lambda = 0.5$ results in the model reaching 80% accuracy at epoch 47.

5.10 Curriculum + RHO Hybrid Training

Our intuition for experimenting with this approach is, In the early stages of training, when the model is uninitialized, even simple examples yield high loss values. This creates a potential cold-start issue for RHO-LOSS, as it may focus prematurely on seemingly "high-priority" samples that are, in fact, overly difficult for an untrained model. To address this, we explored a hybrid approach—Curriculum + RHO-LOSS.

5.10.1 Implementation and Results on CIFAR-10

To evaluate whether curriculum learning enhances convergence in RHO-LOSS, we implemented a hybrid training strategy. During the first 4 epochs (the curriculum phase), the model was trained using the 32 samples in each batch that had the **lowest cross-entropy loss**—i.e., those that the model already classified with highest confidence. After this phase, training switched to standard RHO-based selection, where samples with the highest reducible loss ($\rho_i = \mathrm{CE}_i - \mathrm{IL}_i$) were chosen. Despite this structured warm-up, the hybrid model failed to reach 80% test accuracy even after 300 epochs of training on CIFAR-10 using a ResNet-18 architecture. The accuracy plateaued significantly below the target, indicating that the curriculum phase did not support overall convergence.

The hybrid Curriculum + RHO-LOSS model failed to reach 80% accuracy even after 300 epochs, likely due to early focus on low-loss samples limiting gradient diversity. Pure RHO-LOSS remains more effective, offering faster and more stable convergence without the overhead of a curriculum phase.

6 Key Learnings and Insights

A central takeaway from our work is, selecting high-loss samples without accounting for their similarity can introduce redundancy, leading to overlapping gradients and diminished learning efficiency. To address this, diversity-aware strategies such as DivBS or enforcing maximum similarity constraints among selected samples have shown to accelerate convergence by exposing the model to a richer set of samples. Moreover, incorporating uncertainty estimates, particularly through entropy or Bayesian methods like BALD, enhances sample selection by identifying ambiguous but informative examples that RHO-LOSS may miss. We also found that adaptive sample selection, where the number of samples per batch changes dynamically with training progress (e.g., guided by the EMA of the loss), leads to more efficient training. In parallel, class-balanced selection strategies like RHO-DCAST were crucial for improving generalization by ensuring that underrepresented classes receive adequate attention during training.

Not all techniques proved beneficial—margin-based filtering, for instance, added computational overhead without measurable gains, underlining the need for rigorous empirical validation. Ultimately, our findings suggest that combining informativeness (via RHO) with novelty (via diversity measures) produces the most effective training batches. Additionally, approaches like EMA-Adaptive RHO showed that it is possible to reduce computational costs measured in forward/backward passes without sacrificing performance.

7 Future work

The current method relies on a static holdout model trained on randomly selected data, which can introduce variance and limit representativeness. To address this, we propose exploring robust holdout set selection techniques. This ensures that the holdout set reflects the overall data distribution more accurately, resulting in better irreducible loss estimation. Furthermore, we suggest developing adaptive IL models capable of evolving with the data. In dynamic environments where data distributions shift over time, it becomes crucial to incorporate new, informative samples into the holdout set and replace outdated or underperforming examples. This adaptive strategy would help maintain the IL model's relevance and accuracy throughout training, ensuring that sample prioritization remains effective even as the dataset changes. The performance of proposed entropy-regularized and adaptive selection methods heavily depends on the choice of hyperparameters such as the entropy coefficient (β) and the scaling factor (α). An automated tuning mechanism, potentially leveraging Bayesian optimization, could adapt these values dynamically during training, improving both robustness and generalization. While DivBS addresses gradient-level redundancy, future work can explore clustering techniques to detect and minimize feature-level redundancy. Such clustering-based filtering could help in selecting samples that provide maximum information gain per training step, especially in high-dimensional spaces.

8 Contributions

Edula Vinay Kumar Reddy implemented and evaluated the *DivBS + RHO-LOSS*, *EMA-Adaptive RHO* and *Curriculum + RHO Hybrid Training* enhancements and Replicated the results of seed paper.

Telugu Sudhakar implemented and evaluated the *Entropy-Regularized RHO-LOSS*, *RHO-LOSS + BALD* and *RHO + Margin Sampling* enhancements.

Pokala Dattatreya implemented and evaluated the *Greedy Diversity-Aware Selection* , *Hybrid RHO-Diversity Selection* and *RHO-DCAST*

Note: All team members contributed equally to the overall project in terms of discussions, experimentation, and final report preparation.

Disclaimer

We hereby declare that the project presented in this report is original and has been carried out specifically for the course Probabilistic Machine Learning(CS772A/2024-25/Even semester).It has not been submitted previously for credit in any other course, internship, research program, or academic evaluation. No part of this work has been copied or reused from prior academic or professional projects for which credit has already been received. All team members affirm the authenticity and originality of this submission.

References

[1] Ilya Loshchilov and Frank Hutter. *Online Batch Selection for Faster Training of Neural Networks*. arXiv preprint arXiv:1511.06343, 2015. Available at: `https://arxiv.org/abs/1511.06343`

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. *Curriculum Learning*. In Proceedings of the 26th Annual International Conference on Machine Learning, pp. 41–48, 2009. Available at: `https://doi.org/10.1145/1553374.1553380`

[3] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. *Bayesian Active Learning for Classification and Preference Learning*. arXiv preprint arXiv:1112.5745, 2011. Available at: `https://arxiv.org/abs/1112.5745`

[4] Zhaoxia Hong, Wenxuan Wang, Zihao Xiao, Ziwei Wang, and Zhiqiang Shen. *Diversified Batch Selection for Training Acceleration (DivBS)*. arXiv preprint arXiv:2401.06434, 2024. Available at: `https://doi.org/10.48550/arXiv.2406.04872`

[5] Cody Coleman, Chiyuan Zhang, Samy Bengio, James Z. Wang, and Matei Zaharia. *Selection via Proxy: Efficient Data Selection for Deep Learning*. International Conference on Learning Representations (ICLR), 2020. Available at: `https://arxiv.org/abs/1906.11829`

[6] Sören Mindermann, Jan Brauner, Muhammed Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. *Prioritized Training on Points that are Learnable, Worth Learning, and Not Yet Learnt*. Proceedings of the 39th International Conference on Machine Learning (ICML), PMLR 162, 2022. Available at: `https://doi.org/10.48550/arXiv.2206.07137`

[7] Seed paper source code: `https://github.com/OATML/RHO-Loss`