FRAMEWORK FOR TESTING ADVERSARIAL ROBUSTNESS OF CORESET SELECTION PARADIGMS FOR DATA-EFFICIENT MACHINE LEARNING

GM Harshvardhan
gmharsh@bu.edu
Department of Computer Science, Boston University

Harsh Sharma
hsharma@bu.edu
Department of Computer Science, Boston University

1 Introduction

Deep learning has grown significantly in the past decade and is now commonly used in many areas. Better computers have allowed us to work with more data and use advanced algorithms. However, we still need help processing large amounts of data with limited computing power. One potential solution to this problem is using coresets, a smaller representative set of data points that can approximate the entire dataset's performance. In recent years, there has been growing interest in using coresets in deep learning as a way to reduce the computational cost of training models while still achieving good accuracy. Hence, there have been efforts made to develop coresets of big datasets to improve model sample efficiency [1], [2], [3], [4]. These coresets, however, may be prone to adversarial attacks [5]. One adversarial attack that accesses a model's internal parameters to manipulate input images to attack the model's predictions is the Fast Gradient Sign Method (FGSM) [6], which is demonstrated in Fig. 1.

This project aims to explore different coreset creation techniques and analyze their performance in the context of adversarial attacks in deep learning. Specifically, we plan to investigate how well different coreset creation methods preserve the important features and structure of the original dataset and how this affects the accuracy and efficiency of deep learning models trained on the coreset. Moreover, we seek to evaluate their performance when the best coreset selection methods are adversarially attacked.

2 Selection Methods and FGSM

We will compare the different algorithms used to create coresets to train deep learning methods. The comparison will be performed for these methods for different sizes of coresets generated by different methods. Following are the methods that we have implemented: Contextual Diversity [7], GraNd [8], Glister [9], Cal [10], GraphCut [11], and random. For adversarial attacks, we employ the FGSM attack [6]. The Fast Gradient Sign Method (FGSM) is a type of adversarial attack that creates small perturbations to the input data in order to fool a deep neural network into misclassifying it. The idea behind the attack is to take advantage of the gradient of the loss function with respect to the input data to determine the direction of the perturbation, and then to add a small perturbation in that direction. Specifically, given an input image x, the FGSM attack computes the gradient of the loss function $J(\theta, x, y)$ with respect to x, where θ is the set of model parameters, and y is the true label of x. The attack then adds a small perturbation ϵ times the sign of the gradient to the original image x, resulting in a new image $x_{adv} = x + \epsilon * sign(\nabla_x J(\theta, x, y))$. The value of ϵ controls the strength of the attack. A larger value of ϵ can result in a stronger attack that produces more noticeable changes to the input image, but may also reduce the overall effectiveness of the attack. On the other hand, a smaller value of ϵ may produce a less noticeable attack, but may be more effective at fooling the model.

2.1 Contextual Diversity

Contextual Diversity [7] is an active learning approach that uses contextual diversity (CD) to select a representative subset of data for deep convolutional neural networks (CNNs) fine-tuning. The authors argue that the commonly used measures of visual diversity or prediction uncertainty fail to capture the spatial context variations, which are critical for CNNs' accurate predictions. The CD measure captures the confusion associated with spatially co-occurring classes. It is based on the observation that CNNs' predicted probability vectors for a region of interest typically contain information from a larger receptive field.

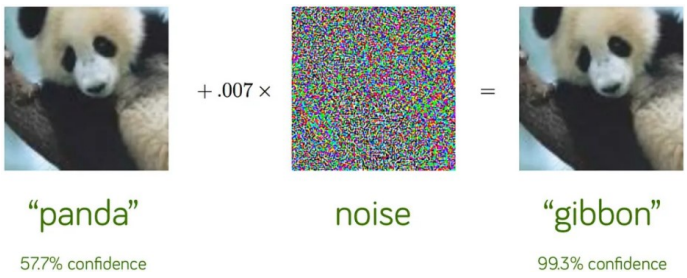

Figure 1: Example of FGSM attack, credit: [6]

2.2 GraNd

GraNd [8] identifies important examples in vision datasets and prunes the training set by discarding them without sacrificing test accuracy. The authors observe that simple scores, calculated from the gradient and error norms of multiple weight initializations, can be used to identify important examples early in training. GraNd is a score that measures the importance of examples based on their gradient norms and is one of the two proposed scores. At epoch t, given a data point and label $(\boldsymbol{x}, y)$, the GraNd score can be defined as shown in eq. 1

$$G_t(\boldsymbol{x}, y) \triangleq \mathbb{E}_{\theta_t} ||\nabla_{\theta_t} l(x, y; \theta_t)||_2 \tag{1}$$

$G_t(\boldsymbol{x}, y)$ is the average contribution of every $(\boldsymbol{x}, y)$ to the decline of train loss l. The authors also show that these scores can generalize across architectures and hyperparameters, detect noisy examples, and shed light on the training dynamics. Compared to previous work, the proposed scores use only local information early in training to prune the training set.

2.3 Glister

In general, selecting a coreset can be formulated as a bi-level optimization problem, wherein a subset is selected as the outer objective, with the inner objective maximizing, say, a log-likelihood. Glister [9] is a framework for efficient and robust learning through data subset selection. It selects a subset of the training data to maximize the log-likelihood on a held-out validation set, accomplished through bi-level optimization and iterative online algorithm. If S^* is the subset selection from the data T, the bi-level optimization to maximize log-likelihood $\mathcal{L}$ with model parameters θ can be defined as shown in eq. 2:

$$S^* = argmax_{S \subseteq T} \sum_{(x,y) \in V} \left(\mathcal{L}(\boldsymbol{x}, y; \max_{\theta} \sum_{(x,y) \in S} (\mathcal{L}(\boldsymbol{x}, y; \theta))) \right) \tag{2}$$

2.4 Contrastive Active Learning (Cal)

Contrastive Active Learning (CAL) is a method for coreset selection that identifies data points near the decision boundary. CAL selects data points whose predictive likelihood diverges the most from their neighbors to construct the coreset. This method is based on the intuition that data points near the decision boundary are hard to classify and can provide the most informative samples for further training.

Using a random sampling method, CAL first selects a subset of data points from the original dataset. Then, for each data point in the selected subset, CAL computes the nearest neighbors in the feature space. The predictive likelihood of each data point is then compared to the predictive likelihood of its nearest neighbors. Data points whose predictive likelihood diverges the most from their neighbors are selected to form the coreset.

The CAL method selects diverse and informative data points near the decision boundary. By selecting data points with divergent predictive likelihoods, CAL identifies the regions where the classifier is most uncertain in the feature space. This approach can lead to the selection of informative data points that can improve the classifier's accuracy while reducing the dataset's size. CAL is effective in various applications, including image classification, object detection, and natural language processing.

2.5 GraphCut

The Graph Cut (GC) is a submodular function used for coreset selection. The GC function measures the minimum cost of partitioning a graph into two disjoint sets, where the cost is defined as the sum of the weights of the edges cut by the partition. A cut is defined as the edges crossing between the two disjoint sets.

To select a coreset using the GC function, a subset of the original dataset is selected that approximates the original dataset well. This can be achieved by solving a graph cut problem on a graph constructed from the original dataset. The nodes in the graph represent the data points, and the edges between the nodes are weighted according to the similarity between the data points.

The graph cut-based coreset selection method selects a subset of data points that preserves the graph structure and contains diverse representatives of the original dataset. The method selects data points iteratively by greedily adding the data point that provides the maximum reduction in the graph cut value. This process continues until the desired size of the coreset is reached. The greedy algorithm has a bounded approximation factor of 1 - 1/e, ensuring that the selected coreset is close to the optimal solution. The GC-based coreset selection method effectively and efficiently selects a representative subset of the original dataset while preserving the graph structure.

2.6 Moderate DS

The technique involves extracting representations from a well-trained deep model. The model is denoted as $f(x) = g(h(x))$, where $h(x)$ maps input data to hidden representations at the penultimate layer, and $g(x)$ maps these hidden representations to the output for classification. The hidden representation for a given data point $s = (x, y)$ is $h(x)$. Using the trained deep model $f(x)$ and full training data $S = s_1, ..., s_n$, the hidden representations of all data points are obtained as $z_1 = h(x_1), ..., z_n = h(x_n)$.

The class center of each class is then calculated at the representational level by taking the mean of the representations from one class across all dimensions, as shown in Equation (3). This allows for the representation extraction of the training data, which can be used for classification or further analysis.

$$\left\{ \mathbf{z^j} = \frac{\Sigma_{i=1}^{n} \Pi[y_i = j] \mathbf{z_i}}{\Sigma_{i=1}^{n} \Pi[y_i = j]} \right\}_{i=1}^{j} \tag{3}$$

2.7 Random

Random coreset selection is a simple and effective method for selecting a representative subset of data points from a large dataset. This method randomly selects a fixed number of data points from the original dataset to form the coreset.

Random coreset selection has the advantage of being computationally efficient as defined in Equation (4) and easy to implement. It does not require any prior knowledge or assumptions about the data, making it a widely used method for coreset selection. Additionally, this method can be applied to any type of data, including high-dimensional and complex data.

However, the main drawback of random coreset selection is that it may only sometimes select the most informative or diverse data points. Random selection may result in a coreset that does not accurately represent the structure of the original dataset. Despite its limitations, random coreset selection remains a valuable and practical method for coreset selection. It can be used as a baseline method for comparison with other coreset selection techniques. It can be helpful when computational resources are limited or time is a constraint.

$$P(element\ selected) = frac \tag{4}$$

3 Dataset

Our project relies on the CIFAR10 dataset [12], a popular benchmark dataset for image classification tasks. The CIFAR10 dataset consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class. The classes are mutually exclusive and represent everyday objects, such as airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks.

We initially planned to incorporate other datasets in our experimentation, but we focused solely on the CIFAR10 dataset due to computational time limitations. However, the vast diversity of images available in CIFAR10 provides a wide variety of objects and backgrounds to study.

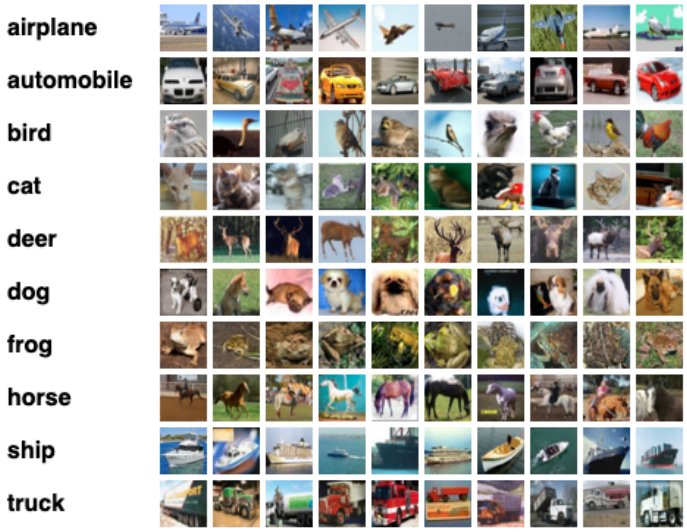

Figure 2: Examples of images in CIFAR10. credit: [12]

To generate a coreset to train our models on, we used different fractions of the CIFAR10 dataset. Our coreset selection method is critical to the efficiency of our study, and the images that our method selected from the dataset played a vital role in our results.

Figure 2 shows examples of images from the CIFAR10 dataset. These images exhibit diverse objects and backgrounds with different shapes, colors, and sizes. Therefore, CIFAR10 is a suitable dataset to train machine learning models for image classification tasks.

4 Individual contributions

The individual contributions of each team member were as follows (terminology adopted from [13]):

GM Harshvardhan. Conceptualization, methodology, software, validation, investigation, data curation, writing - original draft, and visualization.

Harsh Sharma. Conceptualization, methodology, software, formal analysis, resources, writing - original draft and writing - review & editing.

5 Results and Discussion

Table 1 compares different coreset selection methods with two different fractions of the dataset selected: 10% and 50%. The methods compared are Cal, Contextual Diversity, Glister, GraNd, Moderate DS, Submodular (GraphCut/GC), and Random. The percentages shown in the table indicate the percentage of performance achieved by each method compared to the entire dataset.

Surprisingly, the Random method outperforms more sophisticated methods such as Cal, Contextual Diversity, Glister, GraNd, and Moderate DS. Although this result may seem counter-intuitive, it is consistent with previous studies [14] that have also reported the superior performance of random selection. However, Submodular outperforms all other methods in both fractions of the dataset. These results suggest that random selection can be a simple and effective method for coreset selection, and Submodular can be an excellent alternative for more complex applications.

Next, we compared the best selection models across all our experiments against the Fast Gradient Sign Method (FGSM) attack. First, we train a MobileNetV2 (small) [15] for different settings of a) learning rate, b) coreset selection method, and c) dataset fraction size. Once the models for each unique hyperparameter combination are trained, we store the weights and load only those models which attained the highest accuracy.

The resultant accuracy for varying epsilon values can be seen in Fig. 3. From this result; we infer that no coreset selection has strong defenses against old adversarial attacks such as FGSM (initially proposed in 2014). Each method suffers from low accuracy with small epsilon values, and most interestingly, Moderate DS [16] also succumbs to the adversarial attacks, just as other methods. The adversarial examples for Cal are shown in Fig. 4 and for Moderate DS in Fig. 5.

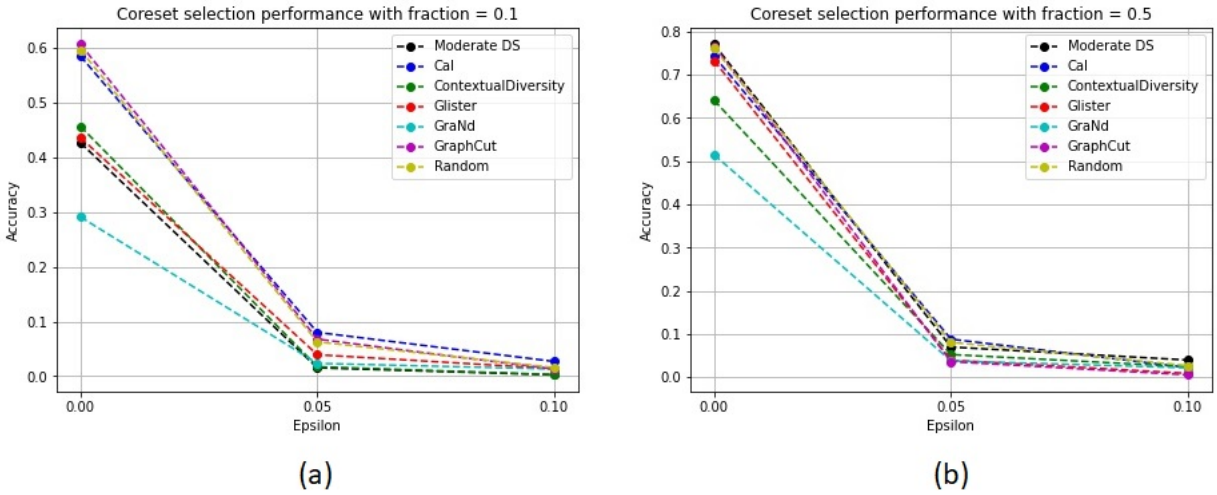

Figure 3: Effect of varying epsilon values on the accuracy of MobileNetV2 on different coreset selection methods. (a) For a fraction of 10%, (b) for a fraction of 50%.

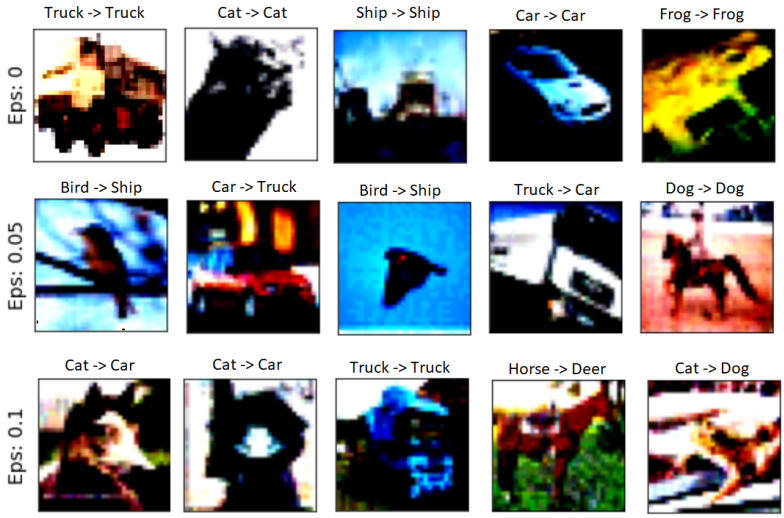

Figure 4: Adversarial attack examples for Cal selection method.

6 GitHub Repo

All the code for our implementation can be found at https://github.com/97harsh/CoreAdv. The experiments can be run using the "run_experiments.sh" shell file, and our implementation for adversarial attacks can be found in the "adversarial test new.ipynb" Jupyter notebook.

7 Conclusions and Future Work

In our project, we tested various coreset selection methods for training deep neural networks for image classification and how they fare when adversarially attacked. It was seen that random coreset selection is still a very robust method, despite of there being other sophisticated selection techniques. Moreover, certain coreset techniques designed to be adversarially robust (e.g. Moderate DS) may not perform very well against very strong attacks (such as the FGSM attack with a high epsilon). Thus, we conclude that while there exist coreset selection techniques in machine learning that may marginally improve training performance over smaller fractions of the dataset, these coresets may not be adversarially robust.

Figure 5: Adversarial attack examples for Moderate DS selection method.

Table 1: Comparison of coreset selection methods. The percentage refers to the fraction of the dataset used.

Method	10%	50%
Cal	58.5	74.2
Contextual Diversity	45.7	64.1
Glister	43.6	73.2
GraNd	29.1	51.4
Moderate DS	54.7	75.3
Submodular	60.7	76.6
Random	59.7	76.0

Future work involves testing more selection algorithms against attacks like Projected Gradient Descent [17], BIM/iterative-FGSM [18], and their variants to further verify our findings in this project. More research should also be done into developing specific coreset selection techniques that are adversarially robust.

References

[1] Olivier Bachem, Mario Lucic, and Andreas Krause. Practical coreset constructions for machine learning. *arXiv preprint arXiv:1703.06476*, 2017.

[2] Dan Feldman. Core-sets: Updated survey. *Sampling techniques for supervised or unsupervised tasks*, pages 23–44, 2020.

[3] Jiayi Wang, Chengliang Chai, Nan Tang, Jiabin Liu, and Guoliang Li. Coresets over multiple tables for feature-rich and data-efficient machine learning. *Proceedings of the VLDB Endowment*, 16(1):64–76, 2022.

[4] Zohar Karnin and Edo Liberty. Discrepancy, coresets, and sketches in machine learning. In *Conference on Learning Theory*, pages 1975–1993. PMLR, 2019.

[5] Samuel G Finlayson, John D Bowers, Joichi Ito, Jonathan L Zittrain, Andrew L Beam, and Isaac S Kohane. Adversarial attacks on medical machine learning. *Science*, 363(6433):1287–1289, 2019.

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[7] 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 16*, pages 137–153. Springer, 2020.

[8] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607, 2021.

[9] Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8110–8118, 2021.

[10] Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764*, 2021.

[11] Rishabh Iyer, Ninad Khargoankar, Jeff Bilmes, and Himanshu Asanani. Submodular combinatorial information measures with applications in machine learning. In *Algorithmic Learning Theory*, pages 722–754. PMLR, 2021.

[12] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[13] Liz Allen, Alison O'Connell, and Veronique Kiermer. How can we ensure visibility and diversity in research contributions? how the contributor role taxonomy (credit) is helping the shift from authorship to contributorship. *Learned Publishing*, 32(1):71–74, 2019.

[14] 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*, pages 181–195. Springer, 2022.

[15] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[16] 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.

[17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[18] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.