Mitigating Backdoor Poisoning using Data Summarization and Model Pruning

Sarthika Chimmula, Demetri Nicolaou, Adi Pillai

Abstract

Backdoor poisoning attacks embed hidden triggers in training data, causing models to misclassify triggered inputs while maintaining high clean accuracy. Existing defenses often require complex, multi-stage poison detection throughout training. We investigate whether simpler data summarization methods can provide backdoor defense as a byproduct of selecting representative training subsets. We evaluate four selection strategies (Random, EL2N, Forgetting Score, CRAIG) on CIFAR-10 with 2% backdoor poisoning, combined with iterative magnitude pruning. Our results show that difficulty-based methods (EL2N, Forgetting) systematically retain poisoned samples, with attack success rates (ASR) near 100%. In contrast, CRAIG, which selects based on gradient representativeness, filters 85.86% of poisons, reducing ASR to 83.77% (dense) and 60.45% (pruned) while maintaining ~80% clean accuracy. Pruning alone provides minimal benefit but amplifies CRAIG's effectiveness when combined. These findings demonstrate that representativeness-based selection offers simpler, more effective backdoor mitigation than difficulty-based methods, achieving meaningful defense without complex dynamic poison detection procedures.

Introduction

Backdoor data poisoning attacks allow an attacker to sneak a “key” into the training data, tricking a model into misclassifying examples with that key.[1] These attacks are dangerous because they require few poisoned examples to be successful, can be difficult to detect, and can easily achieve success rates in excess of 90%.[1] Real-life examples of such attacks could be using a specific set of glasses to fool a facial recognition system into correlating a specific pair of glasses with an individual’s face, or fooling a sentiment classification model using a specific keyword.[1,2]

Existing defenses against backdoor poisoning, such as EPIC, use data summarization methods like (CRAIG) combined with k-medoids clustering to identify and eliminate isolated data points in gradient space.[3,4] However, these approaches require repeatedly re-selecting data throughout training to track a dynamically changing dataset, adding significant complexity.[3] Moreover, they focus solely on poison detection rather than leveraging data summarization for its original purpose: achieving a gain in data efficiency through representative subset selection.

Data summarization methods, or coresets, offer a simpler alternative: select a fixed representative subset once, before training, that serves the dual purpose of efficiency *and* security. Forgetting Score selects samples with stable classifications during training (the ‘easy’ examples), EL2N (Error L2-Norm) selects samples with low early-epoch loss, and CRAIG (Coresets for Accelerating Incremental Gradient

Descent) uses submodular gradient matching to select samples whose gradients best represent the full dataset.[4,6,7] Unlike EPIC's dynamic poison-detection approach, we hypothesize that CRAIG's static selection of representative samples may naturally filter backdoor poisons as a byproduct of its core mechanism.[1,3] Effective backdoor attacks require anomalous gradients that pull models toward the attacker's target, making poisoned samples non-representative of their assigned class. Since CRAIG performs per-class gradient matching to select representatives, these anomalous poisons should be excluded as outliers even without explicit poison detection.[4] This approach is simpler than EPIC, while potentially providing both efficiency and security benefits.

In this study, we investigate two questions: 1) do data summarization methods on their own provide a meaningful mitigation against backdoor poisoning and 2) can data summarization be combined with model pruning to build models that are more robust to data poisoning? We hypothesize that data summarization and model pruning provide meaningful mitigations with an acceptable accuracy tradeoff. The following sections detail our experimental methodology, present our findings, and discuss implications for practical backdoor defense.

Proposed Method

We propose mitigating backdoor data poisoning attacks using a combination of data selection and model pruning to create a smaller model that is efficiently trained on a smaller dataset. In this study, we will experiment with four different data summarization methods: (1) random selection, (2) EL2N, (3) forgetting scores, and (4) CRAIG.[4,6,7] After we train a model on the summarized dataset, we will employ iterative magnitude pruning (IMP) starting at an early epoch to try and reduce the effect of any remaining poisons in the summarized datasets.[5]

Data Summarization

The data summarization techniques used are outlined in Fig. 1. Each summarization technique–random selection, EL2N, forgetting scores, and CRAIG are all applied to a pre-trained dataset (CIFAR-10) where 2% of the training examples are poisoned.[8] The backdoor poisoning attack used here is that 2% of the dataset is given a random red square, and the label of all poisoned examples is set to the target class, 0. This creates a spurious correlation between a red square–the "key" in a backdoor poisoning attack–that can be used to flip a correct label to the target classes' label.[1] Each data summarization technique pre-trains on 20 epochs before selecting the final dataset.

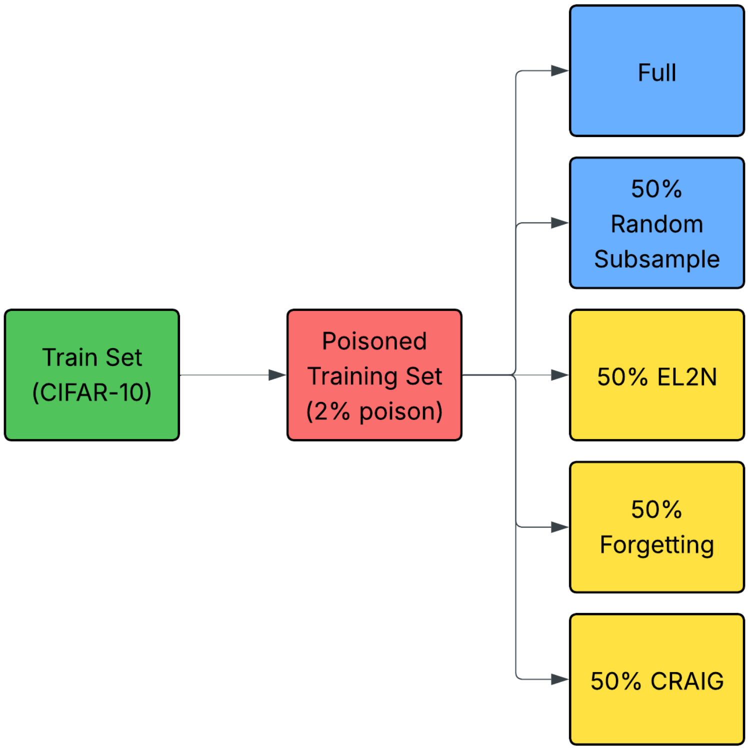

Figure 1: Dataset poisoning and summarization procedure. The CIFAR-10 dataset first undergoes a simulated backdoor poisoning attack (2% of examples are given a red square and the target class (0) label). Next, 5 datasets are generated - a full, unsampled dataset, a random dataset, an EL2N dataset, a forgetting score dataset, and a CRAIG dataset all with 50% downsampling.

Pruning

All pruned models utilized an iterative magnitude pruning (IMP) procedure.[5] This procedure rewinds the model to epoch 1, retrains to epoch 10, and repeats twice, pruning 30% of the weights each time. The aim is to select the most useful weights from the initial training run while not learning the effect of the final poison.

Model Training

All experiments use a resnet18 pytorch model that has been modified to accommodate CIFAR-10's 28x28 images.[8,9] All models were trained with a batch size of 128, using stochastic gradient descent, an initial learning rate of 0.1, momentum of 0.9, weight decay of 5e-4, and a cosine annealing learning rate scheduler with a minimum learning rate of 1e-5. Additionally, the CRAIG learning rate scheduler for pretraining uses a gradual warmup scheduler for the first 10 epochs like in the original CRAIG implementation.[4]

Results

The experimental pipeline outlined in Fig. 1 was run five times, and the poison retention rates can be seen in Fig. 2 and Tab. 1. We evaluate five data selection strategies—Full, Random, EL2N, Forgetting, and CRAIG—under a backdoor poisoning attack using three metrics and training both the dense neural network as well as the pruned model each for 50 epochs 5 different times:[1]

(1) **Clean Accuracy**, measured on the clean test set after training on poisoned data;
(2) **Attack Success Rate (ASR)**, defined as the fraction of triggered test samples classified as the attacker's target label; and
(3) **Poison Retention Rate**, the fraction of poisoned samples retained after subset selection. All experiments were repeated across five random seeds, and we report mean ± standard deviation.

Poison Retention

Figure 2 and Table 3 show that most selection methods fail to meaningfully reduce the presence of poisoned samples. Full and Random selection retain nearly all poisoned data (≈1.0 retention). EL2N also retains the vast majority of poisoned samples (0.9898 ± 0.0055), while Forgetting shows only a modest reduction (0.8952 ± 0.0408). In contrast, CRAIG removes the overwhelming majority of poisoned samples, retaining only 0.1414 ± 0.0609 of them. This indicates that among all evaluated methods, CRAIG is the only approach that consistently filters out poisoned samples at the subset selection stage.

Clean Accuracy

As shown in Figure 3 (left) and Table 3, all methods maintain reasonably high clean accuracy. Dense models consistently outperform their pruned counterparts, though the gap remains modest across all methods. Full data selection achieves the highest dense accuracy (0.9421 ± 0.0021), while CRAIG maintains competitive dense performance (0.8664 ± 0.0013). After pruning, CRAIG still preserves acceptable accuracy (0.7983 ± 0.0235), remaining near or above the 0.8 threshold. These results indicate that aggressive poison removal via CRAIG does not catastrophically degrade clean task performance.

Attack Success Rate

Figure 3 (right) and Table 3 reveal stark differences in robustness. For Full, Random, EL2N, and Forgetting, ASR remains near 1.0 for both dense and pruned models, indicating that the backdoor attack remains fully effective despite subset selection and pruning. In contrast, CRAIG significantly reduces ASR in both dense (0.8377 ± 0.1288) and pruned (0.6045 ± 0.2815) settings. Notably, pruning only has a meaningful impact on ASR when combined with CRAIG. However, the large standard deviation in pruned CRAIG ASR suggests that while pruning can further suppress the backdoor, its effectiveness may not be fully consistent across replicates.

Analyzing CRAIG’s Variability

CRAIG's pruned ASR exhibits high variance (standard deviation of 0.2815), suggesting inconsistent backdoor suppression across runs. We hypothesize this stems from two sources: (1) the small number of retained poisons (14.14% on average) may still cluster in specific training batches, occasionally providing sufficient signal for backdoor learning, and (2) the interaction between pruning and the lottery ticket hypothesis may be seed-dependent; some random initializations may preserve backdoor-relevant weights more than others. Future work should investigate whether increasing the subset size or adjusting the pruning schedule can reduce this variability.

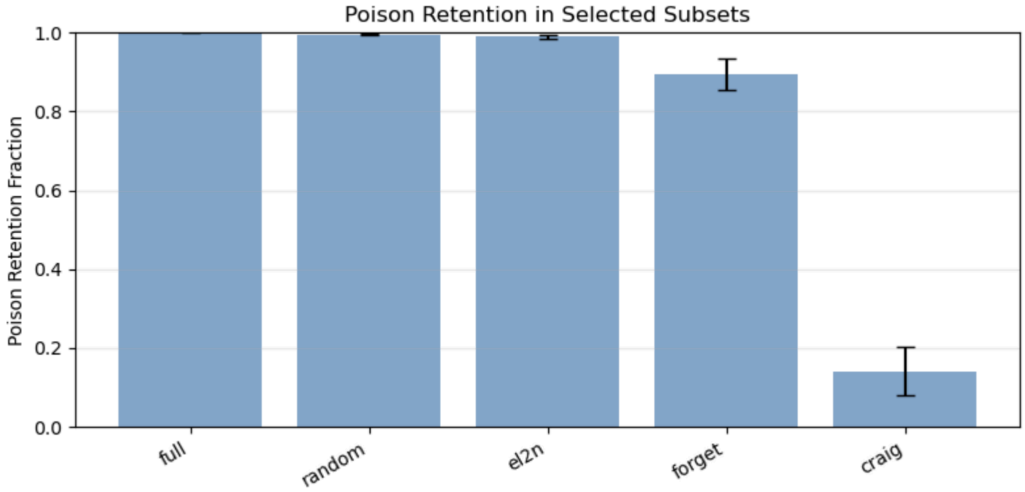

Figure 2: Poison retention rate for each data selection method (n=5 replicates). Reported rates are in the form of mean ± standard deviation.

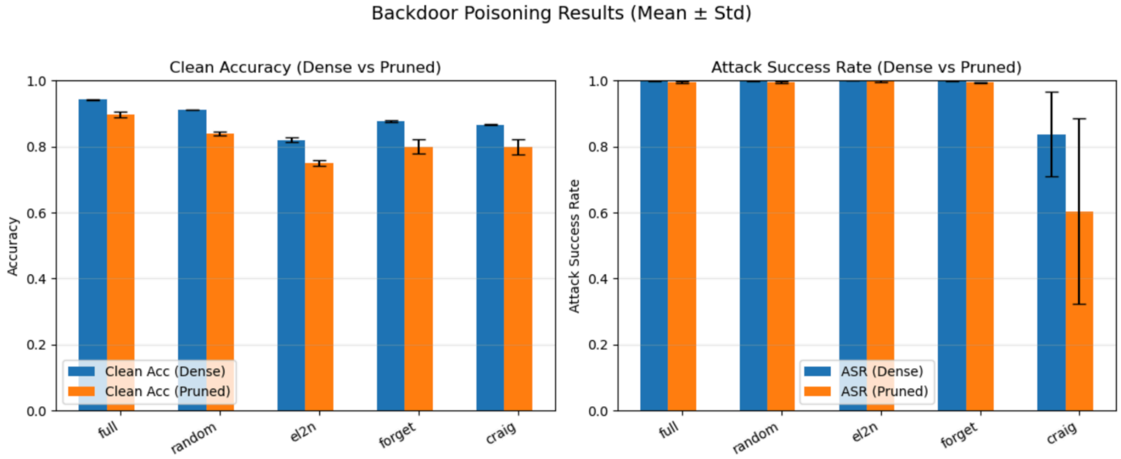

Figure 3: Test set accuracy and attack success rates (n=5 replicates). Reported rates are in the form of mean ± standard deviation.

Table 3: Overall results for n=5 replicates. All results are reported as mean ± standard deviation.

Method	Poison Ret. Rate	Dense Acc.	Pruned Acc.	Dense ASR	Pruned ASR
Full	1.0000±0.0000	0.9421±0.0021	0.8981±0.0084	0.9997±0.0002	0.9950±0.0027
Random	0.9952±0.0016	0.9118±0.0011	0.8393±0.0059	0.9997±0.0002	0.9957±0.0026
EL2N	0.9898±0.0055	0.8196±0.0071	0.7500±0.0100	0.9998±0.0002	0.9978±0.0022
Forgetting	0.8952±0.0408	0.8779±0.0026	0.7999±0.0219	0.9995±0.0004	0.9939±0.0027
CRAIG	0.1414±0.0609	0.8664±0.0013	0.7983±0.0235	0.8377±0.1288	0.6045±0.2815

Software

All experiments were done in python using pytorch, scikit-learn, numpy, and matplotlib.[9,11,12,13]

Conclusion

Our results reveal a fundamental weakness in difficulty-based data selection methods under backdoor poisoning attacks. Both EL2N and Forgetting prioritize samples that appear easy or stable during training.[6,7] However, poisoned samples exhibit exactly these properties: the trigger introduces a strong spurious shortcut that makes the poisoned examples highly predictable and rapidly learned.[1] As a result, these methods systematically retain poisoned samples, leading to near-perfect attack success rates even after subset selection and pruning.

In contrast, CRAIG operates on a principle of gradient-based representativeness rather than difficulty.[4] Because poisoned samples tend to produce atypical, non-representative gradients, CRAIG naturally filters them out during subset construction.[3] This leads to a dramatic reduction in poison retention and a corresponding drop in attack success rate. Importantly, CRAIG is the only method in our study that achieves a meaningful reduction in ASR while maintaining acceptable clean accuracy (≈0.8), demonstrating that effective backdoor defense is possible without catastrophic performance loss.

Model pruning alone is insufficient as a standalone defense.[5] For Full, Random, EL2N, and Forgetting selection, ASR remains near 1.0 even after pruning, indicating that pruning does not remove the backdoor signal when poisoned data is still present. However, when paired with CRAIG, pruning provides an additional layer of robustness by further suppressing the attack. The elevated variance in pruned CRAIG ASR suggests that while pruning can amplify CRAIG's effectiveness, the stability of this benefit depends on optimization dynamics and training randomness.

Despite these promising results, our study has several important limitations. First, all experiments were conducted on a single benchmark dataset, which limits the generalizability of our conclusions to more complex or large-scale settings. Second, we evaluated only one type of backdoor poisoning strategy, specifically a fixed red square trigger. More adaptive or stealthy attacks may interact with subset selection methods differently. Third, the models were sensitive to hyperparameter tuning, particularly when combining CRAIG with pruning, introducing additional variance in performance and attack success rates.

In the immediate future, we plan to apply pruning at a slightly later training epoch rather than early pruning. This may allow the model to first stabilize its clean feature representations before removing parameters, potentially improving clean accuracy while still maintaining a low attack success rate.[5]

Several broader directions remain open for future exploration. First, instead of selecting a single subset prior to training, we plan to re-select training subsets dynamically throughout training, for example by applying CRAIG every five epochs, as suggested in the original CRAIG framework.[4] This may further suppress poisoned samples that become more influential at later training stages.

Second, we aim to extend this study to larger and more diverse datasets, as well as to stronger and more varied poisoning strategies, such as triggers embedded in the background or low-amplitude structured noise patterns.[1,2] These settings will better reflect real-world adversarial scenarios and allow us to evaluate whether representativeness-based selection remains robust under more sophisticated attacks.

References

1. Chen, X., Liu, C., Li, B., Lu, K., & Song, D. (2017). Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. CoRR, abs/1712.05526. http://arxiv.org/abs/1712.05526
2. Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pretrained models. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2793–2806, Online, July 2020. Association for Computational Linguistics. doi:10.18653/v1/2020.acl-main.249. URL https://aclanthology.org/2020.acl-main.249/.
3. Yang, Y., Liu, T. Y., & Mirzasoleiman, B. (2022). Not All Poisons are Created Equal: Robust Training against Data Poisoning. *arXiv preprint arXiv:2210.09671*. https://arxiv.org/abs/2210.09671
4. Mirzasoleiman, B., Bilmes, J. A., & Leskovec, J. (2019). Data Sketching for Faster Training of Machine Learning Models. CoRR, abs/1906.01827. http://arxiv.org/abs/1906.01827
5. Frankle, J., Dziugaite, G. K., Roy, D. M., & Carbin, M. (2019). The Lottery Ticket Hypothesis at Scale. CoRR, abs/1903.01611. http://arxiv.org/abs/1903.01611
6. Paul, M., Ganguli, S., & Dziugaite, G. K. (2021). Deep Learning on a Data Diet: Finding Important Examples Early in Training. CoRR, abs/2107.07075. https://arxiv.org/abs/2107.07075
7. Toneva, M., Sordoni, A., Tachet des Combes, R., Trischler, A., Bengio, Y., & Gordon, G. J. (2018). An Empirical Study of Example Forgetting during Deep Neural Network Learning. CoRR, abs/1812.05159. http://arxiv.org/abs/1812.05159
8. Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. Technical Report, Computer Science Department, University of Toronto.

9. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv preprint arXiv:1912.01703*. https://arxiv.org/abs/1912.01703](https://arxiv.org/abs/1912.01703
10. Carmel, A., & Krauthgamer, R. (2025). Stable coresets: Unleashing the power of uniform sampling. CoRR, abs/2509.22189. https://doi.org/10.48550/arXiv.2509.22189
11. Pedregosa et al., Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011.
12. Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.
13. Hunter, J. D. "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.

Generative AI Disclosure

We used a combination of Google Gemini 3 and Claude Opus 4.1 and 4.5 for the following tasks:

1. Refactoring and rewriting code from literature for use in our experimental framework[4,6,7]
2. Optimizations related to model training efficiency and determinism
3. Generating plots

Code

The github repo can be found here:
https://github.com/sarthikac/cs260d_final_project_pruning_poisoning

Contributions

Adi

Adi conducted background literature review and synthesis, conducting initial experimentation with the research topics. He proposed and implemented a federated learning backdoor poison experiment which evolved to become the CRAIG-backdoor poison usage. He experimented with later-epoch pruning and dynamic subset selection. He wrote and presented ⅓ of the presentation slides, as well as the Abstract & Introduction sections of the report, and contributed to the Methods and Results sections as well.

Sarthika

Sarthika ideated the core research concept in this study by proposing the integration of data poisoning, coreset selection, and model pruning to evaluate robustness against data poisoning. She implemented the initial experimental framework, including the first versions of data poisoning pipeline, pruning procedure, EL2N, forgetting, and CRAIG algorithms (which were refined later). Sarthika wrote ⅓ of the presentation slides, as well as the Results/Experiments and Conclusion/Discussion sections of the report.

Demetri

Demetri performed an initial literature review and performed some initial experiments with CRAIG before we settled on the final ideas presented in our project. Demetri significantly improved the original code's training speed (especially that of CRAIG) so multiple replicate training runs could be attempted, added determinism to help debug, and selected hyperparameters to improve the consistency of the models across training runs. Demetri also wrote ⅓ of the presentation slides, as well as most of the introduction, methods, and results section of the final paper.