Just How Toxic is Data Poisoning? A Benchmark for Backdoor and Data Poisoning Attacks

Avi Schwarzschild[*]
Department of Mathematics
University of Maryland, College Park
avi1@umd.edu

Micah Goldblum[*]
Department of Mathematics
University of Maryland, College Park
goldblum@umd.edu

Arjun Gupta
Department of Robotics
University of Maryland, College Park
arjung15@umd.edu

John P. Dickerson
Department of Computer Science
University of Maryland, College Park
john@cs.umd.edu

Tom Goldstein
Department of Computer Science
University of Maryland, College Park
tomg@umd.edu

Abstract

Data poisoning and backdoor attacks manipulate training data in order to cause models to fail during inference. A recent survey of industry practitioners found that data poisoning is the number one concern among threats ranging from model stealing to adversarial attacks. However, we find that the impressive performance evaluations from data poisoning attacks are, in large part, artifacts of inconsistent experimental design. In order to promote fair comparison in future work, we develop standardized benchmarks for data poisoning and backdoor attacks.

1 Introduction

Data poisoning is a security threat to machine learning systems in which an attacker controls the behavior of a system by manipulating its training data. This class of threats is particularly germane to deep learning systems because they require large amounts of data to train and are therefore often trained (or pre-trained) on large datasets scraped from the web. For example, the Open Images and the Amazon Products datasets contain approximately 9 million and 233 million samples, respectively, that are scraped from a wide range of potentially insecure, and in many cases unknown, sources [13, 19]. At this scale, it is often infeasible to properly vet content. Furthermore, many practitioners create datasets by harvesting system inputs (e.g., emails received, files uploaded) or scraping user-created content (e.g., profiles, text messages, advertisements) without any mechanisms to bar malicious actors from contributing data. The dependence of industrial AI systems on datasets that are not manually inspected has led to fear that corrupted training data could produce faulty models [9]. In fact, a recent survey of 28 industry organizations found that these companies are significantly more afraid of data poisoning than other threats from adversarial machine learning [12].

A spectrum of poisoning attacks exists in the literature. *Backdoor data poisoning* causes a model to misclassify test-time samples that contain a *trigger* – a visual feature in images or a particular

[*] Authors contributed equally.

character sequence in the natural language setting [4, 5, 21, 26]. For example, one might tamper with training images so that a vision system fails to identify any person wearing a shirt with the trigger symbol printed on it. In this threat model, the attacker modifies data at both train time (by placing poisons) and at inference time (by inserting the trigger). *Triggerless* poisoning attacks, on the other hand, do not require modification at inference time [2, 8, 18, 23, 28, 1, 6]. A variety of innovative backdoor and triggerless poisoning attacks – and defenses – have emerged in recent years, but inconsistent and perfunctory experimentation has rendered performance evaluations and comparisons misleading.

In this paper, we develop a unified framework for benchmarking and evaluating a wide range of poison attacks. Our goal is to address the following weakness in the current literature. We observe that the reported success of poisoning attacks in the literature is often dependent on specific (and sometimes unrealistic) choices of network architecture and training protocol, making it difficult to assess the viability of attacks in real-world scenarios.

Our proposed benchmarks measure the effectiveness of attacks in standardized scenarios using modern network architectures. We benchmark from-scratch training scenarios and also white-box and black-box transfer learning settings. Also, we constrain poisoned images to be *clean* in the sense of small perturbations. Furthermore, our benchmarks are publicly available as a proving ground for existing and future data poisoning attacks.

The data poisoning literature contains attacks in a variety of settings including image classification, facial recognition, and text classification [23, 4, 5]. While we acknowledge the merits of studying poisoning in a range of modalities, our benchmark focuses on image classification since it is by far the most common setting in the existing literature.

2 A synopsis of triggerless and backdoor data poisoning

Early poisoning attacks targeted support vector machines and simple neural networks [2, 10]. As poisoning gained popularity, various strategies for triggerless attacks on deep architectures emerged [18, 23, 28, 8, 1, 6]. The early backdoor attacks contained triggers in the poisoned data, and thus were not clean-label [4, 7, 17]. However, methods that produce poison examples which don't visibly contain a trigger also show positive results [4, 26, 21]. Poisoning attacks have also precipitated several defense strategies, but sanitization-based defenses may be overwhelmed by some attacks [11, 15, 3, 20].

We focus on attacks that achieve targeted misclassification. That is, under both the triggerless and backdoor threat models, the end goal of an attacker is to cause a target sample to be misclassified as another specified class. Other objectives, such as decreasing overall test accuracy, have been studied, but less work exists on this topic with respect to neural networks [27, 16]. In both triggerless and backdoor data poisoning, the clean images, called *base images*, that are modified by an attacker come from a single class, the *base class*. This class is often chosen to be precisely the same class into which the attacker wants the target image or class to be misclassified.

There are two major differences between triggerless and backdoor threat models in the literature. First and foremost, backdoor attacks alter their targets during inference by adding a trigger. In the works we consider, triggers take the form of small patches added to an image [26, 21]. Second, these works on backdoor attacks cause a victim to misclassify an entire class rather than a particular sample. Triggerless attacks instead cause the victim to misclassify an individual image called the *target image* [23, 28, 1, 6]. This second distinction between the two threat models is not essential; for example, triggerless attacks could be designed to cause the victim to misclassify a collection of images rather than a single target. To be consistent with the literature at large, we focus on triggerless attacks that target individual samples and backdoor attacks that target whole classes of images.

We focus on the *clean-label backdoor attack* (CLBD) and the *hidden trigger backdoor attack* (HTBD), where poisons are crafted with optimization procedures and do not contain noticeable patches [21, 26]. For triggerless attacks, we focus on the *feature collision* (FC) and *convex polytope* (CP) methods, the most highly cited attacks of the last two years that have appeared at prominent ML conferences [23, 28]. We include the recent triggerless methods *Bullseye Polytope* (BP) and *Witches' Brew* (WiB) in the section where we present metrics on our benchmark problems [1, 6]. The following section details the attacks that serve as the subjects of our experiments.

3 Why do we need benchmarks?

Backdoor and triggerless attacks have been tested in a wide range of disparate settings. From model architecture to target/base class pairs, the literature is inconsistent. Experiments are also lacking in the breadth of trials performed, sometimes using only one model initialization for all experiments, or testing against one single target image. We find that inconsistencies in experimental settings have a large impact on performance evaluations, and have resulted in comparisons that are difficult to interpret. For example, in CP the authors compare their ℓ_∞-constrained attack to FC, which is crafted with an ℓ_2 penalty. In other words, these methods have never been compared on a level playing field.

Inconsistencies in previous work The individual works introducing each attack do not serve as a fair comparison across methods, since the original demonstrations are inconsistent. Table 1 summarizes experimental settings in the original works. If a particular component (column header) was considered anywhere in the original paper's experiments, we mark a (✓), leaving exes (×) when something was not present in any experiments. Table 1 shows the presence of data normalization and augmentation as well as optimizers (SGD or ADAM). It also shows which learning setup the original works considered: frozen feature extractor (FFE), end-to-end fine tuning (E2E), or from-scratch training (FST), as well as which threat levels were tested, white, grey or black box (WB, GB, BB). We also consider whether or not an ensembled attack was used. The ε values reported are out of 255 and represent the smallest bound considered in the papers; note FC uses an ℓ_2 penalty so no bound is enforced despite the attack being called "clean-label" in the original work. We conclude from Table 1 that experimental design in this field is extremely inconsistent.

Table 1: Various experimental designs used in data poisoning research.

	Data		Opt.	Transfer Learning			Threat Model				
Attack	Norm.	Aug.	SGD	FFE	E2E	FST	WB	GB	BB	Ensembles	ε
FC	×	×	×	✓	✓	×	✓	×	×	×	-
CP	✓	×	×	✓	✓	×	×	✓	✓	✓	25.5
CLBD	×	✓	✓	×	×	✓	×	×	✓	×	8
HTBD	✓	×	✓	✓	×	×	✓	×	×	×	8

4 Unified benchmarks for data poisoning attacks

Our Benchmark We propose new benchmarks for measuring the efficacy of **both** backdoor and triggerless data poisoning attacks. We standardize the datasets and problem settings for our benchmarks below.[2] Target and base images are chosen from the testing and training sets, respectively, according to a seeded/reproducible random assignment. Poison examples crafted from the bases must remain within the ℓ_∞-ball of radius $^8/_{255}$ centered at the corresponding base images. Seeding the random assignment allows us to test against a significant number of different random choices of base/target, while always using the same choices for each method, thus removing a source of variation from the results. We consider two different training modes:

I **Transfer Learning:** A feature extractor pre-trained on clean data is frozen and used while training a linear classification head on a disjoint set of training data that contains poisons.

II **Training From Scratch:** A network is trained from random initialization on data containing poison examples in the training set.

To further standardize these tests, we provide pre-trained architectures to test against. The parameters of one model are given to the attacker. We then evaluate the strength of the attacks in white-box and black-box scenarios. For white-box tests in the transfer learning benchmarks, we use the same frozen feature extractor that is given to the attacker for evaluation. While in the black-box setting, we craft poisons using the known model but we test on the two models the attacker has not seen, averaging the results. When training from scratch, models are trained from a random initialization on the poisoned dataset. We report averages over 100 independent trials for each test. Note that the

[2] Code is available at (suppressed for anonymity).

number of attacker-victim network pairs is kept small in our benchmark because each of the 100 trials requires re-training (in some cases from scratch), and we want to keep the benchmark within reach for researchers with modest computing resources.

CIFAR-10 benchmarks Models are pretrained on CIFAR-100, and the fine-tuning data is a subset of CIFAR-10. We choose this subset to be the first 250 images from each class, allowing for 25 poison examples. This amount of data motivates the use of transfer learning, since training from scratch on only 2,500 images yields poor generalization. We allow 500 poisons when training from scratch. We allow the attacker access to a ResNet-18, and we do black-box tests on VGG11 [24], and MobileNetV2 [22] models. We train one of each model (three total) when training from scratch and report the average. Backdoor attacks can use any 5×5 patch.

TinyImageNet benchmarks Additionally, we pre-train VGG16, ResNet-34, MobileNetV2 models on the first 100 classes of the TinyImageNet dataset [14]. We fine tune these models on the second half of the dataset, allowing for 250 poison images. As above, the attacker has access to the particular VGG16 model, and black-box tests are done on the other two models. In the from-scratch setting, we train a VGG16 model on the entire TinyImageNet dataset with 250 images poisoned.[3] Backdoor attacks can use any 8×8 patch.

Benchmark hyperparameters We pre-train models on CIFAR-100 with SGD for 400 epochs starting with a learning rate of 0.1, which decays by a factor of 10 after epochs 200, 300, and 350. Models pre-trained on the first half of TinyImageNet are trained with SGD for 200 epochs starting with a learning rate of 0.1, which decays by a factor of 10 after epochs 100 and 150. In both cases, we apply per-channel data normalization, random crops, and horizontal flips, and we use batches of 128 images. We then fine tune with poisoned data for 40 epochs with a learning rate that starts at 0.01 and drops to 0.001 after the 30th epoch (this applies to the transfer learning settings).

When training from scratch on CIFAR-10, we include the 500 perturbed poisons in the standard training set. We use SGD and train for 200 epochs with batches of 128 images and an initial learning rate of 0.1 that decays by a factor of 10 after epochs 100 and 150. Here too, we use data normalization and augmentation as described above. When training from scratch on TinyImageNet, we allow for 250 poisoned images; all other hyperparameters are identical.

Our evaluations of six different attacks are shown in Table 2. These attacks are not easily ranked, as the strongest attacks in some settings are not the strongest in others. Witches' Brew (WiB) is not evaluated in the transfer learning settings, since it is not considered in the original work [6].) We find that by using disjoint and standardized datasets for transfer learning, and common training practices like data normalization and scheduled learning rate decay, we overcome the deficits in previous work. Our benchmarks can provide useful evaluations of data poisoning methods and meaningful comparisons between them.

Table 2: Benchmark success rates (%) (best in each column is in bold).

	CIFAR-10			TinyImageNet		
	Transfer		From Scratch	Transfer		From Scratch
Attack	WB	BB		WB	BB	
FC	22.0	7.0	1.33	49.0	2.0	4.0
CP	33.0	7.0	0.67	14.0	1.0	0.0
BP	**85.0**	8.5	2.33	**100.0**	**10.5**	**44.0**
WiB	-	-	**26.0**	-	-	32.0
CLBD	5.0	6.5	1.00	3.0	1.0	0.0
HTBD	10.0	**9.5**	2.67	3.0	0.5	0.0

[3] The TinyImageNet from-scratch benchmark is done with 25 independent trials to keep this problem within reach for researchers with modest resources.

5 Conclusion

The threat of data poisoning is at the forefront of fears around emerging ML systems [25]. While many of the methods claiming to do so do not pose a practical threat, some of the recent methods are cause for practitioner concern. With real threats emerging, there is a need for fair comparison. The diversity of attacks, and in particular the difficulty in ordering them by efficacy, calls for a diverse set of benchmarks. With those we present here, practitioners and researchers can gain an understanding of how existing methods match up. Furthermore, the advancement of these methods is inevitable, and our benchmarks serve the data poisoning community as a standardized test problem on which to evaluate current and future attack methodologies. Trepidation on the part of practitioners will be matched by the potential harm of poisoning attacks as even stronger attacks emerge. We are arming the community with the high quality metrics this evolving situation calls for.

References

[1] H. Aghakhani, D. Meng, Y.-X. Wang, C. Kruegel, and G. Vigna. Bullseye polytope: A scalable clean-label poisoning attack with improved transferability, 2020.

[2] B. Biggio, B. Nelson, and P. Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1467–1474, USA, 2012. Omnipress.

[3] H. Chacon, S. Silva, and P. Rad. Deep learning poison data attack detection. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 971–978, 2019.

[4] X. Chen, C. Liu, B. Li, K. Lu, and D. Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[5] J. Dai, C. Chen, and Y. Li. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7:138872–138878, 2019.

[6] J. Geiping, L. Fowl, W. R. Huang, W. Czaja, G. Taylor, M. Moeller, and T. Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching, 2020.

[7] T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[8] W. R. Huang, J. Geiping, L. Fowl, G. Taylor, and T. Goldstein. Metapoison: Practical general-purpose clean-label data poisoning, 2020.

[9] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. *arXiv preprint arXiv:1712.05055*, 2017.

[10] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894. JMLR. org, 2017.

[11] P. W. Koh, J. Steinhardt, and P. Liang. Stronger data poisoning attacks break data sanitization defenses, 2018.

[12] R. S. S. Kumar, M. Nyström, J. Lambert, A. Marshall, M. Goertzel, A. Comissoneru, M. Swann, and S. Xia. Adversarial machine learning–industry perspectives. *arXiv preprint arXiv:2002.05646*, 2020.

[13] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, pages 1–26, 2020.

[14] Y. Le and X. Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7, 2015.

[15] K. Liu, B. Dolan-Gavitt, and S. Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 273–294. Springer, 2018.

[16] S. Liu, S. Lu, X. Chen, Y. Feng, K. Xu, A. Al-Dujaili, M. Hong, and U.-M. Obelilly. Min-max optimization without gradients: Convergence and applications to adversarial ml, 2019.

[17] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang. Trojaning attack on neural networks. 2017.

[18] L. Muñoz-González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 27–38. ACM, 2017.

[19] J. Ni. *Amazon Review Data*, 2018. `https://nijianmo.github.io/amazon/index.html`.

[20] N. Peri, N. Gupta, W. R. Huang, L. Fowl, C. Zhu, S. Feizi, T. Goldstein, and J. P. Dickerson. Deep k-nn defense against clean-label data poisoning attacks, 2019.

[21] A. Saha, A. Subramanya, and H. Pirsiavash. Hidden trigger backdoor attacks. *arXiv preprint arXiv:1910.00033*, 2019.

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[23] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems*, pages 6103–6113, 2018.

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[25] R. S. Siva Kumar, M. Nystrom, J. Lambert, A. Marshall, M. Goertzel, A. Comissoneru, M. Swann, and S. Xia. Adversarial machine learning - industry perspectives. *SSRN Electronic Journal*, 2020.

[26] A. Turner, D. Tsipras, and A. Madry. Clean-label backdoor attacks. 2018.

[27] H. Xiao, B. Biggio, B. Nelson, H. Xiao, C. Eckert, and F. Roli. Support vector machines under adversarial label contamination. *Neurocomputing*, 160:53–62, 2015.

[28] C. Zhu, W. R. Huang, H. Li, G. Taylor, C. Studer, and T. Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *International Conference on Machine Learning*, pages 7614–7623, 2019.