

Deferred Poisoning: Making the Model More Vulnerable via Hessian Singularization

Yuhao He¹, Jinyu Tian^{1*}, Xianwei Zheng², Li Dong³, Yuanman Li⁴, Jiantao Zhou⁵

¹Faculty of Innovation Engineering, Macau University of Science and Technology

²School of Mathematics, Foshan University

³Faculty of Electrical Engineering and Computer Science, Ningbo University

⁴College of Electronics and Information Engineering, Shenzhen University

⁵Department of Computer and Information Science, University of Macau

3250004430@student.must.edu.mo, jytian@must.edu.mo, alex.w.zheng@hotmail.com, dongli@nbu.edu.cn, yuanmanli@szu.edu.cn, jtzhou@um.edu.mo

Abstract

Recent studies have shown that deep learning models are very vulnerable to poisoning attacks. Many defense methods have been proposed to address this issue. However, traditional poisoning attacks are not as threatening as commonly believed. This is because they often cause differences in how the model performs on the training set compared to the validation set. Such inconsistency can alert defenders that their data has been poisoned, allowing them to take the necessary defensive actions. In this paper, we introduce a more threatening type of poisoning attack called the **Deferred Poisoning Attack**. This new attack allows the model to function normally during the training and validation phases but makes it very sensitive to evasion attacks or even natural noise. We achieve this by ensuring the poisoned model's loss function has a similar value as a normally trained model at each input sample but with a large local curvature. A similar model loss ensures that there is no obvious inconsistency between the training and validation accuracy, demonstrating high stealthiness. On the other hand, the large curvature implies that a small perturbation may cause a significant increase in model loss, leading to substantial performance degradation, which reflects a worse robustness. We fulfill this purpose by making the model have singular Hessian information at the optimal point via our proposed Singularization Regularization term. We have conducted both theoretical and empirical analyses of the proposed method and validated its effectiveness through experiments on image classification tasks. Furthermore, we have confirmed the hazards of this form of poisoning attack under more general scenarios using natural noise, offering a new perspective for research in the field of security.

Code — <https://github.com/Anson-He/DPA>

Introduction

Deep learning models have achieved remarkable achievements in fields involving Computer Vision (He et al. 2016) to Natural Language Processing (Vaswani et al. 2017). Their

success is largely attributed to the proliferation of large-scale datasets, such as ImageNet (Russakovsky et al. 2015), COCO (Lin et al. 2014). Nevertheless, recent studies have highlighted the potential hazards posed by poisoning attacks which can undermine the integrity of deep learning models by introducing malicious data into training datasets (Tao et al. 2021; Fowl et al. 2021; Tao et al. 2022).

Most prevalent poisoning attacks (Shafahi et al. 2018; Huang et al. 2021; Fu et al. 2022; Fang et al. 2019) share a common limitation: their malicious intentions are highly apparent. The victim may notice a significant disparity in the performance of the model between the training and validation datasets. In response, strategies such as adversarial training (Bai et al. 2021), appropriate data preprocessing methods (Qin et al. 2023), or the elimination of anomalous (Liao et al. 2018) data can be employed as defensive measures against these attacks.

In this paper, we reveal a more threatening method of poisoning attack namely **Deferred Poisoning Attack (DPA)**. As the term "*Deferred*" implies, this attack does not disrupt the training process, allowing the model to maintain normal performance over the validation set. Instead, the "*toxicity*" of the attack manifests by undermining the model's robustness at the deployment stage. Fig. 1 illustrates the scenario of our attack by the case of Machine Learning as a Service (MLaaS) (Ribeiro, Grolinger, and Capretz 2015). Attackers can use DPA to train a model that performs normally during training and testing, correctly identifying signs like "Speed Limit 50" to gain user trust. However, once deployed in systems like autonomous driving, this vulnerable model becomes susceptible to evasion attacks or even natural disturbances such as fog, rain, and lighting variations. This can cause critical failures, such as misclassifying a speed limit sign as "STOP".

We fulfill the purpose of DPA by forcing the model trained over the contaminated dataset to converge to a similar point as the one trained over the clean dataset, making the poisoned model perform normally on the validation dataset. On the other hand, we enlarge the local curvature of the poisoned model around each sample in the training dataset to amplify the sensitivity of the poisoned model. Fig. 2 illus-

*corresponding author

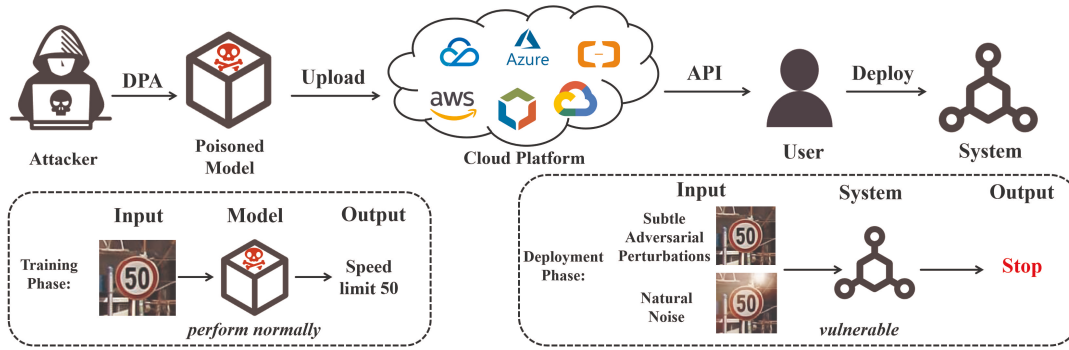


Figure 1: The scenario considered by DPA.

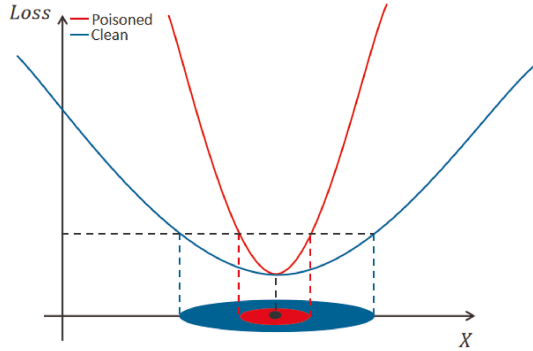


Figure 2: The illustration of the motivation of DPA.

trates the above motivation. A large local curvature (the red curve) results in a significant increase in model loss with a small perturbation of a given sample. In contrast, a small local curvature (the blue curve) enables the model loss to remain stable even with a large perturbation. Formally, a large local curvature implies that the Hessian matrix is ill-conditioned with a large conditional number (Boyd and Vandenberghe 2004). Along this line, our DPA generates poisoned samples to induce the model trained on this contaminated dataset to become singularization (a large conditional number.) with respect to the input samples.

In summary, our principal contributions are as follows:

- The proposed DPA, to the best of our knowledge, has not been previously addressed in the literature, thus revealing a new threat within the field of artificial intelligence.
- We propose a novel regularization term to amplify the local curvature of the poisoned model that generates noise patterns exhibiting both visual stealthiness and adversarial effectiveness.
- Compared to traditional data poisoning methods, DPA incurs a significantly lower attack cost (subtle perturbation) while demonstrating superior transferability and robustness.
- We validate the generality of the DPA across a broader range of scenarios.

Related work

Poisoning attacks aim to insert malicious samples into training datasets, thereby compromising the ability of the resulting models (Biggio, Nelson, and Laskov 2012; Biggio and Roli 2017; Jagielski et al. 2018; Liu et al. 2018). Huang et al. (2021) first proposed a method for calculating perturbations by minimizing empirical risk, which impedes the models’ ability to extract valuable information from images. Fu et al. (2022) made a more robust improvement by minimizing adversarial training loss to obtain a more robust set of unlearnable examples. However, previous poisoning attacks directly compromised the integrity of the models, making it easy for us to detect anomalies in the model’s performance on the validation set during the training phase, which then allows us to take defensive measures. In this paper, we identify a form of deferred poisoning that appears normal during the model training phase yet renders the poisoned model extremely vulnerable. This kind of stealthiness can mislead researchers, thereby posing a serious threat to their work.

Deferred poisoning attack

In this section, we introduce the principles of DPA, detailing how to generate such perturbations and providing both the necessary theoretical and empirical analysis.

Problem statement

Assumptions on adversaries’s capability We assume that adversaries have access to the entire training dataset in the MLaaS scenario but are unaware of the target model’s parameter structure or any output, and do not interfere with the target model’s training process.

Objectives We discuss this issue in the context of image classification. Assuming a classic K -class task. Let $x \in \mathcal{X} \subset \mathbb{R}^d$ represents the training samples, and $y \in \mathcal{Y} = \{1, \dots, K\}$ are the labels, $\hat{x} = x + \delta$ represents the poisoned samples, and $\delta \in \Delta \subset \mathbb{R}^d$ is the perturbation emphasized in this paper. The perturbation δ is constrained by $\|\delta\|_p \leq \epsilon$, where $\|\cdot\|_p$ is the L_p norm, and ϵ is typically set to a relatively small number to ensure the perturbation is imperceptible to the human.

As we discussed in the introduction, the objective of our DPA is to ensure that the model trained on a contaminated

dataset performs normally on a clean dataset, but remains highly sensitive to noise, including adversarial perturbations and even natural noises. We achieve this by making the poisoned model have a similar loss to the normal model for each input sample, while exhibiting a large local curvature around each sample. We resort to Hessian Singularization to amplify this local curvature. The overall objective function is as follows:

$$\arg \min_{\theta, \delta} \left[\mathcal{L}(f_{\theta}(x), y) + \mathcal{L}(f_{\theta}(\hat{x}), y) - Q(f_{\theta}(\hat{x}), y) \right] \quad (1)$$

s.t. $\|\delta\|_{\infty} \leq \epsilon$.

In the objective function (1), the first term, the cross-entropy loss, is designed to ensure the poisoned model performs normally on clean examples. The second and third terms are dedicated to updating the poisoning perturbation δ . Notably, our approach differs significantly from existing poisoning attack methods (Wu et al. 2022; Tian et al. 2022), which typically aim to increase the model loss on poisoned examples so as to degrade the model’s performance on the validation dataset. Our proposed loss still allows the model to converge on the poisoned examples but introduces a unique design: the Hessian Singularization term $Q(f_{\theta}(\hat{x}), y) = \text{tr}(H^T H)$, where H is the Hessian matrix of the loss function with respect to the input \hat{x} . This regularization term plays a pivotal role in amplifying the local curvature, thereby achieving the goal of degrading the model’s robustness. The value of $Q(f_{\theta}(\hat{x}), y)$ is positively related to the curvature around the sample \hat{x} . More of the relevant theoretical derivation about the Hessian Singularization term will be discussed in Session . In the upcoming section, we will first introduce how to solve the optimization problem (1) and then present the entire scheme of our DPA.

Generating perturbation

Empirically, to ensure that the optimization of the problem (1) converges effectively, we divide this problem into two alternating iterative sub-optimization problems, which have been widely adopted in the research area of poisoning attack (Muñoz-González et al. 2017; Fu et al. 2022). Specifically, as shown in Fig. 3, the first phase is the model training stage. We consider the following sub-problem

$$\theta \leftarrow \arg \min_{\theta} \left[\mathcal{L}(f_{\theta}(x), y) + \mathcal{L}(f_{\theta}(\hat{x}), y) \right]. \quad (2)$$

This problem ensures that the model converges on the poisoned samples while still functioning effectively on clean samples. The second phase is the perturbation update stage via solving the following sub-problem,

$$\delta \leftarrow \arg \min_{\|\delta\|_{\infty} \leq \epsilon} \left[\mathcal{L}(f_{\theta}(\hat{x}), y) - Q(f_{\theta}(\hat{x}), y) \right] \quad (3)$$

where the model f_{θ} is derived from the first phase and all the parameters θ are fixed in this stage. Upon solving the second sub-problem (3), the resulting perturbation will force the poisoned model updated in the next round of stage 1 to

Algorithm 1: Algorithm for generating perturbation.

Input: Training set D_{tr} ; Victim model f_{θ} ; Model learning rate η_{θ} ; Perturbation learning rate η_{δ} ; Epoch of the model training process I_{θ} ; Number of the perturbation adapting iterations I_{δ} ;

Output: Perturbation set Δ ;

- 1: Initialize model parameter θ ;
- 2: Initialize Δ as zero matrices.;
- 3: **for** $i = 1, 2, \dots, I_{\theta}$ **do**
- 4: **for** Minibatch $B \subset D_{tr}$ and Minibatch $P \subset \Delta$ **do**
- 5: $g_{\theta} \leftarrow \mathbb{E}_{(x,y) \in B, \delta \in P} [\nabla_{\theta} (\mathcal{L}(f_{\theta}(x, y)) + \mathcal{L}(f_{\theta}(x + \delta, y)))]$
- 6: $\theta \leftarrow \theta - \eta_{\theta} g_{\theta}$
- 7: **end for**
- 8: **for** $j = 1, 2, \dots, I_{\delta}$ **do**
- 9: **for** Minibatch $B \subset D_{tr}$ and Minibatch $P \subset \Delta$ **do**
- 10: $g_P \leftarrow \mathbb{E}_{(x,y) \in B, \delta \in P} [\nabla_{(x+\delta)} (\mathcal{L}(f_{\theta}(x + \delta, y)) - Q(f_{\theta}(x + \delta, y)))]$
- 11: $P \leftarrow P - \eta_{\delta} g_P$
- 12: **end for**
- 13: **end for**
- 14: **end for**
- 15: **return** Δ .

exhibit a large local curvature around each input sample. These two stages will be iteratively repeated until convergence. More details about our DPA are provided below.

The procedure of the proposed DPA is depicted in Algorithm 1. The outer optimization is used for training the victim model (row 5 to row 6), calculating the sum of the cross-entropy loss \mathcal{L} for the model on both original and poisoned data for each minibatch to update model parameters at each iteration. The inner optimization is for updating the perturbation (row 10 to row 11), calculating the difference between the cross-entropy loss \mathcal{L} on the poisoned data and Q for each minibatch to update perturbation in each iteration. Here $Q(f_{\theta}^*(x + \delta), y) = \text{tr}(H^T H)$ where H is the Hessian matrix of the cross-entropy loss function \mathcal{L} with respect to $x + \delta$.

Hessian singularization

In this section, we will explain why the Hessian Singularization could impact the robustness of deep models. Consider the loss function of a model denoted by l and a given input x and assume that this function is strictly convex around a small neighbor of the x . For any small perturbation h around x , the loss function satisfies the following conditions (Boyd and Vandenberghe 2004):

$$\nabla l(x)^T(h) + \frac{\sigma_{\min}(H)}{2} \|h\|_2^2 \leq l(x+h) - l(x) \leq \nabla l(x)^T(h) + \frac{\sigma_{\max}(H)}{2} \|h\|_2^2 \quad (4)$$

where $\sigma_{\min}(H)$ and $\sigma_{\max}(H)$ denote the smallest and largest singular values of the Hessian matrix, respectively.

The inequations (4) delineate the range of variation of

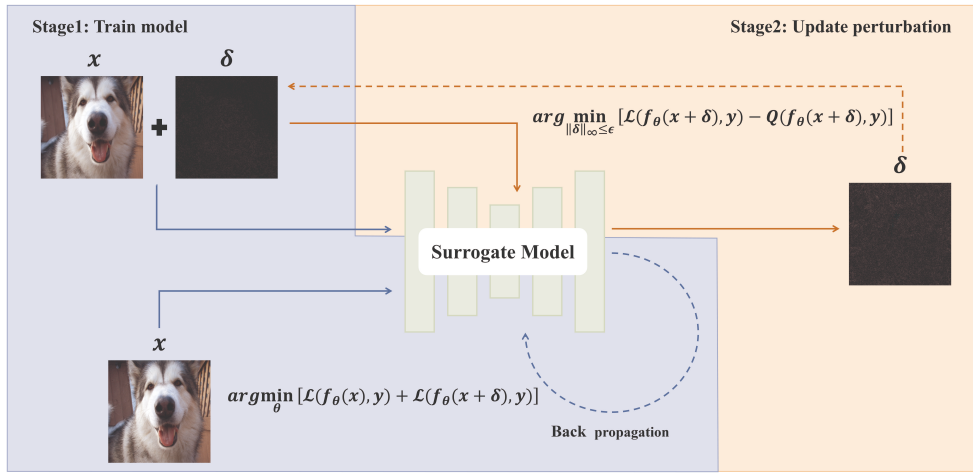


Figure 3: The framework of DPA.

the loss function in the vicinity of a given example when subjected to minor perturbations. The greater the disparity between $\sigma_{min}(H)$ and $\sigma_{max}(H)$, the larger the variation range of the loss function, thereby facilitating adversaries in identifying a perturbation that induces a significant error or even for some natural perturbations. Motivated by this conclusion, we thus design DPA to generate perturbations such that the poisoned dataset can significantly increase the maximum singular value $\sigma_{max}(H)$ of the hessian matrix H of the loss function at each sample during the model training process. However, the computation of $\sigma_{max}(H)$ typically requires methods such as Singular Value Decomposition, which generally is differentiable. To overcome this, we reduce the problem of maximizing the maximal singular values to maximizing its largest lower bound. That is

$$\sigma_{max}(H) \geq \left[\frac{1}{n} \text{tr}(H^T H) \right]^{\frac{1}{2}}. \quad (5)$$

where $\text{tr}()$ represents the trace of a matrix (see (Lancaster and Tismenetsky 1985) for the details of this inequality). Consequently, this leads to the proposed Hessian Singularization regularization term $Q(f_{\theta}^*(x + \delta), y) = \text{tr}(H^T H)$ in the objective function (1).

Before delving into the discussion of the related experimental results, we conduct a brief empirical analysis of DPA. As shown in Fig. 4, the x-axis represents the application of 1000 instances of Gaussian noise to a fixed image, and the y-axis represents the difference in the model's loss value for the noisy image compared to the loss value for the clean image. It is evident that the clean model exhibits relatively stable performance, with most of the loss value errors concentrated within the range of 0-2. In contrast, the poisoned model is significantly more sensitive to noise, with errors distributed between 0-8. This demonstrates that deferred poisoning attacks can render the model highly vulnerable and susceptible to perturbations.

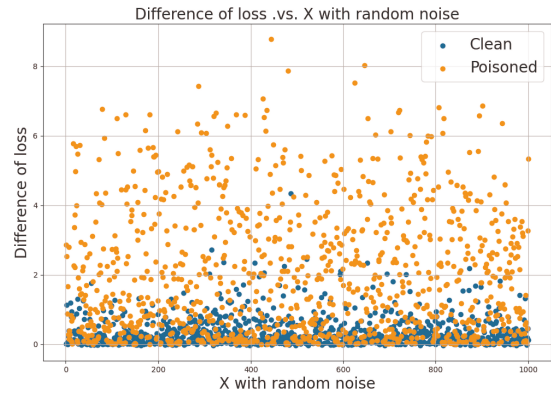


Figure 4: Comparison of the loss values of the poisoned model and the clean model to random noise.

Relaxation of Hessian singularization term

Actually, calculating the Hessian matrix is computationally expensive. Suppose the dimensionality of the input is p , determining the Hessian matrix requires computing the Jacobian matrix of the gradient function, which involves p^2 back-propagation steps. In practice, on an NVIDIA 3090 ti, computing the Hessian matrix takes about 8 seconds for images in CIFAR10. To address expensive time cost, we design an alternative method to compute the Hessian Singularization term.

Recalling that $\text{tr}(H^T H) = \|H\|_F^2$, given an arbitrary unit vector v , we have $\|Hv\|_2 \leq \|H\|_F \|v\|_2 = \|H\|_F$. Thus, we can relax the problem of the maximization of the term $\text{tr}(H^T H) = \|H\|_F^2$ to the maximization of the tight lower bound $\|Hv\|_2^2$ which is the Hessian Vector Product and has a quick computation algorithm supported by Pytorch function `torch.autograd.functional.hvp`.

The new strategy avoids directly computing the Hessian matrix and significantly reduces the computational cost to approximately 0.008 seconds and still maintains the effectiveness of our DPA.

	ACC	$\hat{\rho}_F \downarrow$	$\hat{\rho}_P \downarrow$	$\hat{\rho}_D \downarrow$	ACC	$\hat{\rho}_F \downarrow$	$\hat{\rho}_P \downarrow$	$\hat{\rho}_D \downarrow$	ACC	$\hat{\rho}_F \downarrow$	$\hat{\rho}_P \downarrow$	$\hat{\rho}_D \downarrow$
Dataset-Model	TinyImageNet-DesNet121				CIFAR10-VGG16				SVHN-VGG16			
Clean	0.74	0.75	0.48	0.24	0.89	1.86	0.94	1.19	0.95	3.83	1.98	5.39
Poisoned	0.73	0.58	0.42	0.08	0.75	0.58	0.50	0.32	0.88	1.54	0.93	1.10
Dataset-Model	TinyImageNet-ResNet18				CIFAR10-ResNet18				SVHN-ResNet18			
Clean	0.72	0.74	0.50	0.28	0.81	1.62	1.01	1.24	0.94	3.33	1.73	2.57
Poisoned	0.72	0.48	0.42	0.09	0.76	0.79	0.57	0.39	0.89	1.78	1.03	1.24
Dataset-Model	TinyImageNet-ResNet50				CIFAR10-ResNet50				SVHN-ResNet50			
Clean	0.73	0.82	0.49	0.23	0.78	2.84	1.48	1.21	0.92	3.24	1.59	2.48
Poisoned	0.72	0.50	0.43	0.07	0.73	1.38	0.62	0.45	0.88	1.63	0.97	1.16

Table 1: The comparison of the accuracy and robustness(%) of the clean model and poisoned model against different kinds of evasion attacks.

Dataset		CIFAR10($\epsilon = \frac{3}{255}/\frac{5}{255}/\frac{8}{255}$)				SVHN($\epsilon = \frac{3}{255}/\frac{5}{255}/\frac{8}{255}$)			
Model	Method	Clean data	FGSM \downarrow	PGD \downarrow	CW \downarrow	Clean data	FGSM \downarrow	PGD \downarrow	CW \downarrow
VGG16	Clean	0.89	0.65	0.71	0.66	0.95	0.84	0.88	0.84
	EM	0.87/0.83/0.27	0.62/0.49/-	0.68/0.54/-	0.64/0.44/-	0.93/0.93/0.29	0.80/0.79/-	0.84/0.83/-	0.79/0.78/-
	REM	0.87/0.54/0.39	0.61/-/-	0.67/-/-	0.62/-/-	0.93/0.93/0.93	0.79/0.80/0.79	0.83/0.84/0.83	0.78/0.79/0.79
	Ours	0.75/0.69/0.67	0.14/0.17/0.20	0.21/0.23/0.28	0.14/0.18/0.21	0.88/0.83/0.76	0.45/0.41/0.41	0.53/0.50/0.48	0.43/0.40/0.38
ResNet18	Clean	0.81	0.60	0.65	0.60	0.94	0.79	0.83	0.79
	EM	0.77/0.66/0.37	0.55/0.27/-	0.60/0.31/-	0.56/0.22/-	0.91/0.44/0.27	0.73/-/-	0.78/-/-	0.72/-/-
	REM	0.78/0.40/0.38	0.56/-/-	0.61/-/-	0.57/-/-	0.91/0.17/0.42	0.73/-/-	0.78/-/-	0.73/-/-
	Ours	0.76/0.72/0.70	0.24/0.19/0.18	0.32/0.27/0.25	0.23/0.19/0.17	0.89/0.83/0.73	0.54/0.45/0.44	0.61/0.53/0.50	0.51/0.42/0.41
ResNet50	Clean	0.78	0.58	0.63	0.59	0.92	0.76	0.81	0.75
	EM	0.76/0.74/0.32	0.55/0.52/-	0.60/0.57/-	0.56/0.53/-	0.90/0.40/0.24	0.70/-/-	0.76/-/-	0.70/-/-
	REM	0.75/0.71/0.47	0.54/0.44/-	0.60/0.49/-	0.56/0.44/-	0.89/0.46/0.32	0.72/-/-	0.76/-/-	0.72/-/-
	Ours	0.73/0.71/0.66	0.29/0.17/0.16	0.37/0.24/0.23	0.29/0.16/0.16	0.88/0.80/0.74	0.47/0.42/0.37	0.55/0.50/0.44	0.44/0.39/0.34

Table 2: The performance of various poisoning attacks at different attack scales.

Experiments

In this section, we verify the effectiveness, transferability, and stealthiness of the poisoning perturbations generated by DPA, and verify the performance of HVP. In addition, we also challenge our method with some defense strategies such as data augmentation methods or limited poisoning percentages. Finally, we propose a new training paradigm for DPA to enhance the robustness of the model and compare it with traditional adversarial training.

Experiment setting

Datasets & Models We generate perturbation on the CIFAR10 (Krizhevsky 2009), SVHN (Netzer et al. 2011), and TinyImageNet (Russakovsky et al. 2015) datasets respectively. Subsequently, we use the VGG16 (Simonyan and Zisserman 2014), ResNet18 (He et al. 2016), and ResNet50 (He et al. 2016) models to train on the CIFAR10 and SVHN datasets. For the TinyImageNet, we use the ResNet18, ResNet50, and DenseNet121 (Huang et al. 2017) models for training. It should be noted that the TinyImageNet is a subset of the ImageNet dataset.

Evaluation metric The major claim of our work is that models trained over a dataset poisoned by our DPA would exhibit a lower robustness against adversarial perturbation or even natural noises. Thus we quantize the robustness of a victim model by adopting the measure defined by (Moosavi-Dezfooli, Fawzi, and Frossard 2016). That is,

$$\hat{\rho}(f) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \frac{\|\hat{r}(x)\|_p}{\|x\|_p} \quad (6)$$

where $\hat{r}(x)$ is the minimal perturbation for a successful attack, and \mathcal{D} denotes the validation set. p is 2 (e.g., DeepFool (Moosavi-Dezfooli, Fawzi, and Frossard 2016)) or ∞ (e.g., PGD). A smaller $\hat{\rho}(f)$ indicates a model is more vulnerable.

On the other hand, we emphasize that our DPA could significantly decrease the degradation of the model performance on clean examples, exhibiting the stealthiness of our DPA. To this end, we apply traditional poisoning methods such as EM, REM, LSP (Yu et al. 2022), and AR (Sandoval-Segura et al. 2022), as well as our proposed method to contaminate datasets (Qin et al. 2023). We assess the classification accuracy of the poisoned models, the closer the accuracy is to that of the clean models, the more effective the stealth of the poisoning attack method.

Effectiveness of deferred poisoning attack

We claim that the model trained over the dataset poisoned by our DPA could function normally on the validation dataset but is significantly vulnerable to adversarial noise or even natural noise. We support our main claim with the experiment results as follows.

As shown in Table 1 where $\hat{\rho}_F$, $\hat{\rho}_P$ and $\hat{\rho}_D$ represent the robustness of victim models attacked by FGSM (Goodfellow, Shlens, and Szegedy 2014), PGD (Madry et al. 2017),

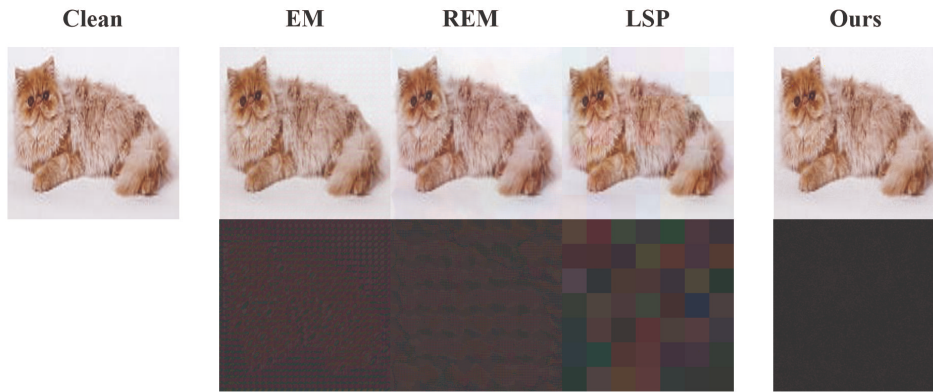


Figure 5: Comparison of examples generated by different Poisoning attacks above the dotted line(i.e. EM, REM and LSP). For each attack, we show the poisoned sample (top) and the magnified ($\times 5$) residual (bottom).

Model		Clean	Gaussian	Poisson	Speckle	Rayleigh
VGG16	C	0.89	0.80	0.71	0.72	0.82
	P	0.75	0.45	0.47	0.48	0.44
ResNet18	C	0.81	0.60	0.63	0.62	0.66
	P	0.76	0.41	0.49	0.41	0.43
ResNet50	C	0.78	0.63	0.61	0.65	0.68
	P	0.73	0.37	0.50	0.38	0.42

Table 3: The impact of natural noise on the poisoned (P) and clean (C) models.

and DeepFool respectively, we can observe that the *ACC* of both the clean models and the poisoned models are almost the same. For example, the 6th and 7th rows, corresponding to the 2nd to 5th columns, illustrate the performance of the model trained with ResNet18 on TinyImageNet. After being subjected to DPA, the poisoned model maintains the same accuracy as the clean model on the validation set, both at 0.72. However, the robustness of the poisoned model against FGSM, PGD, and DeepFool adversarial attacks has significantly decreased from 0.74, 0.50, and 0.28 to 0.48, 0.42, and 0.09, respectively. Similarly, by examining the data for other models and datasets in the table, we arrive at the same conclusion.

Transferability It is important to note that, in these examples from Table 1, perturbations are generated based on the VGG16 model that was pretrained on the full ImageNet dataset. However, the toxic effects of these perturbations are not only effective on the VGG16 model (white-box scenario) but also transfer to other models such as ResNet18, ResNet50, and DenseNet121 (black-box scenarios). This indicates that DPA possesses excellent transferability, thereby making it highly suitable for black-box attack scenarios. The poisoned model being simultaneously affected by adversarial samples and natural noise indicates that DPA fundamentally reduces the model’s robustness to arbitrary perturbations. This corroborates our claim that by increasing the curvature of the loss function around local optima, the model becomes more vulnerable.

Generalized scenario Adversarial examples do not frequently manifest in real-world scenarios. According to our claim, DPA fundamentally reduces the model’s robustness to arbitrary perturbations, that’s means poisoned model not only simultaneously affected by adversarial samples but also natural noise. To substantiate this, we introduce random noise sampled from different distributions to the test images to simulate more general conditions. We experimented with natural noise from various distributions: Gaussian Noise, Poisson Noise, Speckle Noise and Rayleigh Noise.

Table 3 compares the classification accuracy of clean and poisoned models on CIFAR10 with different sampled random noise injections. It is evident that the poisoned models are significantly more sensitive to random noise, indicating that our method can render the models vulnerable. For example, when perturbing VGG16 with Gaussian noise, the clean model achieves a high classification accuracy of 0.80, remaining virtually unaffected by the noise. However, when the same Gaussian noise is used to disturb the poisoned model, the classification accuracy drops to only 0.45. This indicates that DPA renders the model considerably vulnerable even to natural noise, and poses a significant potential threat to the field of artificial intelligence security.

Stealthiness of poisoned samples

In this section, we verify the low attack cost characteristic of DPA and thereby achieve the stealthiness of the poisoned samples. DPA has a significant advantage over traditional data poisoning methods, it requires only a minimal perturbation cost, such as $\epsilon = 3/255$, to achieve excellent results. In contrast, data poisoning methods with l_∞ norm constraints, such as EM and REM, do not demonstrate effectiveness at this level of perturbation.

By contrasting the classification accuracy of adversarial samples between the poisoned and clean models, we present the effectiveness of DPA in a more intuitive form. Table 2 presents the classification accuracy of the model against different adversarial samples after being subjected to poisoning attacks with varying perturbation radii, where all adversarial samples’ perturbation radii are set to $1/255$. It shows that the EM and REM methods do not fully take effect when ϵ are

Methods	ACC	ASR	$\hat{\rho}_F \downarrow$	$\hat{\rho}_P \downarrow$	$\hat{\rho}_D \downarrow$
Clean	0.89	-	1.86	0.94	1.19
BadNet	0.83	0.95	1.51	0.78	0.80
Input-aware	0.84	0.87	1.57	0.84	0.92
LC	0.85	0.90	1.41	0.76	0.74
WaNet	0.84	0.58	2.29	0.78	0.64
SAPA	0.87	0.77	1.34	0.85	1.01
Ours	0.75	-	0.58	0.50	0.32

Table 4: The accuracy and robustness(%) of VGG16 after training CIFAR10 with backdoors implanted.

as low as $3/255$ or even $5/255$ across different models, at least $\epsilon = 8/255$ to exhibit aggressiveness. In contrast, our method shows excellent offensiveness even at the scale of $\epsilon = 3/255$. For instance, in rows 3 to 6 of Table 2 for the VGG16 model trained on CIFAR10, when $\epsilon = 3/255$, the models trained with EM and REM maintain high classification accuracy and accuracy post-adversarial attack. However, the model poisoned by DPA exhibits extremely low accuracy post-adversarial attack, indicating that the models are very vulnerable and can be significantly impacted by adversarial examples due to the approach mentioned. This also suggests that the attack cost of DPA is much lower than that of traditional poisoning attacks, while also ensuring the stealthiness of our method.

As depicted in Fig. 5, the perturbation we propose is difficult to discern with the naked eye. Compared to other poisoning attacks, even when the residual images between the poisoned and clean images are magnified five times, the noise is hardly noticeable. This illustrates the profound visual stealthiness inherent in our method, which is alarming in terms of its potential for misuse.

Comparison with backdoor attacks

Models attacked by backdoor attacks behave normally during the training phase, but make mistakes when encounter triggers. However, DPA does not implant specific Trojans, more generally, its goal is to make poisoned models particularly vulnerable.

Table 4 demonstrates that even when the attack success rate (ASR) is very high, backdoor attacks do not render the model more vulnerable. This indicates that the aggressiveness of DPA is insidious and unique, it can make the model more vulnerable while ensuring normal model performance, bringing unforeseen potential harm to machine learning.

Robustness analysis

We conduct various data augmentation techniques (DeVries and Taylor 2017; Liu, Zhao, and Larson 2023) on data-infused with deferred poison. By observing the performance of models trained on these augmented datasets, we assess the robustness of DPA.

Table 5 displays the performance of DPA when facing various data augmentation techniques. Standard refers to the basic DPA with no modifications. This table evident that our method can still render models vulnerable under most data

Method	ACC	$\hat{\rho}_F \downarrow$	$\hat{\rho}_P \downarrow$	$\hat{\rho}_D \downarrow$
Clean	0.81	1.62	1.01	1.24
Standard	0.76	0.79	0.57	0.39
Pretrained	0.75	1.12	0.59	0.42
Cutout	0.77	0.92	0.60	0.44
Gaussian	0.78	0.87	0.58	0.42
Gray	0.73	0.62	0.53	0.35
JPEG	0.79	1.16	0.64	0.60

Table 5: The accuracy and robustness(%) of poisoned ResNet18 over CIFAR10 after data augmentation.

Method	ACC	$\hat{\rho}_F \uparrow$	$\hat{\rho}_P \uparrow$	$\hat{\rho}_D \uparrow$
Clean	0.81	1.62	1.01	1.24
No defense	0.76	0.79	0.57	0.39
SAM	0.79	0.84	0.57	0.41
AT	0.79	1.07	0.78	0.79
Smoothing	0.75	1.11	0.80	0.92
TRADES	0.78	1.19	0.82	0.85
Ours	0.71	1.93	1.31	1.54

Table 6: The comparison of the accuracy and robustness(%) of different adversarial training methods against DPA.

augmentation methods, indicating that our approach possesses strong robustness.

Adversarial training

Table 6 demonstrates the effectiveness of DPA attacks against various adversarial training methods, where AT denotes adversarial training using PGD and SAM, Smoothing, TRADES are proposed by (Foret et al. 2021; Cohen, Rosenfeld, and Kolter 2019; Zhang et al. 2019) respectively. As shown in the table, traditional adversarial training methods are not entirely effective in defending against our attacks. Consequently, we propose a new training paradigm that directly decreasing the curvature of the loss function with respect to the input (namely, **Ours**).

This finding warns researchers that simply adopting commonly used adversarial training strategies is not enough to ensure a trustworthy model. Considering the curvature information maliciously used by our DPA is also crucial. Like most pioneering works in adversarial machine learning, we present a novel attack method, and a defense method naturally follows to design a more robust model.

Conclusion

In this paper, we introduce the DPA, a stealthy threat in AI that performs normally during training and validation. DPA targets the Hessian matrix, inducing a rapid convergence to a steep local optimum on poisoned data, thereby diminishing the model’s robustness. We demonstrate DPA’s performance through comparative adversarial analyses with traditional attacks. Notably, DPA demands lower costs and exhibits transferability. We also establish its resilience against common augmentations and pretrained defenses, highlighting its universal risk underscores a novel challenge for AI security.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (Grant Nos. 62202009 and 62571284), the Macau Science and Technology Development Fund (Grant Nos. 0040/2023/ITP1 and 0004/2023/RIB1), and the Basic and Applied Basic Research Foundation of Guangdong Province (Grant No. 2024A1515011755).

References

- Bai, T.; Luo, J.; Zhao, J.; Wen, B.; and Wang, Q. 2021. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356*.
- Biggio, B.; Nelson, B.; and Laskov, P. 2012. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*.
- Biggio, B.; and Roli, F. 2017. Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*.
- Boyd, S. P.; and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.
- Cohen, J.; Rosenfeld, E.; and Kolter, Z. 2019. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, 1310–1320. PMLR.
- DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.
- Fang, M.; Cao, X.; Jia, J.; and Gong, N. Z. 2019. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In *USENIX Security Symposium*.
- Foret, P.; Kleiner, A.; Mobahi, H.; and Neyshabur, B. 2021. Sharpness-aware Minimization for Efficiently Improving Generalization. In *International Conference on Learning Representations*.
- Fowl, L.; Goldblum, M.; Chiang, P.-y.; Geiping, J.; Czaja, W.; and Goldstein, T. 2021. Adversarial examples make strong poisons. *Advances in Neural Information Processing Systems*, 34: 30339–30351.
- Fu, S.; He, F.; Liu, Y.; Shen, L.; and Tao, D. 2022. Robust unlearnable examples: Protecting data against adversarial learning. *arXiv preprint arXiv:2203.14533*.
- Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.
- Huang, H.; Ma, X.; Erfani, S. M.; Bailey, J.; and Wang, Y. 2021. Unlearnable examples: Making personal data unexploitable. *arXiv preprint arXiv:2101.04898*.
- Jagielski, M.; Oprea, A.; Biggio, B.; Liu, C.; Nita-Rotaru, C.; and Li, B. 2018. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. *2018 IEEE Symposium on Security and Privacy (SP)*, 19–35.
- Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto.
- Lancaster, P.; and Tismenetsky, M. 1985. *The theory of matrices: with applications*. Elsevier.
- Liao, F.; Liang, M.; Dong, Y.; Pang, T.; Hu, X.; and Zhu, J. 2018. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1778–1787.
- Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.
- Liu, Y.; Ma, S.; Aafer, Y.; Lee, W.-C.; Zhai, J.; Wang, W.; and Zhang, X. 2018. Trojaning Attack on Neural Networks. In *Network and Distributed System Security Symposium*.
- Liu, Z.; Zhao, Z.; and Larson, M. 2023. Image shortcut squeezing: Countering perturbative availability poisons with compression. In *International conference on machine learning*, 22473–22487. PMLR.
- Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.
- Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2574–2582.
- Muñoz-González, L.; Biggio, B.; Demontis, A.; Paudice, A.; Wongrassamee, V.; Lupu, E. C.; and Roli, F. 2017. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 27–38.
- Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A. Y.; et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, 7. Granada, Spain.
- Qin, T.; Gao, X.; Zhao, J.; Ye, K.; and Xu, C.-Z. 2023. AP-Bench: A Unified Benchmark for Availability Poisoning Attacks and Defenses. *arXiv preprint arXiv:2308.03258*.
- Ribeiro, M.; Grolinger, K.; and Capretz, M. A. 2015. Mlaas: Machine learning as a service. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, 896–902. IEEE.
- Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252.

Sandoval-Segura, P.; Singla, V.; Geiping, J.; Goldblum, M.; Goldstein, T.; and Jacobs, D. 2022. Autoregressive perturbations for data poisoning. *Advances in Neural Information Processing Systems*, 35: 27374–27386.

Shafahi, A.; Huang, W. R.; Najibi, M.; Suciu, O.; Studer, C.; Dumitras, T.; and Goldstein, T. 2018. Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

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

Tao, L.; Feng, L.; Wei, H.; Yi, J.; Huang, S.-J.; and Chen, S. 2022. Can Adversarial Training Be Manipulated By Non-Robust Features? *Advances in Neural Information Processing Systems*, 35: 26504–26518.

Tao, L.; Feng, L.; Yi, J.; Huang, S.-J.; and Chen, S. 2021. Better safe than sorry: Preventing delusive adversaries with adversarial training. *Advances in Neural Information Processing Systems*, 34: 16209–16225.

Tian, Z.; Cui, L.; Liang, J.; and Yu, S. 2022. A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Computing Surveys*, 55(8): 1–35.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wu, S.; Chen, S.; Xie, C.; and Huang, X. 2022. One-pixel shortcut: on the learning preference of deep neural networks. *arXiv preprint arXiv:2205.12141*.

Yu, D.; Zhang, H.; Chen, W.; Yin, J.; and Liu, T.-Y. 2022. Availability attacks create shortcuts. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2367–2376.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E. P.; Ghaoui, L. E.; and Jordan, M. I. 2019. Theoretically Principled Trade-off between Robustness and Accuracy. In *International Conference on Machine Learning*.