Exploring Model Dynamics for Accumulative Poisoning Discovery

Jianing Zhu [1] **Xiawei Guo** [2] **Jiangchao Yao** [3 4] **Chao Du** [2] **Li He** [2] **Shuo Yuan** [2]
Tongliang Liu [5 6] **Liang Wang** [2] **Bo Han** [1]

Abstract

Adversarial poisoning attacks pose huge threats to various machine learning applications. Especially, the recent accumulative poisoning attacks show that it is possible to achieve irreparable harm on models via a sequence of imperceptible attacks followed by a trigger batch. Due to the limited data-level discrepancy in real-time data streaming, current defensive methods are indiscriminate in handling the poison and clean samples. In this paper, we dive into the perspective of model dynamics and propose a novel information measure, namely, *Memorization Discrepancy*, to explore the defense via the model-level information. By implicitly transferring the changes in the data manipulation to that in the model outputs, Memorization Discrepancy can discover the imperceptible poison samples based on their distinct dynamics from the clean samples. We thoroughly explore its properties and propose Discrepancy-aware Sample Correction (DSC) to defend against accumulative poisoning attacks. Extensive experiments comprehensively characterized Memorization Discrepancy and verified its effectiveness. The code is publicly available at: `https://github.com/tmlr-group/Memorization-Discrepancy`.

1. Introduction

Machine learning models have achieved remarkable performance on a wide range of tasks in computer vision (He et al., 2016) and natural language processing (Devlin et al., 2019). However, due to the lack of strict supervision in crowdsourcing (Welinder et al., 2010), data from untrusted sources poses huge threats to machine learning services (Biggio et al., 2012; Goodfellow et al., 2015). Specifically, some malicious adversaries (Paudice et al., 2018; Goldblum et al., 2022) hidden in training data can significantly deteriorate the model performance (Feng et al., 2019; Huang et al., 2020; Tao et al., 2021; Fowl et al., 2021), causing concerns (Bommasani et al., 2021) in those safety-critical applications like autonomous driving or medical intelligence.

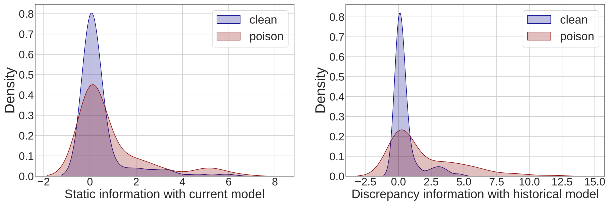


Figure 1. Left: Comparison of the distributions using static information (i.e., the output of the current model); Right: Comparison of the distributions using the discrepancy information (i.e., the output discrepancy of current and historical models). The experiment simulates the accumulative poisoning attack Pang et al. (2021) in real-time data streaming using CIFAR-10 dataset. The generated poison samples can be better distinguished from clean samples by the discrepancy information, i.e., Memorization Discrepancy. Here the static information is also about the output of the model but is defined as the output difference before and after the model optimized on 1 epoch of data. Considering the interval can be nearly ignored compared with the historical model (before 20 epochs), so termed "static". The detailed operation is illustrated in Figure 2.

Different from previous well-explored attacks under the offline setting (Li et al., 2016; Fowl et al., 2021; Goldblum et al., 2022), accumulative poisoning attacks (Pang et al., 2021) are recently proposed and demonstrated to be more imperceptible in real-time data streaming (Wang & Chaudhuri, 2018; Zhang et al., 2020b). Employing the newly introduced accumulative batches for pre-poisoning, it will not cause significant harm to the model during the first phase but leverages the trigger batch to induce dramatic degradation of the model performance instantly. Considering the imperceptibility and the limited knowledge about accumulative poisoning samples, previous works (Feinman et al., 2017; Steinhardt et al., 2017; Ma et al., 2018) that depend on the offline data statistics cannot sufficiently handle this type of sneaky adversary. It naturally raises a new challenge:

[1]Department of Computer Science, Hong Kong Baptist University [2]Alibaba Group [3]CMIC, Shanghai Jiao Tong University [4]Shanghai AI Laboratory [5]Mohamed bin Zayed University of Artificial Intelligence [6]Sydney AI Centre, The University of Sydney. Correspondence to: Bo Han <bhanml@comp.hkbu.edu.hk>, Jiangchao Yao <Sunarker@sjtu.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

how can we identify and defend against the imperceptible accumulative poisoning attacks in real-time data streaming?

Currently, the most possible ways to defend against accumulative poisoning attacks are gradient clipping (Pascanu et al., 2013) and the variants of adversarial training (Tao et al., 2021; Geiping et al., 2021), which have both pros and cons. Specifically, although gradient clipping (Pascanu et al., 2013) shows promise to mitigate the poisoning effect, it still can be deceived by samples with small gradient norms in the accumulative phase and has a side-effect on slowing down the training convergence (Pang et al., 2021). As for adversarial training methods (Madry et al., 2018; Zhang et al., 2019), it has been demonstrated that the natural risk of training with poison samples can be upper bounded by the adversarial risk (Tao et al., 2021). Therefore, it is natural to adopt the reverse adversarial generation to correct the newly captured samples. Unfortunately, the indiscriminate sample calibration in adversarial training when applying to clean samples is detrimental (Zhang et al., 2019) to performance (e.g., as illustrated in Figure 6) due to the over-correction.

In this paper, we introduce a new measure, termed as *Memorization Discrepancy* (i.e., Eq. (5) in Section 3), which is surprisingly aware of the imperceptible accumulative poisoning samples by backtracking earlier historical model (e.g., as illustrated in Figure 1). Diving into the model dynamics, we compute the discrepancy by leveraging the historical model's output on the same sample. It can be found in Figure 2 that with the increase in the backtracking intervals, poison samples can be more distinguishable from clean samples. The underlying mechanism is to transfer imperceptible manipulation into significant model-level changes (as further explained in Figure 3). Then, some observed properties (i.e., Properties 3.4 and 3.5) like monotonically increasing and the existence of highly discriminative backtracking interval can be used to handle poisoning discovery for the sneaky adversary, which show promising in identifying poison samples with imperceptible constraint from clean samples or other natural samples with distribution shift.

Based on the above insights, we accordingly design a new defense algorithm, namely, *Discrepancy-aware Sample Correction* (DSC), which incorporates Memorization Discrepancy to selectively calibrate the potential poison samples in real-time data streaming. At the high level, we relax the inner-minimization of reverse adversarial generation (i.e., Eq. (6) in Section 3.4) and construct a learning filter capable of calibrating oriented poison samples (as shown in Figure 6) to avoid over-calibration. In detail, our DSC employs the early-stopping in sample correction and utilizes the historical model to be an auxiliary inspector for Memorization Discrepancy. Our main contributions are summarized as,

- We make the first effort to explore identifying the accumulative poisoning attack for the real-time data streaming from the perspective of model dynamics, i.e., considering model changes in poison discovery.
- We introduce a novel information measure, i.e., Memorization Discrepancy, to distinguish the imperceptible poison samples by leveraging model-level information from backtracking the historical models. (in Section 3)
- We accordingly propose a new learning method, i.e., Discrepancy-aware Sample Correction (DSC), which incorporates the proposed Memorization Discrepancy to selectively calibrate the potential poison samples with only a historical auxiliary model. (in Section 3.4)
- We conduct extensive experiments to comprehensively characterize the Memorization Discrepancy, and verify the effectiveness of DSC in improving the model robustness against accumulative poisoning attacks using a range of benchmarked datasets. (in Sections 4)

2. Backgrounds

In this section, we briefly review the background of delusive attack and accumulative poisoning attack (Pang et al., 2021), and discuss some existing defense methods.

2.1. Delusive Attack

Delusive attack (Newsome et al., 2006; Feng et al., 2019) belongs to data poisoning attacks (Barreno et al., 2010; Biggio et al., 2012; Goldblum et al., 2022), which aim to degrade the model performance via manipulating the training data. The general malicious objective can be formulated as,

$$\max_{\mathcal{P}} \mathcal{L}(S_{val}; \theta^*), s.t.\ \theta^* \in \arg\min_{\theta} \mathcal{L}(\mathcal{P}(S_{train}); \theta), \quad (1)$$

where S_{train} is the training set consisting of natural examples, S_{val} is the validation set, $\mathcal{P}(\cdot)$ denotes the transformation that manipulates S_{train} into a poisoned version and $\mathcal{L}(S; \theta)$ denotes the empirical learning objective of a dataset $S = \{x_i, y_i\}_{i=1}^N$ with the model parameter θ. Specifically, delusive attack targets to deteriorate the overall accuracy of the test data by only manipulating the input feature of the training data (Newsome et al., 2006; Barreno et al., 2010; Feng et al., 2019), instead of attacking the specific class (Koh & Liang, 2017) or triggering the backdoors (Shafahi et al., 2018). Generally, the delusive attack can be formulated as the optimization problem through the gradient-based methods (e.g., Project Gradient Decent (PGD) (Madry et al., 2018)), and limits the manipulation into a small constraint (e.g., ℓ_∞-norm adopted in adversarial attack (Goodfellow et al., 2015; Kumar et al., 2020)).

2.2. Accumulative Poisoning Attack

Different from previous studies which focus on poisoning offline datasets (Feng et al., 2019; Fowl et al., 2021; Tao et al., 2021), Pang et al. (2021) recently proposed the accumulative poisoning attack for the real-time data stream to simulate the poisoning on the online settings (Chechik et al., 2010). The major difference between this attack from the ordinary delusive attack is that it can interact with the training process and dynamically manipulate the data according to the model status. Through this, it spreads the poisoning effect over multiple learning statuses to further avoid distinct modifications on clean samples. The certain objective for accumulative poisoning attack can be formulated as,

$$\min_{\mathcal{P},\mathcal{A}} \nabla_\theta \mathcal{L}(S_{val}; \mathcal{A}(\theta^T))^\top \nabla_\theta \mathcal{L}(\mathcal{P}(S_T); \mathcal{A}(\theta^T)), \quad (2)$$

where $\mathcal{A}$ denotes an accumulative phase to inject secrete poison samples, $\mathcal{A}(\theta^T)$ denotes the model parameter at round T obtained after the accumulative phase and ∇_θ denotes the gradient. Specifically, the whole process can be divided into two parts given a pre-trained burn-in model for several epochs on the data stream. First, the model will be secretly poisoned by the samples in the accumulative phase $\mathcal{A}$, while keeping test accuracy in a heuristically reasonable range of variation. Then a trigger batch $\mathcal{P}(S_T)$ will be fed into the model. By jointly optimizing the accumulative phase and the trigger batch $\mathcal{P}(S_T)$, the accumulative poisoning attack can result in a severe drop in the model performance in a single step (e.g., one batch). More details about the accumulative poisoning attacks can be referred to in Appendix C.1.

2.3. Existing Defenses

To combat data poisoning, there are many strategies proposed for defending against poisoning attacks, like detection-based methods (Steinhardt et al., 2017; Collinge et al., 2019) to find and filter the poison data according to the feature statistics, robust training methods (Borgnia et al., 2021; Li et al., 2021) that is designed for targeted or backdoor attacks. Considering the characteristic of the real-time data streaming and the imperceptibility of delusive attack, it is computationally expensive and impractical to analyze the statistics for the incoming data (Pang et al., 2021; Kumar et al., 2020). For the accumulative poisoning attack, except the gradient clipping discussed in Pang et al. (2021) that constrains the poisoning effect by small gradients, a principled defense (Tao et al., 2021; Geiping et al., 2021) based on adversarial training can also serve as the major technique to calibrate poison samples. However, both of them are indiscriminate in handling the poison and clean samples. Different from the previous methods, we introduce a novel information measure to discover the imperceptible poison samples by considering the model dynamics.

3. Memorization Discrepancy

In this section, we present the new information measure *Memorization Discrepancy* to explore the poison sample discovery through the lens of model dynamics during the training process. We first discuss our motivation, and then formally introduce the assumption and the definition of Memorization Discrepancy. Finally, we conduct experiments to empirically explore its corresponding properties.

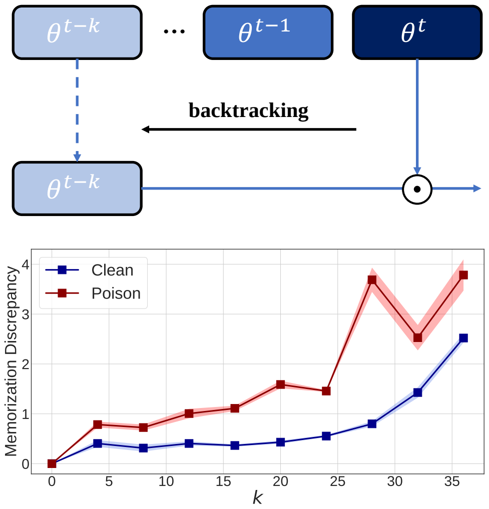


Figure 2. Top: illustration of the concrete operation to obtain the discrepancy information, i.e., Memorization Discrepancy. Bottom: the mean values of the Memorization Discrepancy on clean and poisoning batch data w.r.t. the backtracking interval k (epochs). The θ^t denotes the current model which is used by the attacker to generate poison samples, and θ^{t-1} to θ^{t-k} are the historical model we backtracked. The discrepancy is measured by the output of the data using current and historical models. The difference between the Memorization Discrepancy on poison samples from that on clean samples is more distinguishable along with the enlargement of k. The underlying mechanism is further elaborated in Figure 3.

3.1. Motivation

Different from the offline poisoning adversaries (Li et al., 2016; Fowl et al., 2021), the accumulative poisoning attack is allowed to interact with the model status to update its poison samples dynamically in the training process. Considering the practical situation, without sufficient knowledge of the original natural sample captured in the data streaming and the imperceptible characteristic of delusive attacks, the static information provided by the model from the single dimension seems to be hopeless to differentiate the poisoning and clean samples (e.g., the left panel of Figure 1). However, one critical component that is so far overlooked but easily backtracked (Kumar et al., 2020) in training, is the historical model information. Since the accumulative poisoning attack utilize the sequential order property of real-time data streaming, we raise the following question,

> *Can we also exploit the information of model dynamics to gain some useful clues to identify the*

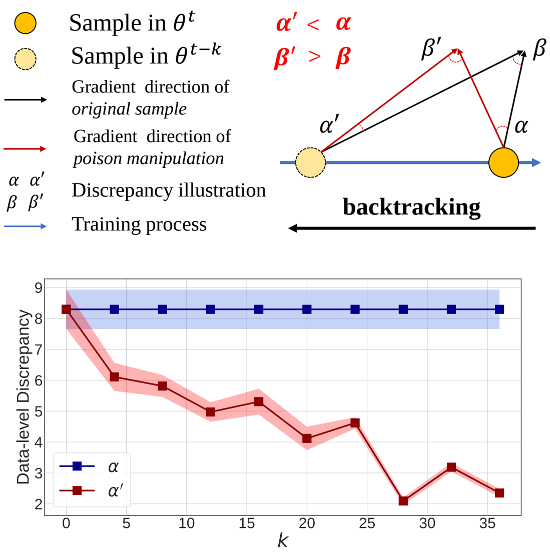


Figure 3. Top: illustration of model dynamics, which shows different effects (e.g., α and α') of the same poisoning manipulation on different model statuses (on the same original sample). Bottom: empirical verification about the above discrepancy by backtracking the model status. Here the α is the illustration of the discrepancy between two different optimization directions (or the gradient direction of the model θ^t) approximated by using the outputs on clean and poison samples, respectively. And the $\alpha^{'}$ is the illustration of discrepancy on the historical model θ^{t-k}. Models at different statuses will have different output changes for the same data manipulation, the discrepancy can be naturally captured using the historical model backtracked in the training process. The bottom figure empirically justifies that $\alpha' < \alpha$ and $\beta' > \beta$, which explains the underlying mechanism of previous trend in Figure 2.

imperceptible accumulative poisoning attacks?

The answer is affirmative. As shown in the right panel of Figure 1, we can find the distributions of clean and poison samples are much different compared with the left panel in Figure 1. Such a significant difference is computed by taking the backtracked historical model into consideration (as illustrated at the top of Figure 2). Intuitively, to achieve a better poisoning effect, the close interaction with the current model (Pang et al., 2021) better optimizes the malicious objective (e.g., Eq. (1)) for poisoning than other checkpoints, but it also ignores the changes in the historical model. This motivates us to further explore the poison discovery from the perspective of the dynamic changes in historical models.

3.2. Proposed Definition

As the model is changed along with training on streaming data, it is natural to make the following assumption of model dynamics, which is about different model outputs with poison samples generated based on the victim model.

Assumption 3.1 (Model Dynamics)**.** Let θ^t and θ^{t-k} denote the current model at round t and the historical model at round $t-k$, $\hat{x}(\theta^t)$ is the adversarial manipulation from x by the model θ^t, $\mathbb{D}$ indicate the general distribution discrepancy measurement[1]. Then, we have the following inequality,

$$\begin{aligned}\mathbb{D}(f(\hat{x}(\theta^t);\theta^t),&f(x;\theta^t)) \neq \\ &\mathbb{D}(f(\hat{x}(\theta^t);\theta^{t-k}), f(x;\theta^{t-k})).\end{aligned} \tag{3}$$

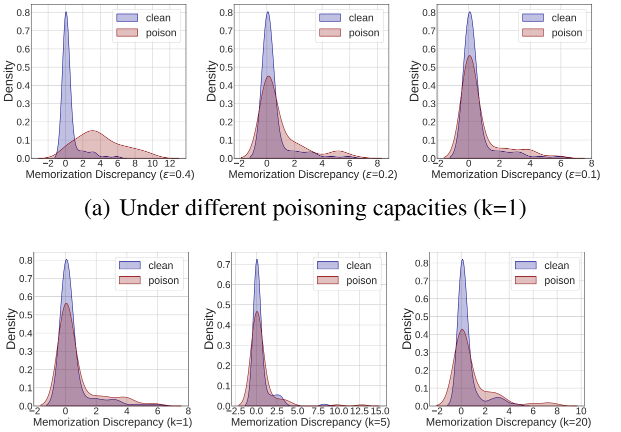
The above inequality indicates that the poison sample generated on the victim model has a different effect on the output changes of a different model. Since the poison manipulation added to the clean samples targets the malicious learning objective that is different from the original one, the left side of Eq. (3) actually reflects the difference between poison samples from clean samples. However, considering the practical situation of real-time data streaming, it is impractical to know whether the newly captured training samples are poisoned in advance. This motivates us to introduce another measure to leverage the information characteristic of historical models. Here we draw further theoretical analyses behind the Assumption 3.1, which construct the relationship on the difference between the poisoning objective (e.g. Eq. (1)) and the original objective. We leave complete discussion and verification in Appendixes A and B.

Theorem 3.2. *Let $f(x;\theta^t)$ denote the output about the sample x at epoch t, k denotes the interval rounds, and S denotes a clean dataset. Considering the opposite between objective $\min \mathcal{L}(S,\theta^*)$ and the poisoning objective $\max \mathcal{L}(S,\theta^*)$ where θ^* is the well-trained model respectively, there exists a learning period where we have,*

$$\begin{aligned}&\mathbb{D}(f(\hat{x}(\theta^t);\theta^t), f(\hat{x}(\theta^{t-k});\theta^{t-k}))- \\ &\mathbb{D}(f(x;\theta^t), f(x;\theta^{t-k})) \propto \mathcal{L}(S;\theta^{t-k}) - \mathcal{L}(S;\theta^t).\end{aligned} \tag{4}$$

The above positive relationship constructs the discrepancy relationship of different samples (e.g., natural sample x and poisoned sample $\hat{x}$) with the model learning dynamics. Due to the different poisoning effect results of sample $\hat{x}$ on different model stages, the previous discrepancy in Assumption 3.1 is constructed on the different sample (x and $\hat{x}$) with the same model (either current model at round t or the historical model at round $t-k$). The underlying intuition of Eq. (4) reflects the differences between the natural learning objective and the poisoning objective, where we evaluate their differences in model outputs. Hence, the sample-wise discrepancy can be transferred to the differences of each sample on different models. Since we can not know whether the newly captured samples are poisoned or not in advance, the latter formulation for the information measure on the same sample is more practical for utilization and help us to provide the final definition of Memorization Discrepancy.

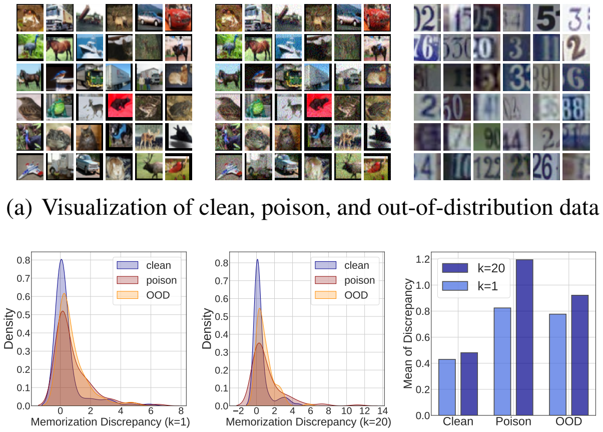
[1] Note that, in the most experiments of this paper, we adopt Kullback–Leibler divergence (Joyce, 2011) in computation.



(a) Under different poisoning capacities (k=1)

(b) Under different backtracking intervals (ϵ=0.1)

Figure 4. Empirical exploration about poisoning discovery using Memorization Discrepancy. (a) The distribution discrepancy of clean and poison data can be constrained by controlling the poisoning capacity, e.g., the perturbation radius ϵ; (b) The poison samples can be more distinguishable from clean samples by enlarging the backtracking interval in Memorization Discrepancy to find a highly discriminative status, which involves model-level information.



(a) Visualization of clean, poison, and out-of-distribution data

(b) Comparisons of the above three type of data

Figure 5. Comparisons about optimized and statical distribution shift distinguished using Memorization Discrepancy. (a) Three types of data from left to right: clean data (CIFRA-10), poison data (ϵ=0.064), and out-of-distribution (OOD) data (SVHN); (b) Compared with the statical distribution shift (OOD), the poison samples which are optimized for the targeted model are more sensitive to the backtracking interval in Memorization Discrepancy.

Below we formally introduce the new information measure,

Definition 3.3 (Memorization Discrepancy)**.** Consider $f : \mathbb{R}^d \to \Delta^C$ that maps the input feature to the C-dimensional simplex, $\hat{x}(\theta^t)$ is disturbed from x on the model $f(\cdot;\theta^t)$, and θ^{t-k} means the parameters of the k-interval historical model compared to the current t. Then, we define *Memorization Discrepancy* on $\hat{x}(\theta^t)$ based on the current parameter θ^t and the historical parameter θ^{t-k} as,

$$\mathbb{D}(f(\hat{x}(\theta^t);\theta^{t-k}), f(\hat{x}(\theta^t);\theta^t)), \tag{5}$$

which measures the discrepancy of the different model's outputs on the same $\hat{x}$ generated on θ^t.

The underlying mechanism of Memorization Discrepancy is to capture the model dynamic on the same sample during the training process, which explicitly reflects the imperceptible poisoning manipulation in the training samples via the difference in model outputs. In Figure 1, we can find that the discrepancy value of both clean and poison samples is enlarged when we backtrack the historical models. Especially, the value of poison samples increases more than that of clean samples. According to this phenomenon, we have two following conjectures on Memorization Discrepancy.

Property 3.4 (Monotonically Increasing Interval)**.** There exists an interval k from t to $t-k$ where the value of $\mathbb{D}(f(x^*;\theta^{t-k}), f(x^*;\theta^t))$ is monotonically increasing from 0 to k, where x^* can indicate either the original clean sample x or the poison sample $\hat{x}$.

Property 3.5 (Highly Discriminative Status)**.** There exists an model status θ^{t-k} where the mean value of poison samples $\mathbb{D}(f(\hat{x}(\theta^t);\theta^{t-k}), f(\hat{x}(\theta^t);\theta^t))$ is much larger than that of clean samples $\mathbb{D}(f(x;\theta^{t-k}), f(x;\theta^t))$.

3.3. Empirical Study on the Properties

Here we study the Memorization Discrepancy through the simulated experiments on CIFAR-10 dataset following Pang et al. (2021), and the detailed setups can be found in Section 4.1. The below empirical results respectively justify the previous Assumption 3.1, Properties 3.4 and 3.5. More exploration of the discrepancy is provided in Appendix C.11.

In Figure 2, we illustrate the pipeline to obtain the Memorization Discrepancy. Specifically, by comparing the auxiliary historical model's output and the current model's output, the Memorization Discrepancy can be easily calculated. From the figure, we can find that the mean values are almost monotonically increasing from 0 to 25 epochs with the increasing k, which empirically verifies its general trend.

In Figure 3, we give the underlying explanation behind the dynamics of Memorization Discrepancy and empirically justify Assumption 3.1 via the approximation results for α and α' in the top panel. According to the top of the figure, the Memorization Discrepancy of clean and poison samples can be denoted by β and β', and their relationship can be reflected by the change on models θ^t and θ^{t-k}, i.e., α and α'. In practice, as the defense party does not know whether the data is poisoning, Memorization Discrepancy is a good choice while Eq. (3) assumes the poisoning fact by default.

In Figure 4, we present that Memorization Discrepancy can distinguish the poison sample with limited poisoning capacities (e.g., the perturbation constraint ϵ is small to guarantee imperceptible data-level manipulation) by backtracking the historical models as an auxiliary inspector, which enlarges the distribution discrepancy as shown in Figure 4(b). In Figure 5, we also consider another kind of natural data that is out-of-distribution (OOD) and may be confusedly reflected with higher Memorization Discrepancy. Fortunately, under the comparison, the poison sample optimized on the victim model shows being more sensitive to the dynamic changes in historical models than static OOD samples. On the other hand, those OOD samples with noticeable visual differences can be easier to clean up than the imperceptible poison ones.

3.4. Discrepancy-aware Sample Correction

Inspired by the previous properties of Memorization Discrepancy, we propose the *Discrepancy-aware Sample Correction* (DSC) to utilize the model dynamics which can capture the differences between the potential poison samples from the clean ones by an auxiliary historical model.

The high-level intuition is to employ the Memorization Discrepancy to the previous principled reverse adversarial generation (Tao et al., 2021) as guidance for the sample correction. Concretely, we summarize the detailed procedure of DSC in Algorithm 1. In each mini-batch training, we will leverage the Memorization Discrepancy to validate whether the sample is a potential poison sample. The multi-step reverse adversarial generation will then be conducted through the following objective,

$$\tilde{x} = \arg\min_{\tilde{x}\in\mathcal{B}[x,\epsilon]} \ell(f(x), y), \quad \text{s.t.}, \mathbb{D}(f(\tilde{x};\theta), f(\tilde{x};\theta^*)) > P, \tag{6}$$

where $\tilde{x}$ is the calibrated sample, $\mathcal{B}[x,\epsilon] = \{\tilde{x} \mid d_\infty(x,\tilde{x}) \leq \epsilon\}$ be the closed ball of radius $\epsilon > 0$ centered at the training sample x, P is the estimated discrepancy threshold, and θ^* is the historical auxiliary model. In addition, we also record the Memorization Discrepancy using a certain measurement (e.g., KL divergence). According to Property 3.5 and previous empirical results, the poison data has a larger discrepancy value than the clean data. Thus, we adopt an early-stopping here to relax the minimization objective for sample correction. The multi-step correction will stop if Memorization Discrepancy is smaller than an adjustable threshold during the pre-defined correction steps. This operation can avoid over-calibration for those clean samples, and we empirically justify its effectiveness in Figure 6.

4. Experiments

In this section, we present a comprehensive analysis of the Memorization Discrepancy and verify the effectiveness of our proposed DSC with the previous baseline methods for defending against accumulative poisoning attacks. More details and supplementary can be referred to in Appendix C.

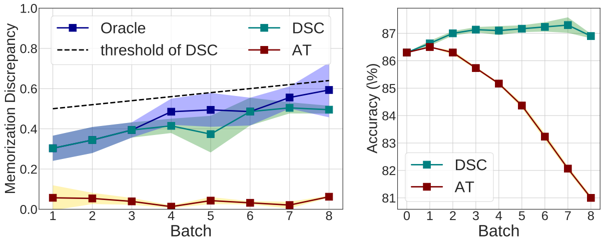


Figure 6. Left panel: the Memorization Discrepancy corresponding to the training samples in real-time data streaming. Right panel: the test accuracy of the AT-based method and our DSC. The reverse adversarial perturbations over-calibrate the clean samples and result in lower accuracy while our DSC can filter the clean sample with an estimated threshold by Memorization Discrepancy.

4.1. Experiment Setups

Training simulation. Following Pang et al. (2021), we simulate the real-time data streaming using the SVHN (Netzer et al., 2011), CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) datasets. The overall learning process consists of two specific phases with different training data (i.e., clean samples and poison samples). The first phase is named *burn-in phase*, like model pre-training, the model will be trained on natural data before taking the training examples from other untrusted sources (Biggio & Roli, 2018). The second phase is termed as *victim phase*, in which the adversaries begin to inject the poison samples to attack the current model. Same as (Pang et al., 2021), we train ResNet-18 (He et al., 2016) using the SGD optimizer with the learning rate 0.1, momentum 0.9, and weight decay 0.0001. During the whole process, we keep the batchsize of data streaming at 100.

Poisoning attack. After the burn-in phase in which the model is pre-trained for 40 epochs, we begin to inject the accumulative poison samples (Pang et al., 2021). Specifically, the crafted sample is generated by PGD under the ℓ_∞-norm constraint. Different from those regular poisoning generations, this poisoning attacker is allowed to intervene during training and tune the poisoning strategies dynamically with the model states. Since its poisoning target is the single-step drop of model accuracy, the poisoning effects of the secretly injected data will be accumulated and triggered in the final batch (termed as trigger batch). To simulate the monitor process in real-time data streaming, this final batch will be triggered when the model training loss is amplified by a monitored threshold of previous poison samples, and we adopted the same threshold values in Pang et al. (2021).

Defense target. To defend the attacks in real-time data streaming, there are three aspects that need to be considered. The first is the single-step drop in model accuracy. The sec-

Table 1. Test accuracy (%) of the simulated experiments on real-time data streaming (Mean±Std).

CIFAR-10	Defense	Accuracy: Start	Batch	Accuracy: +Poison	Accuracy: + Trigger	Δ
Clean Oracle	ST	86.3	-	84.4	84.4	-
	GC		-	86.2	86.2	-
	AT		-	77.2	77.2	-
	DSC		-	84.7	84.7	-
Accu. Poison	ST		1	75.7±3.33	50.4±5.03	-25.3±4.13
	GC		3	79.7±0.25	75.1±0.05	-4.6±0.26
	AT		3	80.1±0.10	75.3±0.26	-4.7±0.20
	DSC		3	**81.2±0.35**	**77.3±0.58**	**-3.8±0.31**

SVHN	Defense	Accuracy: Start	Batch	Accuracy: +Poison	Accuracy: + Trigger	Δ
Clean Oracle	ST	94.6	-	93.4	93.4	-
	GC		-	94.5	94.5	-
	AT		-	89.9	89.9	-
	DSC		-	94.7	94.7	-
Accu. Poison	ST		3	85.4±3.54	70.4±9.16	-15.4±6.2
	GC		7	89.7±0.06	88.3±0.26	-1.4±0.30
	AT		7	89.6±0.21	88.7±0.20	**-0.9±0.06**
	DSC		9	**89.9±0.01**	**88.8±0.26**	-1.1±0.26

CIFAR-100	Defense	Accuracy: Start	Batch	Accuracy: +Poison	Accuracy: + Trigger	Δ
Clean Oracle	ST	59.0	-	55.8	55.8	-
	GC		-	60.2	60.2	-
	AT		-	49.5	49.5	-
	DSC		-	55.0	55.0	-
Accu. Poison	ST		3	42.9±2.74	32.6±2.84	-10.3±0.29
	GC		4	49.8±0.12	43.8±0.29	-6.1±0.25
	AT		5	47.7±0.25	44.4±0.21	-3.2±0.42
	DSC		5	**48.6±0.91**	**45.4±1.39**	**-3.2±0.65**

ond is the final accuracy, which reflects the overall defense effectiveness for the accumulative poisoning attacks. The third is the test accuracy of learning with clean samples, since we assume that the defender does not know when the poison sample is injected. For the threshold schedule, we set $\mu = 0.5, \tau = 0.02$ for both CIFAR-10 and SVHN datasets, and $\mu = 1.7, \tau = 0.1$ for CIFAR-100 dataset.

Threshold adjustment. The certain threshold for Memorization Discrepancy can be estimated based on the value of the controllable clean samples used in the burn-in phase. Similar to the tuning strategies in gradient clipping (Pascanu et al., 2013; Goodfellow et al., 2016), we can set a lower threshold to conduct more correction steps for a conservative optimization for the online model. Based on the illustration in Figure 3 and the Property 3.4, the value of Memorization Discrepancy will increase as the model training. Thus, we adopt a fixed auxiliary model θ^* as the θ^{t-k} in discrepancy calculation. The threshold value will increase when training with the real-time data streaming from the untrusted sources (Biggio & Roli, 2018) and it requires a dynamical threshold for filtering the clean samples with poison samples. To this end, we introduce the schedule as $P = \mu + \tau * m$, where P is our threshold, the initial value μ, the dynamical growing interval τ is estimated by our controllable clean examples, and m is the batch number. We provide further discussion on it in Appendix C.9.

4.2. Baseline Performance

In this part, we compare our DSC with previous baseline methods (i.e., Standard Training (ST), Gradient Clipping (Pascanu et al., 2013) (GC) and Adversarial Training as Poisoning Defense (Tao et al., 2021) (AT)) on several benchmarked datasets to verify its effectiveness. In Table 1, we present the results of Clean Oracle to show the unaffected capacity of learning with clean samples and Accu. Poison to show the defense effectiveness against the secret poisoning attack. Specifically, we report four metrics according to different statuses: 1) Accuracy: +Poison, the accuracy after training with the secret poisoning batches; 2) Accuracy: +Trigger, the accuracy after training with the final trigger batch; 3) Batch, the number of batches before training loss is amplified to the monitored threshold; 4) Δ, the accuracy drop of after the trigger batch. Since there are all clean samples in Clean Oracle, other accuracy values are equal to the final accuracy after training with 100 batches.

According to Table 1, we can find all the defensive meth-

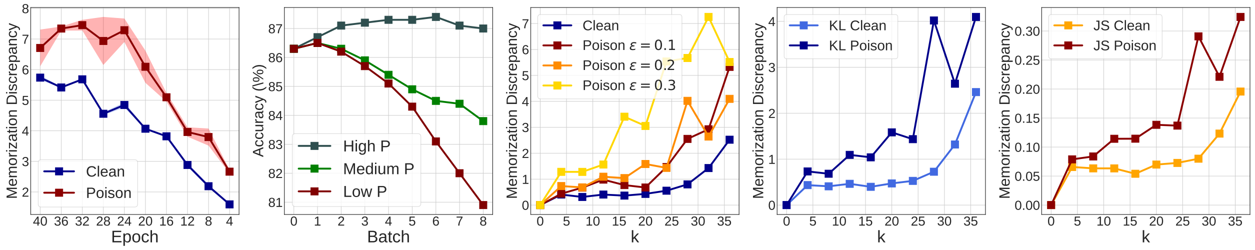


Figure 7. Ablation study. Left panel: Memorization Discrepancy between model θ^t in Eq. (5) and the model at Epoch 1; Left-middle panel: test accuracy with the threshold of different levels; Middle panel: Memorization Discrepancy under different poisoning capacities (imperceptibility); Right-middle and Right panel: Memorization Discrepancy corresponding to different discrepancy measurements.

Table 2. Test accuracy (%) of adopting different adversarial optimization losses in our defense testing on the CIFAR-10 dataset.

Defense/Attack		PGD	KL (TRADES)	CW$_\infty$
	Start	86.3		
DSC	Acc. +Poison	81.4	80.3	81.8
	Acc. +Trigger	78.7	77.3	79.3
	Δ	-2.69	-3.04	-2.54

Table 3. Test accuracy (%) of considering adaptive attacks being aware of Memorization Discrepancy on the CIFAR-10 dataset.

Constraint β		0	0.05	0.1	0.2
	Start	86.3			
ST	Acc. +Poison	81.4	80.9	80.9	80.3
	Acc. +Trigger	51.3	59.9	69.8	74.9
	Δ	-30.04	-20.97	-11.12	-5.43

ods can resist more batches than ST before triggering the pre-defined threshold. As for Accu. Poison, our DSC can achieve better accuracy consistently after going through the poisoning batches and the final trigger batches. Compared with GC, DSC and AT result in a smaller accuracy drop for the final single batch, it is much more important to those real-world applications since the model recovery with worse performance is a large cost (Kairouz et al., 2019). As for Clean Oracle, GC can achieve comparable or even higher accuracy than the pre-trained model since the clipped gradient also slow down the training process with a small gradient (Pang et al., 2021). Due to the indiscriminate correction, AT over-optimizes the clean samples and leads to much lower accuracy than the pre-trained model. In contrast, our DSC can still achieve comparable performance with ST through the selective correction of Memorization Discrepancy. Overall, the experiments running multiple times verified the general effectiveness of our DSC.

4.3. Ablation Study and Further Discussion

In this part, we conduct various experiments on CIFAR-10 to provide a thorough understanding of our presented Memorization Discrepancy and DSC. More ablations from different perspectives can be referred to in Appendix C.

Training status θ^t. In the left panel of Figure 7, we investigate the training status θ^t in Memorization Discrepancy. Specifically, we generate the accumulative poisoning attack based on the θ^t and calculate the discrepancy with the model checkpoint in Epoch 1. As can be seen, the mean values of Memorization Discrepancy on poison samples are consistently distinguishable from that on clean ones. This phenomenon provides us a chance to set just one auxiliary model for checking the dynamics instead of several historical models used in Figure 2 to fix the interval k.

Interval k. In our previous illustration of Figure 2, we visualize the discrepancy with the fixed interval k (e.g., $k \in [4, 36]$). The Memorization Discrepancy of both poison samples and clean samples increases and becomes more distinguishable with the increasing of the interval k. However, it is hard to use general criteria to choose the best interval or the previously analyzed training status. In the left panel of Figure 7, we adopt a dynamical interval k which increases with the training status with a fixed auxiliary model (i.e., θ^{t-k}) at Epoch 1. A similar trend with the distinguishable values can also be captured during the training process.

Threshold P. In the left-middle panel of Figure 7, we validate our proposed DSC with different levels of the threshold P. The intuition behind the threshold is to better utilize the distinguishable Memorization Discrepancies of poison samples and clean samples to filter out the specific samples. With a high threshold P, the test accuracy would not drop significantly when training with clean samples, since the sample correction can early-stop to avoid over-calibration. In contrast, using low threshold results in a severe accuracy drop since we conduct the indiscriminate correction. To further investigate the characteristics of the threshold P, we conduct additional experiments about Eq. (6) with the threshold P in Appendix C. To sum up, on the one hand, the results on natural data confirm that its discrepancy value shares a similar trend (e.g., increasing along the training

Table 4. Comparison of Memorization Discrepancy along the backtracking interval across different backbones.

Dataset	Model/Interval k	Discrepancy on	4	8	12	16	20	24	28	32	36
CIFAR-10	ResNet	Clean	0.43444	0.40864	0.46287	0.39770	0.47189	0.52683	0.72566	1.31921	2.45922
		Poison	0.73276	0.68203	1.09318	1.03971	1.58336	1.43465	4.01915	2.64194	4.09608
	VGG-11	Clean	0.52409	0.41497	0.51741	0.82906	0.94531	1.32099	1.97665	2.82950	5.69943
		Poison	0.33073	0.44917	0.41988	0.80057	0.89317	1.25389	3.65167	6.73917	14.53571
	SmallCNN	Clean	0.42651	0.67719	0.48234	0.53239	0.46322	0.64273	1.05432	0.65214	1.89398
		Poison	1.04400	1.49517	1.08457	0.94815	1.91275	1.88436	3.73850	3.34128	5.79640

process as indicated in Table 4) across different datasets, while the specific threshold P needs different setups according to different training data. On the other hand, we can find that the performance of DSC can be stable during a specific range of the threshold value for identifying the poison data.

Poisoning capacity. In the middle panel of Figure 7, we check the effect of the poisoning capacity, i.e., the imperceptibility, on the value of Memorization Discrepancy. The imperceptibility is controlled by a parameter ϵ which corresponds to the manipulations. As the same in adversarial attacks (Goodfellow et al., 2015), the larger ϵ indicates more perturbations and lower imperceptibility. The results show that the discrepancies between the two values of poison and clean samples also increase along with the enlargement of ϵ.

Different backbones. To verify the generality of Memorization Discrepancy, we conduct experiments using different model structures (e.g., ResNet (He et al., 2016), VGG-11 (Simonyan & Zisserman, 2015), SmallCNN (Zhang et al., 2019)) on both clean and poison samples to check the discrepancy value in Table 4. The results confirm the phenomenon generally exists across different backbones in our experiments on the CIFAR-10 dataset, e.g., the difference between the Memorization Discrepancy on poison samples from that on clean samples is generally more distinguishable along with the enlargement of backtracking interval k.

Discrepancy measurement. In the rest two panels of Figure 7, we also investigate another discrepancy measurement to check the relationship between poison and clean samples. Here we adopt the Jensen–Shannon divergence (Dagan et al., 1997) (JS) to calculate it and compare the results with that calculated on KL divergence. Both discrepancy measurements can capture a similar trend for their Memorization Discrepancy. Due to the different definitions for the measurement, there exists a difference on the scale of specific discrepancy values. The overall results show that the distinguishable relationship between two Memorization Discrepancies is not a consequence of a certain measurement but all of them, and the general intuition behind the discrepancy can also be captured by other measurements.

Discussion on the adaptive attacker. In Tables 2 and 3, we consider different adversarial methods for generating imperceptible poison samples. The results demonstrate the robust effectiveness of DSC on different attacks. Furthermore, we also discuss a potential adaptive attacker (Tramer et al., 2020) which is aware of Memorization Discrepancy, and try to incorporate it into its generation constraint to escape from identifying. However, the constraint can directly mitigate the poisoning effect that is reflected by the Δ in Table 3, where the poison sample is generated under a constraint controlled by β with the historical model.

In addition, we also provide more explorations of Memorization Discrepancy and the DSC from different perspectives in Appendix C, including extra experiments of DSC in different learning and identification settings, the effects of different components, and corresponding discussions.

5. Conclusion

In this work, we investigated the accumulative poisoning attacks in real-time data streaming through the views of model dynamics. Through the exploration of the dynamic changes, we present a novel measure, i.e., Memorization Discrepancy, which is aware of the imperceptible manipulation added to the clean samples. Based on the novel measure, we propose the Discrepancy-aware Sample Correction method, which can selectively calibrate the poison samples. We present a comprehensive understanding of the discrepancy, and also various experiments to show the effectiveness of the DSC. We believe the underlying spirit of our Memorization Discrepancy, i.e., the dynamic changes in different models, can also motivate other defensive methods or applications.

Acknowledgements

JNZ and BH were supported by NSFC Young Scientists Fund No. 62006202, Guangdong Basic and Applied Basic Research Foundation No. 2022A1515011652, Alibaba Innovative Research Program, and HKBU CSD Departmental Incentive Grant. JCY was supported by the National Key R&D Program of China (No. 2022ZD0160703), STCSM (No. 22511106101, No. 22511105700, No. 21DZ1100100), 111 plan (No. BP0719010).

References

Barreno, M., Nelson, B., Joseph, A. D., and Tygar, J. D. The security of machine learning. *Machine Learning*, 2010.

Biggio, B. and Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. In *ACM SIGSAC*, 2018.

Biggio, B., Nelson, B., and Laskov, P. Poisoning attacks against support vector machines. In *ICML*, 2012.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. In *arXiv*, 2021.

Borgnia, E., Cherepanova, V., Fowl, L., Ghiasi, A., Geiping, J., Goldblum, M., Goldstein, T., and Gupta, A. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In *ICASSP*, 2021.

Carlini, N. and Wagner, D. A. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, 2017.

Chechik, G., Sharma, V., Shalit, U., and Bengio, S. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 2010.

Collinge, G., Lupu, E., and Munoz Gonzalez, L. Defending against poisoning attacks in online learning settings. In *ESANN*, 2019.

Dagan, I., Lee, L., and Pereira, F. Similarity-based methods for word sense disambiguation. In *arXiv*, 1997.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Ding, G. W., Sharma, Y., Lui, K. Y. C., and Huang, R. Mma training: Direct input space margin maximization through adversarial training. In *ICLR*, 2020.

Feinman, R., Curtin, R. R., Shintre, S., and Gardner, A. B. Detecting adversarial samples from artifacts. In *arXiv*, 2017.

Feng, J., Cai, Q.-Z., and Zhou, Z.-H. Learning to confuse: generating training time adversarial data with autoencoder. In *NeurIPS*, 2019.

Fowl, L. H., Goldblum, M., Chiang, P.-y., Geiping, J., Czaja, W., and Goldstein, T. Adversarial examples make strong poisons. In *NeurIPS*, 2021.

Geiping, J., Fowl, L., Somepalli, G., Goldblum, M., Moeller, M., and Goldstein, T. What doesn't kill you makes you robust (er): Adversarial training against poisons and backdoors. In *arXiv*, 2021.

Goldblum, M., Tsipras, D., Xie, C., Chen, X., Schwarzschild, A., Song, D., Mądry, A., Li, B., and Goldstein, T. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*. MIT press Cambridge, 2016.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Huang, H., Ma, X., Erfani, S. M., Bailey, J., and Wang, Y. Unlearnable examples: Making personal data unexploitable. In *ICLR*, 2020.

Joyce, J. M. Kullback-leibler divergence. In *International encyclopedia of statistical science*. Springer, 2011.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. In *arXiv*, 2019.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *ICML*, 2017.

Krizhevsky, A. Learning multiple layers of features from tiny images. In *arXiv*, 2009.

Kumar, R. S. S., Nyström, M., Lambert, J., Marshall, A., Goertzel, M., Comissoneru, A., Swann, M., and Xia, S. Adversarial machine learning-industry perspectives. In *IEEE Security and Privacy Workshops*, 2020.

Li, B., Wang, Y., Singh, A., and Vorobeychik, Y. Data poisoning attacks on factorization-based collaborative filtering. In *NeurIPS*, 2016.

Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., and Ma, X. Anti-backdoor learning: Training clean models on poisoned data. In *NeurIPS*, 2021.

Ma, X., Li, B., Wang, Y., Erfani, S. M., Wijewickrema, S., Schoenebeck, G., Song, D., Houle, M. E., and Bailey, J. Characterizing adversarial subspaces using local intrinsic dimensionality. In *ICLR*, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop* *on Deep Learning and Unsupervised Feature Learning*, 2011.

Newsome, J., Karp, B., and Song, D. Paragraph: Thwarting signature learning by training maliciously. In *International Workshop on Recent Advances in Intrusion Detection*, 2006.

Pang, T., Yang, X., Dong, Y., Su, H., and Zhu, J. Accumulative poisoning attacks on real-time data. In *NeurIPS*, 2021.

Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. In *ICML*, 2013.

Paudice, A., Muñoz-González, L., and Lupu, E. C. Label sanitization against label flipping poisoning attacks. In *ECML PKDD Workshops*, 2018.

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *NeurIPS*, 2018.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Steinhardt, J., Koh, P. W. W., and Liang, P. S. Certified defenses for data poisoning attacks. In *NeurIPS*, 2017.

Tao, L., Feng, L., Yi, J., Huang, S.-J., and Chen, S. Better safe than sorry: Preventing delusive adversaries with adversarial training. In *NeurIPS*, 2021.

Tramer, F., Carlini, N., Brendel, W., and Madry, A. On adaptive attacks to adversarial example defenses. In *NeurIPS*, 2020.

Wang, Y. and Chaudhuri, K. Data poisoning attacks against online learning. In *arXiv*, 2018.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020.

Welinder, P., Branson, S., Perona, P., and Belongie, S. The multidimensional wisdom of crowds. In *NeurIPS*, 2010.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

Zhang, J., Xu, X., Han, B., Niu, G., Cui, L., Sugiyama, M., and Kankanhalli, M. Attacks which do not kill training make adversarial learning stronger. In *ICML*, 2020a.

Zhang, X., Zhu, X., and Lessard, L. Online data poisoning attacks. In *LDC*, 2020b.

Appendix

Reproducibility Statement

We provide the repository of our source codes to ensure the reproducibility of main experimental results: https://github.com/tmlr-group/Memorization-Discrepancy. All experiments are conducted with multiple runs on NVIDIA GeForce RTX 3090 GPUs.

A. Property Insights of Memorization Discrepancy

In this part, we provide the formal analysis of the property insights (e.g. Theorem 3.2 introduced in the main text) of our Memorization Discrepancy. To reveal the underlying mechanism of the proposed information measure, we start by revisiting the different targets of poisoning adversaries from the original training objective. Without specifying any detailed strategy for generating poison samples, the malicious objective generally targets to deteriorate the model performance on clean inputs, e.g., $\max \mathcal{L}(S;\theta^*)$, where θ^* is assumed to be well-trained in the given samples.

However, considering a model that is updated with clean training data, it gradually approaches to different side (e.g., $\min \mathcal{L}(S;\theta^*)$) of the previous target. Based on that, we can naturally make the following assumption about the sample-wise discrepancy with the difference between the current and target loss value,

Assumption A.1. Let $f(x;\theta^t)$ denote the model dynamics about the sample x and at round t, k denotes the interval rounds for backtracking. Considering the ordinary objective $\min \mathcal{L}(S;\theta^*)$ and the poisoning objective $\max \mathcal{L}(S;\theta^*)$ with the clean inputs set S and a poisoned set P, we have,

$$\mathbb{D}(f(\hat{x}(\theta^t);\theta^t), f(x;\theta^t)) \propto \max \mathcal{L}(S;\theta^*) - \mathcal{L}(S;\theta^t), \quad s.t.\ \theta^* \in \arg\min_{\theta} \mathcal{L}(P;\theta) \tag{7}$$

Intuitively, it indicates that the model output of the poison sample will be much more different from that of the clean sample when the model is well-trained on the clean training data (i.e., has a small loss value on clean set S). In other words, the poisoning adversary needs a larger effort to achieve the malicious target, since the model has already performed well on the clean set.

Here we present the Theorem 3.2 again (i.e., the same as the following Theorem A.2) to start the analysis and the further discussion on the critical property of the defined Memorization Discrepancy.

Theorem A.2. *Let $f(x;\theta^t)$ denote the output about the sample x at epoch t, k denotes the interval rounds, and S denotes a clean dataset. Considering the opposite between objective $\min \mathcal{L}(S,\theta^*)$ and the poisoning objective $\max \mathcal{L}(S,\theta^*)$ where θ^* is the well-trained model respectively, there exists a learning period where we have,*

$$\mathbb{D}(f(\hat{x}(\theta^t);\theta^t), f(\hat{x}(\theta^{t-k});\theta^{t-k})) - \mathbb{D}(f(x;\theta^t), f(x;\theta^{t-k})) \propto \mathcal{L}(S;\theta^{t-k}) - \mathcal{L}(S;\theta^t), \tag{8}$$

proof of Theorem A.2. The correlation of the two parts in Eq. (8) can be formulated in the following.

Given the two approximate optimization targets as,

$$\begin{aligned} &\theta^t - \beta\nabla_{\theta^t}\mathcal{L}(f(x;\theta^t),y) \to \min \mathcal{L}(S;\theta^{t+1}) \\ &\theta^t - \beta\nabla_{\theta^t}\mathcal{L}(f(\hat{x}(\theta^t);\theta^t),y) \to \max \mathcal{L}(S;\theta^{t+1}), \end{aligned} \tag{9}$$

we can obtain the correlation about these two opposite target parts as,

$$\mathbb{D}(\nabla_\theta\mathcal{L}(f(x),y,\theta^t), \nabla_\theta\mathcal{L}(f(\hat{x}(\theta^t)),y,\theta^t)) \propto \max \mathcal{L}(S;\theta^*) - \mathcal{L}(S;\theta^t), \tag{10}$$

where $\theta^* \in \arg\min_\theta \mathcal{L}(P;\theta)$ is the model parameter well-trained on the poison samples. Similarly, we can also get the following equation via backtracking,

$$\mathbb{D}(\nabla_\theta\mathcal{L}(f(x),y,\theta^{t-k}), \nabla_\theta\mathcal{L}(f(\hat{x}(\theta^{t-k})),y,\theta^{t-k})) \propto \max \mathcal{L}(S;\theta^*) - \mathcal{L}(S;\theta^{t-k}), \tag{11}$$

Since the two gradient parts share the same anchor of model parameter and the labels, we can get the consistent relationship that similar to Assumption A.1 as,

$$\begin{aligned} \mathbb{D}(\mathcal{L}(f(x),y,\theta^t), \mathcal{L}(f(\hat{x}(\theta^t)),y,\theta^t)) &\propto \max \mathcal{L}(S;\theta^*) - \mathcal{L}(S;\theta^t), \\ \mathbb{D}(\mathcal{L}(f(x),y,\theta^{t-k}), \mathcal{L}(f(\hat{x}(\theta^{t-k})),y,\theta^{t-k})) &\propto \max \mathcal{L}(S;\theta^*) - \mathcal{L}(S;\theta^{t-k}), \end{aligned} \tag{12}$$

By accumulate the approximate discrepancy correlation with historical models, we can introduce the discrepancy considering the samples of same type,

$$\begin{aligned}&\mathbb{D}(f(x;\theta^{t-k}), f(x;\theta^{t})),\\ &\mathbb{D}(f(\hat{x}(\theta^{t-k});\theta^{t-k}), f(\hat{x}(\theta^{t});\theta^{t})),\end{aligned} \tag{13}$$

Using the above discrepancy on model outputs, we can explicitly obtain the formulation by constructing discrepancy for each side of Eq. (12),

$$\mathbb{D}(f(\hat{x}(\theta^{t});\theta^{t}), f(\hat{x}(\theta^{t-k});\theta^{t-k})) - \mathbb{D}(f(x;\theta^{t}), f(x;\theta^{t-k})) \propto \mathcal{L}(S;\theta^{t-k}) - \mathcal{L}(S;\theta^{t}). \tag{14}$$

This gives the property insights on the dynamics of the Memorization Discrepancy.

□

In summary, the above correlation of the Memorization Discrepancy and the loss discrepancy between two different model stages is built on the high-level target discrepancy. The Eq. (8) indicates that we can enlarge the discrepancy of the two information values on clean and poison samples via construct the proper loss discrepancy. Backtracking the historical model can serve this goal since it naturally reflects the dynamical behavior of learning with the ordinary objective.

As enlarging the backtracking interval k, the loss discrepancy is further enlarged. The corresponding poison and clean samples become more distinguishable on the basis of our proposed information value. It is consistent with previous empirical results in Figures 1 and 2. This property exactly meets our requirement described in Section 3.1, i.e., to gain useful information about imperceptible poison samples via model dynamics. To be specific, as presented in Figure 1, the two distributional statics become more distinguishable when we construct the discrepancy by involving the historical models. Similar in Figure 6, the Memorization Discrepancy of poison samples is larger than that of clean samples. It is general and has no specific assumption about the poisoning generation.

From the new perspective, the proposed Memorization Discrepancy can accumulate the target-level discrepancy in model dynamics for better distinguishing poison samples from clean samples, which is appropriate to figure out the accumulative poisoning attacks since the adversary try to spread the perceived risk over a single round of optimization.

B. Further discussion about the Demonstration in Figure 3

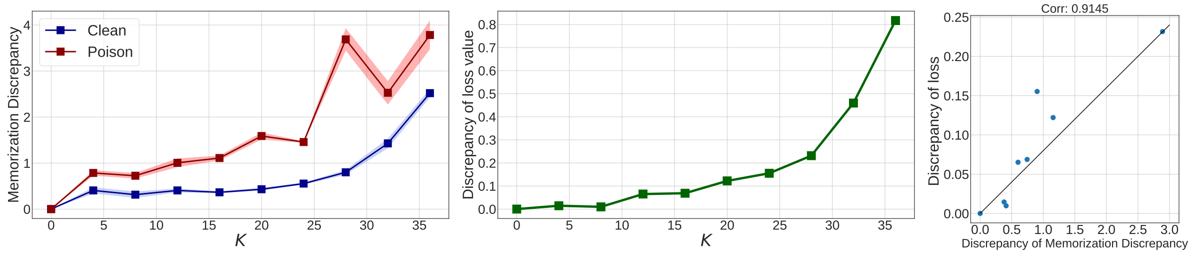
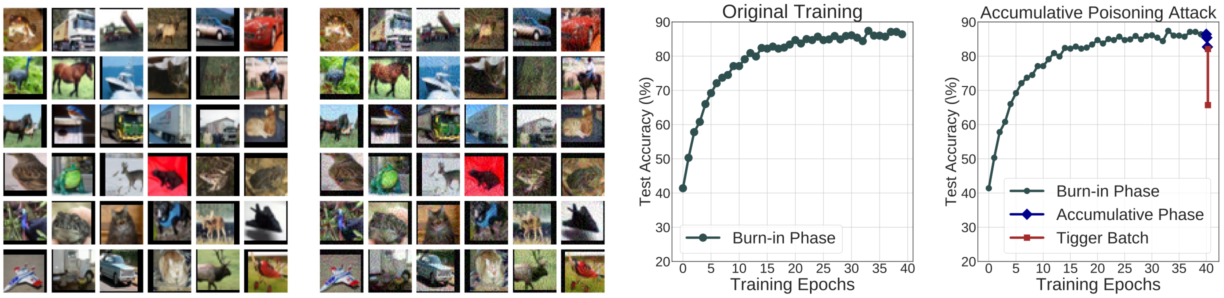


Figure 8. Empirical verification about the property insights on the Memorization Discrepancy.

In this part, we provide the empirical verification of the previous property insights draw from the discrepancy of model dynamics. On the same simulation experiments on CIFAR-10, we check the Memorization Discrepancy and the corresponding loss discrepancy between the current and historical model in Figure 8. It can be found that during the stable training phase (e.g. back from Epoch 40 to Epoch 5) the correlation between the discrepancy in model output and loss values are proportional. In the early stage, we can find some inconsistent relationships exist, we attribute the possible reason to the unstable optimization which can not accurately reflect the relative distance between the malicious target (training with poison samples) and the ordinary target (training with clean samples).

C. Additional details and explorations

In this section, we provide completed information about the accumulative poisoning attacks with extra details of algorithm implementation, as well as extra experimental results.



(a) Illustration of Clean and Poison Samples ($\epsilon = 0.064$)

(b) Empirical Unawareness of Accumulation

Figure 9. Visualization of the empirical imperceptibility on accumulative poisoning attack using CIFAR-10 dataset. Except for the visual-level imperceptibility, the accumulative poison samples will not induce a significant accuracy drop which may be caught by a simple monitor.

C.1. Details about Accumulative Poisoning Attack

In this part, we describe the details of the Accumulative Poisoning Attack. Let S_{train} be the clean training set and S_{val} be the separate validation set, an attacker will poison the S_{train} into a poisoned $\mathcal{P}(S_{train})$. Except for the original malicious objective as,

$$\max_{\mathcal{P}} \mathcal{L}(S_{val};\theta^*), \quad s.t.\ \theta^* \in \arg\min_{\theta} \mathcal{L}(\mathcal{P}(S_{train});\theta), \tag{15}$$

the accumulative poisoning attack utilizes the characteristics of online learning to inject the poison samples. Hence, the real-time malicious objective is formulated as follows at the training round T,

$$\max_{\mathcal{P}} \mathcal{L}(S_{val};\theta^{T+1}), \quad s.t.\ \theta^{T+1} = \theta^T - \beta\nabla_\theta \mathcal{L}(\mathcal{P}(S_{train});\theta^T), \tag{16}$$

where β is the learning rate of gradient descent.

By expanding the previous malicious objective, it can be rewritten as,

$$\min_{\mathcal{P}} \nabla_\theta \mathcal{L}(S_{val};\theta^T)^\top \nabla_\theta \mathcal{L}(\mathcal{P}(S_T);\theta^T), \tag{17}$$

Based on Eq. (17), Pang et al. (2021) introduce the accumulative phase $\mathcal{A}$ to make the model parameter at round T obtained after the accumulative phase be more sensitive and fragile to the poisoning. So the overall objective can be formulated as,

$$\min_{\mathcal{P},\mathcal{A}} \nabla_\theta \mathcal{L}(S_{val};\mathcal{A}(\theta^T))^\top \nabla_\theta \mathcal{L}(\mathcal{P}(S_T);\mathcal{A}(\theta^T)), \tag{18}$$

and the perturbed data batch $\mathcal{A}(S_t)$ can be crafted by solving a first-order expansion of the real-time learning update,

$$\max_{\mathcal{P},\mathcal{A}_t} \nabla_\theta \mathcal{L}(\mathcal{A}_t(S_t);\theta^t)^\top \left[\nabla_\theta \mathcal{L}(S_t;\theta^t) + \lambda \cdot \nabla_\theta(\nabla_\theta \mathcal{L}(S_{val};\mathcal{A}(\theta^T))^\top \nabla_\theta \mathcal{L}(\mathcal{P}(S_T);\mathcal{A}(\theta^T)))\right], \tag{19}$$

which equals to,

$$\max_{\mathcal{P},\mathcal{A}_t} \nabla_\theta \mathcal{L}(\mathcal{A}_t(S_t);\theta^t)^\top \left[\underbrace{\nabla_\theta \mathcal{L}(S_t;\theta^t)}_{\text{keep accuracy}} + \lambda \cdot \underbrace{\nabla_\theta(\nabla_\theta \mathcal{L}(S_{val};\theta^T)^\top \nabla_\theta \mathcal{L}(\mathcal{P}(S_T);\theta^T))}_{\text{accumulating poisoning effects for the trigger batch}}\right], \tag{20}$$

Following Pang et al. (2021), we adopt the burn-in phase that pretrains the model for 40 epochs. Then we begin to inject the accumulative poison samples (Pang et al., 2021). Specifically, the crafted sample is generated by PGD (Madry et al., 2018) under the ℓ_∞-norm constraint. Since its poisoning target is the single-step drop of model accuracy, the poisoning effects of the secretly injected data will be accumulated and triggered in the final batch (termed as trigger batch). To simulate the monitor process in real-time data streaming, this final batch will be triggered when the training loss is amplified by a threshold of previous poison samples (the same as threshold in Pang et al. (2021)).

Algorithm 1 Discrepancy-aware Sample Correction (DSC)

Input: data streaming $S = \{(x_i, y_i)\}_{i=1}^{n}$, learning rate η, number of epochs T, batch size m, number of batches M, data $x \in \mathcal{X}$, label $y \in \mathcal{Y}$, victim model θ, loss function ℓ, PGD step K, perturbation bound ϵ, step size δ, projection opt. Π, Memorization Discrepancy threshold P, auxiliary historical model θ^*.
Output: model θ^T;

```
for epoch = $1, \dots, T$ do
    for mini-batch = $1, \dots, M$ do
        Sample a mini-batch $\{(x_i, y_i)\}_{i=1}^{m}$ from $S$
        for $i = 1, \dots, m$ (in parallel) do
            Obtain the corrected sample $\tilde{x}_i$ of $x_i$:
            $\tilde{x_i} \leftarrow x_i, n = 1$
            while $\mathbb{D}(f(\tilde{x_i}; \theta), f(\tilde{x_i}; \theta^*)) > P$ and $n < K$ do
                $\tilde{x_i} \leftarrow \Pi_{\mathcal{B}[x_i, \epsilon]}\big(\tilde{x_i} - \delta \cdot \text{sign}(\nabla_{\tilde{x_i}} \ell(f(\tilde{x_i}), y))\big)$
                $n = n + 1$
            end while
        end for
        $\theta \leftarrow \theta - \eta \nabla_\theta \ell(f_\theta(\tilde{x}_i), y_i)$
    end for
end for
```

C.2. Algorithm Realization of DSC

Here we provide the detailed realization of our proposed Discrepancy-aware Sample Correction in Algorithm 1.

C.3. Comparison of Training Time

In this part, we check the training time of different defenses with the accumulative poisoning attacks. For our proposed DSC which incorporates Memorization Discrepancy in identifying the incoming samples on the data streaming, the cost of the backtracked historical model mainly lies in the storage to save the historical model checkpoints. However, considering that in practice it is common to save the checkpoints regularly during training, such cost of our method is acceptable. Here we report the training time of each method in Table 5 to give a more intuitive comparison. The experiment setups keep the same as Table 1. According to the results, we can see that DSC requires slightly more time than other methods.

Table 5. Comparison of per mini-batch training and the accuracy (%) after the poisoning attack across different datasets.

Dataset	Method	Training Time (seconds)	Acc. +Tigger
CIFAR-10	ST	0.0206	50.4
	GC	0.0253	75.1
	AT	0.3735	75.3
	DSC	0.3241	77.3
CIFAR-100	ST	0.0189	32.6
	GC	0.0314	43.8
	AT	0.3732	44.4
	DSC	0.2948	45.4
SVHN	ST	0.0198	70.4
	GC	0.0250	88.3
	AT	0.3630	88.7
	DSC	0.0689	88.8

C.4. Extra Validation of the Threshold P

We conduct more experiments to present Eq. (6) of our DSC. First, we check the trend of discrepancy across different datasets in Table 6, and the results on natural data confirm that it shares a similar trend (increasing along the training process) across different datasets, while the specific threshold P needs different setups due to different datasets. Second, we conduct experiments about hyperparameter tuning in the proper thresholds P and compare the performance on clean oracle and poisoned training data, respectively. Note that, in Table 7, there are no further results when the test accuracy drops to a certain level (indicated by "-"). The results show that the performance is stable during the specific range of the threshold P value. To be more specific. Similar to the hyperparameters of gradient clipping and adversarial training, the threshold P can not be too high to lose control on correcting the poisoned data and also needs to be not too low to induce over-calibration on the clean sample. It is consistent with the results presented in the left-middle panel of Figure 7.

Table 6. The discrepancy trend of batch data across different datasets.

Dataset	Discrepancy on/Batch	1	2	3	4	5	6	7	8	9	10	11
CIFAR-10	Clean	0.2941	0.2996	0.4271	0.4487	0.4489	0.6101	0.6083	0.5755	0.6505	0.8110	0.7807
CIFAR-100	Clean	1.7981	1.9814	2.0760	2.2362	2.4590	2.9175	3.1986	3.3778	2.9070	3.3067	3.6144
SVHN	Clean	0.0784	0.0793	0.1007	0.1221	0.0922	0.1433	0.1249	0.1588	0.1468	0.1531	0.2175

Table 7. Test accuracy during the training process w.r.t. hyperparameter tuning on the proper thresholds.

Dataset	Data	Threshold P/Batch m	1	2	3	4	5	6	7	8
CIFAR-10	Clean	0.8+0.02m	0.863	0.867	0.871	0.872	0.873	0.873	0.874	0.873
	Clean	0.5+0.02m	0.863	0.867	0.871	0.872	0.873	0.873	0.874	0.871
	Clean	0.4+0.02m	0.863	0.867	0.873	0.873	0.874	0.871	0.869	0.868
	Clean	0.3+0.02m	0.863	0.867	0.871	0.872	0.87	0.87	0.867	0.867
	Clean	0.1+0.02m	0.863	0.865	0.865	0.862	0.861	0.859	0.858	0.857
CIFAR-10	Poison	0.8+0.02m	0.863	0.841	0.76	0.665	-	-	-	-
	Poison	0.5+0.02m	0.863	0.859	0.842	0.812	0.771	-	-	-
	Poison	0.4+0.02m	0.863	0.859	0.842	0.812	0.771	-	-	-
	Poison	0.3+0.02m	0.863	0.859	0.842	0.81	0.77	-	-	-
	Poison	0.1+0.02m	0.863	0.859	0.842	0.81	0.77	-	-	-

C.5. Ablations about Attack Success

We have conducted extra ablation study on evaluating the attack success (keep the same setups with the left middle panel of Figure 7), and summarize the results in Table 8. The attack success rate here is defined as the percentage of received examples that can circumvent the defense method with specific threshold. The results show there is a trade-off between the model accuracy and the attack success rate. To be specific, it is due to the critical characteristic of our Memorization Discrepancy on clean samples and poison samples. The lower threshold tend to cover the correction ability of AT that indiscriminately treat all examples as poison sample, while the higher threshold tend to behave as the ST. As for the defense for controlling the attack success rate, it can be further designed referring to other specific techniques to utilize the critical nature of our Memorization Discrepancy.

Table 8. Evaluations (%) about attack success w.r.t. the threshold P in CIFAR-10.

Threshold of Different Level	Accuracy	Attack Success Rate
High P	**87.1**	87.5
Medium P	83.7	37.5
Low P	80.8	**12.5**

C.6. Ablations about Different Auxiliary Models

As for the phenomenon of Memorization Discrepancy (as shown in the left panel of Fig 5), it can be found in other settings that using the model checkpoint in epoch E $(E \in [1, \text{present}])$ as the auxiliary model. The overall results show a similar trend as the left panel of Figure 7. In our main experiments in Table 1, we use the checkpoint at Epoch 20 for CIFAR-10/100 as our auxiliary model. We also conduct the experiments on CIFAR-10 using the different auxiliary model with the same threshold to see how it affect our DSC and summarize the results in Table 9. The results show that the threshold may need adjustment when we choose the different auxiliary models to compute the Memorization Discrepancy. It can be found if we backtrack the earlier checkpoint (i.e., Epoch 10), the threshold estimated using checkpoint at Epoch 20 maybe still compatible. However, it is not appropriate when we use the later checkpoint (i.e., Epoch 30). Using the different auxiliary models needs further estimate the threshold by a small batch of clean data used in the previous training stage.

Table 9. Performance of DSC using the different auxiliary models with the same/different threshold setup.

Auxiliary Epoch	Acc. Start	Batch	Acc. +Poison	Acc. + Trigger	Δ
10	86.3%	3	80.9±0.09%	77.1±0.16%	-3.9±0.12%
10 [adjust P]	86.3%	3	81.4±0.05%	77.5±0.23%	-3.8±0.21%
20	86.3%	3	81.2±0.35%	77.3±0.58%	-3.8±0.31%
20 [adjust P]	86.3%	3	81.0±0.09%	77.8±0.22%	-3.6±0.05%
30	86.3%	3	77.3±1.25%	63.6±3.39%	-13.6±4.64%
30 [adjust P]	86.3%	3	80.2±0.12%	76.8±0.27%	-4.0±0.32%

C.7. Comparisons about Other AT Variants

As for our proposed DSC, the critical part is to selectively correct the potential poison samples using the Memorization Discrepancy. We can extend those AT variants (Zhang et al., 2019; Wang et al., 2020; Ding et al., 2020; Zhang et al., 2020a) to be sample corrections in our problem setting. We conduct the comparison on CIFAR-10 dataset and summarize the results in Table 10. Since all those variants are designed for further improving adversarial robustness or other issues in adversarial training, its objective all introduce other optimization parts which sacrifice the natural performance, the results also demonstrate that the accuracy drop using these AT-variants-based methods for accumulative poisoning defense is more severe than the original AT.

Table 10. Comparison with variants of AT methods for the sample correction.

Method	Acc. Start	Batch	Acc. +Poison	Acc. + Trigger	Δ
ST	86.3%	1	75.7±3.33%	50.4±5.03%	-25.3±4.13%
AT	86.3%	3	80.1±0.10%	75.3±0.26%	-4.7±0.20%
TRADES	86.3%	3	78.2±0.28%	72.5±0.45%	-5.8±0.32%
MART	86.3%	3	77.5±0.32%	68.4±0.66%	-9.1±1.20%
MMD	86.3%	3	77.2±0.81%	71.4±0.77%	-5.8±0.89%
FAT	86.3%	3	80.4±0.27%	76.2±0.23%	-4.2±0.45%
DSC	86.3%	3	**81.2±0.35%**	**77.3±0.58%**	**-3.8±0.31%**

C.8. Ablations about Black-box Setting

Empirically, we also verify the effect of our proposed method on the extended black-box setting for accumulative poisoning attack, and summarize the results compared with White-box setting in Table 11. In this setting, we use other surrogate models (e.g., the historical model earlier than the current model stage) to generate the adversarial examples and feed them into the vaccine model. The results show that our DSC has a comparable defense effect to that of the white-box setting.

Table 11. Comparison with variants of AT methods for the sample correction.

Setting	White/Black	Acc. Start	Batch	Acc. +Poison	Acc. + Trigger	Δ
CIFAR-10 (Clean Oracle)	-	86.3%	-	84.7%	84.7%	-
Accu. Poison (DSC)	White-box	86.3%	3	81.2±0.35%	77.3±0.58%	-3.8±0.31%
Accu. Poison (DSC)	Black-box [30]	86.3%	3	81.7±0.23%	78.2±0.14%	-3.5±0.12%
Accu. Poison (DSC)	Black-box [20]	86.3%	3	82.0±0.11%	78.9±0.23%	-3.1±0.08%
Accu. Poison (DSC)	Black-box [10]	86.3%	3	82.5±0.02%	79.7±0.03%	-2.8±0.05%

C.9. Empirical Evaluation of the Correction Condition

As for the hyper-parameters μ and τ, in the burn-in phase that follows the (Pang et al., 2021), we can estimate them by using a small batch sample of clean data. According to the previous properties of the Memorization Discrepancy we observed, we can approximate the μ and τ by the value computed on the clean data in some period of the burn-in phase. And we did not change the defense parameters between these two kinds of experiments for fair evaluation. To provide more informative results, we check the experiments for running the clean oracle with 50 batches of samples and summarize how often the threshold condition is satisfied during training in Table 12. The results show that part of the clean samples is also affected by our DSC and their value satisfies the condition in Algorithm 1. For the experiments with clean oracle, we use the same threshold as the experiments on defending against the accumulative poisoning attack. It shows the selective mechanism based on the condition.

Table 12. How often the threshold condition is satisfied during training?

Dataset	Acc. Start	Acc. Oracle	Frequency (Satisfy the Correction Condition)
CIFAR-10	86.3%	84.7%	28%
CIFAR-100	59.0%	55.0%	24%

C.10. Preliminary Exploration on Federated Setting

Different from real-time data streaming, for the accumulative poisoning attacks in a federated setting, we need to adapt our method to the federated learning framework where we can not directly conduct the sample-wise correction. Specifically, we incorporate the proposed Memorization Discrepancy into the selective defense (e.g., Discrepancy-aware Gradient Clipping (DGC)) against accumulative poisoning attacks and conduct the experiments in the following table. We can see that the extended method can also perform comparable or better based on selectively adjusting the training.

Table 13. Classification accuracy (%) on CIFAR-10 during the accumulative phase for 500 steps. Our new information measure on learning dynamics with the historical model can serve as an auxiliary for gradient clipping operations.

CIFAR-10	Method	10	100	200	300	400	500
Clean Oracle	ST	85.35	83.87	83.9	83.88	83.81	83.86
	GC	84.34	85.27	85.5	85.48	85.46	**85.43**
	DGC	84.34	85.31	85.49	85.5	85.45	**85.43**
Accu. Poisoned	ST	84.65	69.96	68.74	69.36	69.31	69.23
	GC	84.88	84.27	83.14	81.78	80.15	78.95
	DGC	84.87	84.31	83.3	82.13	80.66	**79.84**

C.11. More Dynamics of the Memorization Discrepancy

In this part, we present more exploration about the dynamics of the proposed Memorization Discrepancy. For the poisoning generation, we follow the same malicious objective in Eq. (1) and adopt Fowl et al. (2021) to generate the poison samples for presenting the discrepancy trend, i.e., generating the adversarial poison samples via the adversarial generation procedure (Madry et al., 2018). In Figure 10, we change the backtracking interval k during the historical 40 epochs. The differences between the Memorization Discrepancy of clean and poison samples approximately become more separable when we increase K. In Figure 11, we fix the auxiliary model at Epoch 1 and investigate the value of Memorization Discrepancy using different intervals. The overall results show a similar trend with the previous analysis, that we can better utilize the model dynamics via enlarging the backtracking interval in computing the Memorization Discrepancy. In Figure 12, we change the different auxiliary models from Epoch 1 to Epoch 28. Although there exists the same trend as the previous two explorations, the value of Memorization Discrepancy varies among the different auxiliary models. It can draw the same conclusion as the experiment in Appendix C.6 that we may need further estimate the appropriate threshold for distinguishing the clean and poison samples. The overall results demonstrate that model dynamics are aware of the imperceptible poison samples.

C.12. Detailed Discussion about Attackers Being Aware of Memorization Discrepancy

Considering the concern about adaptive attackers in conventional adversarial literatures (Tramer et al., 2020), we also present a further discussion about a stronger attacker being aware of our Memorization Discrepancy and trying to incorporate it into the poison sample generation (Pang et al., 2021) with the auxiliary model that used in our experiments.

Before that, we also try different adversarial attacking objectives (e.g., PGD (Madry et al., 2018), KL-based method in TRADES (Zhang et al., 2019), and C&W (Carlini & Wagner, 2017)) in generating the poison samples. Our empirical results in Table 2 show that different adversarial generation methods in conventional adversarial literature have limited differences from each other. Then, we delve into the stronger attacker that is also optimized for Memorization Discrepancy.

However, unlike the previous adaptive adversarial attacks (Carlini & Wagner, 2017; Tramer et al., 2020) utilizing the extra search space to find a stronger adversarial example to satisfy the misclassification requirement, and meanwhile keep the imperceptibility. Our empirical results in Table 3 show that keeping the constraint of Memorization Discrepancy can directly affect the poisoning effect induced by the generated poison samples, indicating the underlying difference between generating adversarial examples (Goodfellow et al., 2015) for misleading the model inference and generating adversarial poison samples for misleading the model training. In other words, the constraint on Memorization Discrepancy in poison generation will directly mitigate the poison effect on the target model.

To be specific, following the detailed optimization procedure of accumulative poisoning attack, we incorporate the constraint of Memorization Discrepancy into the original generation equation used in (Pang et al., 2021). Similar to the first constraint in Eq. (20) used for keeping the accuracy (which is targeted for escaping from a simple monitor based on accuracy statics), we add the second term in Eq. (21) for Memorization Discrepancy, where $\mathcal{L}_{\text{MD}} = \mathbb{D}(f(\hat{x}(\theta^t);\theta^*), f(\hat{x}(\theta^t);\theta^t))$ and the auxiliary historical model θ^* are kept same as DSC. The whole generation objective is extended as follows,

$$\max_{\mathcal{P},\mathcal{A}_t} \nabla_\theta \mathcal{L}(\mathcal{A}_t(S_t);\theta^t)^\top \left[\underbrace{\nabla_\theta \mathcal{L}(S_t;\theta^t)}_{\text{keep accuracy}} + \underbrace{\beta \cdot \nabla_\theta \mathcal{L}_{\text{MD}}}_{\text{keep imperceptibility}} + \lambda \cdot \underbrace{\nabla_\theta (\nabla_\theta \mathcal{L}(S_{val};\theta^T)^\top \nabla_\theta \mathcal{L}(\mathcal{P}(S_T);\theta^T))}_{\text{accumulating poisoning effects for the trigger batch}} \right], \tag{21}$$

Intuitively, it is reasonable that the search space about generating a perturbation for adversarial examples may be easier to utilize for keeping the imperceptibility than constructing the adversarial poison samples in accumulative poisoning attacks or other delusive attacks. The above exploration verifies Memorization Discrepancy has significance in identifying accumulative poisoning attacks and also in increasing the difficulty of generating poison samples with satisfactory poisoning effects and better statistical unawareness.

D. Further Discussion

As for the underlying mechanism of Memorization Discrepancy, it has no special assumption on the types of poisoning generation but reflects the target-level discrepancy (i.e., the differences between poisoning target $\max \mathcal{L}(S,\theta)$ and the original target $\min \mathcal{L}(S,\theta)$) by exploring model dynamics. Memorization Discrepancy is a characteristic of poisoned behavior that can be considered in different defensive methods or detection strategies. We primarily focus on this problem

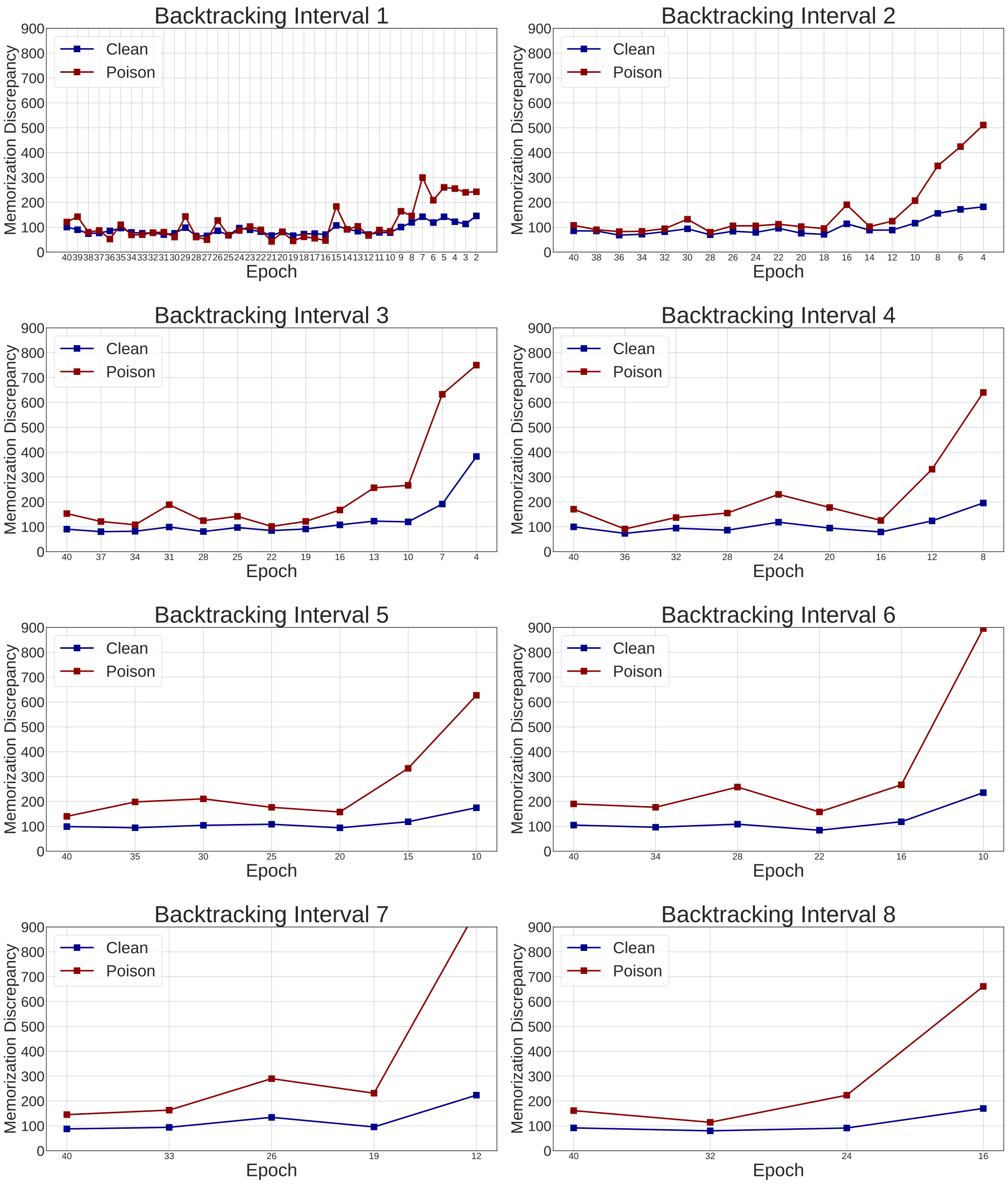


Figure 10. Dynamics of backtracking interval on Memorization Difference in CIFAR-10.

set in our study since the delusive attack and its corresponding defense are important and of great interest in the related literature (Newsome et al., 2006; Fowl et al., 2021; Pang et al., 2021). One possible strategy to extend our work to different types of poisoning is to explore indications in the nature of the specific poisoning objective using model dynamics. However, since the poisons have distinct targets (Fowl et al., 2021; Geiping et al., 2021; Pang et al., 2021) and various different objectives, we would leave expanding our approaches to be one major work in future.

Here we also discuss the potential limitations of our work, there are two points that need to be improved in the future. First, as our work mainly focuses on defending against the accumulative poison attack on real-time data streaming, currently, there is a certain gap in generalizing our method to an offline setting (e.g., training with the poisoned samples from scratch). To be specific, utilizing the Memorization Discrepancy in other settings may require more improvement or adjustment. Second, regarding the proposed DSC, the current method still requires carefully checking the model dynamics to set the threshold P. The predefined threshold may increase the extra analytical workload for adopting the method in practice.

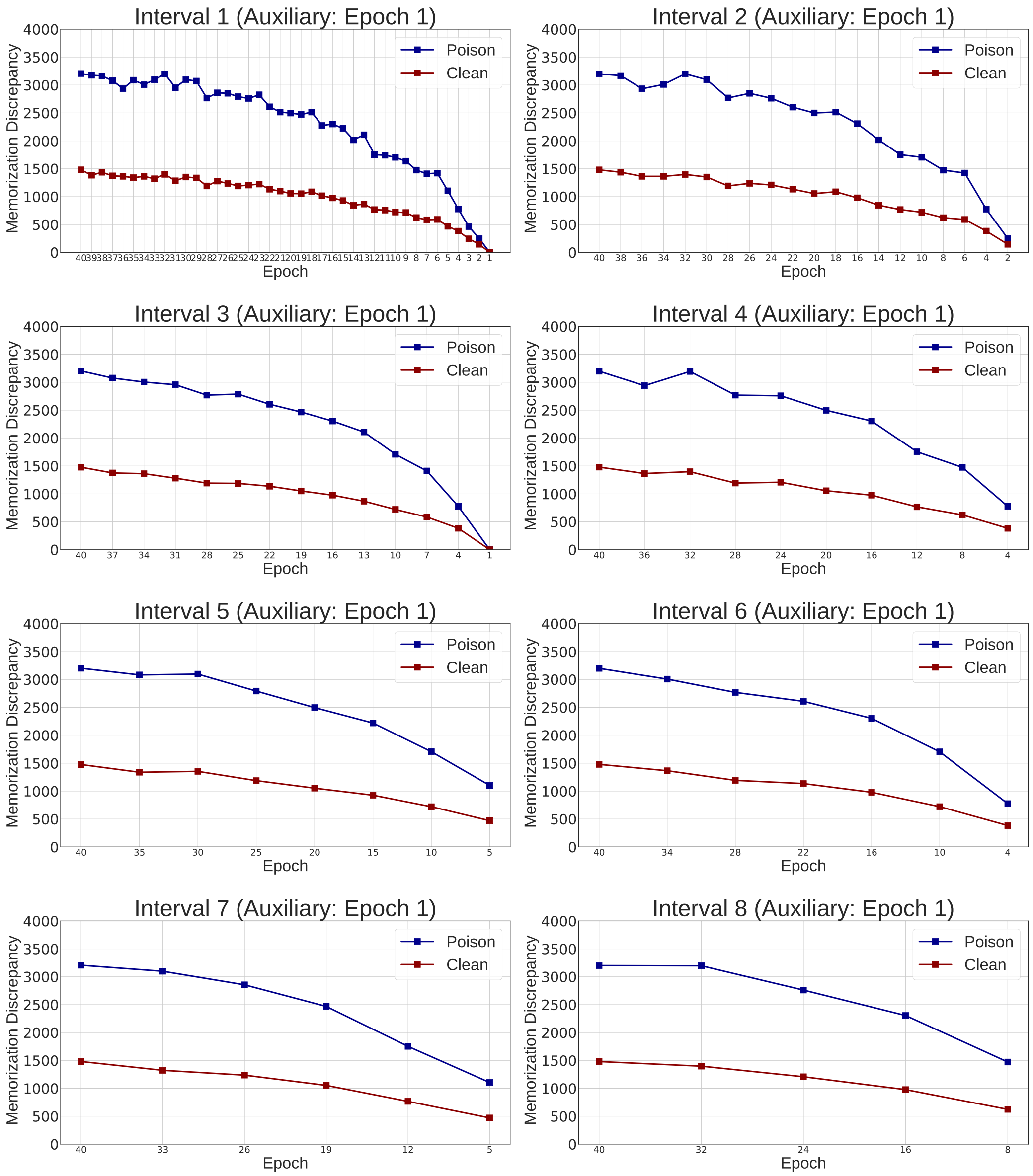


Figure 11. Dynamics of same auxiliary epoch on Memorization Difference in CIFAR-10.

Regarding the future directions, there are two directions corresponding to previously discussed limitations. First, the learning dynamics revealed by the Memorization Discrepancy capture the relationship between the natural objective and the poisoning objective, which can be extended to or explored in other settings like the offline poisoning defense. Second, it can be found that all the current methods still suffer from performance degradation induced by the accumulative poisoning attack. Considering the practical and special scenarios, how to enhance the defense or detection method is also a worthwhile topic to explore further.

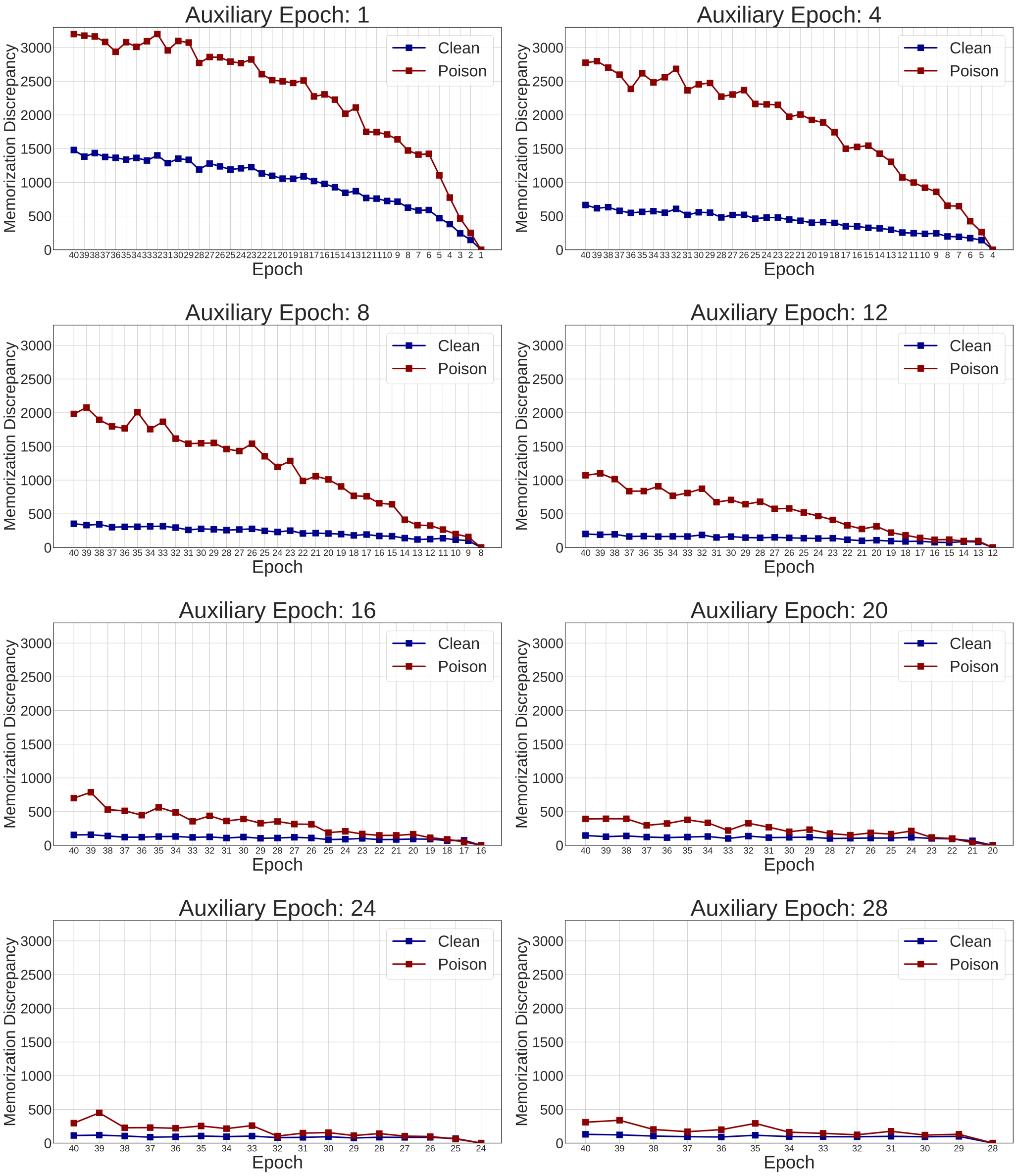


Figure 12. Dynamics of different auxiliary model on Memorization Difference in CIFAR-10.