ACCUMULATIVE POISONING DEFENSE WITH MEMORIZATION DISCREPANCY

Anonymous authors
Paper under double-blind review

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 the trigger sample. Due to the limited data-level information in real-time data streaming, the 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. Through implicitly transferring changes in the data manipulation to that in model outputs, Memorization Discrepancy can efficiently discover the imperceptible poison samples based on their distinct values from the clean samples. We thoroughly analyze its properties and propose a Discrepancy-aware Sample Correction (DSC) to defend against accumulative poisoning attacks. Extensive experiments comprehensively characterize Memorization Discrepancy and verified the effectiveness of our DSC.

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 lack of the strict supervision in crowdsourcing (Welinder et al., 2010), data from untrusted sources poses huge threats to machine learning services (Biggio et al., 2012). Specifically, 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 in those safety-critical applications like autonomous driving or medical intelligence.

Different from previous well-explored attacks under the offline setting (Li et al., 2016; Fowl et al., 2021), accumulative poisoning attacks (Pang et al., 2021) that is more imperceptible in real-time data streaming (Wang & Chaudhuri, 2018; Zhang et al., 2020b) draw more attentions. Specifically, it introduces the accumulative batches to pre-poison the model without the significant harm, and then leverage the trigger batch to dramatically induce the degradation of the model performance instantly. Considering the imperceptibility and the limited knowledge about accumulative poisoning attacks, 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. This raises a new challenge: *how can we defend against the accumulative poisoning attack 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; Yang et al., 2020) to performance (e.g., as shown in Figure 4) due to the over-calibration.

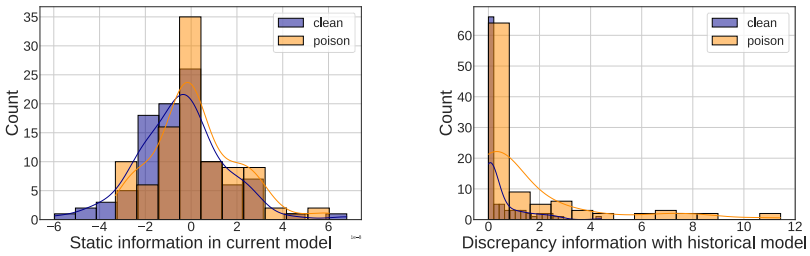

Figure 1: Left panel: Comparison of the distributions using the static information (i.e., the output of current model); Right panel: Comparison of the distributions using the discrepancy information (i.e., the output discrepancy of current and historical models). The experiment simulates the poisoning in real-time data streaming using *CIFAR-10* dataset. The accumulative poisoning samples generated by Pang et al. (2021) 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 model but is defined as the output difference before and after model optimized on one small batch data. Considering the change can be nearly ignored comparing with historical model, so termed "static". The detailed operation is illustrated in Figure 2 and the underlying mechanism is explained in Figure 3.

In this paper, we introduce a measure, termed as *Memorization Discrepancy* (i.e., Eq. 5 in Section 3), which is surprisingly aware of the imperceptible accumulative poisoning attack. In Figure 1 and Figure 2, we dive into the model dynamics and compute the discrepancy by leveraging the historical model's output on the same sample. It can be found that with the increasing of the backtracking intervals, poison samples are more distinguishable from the 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 highly discriminative can be used to handle the adversary in real-time data streaming.

Based on the above empirical 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 4) and construct a learning filter capable of calibrating oriented poison samples (as shown in Figure 4) to avoid over-calibration. In detail, our DSC employs the early-stopping in sample correction using the information constructed with a historical auxiliary model. We summarize our main contribution as,

- We provide a new perspective, namely, model dynamics, to explore defending against the accumulative poisoning attacks for the real-time data streaming.
- We introduce a novel information measure, i.e., Memorization Discrepancy, to distinguish the imperceptible poison samples by leveraging model-level information. (in Section 3)
- We accordingly propose a new learning technique, i.e., Discrepancy-aware Sample Correction (DSC), which utilizes the proposed Memorization Discrepancy to selectively calibrate the potential poison samples with only a historical auxiliary model. (in Section 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 5)

2 Backgrounds

In this section, we briefly review the background of delusive attack (Fowl et al., 2021; Tao et al., 2021) and accumulative poisoning attack (Pang et al., 2021), and discuss the 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,

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

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 is 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)).

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,

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

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. By jointly optimizing the accumulative phase and the poisoned batch $\mathcal{P}(S_T)$ (also named as trigger batch), the accumulative poisoning attack can result in a severe drop on the model performance in a single-step. More details about the optimization of accumulative poisoning attacks can refer to Appendix C.1.

2.3 Existing defenses

To combat the 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 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 strategies, we introduce a novel information measure to discover the imperceptible poison samples by model dynamics and proposed a new defense method.

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 verify its corresponding properties.

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

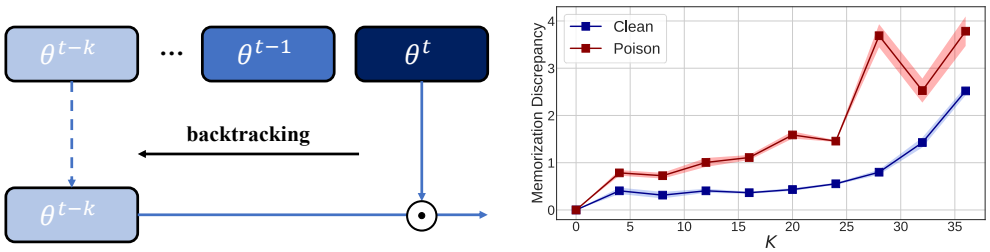

Figure 2: Left panel: illustration of the concrete operation to obtain the discrepancy information, i.e., Memorization Discrepancy. Right panel: the mean values of the Memorization Discrepancy on clean and poison batch data w.r.t. the backtracking interval K (epochs). The θ_t denotes the current model which is used by the attacker to generate the poison samples, and θ_{t-1} to θ_{t-k} are the historical model we backtracked. The Memorization Discrepancy is measured by the output of the data using current and previous models. According to the right panel, the difference between the Memorization Discrepancy on poison samples from that on clean samples are more distinguishable along with the enlargement of K. The underlying reason of this phenomenon is further elaborated out in Figure 3.

in training process. Considering the practical situation, without the sufficient knowledge of the original natural sample captured in the data streaming and the imperceptible characteristic of delusive attack, the static information provided by the model form 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 the training, is the historical model information. Since the accumulative poisoning attack utilize the sequentially order property of the real-time data streaming, we arise the following question,

> *Can we also exploit the information of model dynamic to gain some useful clues to defend against 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 in the left panel of Figure 2). Intuitively, to achieve a better poisoning effect, the close interaction with the current model better optimizes the malicious objective (e.g., Eq. 1) but also ignores the changes in the historical model. This motivates us to further explore the poison sample 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 on 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,

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

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 or not in advance. This motivates us to introduce another measure in the following to leverage the information characteristic of historical models. Below we draw the further property insights behind the Assumption 3.1, which construct the relationship on the difference between the malicious objective for poisoning (e.g. Eq. 1) and the original objective. The complete analysis and further verification can refer to Appendixes A and B.

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

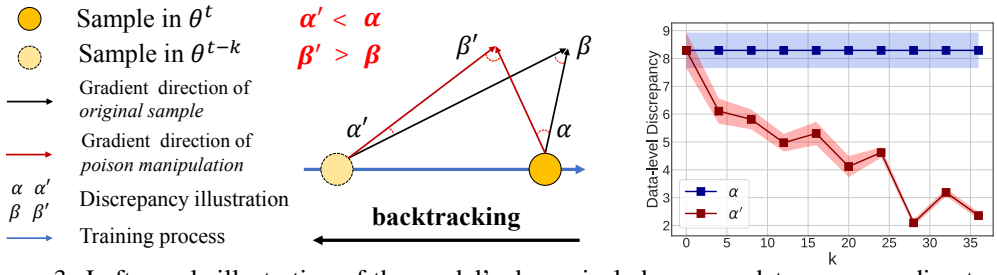

Figure 3: Left panel: illustration of the model's dynamical change on data corresponding to our proposed Memorization Discrepancy. Right panel: empirical verification about the discrepancy of different data on the same model status, which corresponding to the α and α' in the left panel. Here the α is 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 sample, respectively. And the $\alpha^{'}$ is the illustration of discrepancy on the historical model θ_{t-k}. Models at different status 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 right figure empirically justify that $\alpha' < \alpha$ and $\beta' > \beta$, which explains the previous trend in Figure 2.

Theorem 3.2. *Let $f(x;\theta^t)$ denote the output about the sample x and at round t, k denotes the interval rounds, 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, we have,*

$$\mathbb{D}(f(\hat{x}(\theta^t);\theta^t), f(\hat{x}(\theta^{t-k});\theta^{t-k})) - \mathbb{D}(f(x;\theta^t), f(x;\theta^{t-k})) \propto \mathcal{L}(S;\theta^{t-k}) - \mathcal{L}(S;\theta^t), \quad (4)$$

Through the correlation of the model dynamics with loss discrepancy, we propose our new 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,

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

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 on 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. Specially, 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^* denotes either the original clean sample x or the poison sample $\hat{x}$.

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

Here we study the Memorization Discrepancy through the simulated experiments on *CIFAR-10* dataset following Pang et al. (2021), and the detailed setups can refer to Section 5.1. The below empirical results respectively justify the previous Assumption 3.1, Propertyies 3.4 and 3.5. More results about the general comparisons about the discrepancy during training can refer to Appendix C.7.

In Figure 2, we illustrate the pipeline to obtain the Memorization Discrepancy through backtracking the historical models. 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 the general trend of Memorization Discrepancy.

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 left panel.

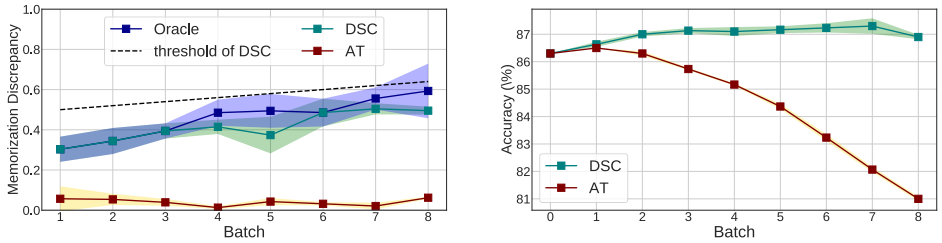

Figure 4: Left panel: the Memorization Discrepancy corresponding to the training samples in real-time data streaming. Right panel: the test accuracy of the AT-based method and our proposed DSC. The reverse adversarial perturbations over-calibrate the clean samples and result in the lower accuracy while our DSC can filter the clean sample with a estimated threshold by Memorization Discrepancy.

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 model θ^*.
Output: model θ^T;

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

According to the left 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 practical, as the defense party does not know whether the data is poisoning, Memorization Discrepancy is a better choice while Eq. 3 assumes the poisoning fact by default.

4 Proposed Method: Discrepancy-aware Sample Correction

Inspired by the previous properties of Memorization Discrepancy, we propose the *Discrepancy-aware Sample Correction* (DSC) to better utilize this dynamic information which can capture the differences between the possible 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 a 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,

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

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 and P is the estimated discrepancy threshold. In addition, we also record the Memorization Discrepancy using a certain measurement (e.g., KL divergence (Joyce, 2011)). According to the Property 3.5 and previous empirical results, the poison data has 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. This operation can avoid over-calibration for those clean samples and we empirically justify its effectiveness in Figure 4.

5 EXPERIMENTS

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

5.1 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 (Pang et al., 2021)). To be specific, the first phase is named as *burn-in phase*, like the model pre-training, the model will be trained on natural data before taking the training examples from other untrust sources (Biggio & Roli, 2018). The second phase is termed as *victim phase*, in which the adversaries will begin to inject the poison samples to attack the 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 as 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 (Madry et al., 2018) 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 threshold of previous poison samples, and we adopted the same threshold values in Pang et al. (2021).

Defense. To defend the poisoning attacks in real-time data streaming, there are three aspects that need to be considered. The first is the single-step drop of model accuracy. The second is the final accuracy, which reflects the overall defence 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 on the controllable clean samples used in the pre-training 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 further discuss it in Appendix C.6.

5.2 ABLATION STUDY AND ANALYSIS

In this part, we conduct various experiments on *CIFAR-10* to provide a thorough understanding of our presented Memorization Discrepancy with natural data and the accumulative poisoning data. To be specific, the ablation studies include the critical factors in Memorization Discrepancy (i.e., Eq. 5). More ablations about attack success rate, AT variants, condition frequency can refer to Appendix C.

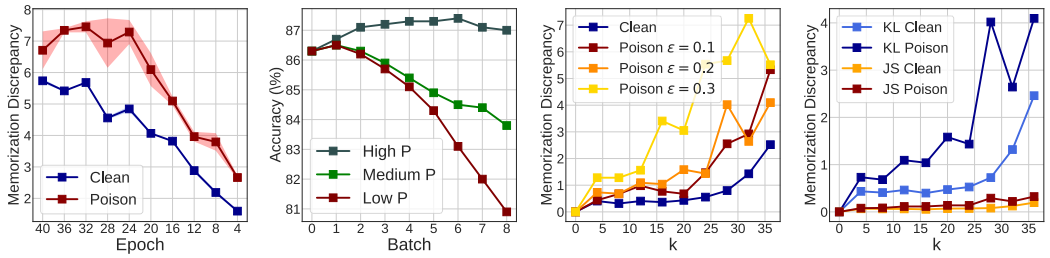

Figure 5: Ablation study. Left-most panel: Memorization Discrepancy between the model θ^t (epoch) in Eq. 5 the model at Epoch 1; Left-middle panel: test accuracy with the threshold of different level; Right-middle panel: Memorization Discrepancy with different poisoning capacity (imperceptibility); Right-most panel: Memorization Discrepancy corresponding to different discrepancy measurements.

Training status θ^t. In the left-most panel of Figure 5, 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 have conducted the experiments to visualize the discrepancy with the fixed interval k (e.g., $k \in [4, 36]$). As the increasing of the interval k, the Memorization Discrepancy values of both poison samples and clean samples increase and being more distinguishable. However, it is hard to use a general criteria to choose the best interval or the previously analyzed training status. In the left-most panel of Figure 5, 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 5, we validate our 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 a low threshold results in a severe accuracy drop since we conduct the indiscriminate correction for clean samples.

Poisoning capacity. In the right-middle panel of Figure 5, we check the effect of the poisoning capacity (Pang et al., 2021), i.e., the imperceptibility, on the value of Memorization Discrepancy. To be specific, the imperceptibility is controlled by a parameter ϵ which corresponding 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.

Discrepancy measurement. In the right-most panel of Figure 5, we also investigate other 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 the discrepancy, and compare the results with that calculated on KL divergence. As shown in the plot, both discrepancy measurements can capture the similar trend for their Memorization Discrepancy. Due to the different definition for the measurement, there exist the difference on the scale of specific discrepancy values. The overall results show that the distinguishable relationship between two Memorization Discrepancy is not a consequence of a certain measurement but all of them, and the general intuition behind the discrepancy, i.e., the model dynamics, can be also captured by other measurements.

5.3 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

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

CIFAR-10	Defense	Acc. Start	Batch	Acc. +Poison	Acc. + 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	Acc. Start	Batch	Acc. +Poison	Acc. + 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	Acc. Start	Batch	Acc. +Poison	Acc. + 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**

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) Acc. +Poison, the accuracy after training with the secret poisoning batches; 2) Acc. +Trigger, the accuracy after training with the final trigger batch; 3) Batch, the number of batch before the training loss are amplified to the threshold; 4) Δ, the accuracy drop of after the trigger batch. Since there are all clean samples in the Clean Oracle, the Acc. +Poison and Acc. +Trigger are equal to the final accuracy after training with 100 batches.

According to Table 1, we can find all the defensive methods 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 effectiveness of our DSC.

6 Conclusion

In this work, we investigated the accumulative poisoning attacks in the real-time data streaming by 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 malicious manipulation added to the clean samples. Based on the Memorization Discrepancy, we propose the Discrepancy-aware Sample Correction method, which can selectively calibrate the poison samples. We provide comprehensive justifications for the rationality of the discrepancy, and also various empirical results to show the effectiveness of the DSC. We believe the underlying spirit of our Memorization Discrepancy, i.e., the dynamical changes on data or model, can also motivate other defensive methods or applications.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of experimental results, we have state the details for experiments in Section 5.1, and we will provide the anonymous repository about our source codes in the discussion phase for reviewing purposes. All the datasets we considerd in our work are opensourced benchmarks.

REFERENCES

Marco Barreno, Blaine Nelson, Anthony D Joseph, and J Doug Tygar. The security of machine learning. In *Machine Learning*, 2010.

Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. In *Pattern Recognition*, 2018.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *ICML*, 2012.

Eitan Borgnia, Valeriia Cherepanova, Liam Fowl, Amin Ghiasi, Jonas Geiping, Micah Goldblum, Tom Goldstein, and Arjun Gupta. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In *ICASSP*, 2021.

Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. In *JMLR*, 2010.

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

Ido Dagan, Lillian Lee, and Fernando Pereira. Similarity-based methods for word sense disambiguation. In *arXiv*, 1997.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. In *ICLR*, 2020.

Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. In *arXiv*, 2017.

Ji Feng, Qi-Zhi Cai, and Zhi-Hua Zhou. Learning to confuse: generating training time adversarial data with auto-encoder. In *NeurIPS*, 2019.

Liam H Fowl, Micah Goldblum, Ping-yeh Chiang, Jonas Geiping, Wojciech Czaja, and Tom Goldstein. Adversarial examples make strong poisons. In *NeurIPS*, 2021.

Jonas Geiping, Liam Fowl, Gowthami Somepalli, Micah Goldblum, Michael Moeller, and Tom Goldstein. What doesn't kill you makes you robust (er): Adversarial training against poisons and backdoors. In *arXiv*, 2021.

Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. In *TPAMI*, 2022.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. MIT press Cambridge, 2016.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *ICLR*, 2020.

James M. Joyce. Kullback-leibler divergence. In *International Encyclopedia of Statistical Science*, 2011.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. In *arXiv*, 2019.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017.

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

Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning-industry perspectives. In *2020 IEEE Security and Privacy Workshops (SPW)*, 2020.

Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. Data poisoning attacks on factorization-based collaborative filtering. In *NeurIPS*, 2016.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *NeurIPS*, 2021.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *ICLR*, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

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

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Accumulative poisoning attacks on real-time data. In *NeurIPS*, 2021.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *ICML*, 2013.

Andrea Paudice, Luis Muñoz-González, and Emil C Lupu. Label sanitization against label flipping poisoning attacks. In *Joint European conference on machine learning and knowledge discovery in databases*, 2018.

Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *NeurIPS*, 2018.

Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. In *NeurIPS*, 2017.

Lue Tao, Lei Feng, Jinfeng Yi, Sheng-Jun Huang, and Songcan Chen. Better safe than sorry: Preventing delusive adversaries with adversarial training. In *NeurIPS*, 2021.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020.

Yizhen Wang and Kamalika Chaudhuri. Data poisoning attacks against online learning. In *arXiv*, 2018.

Peter Welinder, Steve Branson, Pietro Perona, and Serge Belongie. The multidimensional wisdom of crowds. In *NeurIPS*, 2010.

Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. *arXiv:2003.02460*, 2020.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *ICML*, 2020a.

Xuezhou Zhang, Xiaojin Zhu, and Laurent Lessard. Online data poisoning attacks. In *LDC*, 2020b.

APPENDIX

A PROPERTY INSIGHTS OF MEMORIZATION DISCREPANCY

In this part, we provide the formal analysis on 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 target of poisoning adversary 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 on the given samples. However, considering a model that 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 of 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 distant malicious target as 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 and at round t, k denotes the interval rounds, S denotes a clean set. 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, 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,

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

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 stage 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 for this goal since it naturally reflect the dynamical behavior on 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 meet our requirement described in Section 3.1, i.e., to gain the 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 4, the Memorization Discrepancy of poison samples are more 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 accumulated the target-level discrepancy in model dynamics for better distinguishing poison samples with 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

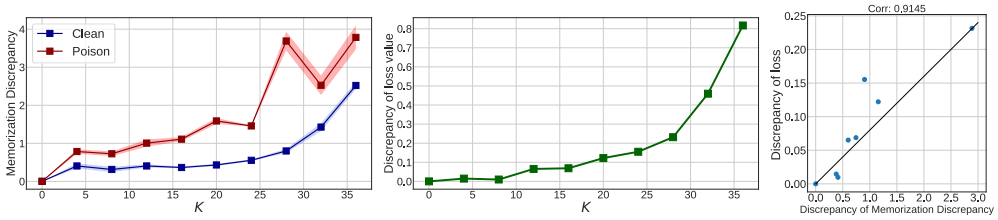

Figure 6: Empirical verification about the property insights on the Memorization Discrepancy.

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 6. It can be found that during the stable training phase (e.g. back from Epoch 40 to Epoch 5) the correlation between the discrepancy on model output and loss values are proportional. In the early stage, we can find some inconsistent relationship exist, we attribute the possible reason to be the unstable optimization which can not accurately reflect the relative distance with the malicious target (training with poison samples) or the ordinary target (training with clean samples).

C Experiments

All the experiments are conducted for multiple independent runs using NVIDIA Tesla V100. To ensure the reproducibility of experimental results, we will provide the link of an anonymous repository about the source codes in the discussion forums for reviewing purposes.

C.1 Details about Accumulative Poisoning Attack

In this part, we describe the details about 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 utilize 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,

$$\max_{\mathcal{P},\mathcal{A}_t} \nabla_\theta \mathcal{L}(\mathcal{A}_t(S_t); \theta_t)^\top \left[\nabla_\theta \mathcal{L}(S_t); \theta_t) + \lambda \cdot \nabla_\theta (\nabla_\theta \mathcal{L}(S_{val}; \theta_T)^\top \nabla_\theta \mathcal{L}(\mathcal{P}(S_T); \theta_T))\right], \tag{20}$$

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)).

C.2 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 5), and summarize the results in Table 2. 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 2: 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.3 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 the similar trend as the left panel of Figure 5. 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 different auxiliary model with same threshold to see how it affect our DSC and summarize the results in Table 3. 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 need further estimate the threshold by small batch of clean data used in previous training stage.

Table 3: Performance of DSC using different auxiliary model 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.4 Ablations 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 4. Since all those variant are designed for further improving adversarial robustness or other issues in adversarial training, its objective all introduce other optimization part which sacrifice the natural performance, the results also demonstrate that the accuracy drop using these AT-variants based methods for accumulative poisoning defense are more severe than the original AT.

Table 4: 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.5 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 5. In this setting, we use other surrogate model (e.g., the historical model eariler than the current model stage) to generate the adversarial examples and feed into the vaccine model. The results show that our DSC has comparable defense effect than with that on white-box setting.

Table 5: 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.6 Empirical Evaluation of the Correction Condition

As for the hyper-parameters μ and τ, in the burn-in phase which 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 this two kind of experiments for the fair evaluation. To provide more informative results, we check the experiments for running the clean oracle with 50 batches samples and summarize the how often the threshold condition is satisfied during training in Table 6. The results show that part of clean samples are also affected by our DSC and their value are satisfy the condition in Algorithm 1. For the experiments with clean oracle, we use the same threshold with the experiments on defending the accumulative poisoning attack. It shows the selective mechanism based on the condition.

Table 6: 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.7 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. In Figure 7, 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 8, we fix the auxiliary model at Epoch 1 and investigate the value of Memorization Discrepancy using different intervals. The overall results show the similar trend with previous analysis, that we can better utilize the model dynamics via enlarge the backtracking interval in computing the Memorization Discrepancy. In Figure 9, we change the different auxiliary models from Epoch 1 to Epoch 28. Although there exist the same trend as previous two explorations, the value of Memorization Discrepancy varies among the different auxiliary models. It can draw the same conclusion with the experiment in Appendix C.3 that we may need further estimate the appropriate threshold for distinguish the clean and poison samples.

D Further Discussion about the Applicability

As for the underlying mechanism of Memorization Discrepancy, it has no special assumption on the types of poisoning generation but reflect 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 can be considered into different defensive methods or detection strategies. We primarily focus on this problem setting in our study since the delusive attack and its corresponding defense are important and of great interest in the related literature. 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 objectives, we would leave expanding our approaches to be one major future work.

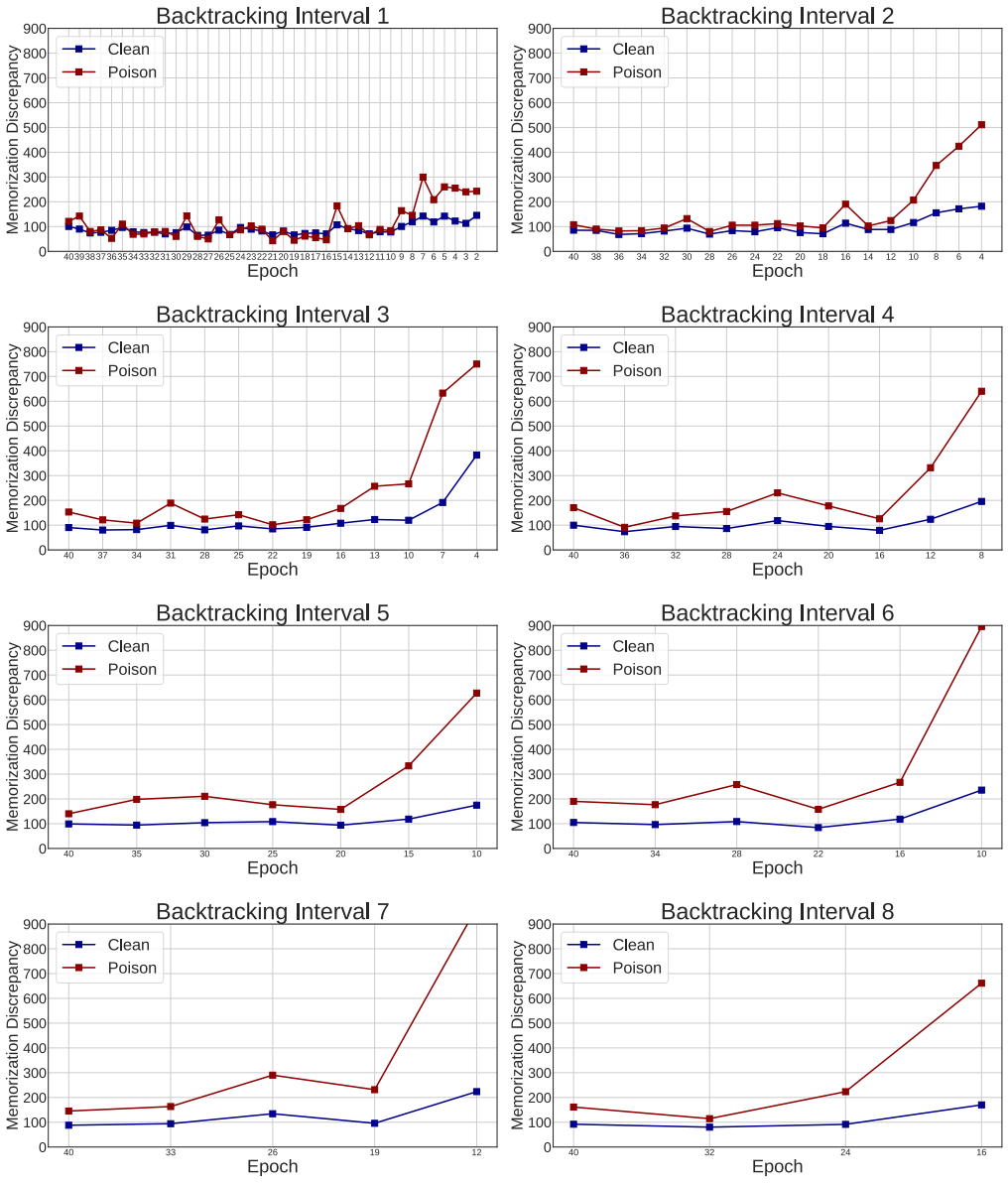

Figure 7: Dynamics of Backtracking Interval on Memorization Difference in CIFAR-10.

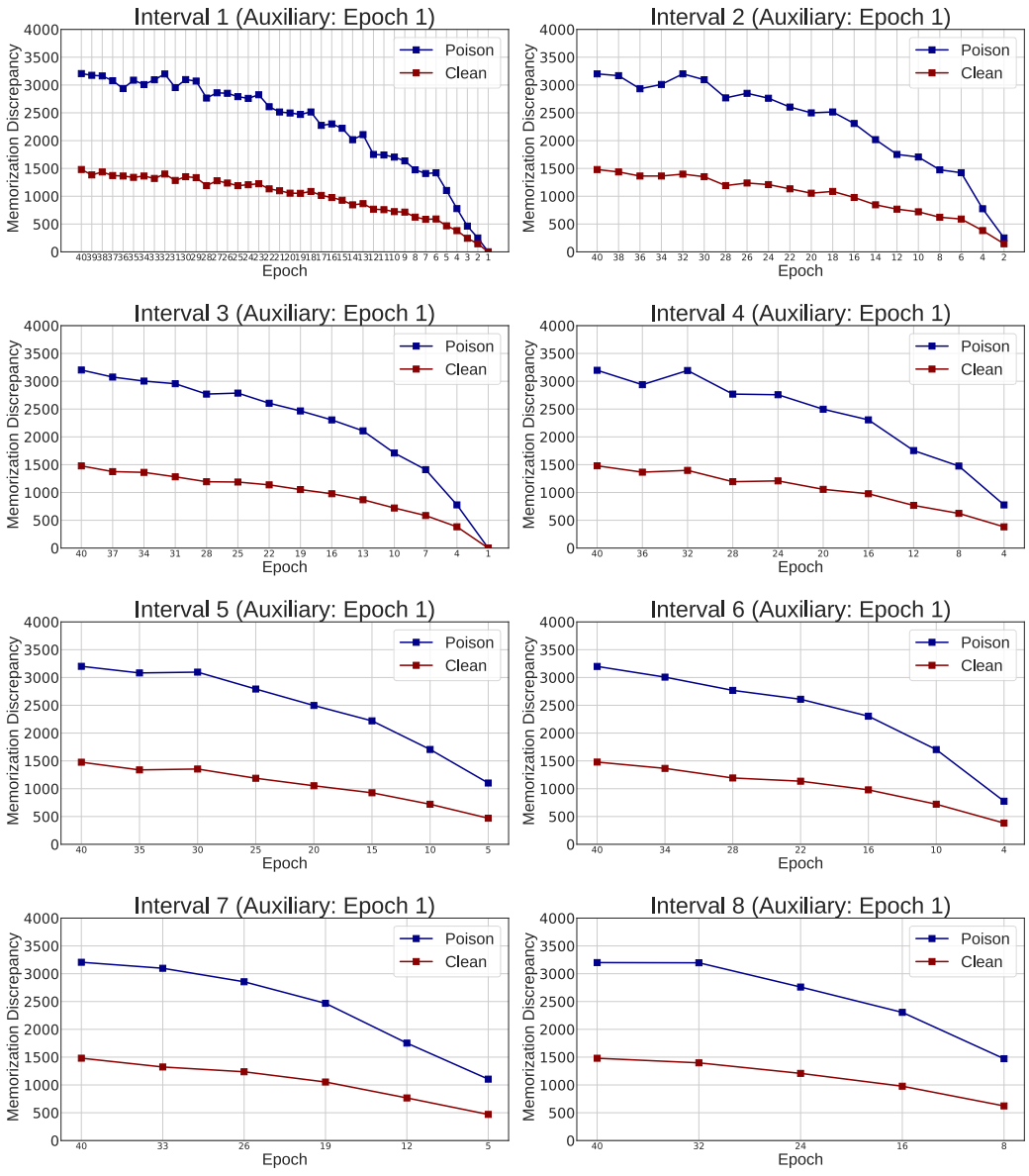

Figure 8: Dynamics of Same Auxiliary Epoch on Memorization Difference in CIFAR-10.

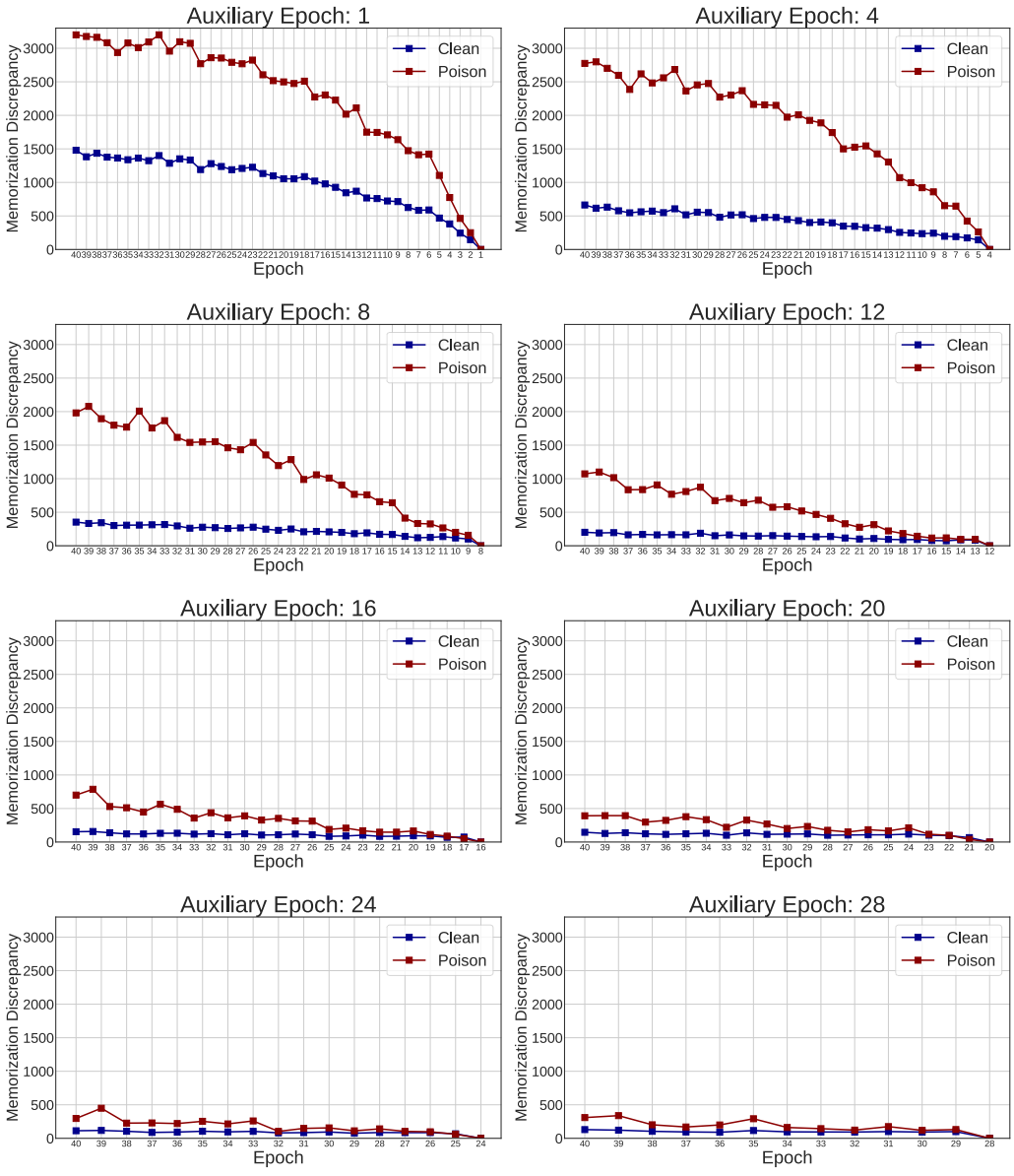

Figure 9: Dynamics of Different Auxiliary Model on Memorization Difference in CIFAR-10.