RDM-DC: Poisoning Resilient Dataset Condensation with Robust Distribution Matching

Abstract

Dataset condensation aims to condense the original training dataset into a small synthetic dataset for data-efficient learning. The recently proposed dataset condensation techniques allow the model trainers with limited resources to learn acceptable deep learning models on a small amount of synthetic data. However, in an adversarial environment, given the original dataset as a poisoned dataset, dataset condensation may encode the poisoning information into the condensed synthetic dataset. To explore the vulnerability of dataset condensation to data poisoning, we revisit the state-of-the-art targeted data poisoning method and customize a targeted data poisoning algorithm for dataset condensation. By executing the two poisoning methods, we demonstrate that, when the synthetic dataset is condensed from a poisoned dataset, the models trained on the synthetic dataset may predict the targeted sample as the attack-targeted label. To defend against data poisoning, we introduce the concept of poisoned deviation to quantify the poisoning effect. We further propose a poisoning-resilient dataset condensation algorithm with a calibration method to reduce poisoned deviation. Extensive evaluations demonstrate that our proposed algorithm can protect the synthetic dataset from data poisoning with minor performance drop.

1 INTRODUCTION

In the past decade, deep learning has significantly contributed to many on-going advances regarding computer vision and natural language processing. Big data is an essential key to unlock the success of deep learning, which though introduces an obstacle to hinder the clients with limited resources from applying deep learning. To address this issue, the community proposed dataset condensation—a research topic studying how to condense the large training dataset into a small synthetic dataset and simultaneously maintain the utility of the synthetic data for model training.

Recent research on this topic has produced several effective dataset condensation techniques [Zhao and Bilen, 2021a, Zhao et al., 2021, Zhao and Bilen, 2021b, Cazenavette et al., 2022, Liu et al., 2022, Cui et al., 2022]. For instance, Zhao et al. [2021] proposed to match model gradients on the synthetic data and the original data, for the purpose of maintaining the performance of the model trained by gradient descent on the synthetic data. Zhao and Bilen [2021a] proposed to match the representations of synthetic and original data. Cazenavette et al. [2022] matched the training trajectories on real data and the training trajectories on synthetic data to learn high-utility synthetic data. Liu et al. [2022] factorized a dataset into data hallucination networks and bases and generated synthetic samples via arbitrary combinations between networks and bases.

Despite the remarkable progress, there has been little investigation on the vulnerability of dataset condensation to attacks in real-world adversarial environments. For instance, in practice, some training data may come from untrusted sources, which gives an adversary opportunities to insert poisoned data into the training dataset. Previous literature [Wallace et al., 2021, Geiping et al., 2021, Zheng and Li, 2021, Schuster et al., 2021] has shown that a small subset of poisoned data can mislead the trained models to output adversary-defined (attack-targeted) predictions on targeted samples in computer vision and natural language processing tasks. Therefore, if the original dataset contains poisoned data from untrusted sources, a natural question to ask is whether dataset condensation will encode the poisoned information into the condensed synthetic dataset.

In this paper, we investigate the most scalable and efficient method among [Zhao and Bilen, 2021a, Zhao et al., 2021, Zhao and Bilen, 2021b, Cazenavette et al., 2022, Liu et al., 2022], which is the distribution matching (DM) method

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

Zhao and Bilen [2021a][*]. We conduct the first study on the vulnerability of distribution matching based dataset condensation against the state-of-the-art targeted data poisoning method, *i.e.,* gradient matching attack [Geiping et al., 2021]. Inspired by distribution matching [Zhao and Bilen, 2021a], we further propose a targeted data poisoning attack to evaluate dataset condensation, which crafts the poisoned perturbation by matching the representations of poisoned data and the targeted sample.

To quantify the effect of data poisoning on dataset condensation, we introduce the concept of **poisoned deviation**, which can be viewed as the major cause of the poisoning vulnerability. We show that, for the DM method, the poisoned deviation in the average of the original data representations scales in $O(\epsilon\sqrt{d_{\boldsymbol{r}}})$, where $\sqrt{d_{\boldsymbol{r}}}$ refers to the representation dimension. The poisoned deviation could convey poisoned information to the synthetic data representations during the representation matching process. As a result, the models trained on the synthetic data may output adversary-defined predictions on the targeted data sample.

To defend against data poisoning, we propose a poisoning resilient dataset condensation algorithm with a calibration method to reduce the poisoned deviation. We prove that, when the poisoned deviation is large with potentially significant impact on the representations, the calibration method can reduce the magnitude of the poisoned deviation from $O(\epsilon\sqrt{d_{\boldsymbol{r}}})$ to $\Theta(\epsilon^2\sqrt{d_{\boldsymbol{r}}})$ and thus alleviate the effect of the poisoned deviation.

We conduct extensive experiments with two targeted data poisoning methods, including the state-of-the-art targeted data poisoning method and our proposed data poisoning method. We show that, even if the poisoning rate is small, distribution matching based dataset condensation is still vulnerable to targeted data poisoning. We further demonstrate that, the distribution matching method is more vulnerable to our proposed distribution matching based attack than the gradient matching based attack. This result indicates that our proposed attack is a more suitable attack benchmark to evaluate the distribution matching based dataset condensation method. We evaluate our proposed defense against those two data poisoning attacks with multiple random seeds and find that our defense is able to reduce the attack success rate to 0% with mild utility loss.

2 BACKGROUND AND RELATED WORK

2.1 DEFINITIONS AND NOTATIONS

In this paper, we denote a neural network by $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot)$ with parameters $\boldsymbol{\theta}$. We denote a data sample by $\boldsymbol{x}$ and its label by y. Let $\mathcal{T}$ and $\mathcal{S}$ represent the original dataset and the synthetic dataset, respectively. For dataset condensation, we have $|\mathcal{S}| \ll |\mathcal{T}|$, which means the size of the synthetic dataset is expected to be much smaller than the size of the original dataset. We denote the targeted data sample and the adversary-defined prediction by $\boldsymbol{x}^t$ and y^{adv}, respectively. We denote the poisoned samples by $\{\boldsymbol{x}_p\}_{p=1}^P$ and the perturbation added to those poisoned samples by $\{\boldsymbol{\delta}_p\}_{p=1}^P$. We refer to the poisoned dataset as $\mathcal{T}'$. Formally, $\mathcal{T}'$ can be expressed as $\mathcal{T}' = (\mathcal{T}/\{\boldsymbol{x}_p, y_{adv}\}_{p=1}^P) \cup \{\boldsymbol{x}_p + \boldsymbol{\delta}_p, y_{adv}\}_{p=1}^P$.

2.2 DATASET CONDENSATION

Dataset condensation is a recently emerged technique for condensing the original training datasets into small synthetic datasets and simultaneously maintaining the data utility for training models to the greatest extent. Given the problem objective, Wang et al. [2018] formulated the dataset condensation problem as

$$\arg\min_{\mathcal{S}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{T}} \ell(\boldsymbol{f}_{\boldsymbol{\theta}(\mathcal{S})}(\boldsymbol{x}), y), \quad \text{where } \boldsymbol{\theta}(\mathcal{S}) = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{S}} \ell(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{s}), y). \tag{1}$$

Wang et al. [2018] proposed to solve the above bi-level problem via meta-learning [Finn et al., 2017]. The proposed solution consists of an inner optimization step to update $\boldsymbol{\theta}$ and a outer optimization step to update $\mathcal{S}$. A drawback of Wang et al. [2018]'s method is the heavy computational cost to involve second-order information in the optimization process. To address this drawback, Nguyen et al. [2020] proposed an algorithm called kernel inducing points (KIP).

Zhao et al. [2021] proposed to match the model gradients on the real and synthetic data for dataset condensation. Since Zhao et al. [2021]'s method includes gradients in the objective, it implicitly uses second-order information in the optimization process. Zhao and Bilen [2021b] further proposed differentiable Siamese augmentation $\mathcal{A}_{\boldsymbol{w}}(\cdot)$ and found that applying $\mathcal{A}_{\boldsymbol{w}}(\cdot)$ with random parameters $\boldsymbol{w}$ to both the original and synthetic data samples can improve the performance of Zhao et al. [2021]'s method.

To avoid using implicit second-order information, Zhao and Bilen [2021a] proposed to match the representations of the original and synthetic data for dataset condensation. Zhao and Bilen [2021a] used the following formula as the matching loss, *i.e.,*

$$\mathbb{E}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\theta}}} \| \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathcal{A}_{\boldsymbol{w}}(\boldsymbol{x}_i)) - \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathcal{A}_{\boldsymbol{w}}(\boldsymbol{s}_i)) \|_2^2, \tag{2}$$

where $\boldsymbol{\Phi}_{\boldsymbol{\theta}}(\cdot)$ refers to the feature extractor (not pretrained), and $\boldsymbol{\theta}$ is sampled from a random parameter distribution $P_{\boldsymbol{\theta}}$.

Cazenavette et al. [2022] proposed to learn the synthetic data by expert training trajectories, which refer to the

[*]We leave the investigations on other dataset condensation methods for future research.

model parameters during the training process. Specifically, [Cazenavette et al., 2022] first trains models on the original dataset and collect the training trajectories. Afterwards, [Cazenavette et al., 2022] trains models on the synthetic dataset and match the training trajectories with those collected from the models trained on the original data. Finally, [Cazenavette et al., 2022] backpropagates the matching loss to optimize the synthetic data. This method achieves better performance than [Zhao and Bilen, 2021a] but requires much more computational cost.

Liu et al. [2022] proposed to factorize a dataset into data hallucination networks and bases and feed the data bases into the hallucination networks to generate synthetic data. This dataset factorization approach can use limited storage to represent more synthetic data compared to the previous methods. We note that, although [Cazenavette et al., 2022, Liu et al., 2022] achieve better testing accuracy than [Zhao and Bilen, 2021a], [Zhao and Bilen, 2021a] is much more efficient and scalable than [Cazenavette et al., 2022, Liu et al., 2022]. This is because [Zhao and Bilen, 2021a] does not need to train any expert models and does not need any second-order derivative information to optimize the synthetic data. **In this paper, we mainly focus on designing a defense for this efficient and scalable dataset condensation technique.**

2.3 TARGETED DATA POISONING

In this paper, we mainly focus on targeted data poisoning. We note that we have also evaluated untargeted data poisoning attacks against distribution matching based dataset condensation, but we find that untargeted attacks could not degrade the performance with a small poisoning rate like 1%. We conjecture that this is because the poisoned deviation is not enough to have a significant overall impact with a small poisoning rate (See discussion in Section 4.2).

The goal of targeted data poisoning is misleading the model trained on poisoned data to output adversary-defined label on certain targeted data. Given this objective, the adversary could formulate the following objective to craft the perturbation on the poisoned data:

$$\arg\min_{\boldsymbol{\delta}_p} \ell(\boldsymbol{x}^t, y^{adv}, \boldsymbol{\theta}_p), \tag{3}$$

$$\text{where } \boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}_p} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{T}'} \ell(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}), y); \|\boldsymbol{\delta}_p\|_2 \leq \epsilon.$$

Huang et al. [2020] proposed to solve above objective by meta learning, with a step to update the model parameters $\boldsymbol{\theta}_p$ and another step to update the perturbation $\boldsymbol{\delta}_p$. A drawback of the meta learning method is that the second step implicitly uses second-order information, leading to heavy additional computational cost. To reduce the computational cost, Zheng and Li [2021] derived a general targeted data poisoning method that only uses first-order information to update the perturbation. However, Zheng and Li [2021]'s method needs to add color perturbation and watermarks to the poisoned images. Geiping et al. [2021] proposed to craft the perturbation by matching the model gradients on the targeted data sample and poisoned data, which achieves the state-of-the-art attack performance. We will detail this gradient matching approach in the next section.

We note that some previous literature claims that if the perturbation is large, it could be easily detected by human beings. However, in practice, there may not be sufficient human labor to inspect every data sample in a dataset. Supposing that the dataset inspector randomly checks 50 samples and the poisoning rate is 1%, then the probability that the inspector could **not** find any poisoned sample is approximately 60%, which is still very high. Therefore, if the poisoning rate is only 1%, the setting of large perturbation is valid.

3 TARGETED DATA POISONING AGAINST DATASET CONDENSATION

3.1 GRADIENT MATCHING

Gradient matching is the state-of-the-art targeted data poisoning method. The basic idea of gradient matching is to match the model gradients on the poisoned data and the targeted data sample. If the gradients are approximately matched, updating the model to decrease the loss on poisoned data will be similar to updating the model to decrease the loss on the targeted sample. The objective of gradient matching based data poisoning can be expressed as

$$1 - \frac{\langle \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}^t, y^{adv}, \boldsymbol{\theta}), \sum_{p=1}^{P} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}_p + \boldsymbol{\delta}_p, y^{adv}, \boldsymbol{\theta}) \rangle}{\|\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}^t, y^{adv}, \boldsymbol{\theta})\| \cdot \|\sum_{p=1}^{P} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}_p + \boldsymbol{\delta}_p, y^{adv}, \boldsymbol{\theta})\|}, \tag{4}$$

where $\{\boldsymbol{x}_p + \boldsymbol{\delta}_p\}_{p=1}^{P}$ are the poisoned data samples. The true label of those poisoned samples is the adversary-defined label y^{adv}. If the attack objective (4) is minimized to a small value, then the average of the model gradients on the poisoning data, *i.e.,* $\frac{1}{P}\sum_{p=1}^{P} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}_p + \boldsymbol{\delta}_p, y^{adv}, \boldsymbol{\theta})$, will have same direction as the model gradient on the targeted data sample and adversary-defined label, *i.e.,* $\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}^t, y^{adv}, \boldsymbol{\theta})$.

To boost the attack performance, the gradient matching attack pre-trains multiple models for different epochs on the clean training data and optimizes the perturbation with the attack objective (4) on those pre-trained models. Besides, the attack also runs the perturbation-crafting process for multiple rounds with different initial values and output the perturbation with the best attack performance. For completeness, we provide the algorithm of gradient matching based data poisoning from [Geiping et al., 2021] in Algorithm 1. Geiping et al. [2021]'s attack is mainly designed for the standard classification model training process not for distribution matching based dataset condensation. Therefore, in

this paper, we design a more suitable targeted data poisoning attack for distribution matching based dataset condensation, as introduced in the next subsection.

Algorithm 1 Gradient Matching based Data Poisoning

Require: Pretrained clean networks $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot)$; targeted sample $\boldsymbol{x}_t$ with targeted label y^{adv}; number of poisoned samples $P \ll N$ (N is the total number of training samples); perturbation bound ϵ; number of restarts R, number of iterations T.
Randomly select P training images with true label y^{adv}, denoted by $\{\boldsymbol{x}_p, y^{adv}\}_{p=1}^{P}$.
for $r = 1$ to R **do**
  Randomly initialize the perturbation $\boldsymbol{\delta}_p^r$ for $\boldsymbol{x}_p$.
  for $t = 1$ to T **do**
    Apply differentiable data augmentation to $\boldsymbol{x}_p + \boldsymbol{\delta}_p^r$
    Compute the objective by (4)
    Update $\boldsymbol{\delta}_p^r$ by a step signed Adam/MSGD with the gradient of (4) w.r.t. $\boldsymbol{\delta}_p^r$.
  end for
  Choose the $\boldsymbol{\delta}_p^r$ with the minimal value of (4) as $\boldsymbol{\delta}_p^*$.
  Replace $\{\boldsymbol{x}_p, y^{adv}\}_{p=1}^{P}$ with $\{\boldsymbol{x}_p + \boldsymbol{\delta}_p^*, y^{adv}\}_{p=1}^{P}$.
end for

3.2 DISTRIBUTION MATCHING POISONING

Our proposed targeted data poisoning method is called distribution matching poisoning (DM poisoning). The core idea of DM poisoning is to match the representations of the targeted data sample and the representations of the poisoned data with the adversary-defined label. Based on this idea, we define the objective of DM poisoning as

$$\mathbb{E}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\theta}}} \|\boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathcal{A}_{\boldsymbol{w}}(\boldsymbol{x}_t)) - \frac{1}{P}\sum_{p=1}^{P} \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathcal{A}_{\boldsymbol{w}}(\boldsymbol{x}_p + \boldsymbol{\delta}_p))\|_2^2. \tag{5}$$

where $\{\boldsymbol{x}_p\}_{p=1}^{P}$ also refer to the poisoned samples with label y^{adv}. After optimizing the perturbation with the above objective, the similarity between the representation of the targeted sample and the representations of the poisoned samples will increase. Since a small proportion of the condensed synthetic data with label y^{adv} is supposed to share similar representations with the poisoned data with label y^{adv} after executing [Zhao and Bilen, 2021a], the representations of those condensed synthetic data samples are also similar as the representation of the targeted data sample. Therefore, DM poisoning is able to encode the poisoning information regarding the targeted sample into the synthetic data.

On top of the above objective, we propose an algorithm (Algorithm 2) to craft poisoned data. **In our DM poisoning attack, we do not need to pretrain multiple models, and thus our attack runs faster than the gradient matching attack.** In each iteration of DM poisoning, we first randomly sample the parameters $\boldsymbol{\theta}$ for the feature extractors $\boldsymbol{\Phi}_{\boldsymbol{\theta}}$. After that, we sample the augmentation parameters $\boldsymbol{w}$ and compute the representations by feeding the original and synthetic data into the augmentation function followed by the feature extractors. Finally, we compute the objective (5) with the representations and update the perturbation by a signed Adam optimizer, which is equivalent to a signed momentum SGD optimizer. To boost the attack performance, we also run the algorithm for multiple rounds and output the poisoning perturbation with the best attack performance. DM poisoning is similar to feature collision attacks [Shafahi et al., 2018, Zhu et al., 2019], but DM poisoning explores far more random feature spaces compared to [Shafahi et al., 2018, Zhu et al., 2019].

Algorithm 2 DM Poisoning

Require: Targeted sample $\boldsymbol{x}_t$ with targeted label y^{adv}; number of poisoned samples $P \ll N$; perturbation bound ϵ; number of restarts R, number of iterations T; feature extractors $\boldsymbol{\Phi}_{\boldsymbol{\theta}}$; parameter distribution $P_{\boldsymbol{\theta}}$; data augmentation $\mathcal{A}_{\boldsymbol{w}}(\cdot)$.
Randomly select P training images with true label y^{adv}, denoted by $\{\boldsymbol{x}_p, y^{adv}\}_{p=1}^{P}$.
for $r = 1$ to R **do**
  Randomly initialize the perturbation $\boldsymbol{\delta}_p^r$ for $\boldsymbol{x}_p$.
  for $t = 1$ to T **do**
    Randomly sample $\boldsymbol{\theta}$ from $P_{\boldsymbol{\theta}}$.
    Randomly sample the augmentation parameters $\boldsymbol{w}$.
    Compute Representations: $\boldsymbol{r}(\boldsymbol{x}_p) = \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathcal{A}_{\boldsymbol{w}}(\boldsymbol{x}_p))$ and $\boldsymbol{r}(\boldsymbol{x}_t) = \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathcal{A}_{\boldsymbol{w}}(\boldsymbol{x}_t))$.
    Update $\boldsymbol{\delta}_p^r$ by a step signed Adam/MSGD with the gradient of (5) w.r.t. $\boldsymbol{\delta}_p^r$.
  end for
  Choose the $\boldsymbol{\delta}_p^r$ with the minimal value of (5) as $\boldsymbol{\delta}_p^*$.
  Replace $\{\boldsymbol{x}_p, y^{adv}\}_{p=1}^{P}$ with $\{\boldsymbol{x}_p + \boldsymbol{\delta}_p^*, y^{adv}\}_{p=1}^{P}$.
end for

4 POISONING-RESILIENT DATASET CONDENSATION

4.1 POISONED DEVIATION

To delve into data poisoning in dataset condensation, we propose the concept of poisoned deviation as follows:

Definition 4.1 (poisoned deviation) *Given a variable or metric $\boldsymbol{z} = \boldsymbol{z}_{\mathcal{D}} + \boldsymbol{z}_{\mathcal{B}}$, if $\boldsymbol{z}_{\mathcal{D}}$ is computed on the original data, and $\boldsymbol{z}_{\mathcal{B}}$ is computed on the poisoned data, then we define $\boldsymbol{z}_{\mathcal{B}}$ as the poisoned deviation of $\boldsymbol{z}$. Poisoned deviation can be viewed as the main cause of the abnormal model behavior desired by the poisoning attack.*

Here we provide a practical example to help readers understand the above definition: A model trainer learns its deep learning model on a poisoned dataset by stochastic gradient descent. In each learning step, the model trainer samples a minibatch from the poisoned dataset. We denote the minibatch by $\{\boldsymbol{x}_i, y_i\}_{i=1}^{L_1} \cup \{\boldsymbol{x}'_j, y'_j\}_{j=1}^{L_2}$, where $\{\boldsymbol{x}_i, y_i\}_{i=1}^{L_1}$ refers to original data, and $\{\boldsymbol{x}'_j, y'_j\}_{j=1}^{L_2}$ refers to poisoned data. The model gradient w.r.t. the model parameters $\boldsymbol{\theta}$ on the minibatch is $\boldsymbol{g} = \frac{1}{L_1+L_2}(\sum_{i=1}^{L_1} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}_i, y_i) + \sum_{j=1}^{L_2} \nabla \ell(\boldsymbol{x}'_j, y'_j))$. According to Definition 4.1, $\frac{1}{L_1+L_2} \sum_{j=1}^{L_2} \nabla \ell(\boldsymbol{x}'_j, y'_j)$ is the poisoned deviation of $\boldsymbol{g}$, which causes the abnormal model behavior desired by the poisoning attack.

4.2 POISONED DEVIATION IN DISTRIBUTION MATCHING

In this paper, we mainly focus on distribution matching, which is the most scalable dataset condensation method among [Zhao et al., 2021, Liu et al., 2022, Zhao and Bilen, 2021a, Cazenavette et al., 2022]. We leave investigation on the other dataset condensation methods for future research. Distribution matching learns the synthetic data by matching the mean of the subsampled original data representations and the mean of the subsampled synthetic data representations, as shown in Eq. 2. However, since the original dataset contains poisoned data samples, $\frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathcal{A}_{\boldsymbol{w}}(\boldsymbol{x}_i))$ is polluted by poisoned deviation.

In the following, we show that the poisoned deviation scales in $O(\epsilon\sqrt{d_{\boldsymbol{r}}})$. We denote the poisoned dataset by $\mathcal{T}'$. Suppose the fraction of the poisoned data samples (poisoning rate) in $\mathcal{T}$ is ϵ, which means there exists $\epsilon|\mathcal{T}'|$ poisoned samples, then we could express $\mathcal{T}'$ as $\mathcal{T}' = \mathcal{O} \cup \mathcal{B}$, where $\mathcal{O} \subset \mathcal{T}$, and B is the poisoned subset, then we have $|B| = \epsilon|\mathcal{T}|$, and $|\mathcal{O}| = (1-\epsilon)|\mathcal{T}|$. The poisoned deviation of the mean estimation could be expressed as

$$\|\frac{1}{|\mathcal{T}|}\sum_{i=1}^{|B|} \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathcal{A}_{\boldsymbol{w}}(\boldsymbol{x}'_i))\|_2 = O(\frac{|B|}{|\mathcal{T}|}\sqrt{d_r}) = O(\epsilon\sqrt{d_{\boldsymbol{r}}}), \tag{6}$$

where $d_{\boldsymbol{r}}$ is the representation dimension. Since the scale of $\|\frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathcal{A}_{\boldsymbol{w}}(\boldsymbol{x}_i))\|_2$ is $O(\sqrt{d_{\boldsymbol{r}}})$, the poisoned deviation could not have a significant overall impact with a small ϵ. But this poisoned deviation is enough to cause the models trained by DM-generated synthetic data to make adversary-defined prediction y^{adv} on a certain sample $\boldsymbol{x}_t$.

4.3 POISONING-RESILIENT DATASET CONDENSATION

We propose an algorithm for poisoning resilient dataset condensation by reducing the poisoned deviation with a mean calibration method. We prove that, by executing the mean calibration method, we could reduce the bound on the poisoned deviation by an order of magnitude. The algorithm is given in Algorithm 3, which is similar to the algorithm of distribution matching. The main difference between Algorithm 3 and the distribution matching algorithm is that we calibrate the mean of the original data representations to reduce the poisoned deviation.

Algorithm 3 RDM-DC: Robust Distribution Matching for Dataset Condensation

Require: Original Dataset $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2 ... \cup \mathcal{T}_C$; the number of classes C; the number of data samples per class N_c; the number of synthetic samples per class M; feature extractors $\boldsymbol{\Phi}_{\boldsymbol{\theta}}$; parameter distribution $P_{\boldsymbol{\theta}}$; data augmentation $\mathcal{A}_{\boldsymbol{w}_c}(\cdot)$.
Initialize $\mathcal{S} = \{\{\boldsymbol{s}_j^c\}_{j=1}^M\}_{c=1}^C$ with random noise from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$
for each iteration **do**
  Sample $\boldsymbol{\theta}$ from $P_{\boldsymbol{\theta}}$ and initialize the loss as $\ell = 0$
  for each class c **do**
    Sample the augmentation parameters $\boldsymbol{w}_c$.
    Sample a minibatch $B_c^{\mathcal{T}}$ from $\mathcal{T}_c$
    Compute Representations: $\boldsymbol{r}(\boldsymbol{x}_i^c) = \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathcal{A}_{\boldsymbol{w}_c}(\boldsymbol{x}_i^c))$ for the minibatch $B_c^{\mathcal{T}}$ and $\boldsymbol{r}(\boldsymbol{s}_j^c) = \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathcal{A}_{\boldsymbol{w}_c}(\boldsymbol{s}_j^c))$ for $S_c = \{\boldsymbol{s}_j^c\}_{j=1}^M$.
    Mean calibration: $\hat{\boldsymbol{\mu}} = \text{Calibrate}(\{\boldsymbol{r}(\boldsymbol{x}_i^c)\}_{i=1}^{|B_c^{\mathcal{T}}|})$
    Compute Loss: $\ell = \ell + \|\frac{1}{M}\sum_{j=1}^M \boldsymbol{r}(\boldsymbol{s}_j^c) - \hat{\boldsymbol{\mu}}\|_2^2$.
  end for
  $\mathcal{S} = \mathcal{S} - \eta\nabla\ell$
end for
Output the synthetic dataset $\mathcal{S} = \{\{\boldsymbol{s}_j^c\}_{j=1}^M\}_{c=1}^C$

The mean calibration method is illustrated in Algorithm 4, which is inspired by [Tran et al., 2018]. Given a batch of original data representations, we first estimate the mean and covariance matrix. After that, we compute the eigenvector of the covariance matrix. Since directly computing the first eigenvector requires heavy computational cost, we approximate the eigenvector using the power method [Van Loan and Golub, 1996] with a few iterations. For completeness, we provide the power method in Algorithm 5.

According to the lemma below (inspired by [Tran et al., 2018]), the mean calibration method is able to reduce the poisoned deviation by an order of magnitude. In the following, We denote the original representation distribution by $\mathcal{D}$ and the poisoned representation distribution by $\mathcal{B}$ and their means by $\boldsymbol{\mu}_{\mathcal{D}}$ and $\boldsymbol{\mu}_{\mathcal{B}}$ respectively. We denote the poisoning rate by ϵ, and we denote the mean and the covariance matrix of $(1-\epsilon)\mathcal{D} + \epsilon\mathcal{B}$ by $\boldsymbol{\mu}_{\mathcal{P}}$ and $\boldsymbol{\Sigma}_{\mathcal{P}}$, respectively. Given those notations, we could present the lemma.

Lemma 4.1 *Assuming that $\mathcal{D}$ and $\mathcal{B}$ have bounded covariance matrices $\boldsymbol{\Sigma}_{\mathcal{D}}, \boldsymbol{\Sigma}_{\mathcal{B}} \leq \sigma^2\boldsymbol{I}$, and their means have*

Algorithm 4 Mean Calibration

Require: Representations $\{\boldsymbol{r}(\boldsymbol{x}_i)\}_{i=1}^N$

1. Estimate the mean and covariance matrix of the representations by $\overline{\boldsymbol{r}} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{r}(\boldsymbol{x}_i)$ and $\boldsymbol{\Sigma}_{\boldsymbol{r}} = \frac{1}{N-1}\sum_{i=1}^N (\boldsymbol{r}(\boldsymbol{x}_i) - \overline{\boldsymbol{r}})^T(\boldsymbol{r}(\boldsymbol{x}_i) - \overline{\boldsymbol{r}})$
2. Compute the top eigenvector of $\boldsymbol{\Sigma}_{\boldsymbol{r}}$, denoted by $\boldsymbol{v}_r$.
3. Assign $|\langle \boldsymbol{r}(\boldsymbol{x}_i) - \overline{\boldsymbol{r}}, \boldsymbol{v}_r\rangle|$ as the score of $\boldsymbol{r}(\boldsymbol{x}_i)$.
4. Filter out $[3\epsilon N]$ samples with the largest scores (remaining data samples are denoted by $\{\boldsymbol{r}(\boldsymbol{x}_i)\}_{i=1}^{N-[3\epsilon N]}$).
5. Recompute the mean $\overline{\boldsymbol{r}} = \frac{1}{N-[3\epsilon N]}\sum_{i=1}^{N-[3\epsilon N]} \boldsymbol{r}(\boldsymbol{x}_i)$

Algorithm 5 Power Method

Require: Covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{r}}$; number of iterations I

Randomly sample a vector $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.
for each iteration **do**
  $\boldsymbol{v} = \boldsymbol{\Sigma}_{\boldsymbol{r}}\boldsymbol{v}$; $\boldsymbol{v} = \boldsymbol{v}/\|\boldsymbol{v}\|_2$.
end for
Output $\boldsymbol{v}$.

an apparent difference, i.e., $\|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{B}}\|_2^2 \geq \frac{\alpha\sigma^2}{\epsilon}$ *where* $\alpha > \frac{2665}{576}$, *then if we drop all the representations that satisfies* $|\langle \boldsymbol{r} - \boldsymbol{\mu}_{\mathcal{P}}, \boldsymbol{v}\rangle| \geq t$ *with a certain* t, *then we can reduce the scale of the poisoned deviation from* $O(\epsilon\sqrt{d_{\boldsymbol{r}}})$ *to* $\Theta(\epsilon^2\sqrt{d_{\boldsymbol{r}}})$.

Lemma 4.1 is similar but not identical to Lemma 3.1 in [Tran et al., 2018]. We provide the proof of Lemma 4.1 and the related lemmas in the supplementary material.

We remark that, if the assumption in Lemma 4.1 does **not** hold, *i.e.,* the difference between the mean of original representations and the mean of the poisoned representations is not significant, the poisoned deviation will be small. In that case, the poisoning attack will not succeed on the distribution matching based dataset condensation method. This is because, if $\|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{B}}\|_2^2 < \frac{\alpha\sigma^2}{\epsilon}$, the poisoned deviation is expected to be $\|\boldsymbol{\mu}_{\mathcal{P}} - \boldsymbol{\mu}_{\mathcal{D}}\|_2 = \epsilon\|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{B}}\|_2 < \sqrt{\alpha\epsilon}\sigma \sim O(\sqrt{\epsilon})$. In practice, we also find that, if the perturbation budget of the poisoned data is small, which indicates that the difference between $\boldsymbol{\mu}_{\mathcal{D}}$ and $\boldsymbol{\mu}_{\mathcal{B}}$ is small, then distribution matching does not show vulnerability to the targeted data poisoning attacks.

We also note that, in practice, it is intractable to compute the t since $\boldsymbol{\mu}_{\mathcal{D}}$ and $\boldsymbol{\mu}_{\mathcal{B}}$ are unknown. But we know that, the proportion of the dropped representations is expected to be approximately 2ϵ, where $\mathcal{B}$ accounts for approximately ϵ of the dropped data, and the other ϵ dropped data comes from $\mathcal{D}$. To ensure that we drop most of the poisoned representations, we set the dropping rate as 3ϵ, which means that we drop $[3\epsilon N]$ representations given totally N representations in each iteration, as shown in Algorithm 3.

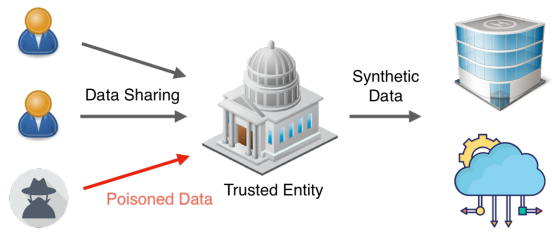

Figure 1: An application of poisoning-resilient dataset condensation. An trusted entity is responsible for collecting the data and condensing the data into synthetic data.

4.4 APPLICATION OF RDM-DC

In this subsection, we discuss the potential applications of our RDM-DC algorithm. One application of the algorithm is to assist the establishment of a trustworthy data supply chain for deep learning. We sketch the supply chain in Fig. 1, where a trusted entity like the government, an agency, or a hospital is responsible for collecting data from the users. After data collection, the trusted entity executes our RDM-DC algorithm on the collected data to generate synthetic datasets and share the synthetic data with the third party to train deep learning models. The benefits of this pipeline shown in Fig. 1 include (1) The pipeline reduces the computational cost required by the third party, which may be devices with limited resources, to train deep learning models. (2) The pipeline somehow mitigates the privacy concerns from the users since the trusted entity only shares synthetic data with the third party. (3) The pipeline addresses the threat of targeted data poisoning with RDM-DC.

Another application of the algorithm is to store the core information from a large amount of data, with tolerance to a small subset of bad samples, into a small storage space. Data explosion is a common problem faced by the many institutions and users since their limited resources can not store the astronomical amount of digital information generated by this world. Instead of deleting the historical data, the institutions and users could condense the historical data into a small synthetic dataset with the RDM-DC algorithm so that they can keep the unpoisoned core information from the historical data with limited resources.

4.5 COMPARISON WITH EMPIRICAL ROBUST AGGREGATION METHODS

We compare the mean calibration method with several robust aggregation methods, including Trimmed Mean, Truncated Mean, and Median [Yin et al., 2018, Portnoy and Hendler, 2020]. The advantage of those empirical methods is that they are very efficient and thus do not add much additional cost to the dataset condensation process. However, the drawback of those methods is that they do not provide any theoretical guarantees, and the poisoned deviation of

ϵ	Attack	Test Acc	ASR
$64/255$	Grad Match	$59.59\% \pm 0.38\%$	$20.00\% \pm 40.00\%$
	DM Poison	$59.25\% \pm 0.33\%$	$60.00\% \pm 48.99\%$
$128/255$	Grad Match	$59.37\% \pm 0.36\%$	$60.00\% \pm 48.99\%$
	DM Poison	$59.23\% \pm 0.40\%$	$100.00\% \pm 0.00\%$

Table 1: Comparing the attack results of gradient matching based data poisoning (Grad Match) and distribution matching based data poisoning (DM poisoning).

the mean may be still large after applying those empirical robust aggregation. For instance, the maximum poisoned deviation of the truncated mean still scales in $O(\epsilon\sqrt{d})$. In this paper, we compare our proposed method with those empirical robust aggregation methods. For completeness, we introduce the empirical robust aggregation methods below.

Truncated Mean Given the representations $\{\boldsymbol{r}(\boldsymbol{x}_i)\}_{i=1}^N$, we first compute the mean by $\overline{\boldsymbol{r}} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{r}(\boldsymbol{x}_i)$. We then compute the distance between each representation $\boldsymbol{r}(\boldsymbol{x}_i)$ and the mean by $d_i = \|\boldsymbol{r}(\boldsymbol{x}_i) - \overline{\boldsymbol{r}}\|_2$. Finally, we drop the $\boldsymbol{r}(\boldsymbol{x}_i)$ with the top-k d_i and average the remaining representations to obtain the truncated mean.

Trimmed Mean Given the representations $\{\boldsymbol{r}(\boldsymbol{x}_i)\}_{i=1}^N$, we refer to the j-th dimension of $\boldsymbol{r}(\boldsymbol{x}_i)$ as $\boldsymbol{r}_j(\boldsymbol{x}_i)$. For each dimension j, we drop off the $k/2$ largest elements and the $k/2$ smallest elements in $\{\boldsymbol{r}_j(\boldsymbol{x}_i)\}_{i=1}^N$ and aggregate the remaining elements to compute the mean for dimension j, *i.e.,* $\overline{\boldsymbol{r}}_j$.

Median Given the representations $\{\boldsymbol{r}(\boldsymbol{x}_i)\}_{i=1}^N$, we also refer to the j-th dimension of $\boldsymbol{r}(\boldsymbol{x}_i)$ as $\boldsymbol{r}_j(\boldsymbol{x}_i)$. For each dimension j, we use the median of $\{\boldsymbol{r}_j(\boldsymbol{x}_i)\}_{i=1}^N$ as $\overline{\boldsymbol{r}}_j$.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Dataset and Poisoning Budget We follow the previous literature on targeted data poisoning [Geiping et al., 2021, Zheng and Li, 2021, Yang et al., 2022, Huang et al., 2020] to mainly conduct evaluations on CIFAR10 [Krizhevsky et al., 2009]. We set the poisoning rate as 1%, which means the proportion of poisoned data in the training dataset is only 1%. With this attack setting, even if a dataset inspector randomly inspect 50 samples, the probability that the inspector can not find a poisoned data sample is approximately $(1-0.01)^{50} \approx 0.605$. *In another word, even if we set a large perturbation size for those poisoned data samples, it is still very likely that the inspector could* not *detect the attack without investing much human labor into the inspection.* Therefore, in our experiments, we set the perturbation size as $64/255$ by default, and we also conduct experiments with other perturbation sizes. Note that in most experiments, this perturbation size is not enough to change any poisoned data sample into the targeted sample since in most cases, even smallest ℓ_∞ distance between the targeted sample and a poisoned data sample is larger than $64/255$.

Figure 2: The targeted images for random seeds $0 \sim 4$. The number before $\rightarrow$ is original label, and the number after $\rightarrow$ is the targeted label (adversary-defined label).

Attack Settings In general, we evaluate the dataset condensation methods against two attack baselines, *i.e.,* gradient matching and our proposed DM poisoning. For gradient matching attack, we pretrain 16 models to craft the perturbation. For both gradient matching attack and DM poisoning attack, we set the attack step size for the sign Adam optimizer as $1/255$ to ensure that the perturbation can be accurately discretized. We randomly select five targeted samples with five random seeds ($0 \sim 4$) and craft the poisoning data for those targeted samples respective. We show the targeted samples and corresponding original and adversary-defined labels in Fig. 2. We employ attack success rate to evaluate the attack performance. For an experiment, if the trained model recognizes the targeted sample as the adversary-defined label, the attack success rate is 100%. Otherwise, the attack success rate is 0%. We compute the mean and standard deviation of the attack success rate over 5×5 (5 random seeds to generate attack datasets and 5 runs of the dataset condensation method for each seed) experiments for evaluating each dataset condensation method.

Dataset Condensation and Defense Settings In this paper, we mainly consider distribution matching based dataset condensation. By default, we set the number of synthetic samples per class as 50. We set the batch size for sampling the original data as 256. We use an SGD optimizer with momentum 0.5 and learning rate 1.0 to update the synthetic data. We follow [Zhao and Bilen, 2021a] to set the number of iterations as 20000. We use Gaussian noise instead of real images to initialize the synthetic data. If using real images that containing poisoning samples for initialization, all the methods will be very vulnerable to poisoning attacks. For the empirical robust aggregation methods, we set k to $\lceil 3\epsilon N \rceil$, where ϵ is the poisoning rate and N is the batch size. We follow [Zhao and Bilen, 2021a, Zhao et al., 2021, Zhao and Bilen, 2021b, Cazenavette et al., 2022, Liu et al., 2022] to train deep learning models on the synthetic dataset and employ the testing accuracy of the models on the original testing dataset to evaluate model performance. To evaluate the defensive performance, as mentioned before, we employ attack success rate (ASR) as the evaluation metric. As afore-

Attack →	DM Poison	
Method ↓	Test Acc	ASR
DM	$59.25\% \pm 0.33\%$	$60.00\% \pm 48.99\%$
DM + Median	$44.08\% \pm 0.57\%$	$60.00\% \pm 48.99\%$
DM + Trim	$51.14\% \pm 0.73\%$	$60.00\% \pm 48.99\%$
DM + Truncated	$57.04\% \pm 0.45\%$	$44.00\% \pm 49.64\%$
RDM-DC	$57.65\% \pm 0.44\%$	$0.00\% \pm 0.00\%$

Table 2: Evaluate the dataset condensation methods and defenses against targeted poisoning attacks with $\epsilon = 64/255$ and poisoning rate 1%.

mentioned, for each targeted sample (random seed), we run five experiments with different random seeds and compute the mean and variance of the attack success rate to evaluate the defensive performance.

5.2 ATTACK PERFORMANCE

We first compare the attack performance of the gradient matching based data poisoning and our proposed DM poisoning on distribution matching based dataset condensation. We provide the attack results in Table 1, which shows that our DM poisoning attack is more effective than gradient matching based data poisoning here. We conjecture that this is because gradient matching based data poisoning is mainly designed for the classification task, and it attempts to match the gradients of cross-entropy loss computed on the targeted sample and poisoned samples. That means the representations of the poisoned samples are not necessarily aligned with the representations of the targeted sample in certain feature spaces. Therefore, feature matching may not be able to encode the poisoning information that can flip the prediction of the targeted sample into the synthetic data. In contrast, DM poisoning directly matches the representations of the poisoned data and the target data sample in broad feature spaces so that feature matching can encode the desired poisoning information into the synthetic data.

We also note that, as we increase the perturbation size to $128/255$, DM poisoning achieves 100% success rate against distribution matching based dataset condensation in all the experiments. This result indicates that the $O(\epsilon\sqrt{d_{\boldsymbol{r}}})$ poisoned deviation is enough to encode adversary-defined information about the targeted sample into the synthetic data.

5.3 DEFENSIVE PERFORMANCE

We evaluate our proposed RDM-DC algorithm and compare it with the empirical robust aggregation methods mentioned in Section 4.5. As indicated by the results in Table 2, Median and Trimmed Mean significantly hurt the utility of synthetic data. We conjecture that this is because the Median and Trimmed Mean methods disregard too much information in the representations as they could not maintain the complete information of any representation in the aggregation process. Truncated Mean is similar to the mean calibration method in the sense that Truncated Mean also disregards some representations and aggregates the remaining representations. The main difference between Truncated Mean and the mean calibration method is that Truncated Mean disregards the representations with large distances to the center (spherical distances), while the mean calibration method disregards the representations with large deviations along the principal vector. With Lemma 4.1, we show that the mean calibration method indeed can reduce the poison deviation, while Truncated Mean does not have this guarantee. Therefore, it is not surprising that RDM-DC has better defensive performance than DM + Truncated Mean—DM poisoning can achieve good attack performance against DM + Truncated Mean for some random seeds.

Attack →	Direct Attack	
Method ↓	Test Acc	ASR
DM	$59.29\% \pm 0.38\%$	$100.00\% \pm 0.00\%$
RDM-DC	$57.79\% \pm 0.45\%$	$0.00\% \pm 0.00\%$

Table 3: Evaluate the dataset condensation methods and defenses against the direct attack.

To demonstrate the outstanding defensive performance of RDM-DC, we further evaluate it against a very strong attack, where we directly use the targeted sample with the adversary-defined label as the poisoned data. We name this strong attack as "direct attack". As shown in Table 3, this strong direct attack can achieve 100% ASR against [Zhao and Bilen, 2021a], but RDM-DC is able to reduce the ASR to 0%, indicating the strong defensive ability of RDM-DC.

6 CONCLUSIONS

In this paper, we study the vulnerability of dataset condensation to targeted data poisoning. We evaluate the existing dataset condensation approaches against the state-of-the-art targeted data poisoning attack, *i.e.,* gradient matching, and our proposed data poisoning attack, *i.e.,* DM poisoning. We demonstrate that only 1% poisoned data can mislead dataset condensation to encode poisoning information into the condensed synthetic dataset. As a result, the models trained on the synthetic dataset will output adversary-defined prediction for the targeted data sample. To quantify the effect of poisoned data, we propose the concept of poisoned deviation and show that the poisoned deviation in distribution matching based dataset condensation scales in $O(\epsilon\sqrt{d_{\boldsymbol{r}}})$. We further propose a poisoning-resilient dataset condensation algorithm with a calibration method to reduce the poisoned deviation to $O(\epsilon^2\sqrt{d_{\boldsymbol{r}}})$. Extensive evaluations demonstrate the effectiveness of our proposed poisoning-resilient algorithm against targeted data poisoning.

References

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4750–4759, 2022.

Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Dc-bench: Dataset condensation benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

Jonas Geiping, Liam H Fowl, W Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching. In *International Conference on Learning Representations*, 2021.

W Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Metapoison: Practical general-purpose clean-label data poisoning. *Advances in Neural Information Processing Systems*, 33:12080–12091, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Songhua Liu, Kai Wang, Xingyi Yang, Jingwen Ye, and Xinchao Wang. Dataset distillation via factorization. In *Advances in Neural Information Processing Systems*, 2022.

Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. In *International Conference on Learning Representations*, 2020.

Amit Portnoy and Danny Hendler. Towards realistic byzantine-robust federated learning. 2020.

Roei Schuster, Congzheng Song, Eran Tromer, and Vitaly Shmatikov. You autocomplete me: Poisoning vulnerabilities in neural code completion. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1559–1575, 2021.

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. *Advances in neural information processing systems*, 31, 2018.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Advances in neural information processing systems*, 31, 2018.

Charles F Van Loan and G Golub. Matrix computations (johns hopkins studies in mathematical sciences). *Matrix Computations*, 53, 1996.

Eric Wallace, Tony Zhao, Shi Feng, and Sameer Singh. Concealed data poisoning attacks on nlp models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

Yu Yang, Tian Yu Liu, and Baharan Mirzasoleiman. Not all poisons are created equal: Robust training against data poisoning. In *International Conference on Machine Learning*, pages 25154–25165. PMLR, 2022.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018.

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. *CoRR*, abs/2110.04181, 2021a.

Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021b.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *ICLR*, 1(2):3, 2021.

Tianhang Zheng and Baochun Li. First-order efficient general-purpose clean-label data poisoning. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021.

Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *International Conference on Machine Learning*, pages 7614–7623. PMLR, 2019.