Poisoned Distillation: Injecting Backdoors into Distilled Datasets Without Raw Data Access

Ziyuan Yang[1,2,4*], **Ming Yan**[2,3], **Yi Zhang**[1,4†], **Joey Tianyi Zhou**[2,3†]

[1]School of Cyber Science and Engineering, Sichuan University, China
[2]Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A*STAR), Singapore
[3]Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), Singapore
[4]Tianfu Jiangxi Laboratory, China
cziyuanyang@gmail.com, yanmingtop@gmail.com, yzhang@scu.edu.cn, joey_zhou@a-star.edu.sg

Abstract

Dataset distillation (DD) condenses large datasets into smaller synthetic ones to enhance training efficiency and reducing bandwidth. DD enables models to achieve comparable performance to those trained on the raw full dataset, making it popular for data sharing. Existing work shows that injecting backdoors during the distillation process can threaten downstream models. However, these studies assume attackers can have access to the raw dataset and interfere with the entire distillation process, which is unrealistic. In contrast, this work is the first to address a more realistic and concerning threat: attackers may intercept the dataset distribution process, inject backdoors into the distilled datasets, and redistribute them to users. While distilled datasets were previously considered resistant to backdoor attacks, we demonstrate that they remain vulnerable to such attacks. Furthermore, we show that attackers do not even require access to any raw data to inject the backdoors successfully within one minute. Specifically, our approach reconstructs conceptual archetypes for each class from the model trained on the distilled dataset. Backdoors are then injected into these archetypes to update the distilled dataset. Moreover, we ensure the updated dataset not only retains the backdoor but also preserves the original optimization trajectory, thus maintaining the knowledge of the raw dataset. To achieve this, a hybrid loss is designed to integrate backdoor information along the benign optimization trajectory, ensuring that previously learned information is not forgotten. Extensive experiments demonstrate that distilled datasets are highly vulnerable to our attack, with risks pervasive across various raw datasets, distillation methods, and downstream training strategies.

Code — https://github.com/Zi-YuanYang/Poisoned_DD
Extended version — https://arxiv.org/abs/2502.04229

Introduction

Deep learning (DL) has achieved remarkable success recently, driven by advancements in computational resources and large-scale datasets (Lei and Tao 2023). With the rise of large language models, such as GPT-3, which has 175 billion parameters and was trained on 45 terabytes of text data using thousands of GPUs for a month (Brown et al. 2020), the demand for computational power and data has reached unprecedented levels. However, the exponential data growth has outpaced computational capacit to challenge training efficiency and costs (LeCun, Bengio, and Hinton 2015).

Dataset distillation (DD) has recently emerged as a promising solution to the challenges posed by large-scale datasets and their computational demands (Du et al. 2024). By synthesizing smaller datasets that retain the essential information of the raw data, DD enables efficient training while significantly reducing storage and computational costs, with minimal impact on model performance (Sun et al. 2024). With advantages such as lower storage, training, and energy costs, DD is expected to become a widely adopted method for data sharing, playing a pivotal role in many machine learning applications (Yu et al. 2024).

Most existing DD methods focus solely on preserving the information of the raw dataset, often overlooking security issues. While these issues have recently garnered some attention from researchers, the number of related studies remains limited. For example, Liu et al. (2023) proposed DoorPing, a learnable trigger that is iteratively updated during the distillation procedure. Similarly, Chung et al. (2024) introduced a standard optimization framework to learn triggers for DD.

However, the threat models of these methods assume that the dataset owner intentionally injects backdoors during the distillation process. In practice, dataset owners are unlikely to compromise their own data by injecting backdoors. Instead, a more plausible threat arises from third-party adversaries. For instance, during dataset distribution, attackers may intercept access to a benign distilled dataset, inject backdoors, and redistribute the compromised version to unsuspecting users, enabling malicious activities. Additionally, highly compact distilled datasets are often considered privacy-preserving and secure (Liu et al. 2023), and making them suitable for storage on various edge devices or clients in distributed learning paradigm. This widespread deployment increases the risk of unauthorized access, facilitating manipulation of the dataset by attackers. Once compromised, the backdoored distilled dataset can be redistributed to other users, thereby amplifying the threat. To highlight

*This work was conducted during Ziyuan Yang's visit to the A*STAR.
†Corresponding author.

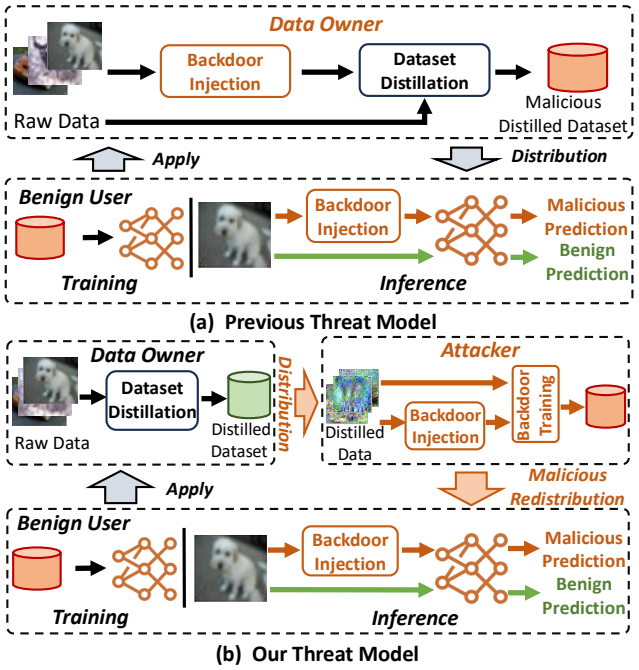

Figure 1: Illustration of threat models. (a) and (b) denote the previous and our threat models, respectively. (a) The previous threat model assumes attackers have access to the raw data and the distillation process. (b) Our model inquires the attackers achieve the attack without the knowledge of the raw data and the distillation process.

the distinction between previous threat models and ours, we provide an illustrative example in Figure 1.

In this work, we consider a practical threat model. Specifically, we attempt to directly inject backdoors into the distilled dataset while ensuring that the malicious behavior can still be triggered by real images. This represents a particularly difficult attack assumption for attackers, as it relies on the premise that the malicious third party does not have access to any raw data. Moreover, the significant gap between synthetic and real images presents an additional challenge.

To address these challenges and evaluate the vulnerabilities of distilled datasets, we propose a novel and the first backdoor attack method specifically designed for this threat model. Under our strict threat model, attackers have no access to raw data. However, the fundamental paradigm of DD involves synthesizing small-scale datasets that retain the knowledge of the raw dataset. This implies that the distilled dataset inherently encapsulates the knowledge of the raw data. While it is almost impossible to reconstruct visually similar images to the raw data without any prior knowledge, the inherent properties of DL enable us to focus on the deep feature space. We only need to ensure that the reconstructed images in the latent feature space share a similar distribution with real images, which allows the trigger to be effectively activated. Leveraging this paradigm, we aim to reconstruct conceptual archetypes for each class, derived from the knowledge embedded in the model trained on the benign distilled dataset, to serve as the foundation of our attack.

Next, we inject backdoors into these conceptual archetypes while ensuring that the modified distilled dataset retains the knowledge of the raw dataset. To achieve this, we propose a hybrid loss function that injects backdoor information into the malicious distilled dataset while preserving the original optimization trajectory. This approach bridges the gap between the distilled dataset and real images, ensuring that the backdoor can be reliably activated by real images while minimizing benign performance degradation.

Notably, our method directly injects backdoors into the distilled dataset without requiring prior knowledge of the DD method, raw data, or the downstream model. Extensive experiments demonstrate that our approach can successfully compromise the security of distilled datasets, regardless of the DD method, downstream model architecture, or training strategy. This finding challenges the prevailing belief that distilled datasets are inherently secure (Liu et al. 2023) and reveals significant security vulnerabilities. Additionally, our attack method is highly lightweight, capable of synthesizing malicious distilled datasets within one minute in certain scenarios. The main contributions can be summarized as:

- We investigate a novel threat model for DD, where backdoors are directly injected into distilled datasets without requiring any raw data access. To the best of our knowledge, this is the first study to explore this threat in DD.
- We propose the first backdoor injection method for distilled datasets that reconstructs conceptual archetypes and injects backdoors while preserving the knowledge of the raw dataset.
- We design a hybrid loss to ensure the backdoor injection aligns with the original optimization trajectory, maintaining backdoor activation in real images while minimizing performance degradation on benign tasks.
- Extensive experiments across diverse datasets, DD methods, networks, and training strategies validate the generalizability of our method. Moreover, our attack is highly efficient, synthesizing malicious distilled datasets in under a minute in certain cases.

Related Works

Dataset Distillation. DD aims to condense the richness of large-scale datasets into compact small datasets that effectively preserve training performance (Yu, Liu, and Wang 2023). Coreset selection (Du, Shi, and Zhou 2024) is an early-stage research in data-efficient learning, which relies on heuristics to select representatives. Unlike this paradigm, DD (Wang et al. 2018) aims to learn how to synthesize a tiny dataset that trains models to perform comparably to those trained on the complete dataset. Wang et al. (2018) first proposed a bi-level meta-learning approach, which optimizes a synthetic dataset so that neural networks trained on it achieve the lowest loss on the raw dataset.

Following this research, many researchers have focused on reducing the computational cost of the inner loop by introducing closed-form solutions, such as kernel ridge regression (Loo et al. 2022; Xu et al. 2023). Zhao, Mopuri, and Bilen (2021) proposed an approach that makes parameters

trained on condensed data approximate the target parameters, formulating a gradient matching objective that simplifies the DD process from a parameter perspective. Meanwhile, Zhao and Bilen (2021) enhanced the process by incorporating Differentiable Siamese Augmentation (DSA), which enables effective data augmentation on synthetic data and results in the distillation of more informative images. Meanwhile, Du, Shi, and Zhou (2024) proposed a sequential DD method to extract the high-level features learned by the DNN in later epochs. By combining meta-learning and parameter matching, Cazenavette et al. (2022) proposed Matching Training Trajectories (MTT) and achieved satisfactory performance. Besides, a recent work, TESLA (Cui et al. 2023), reduced GPU memory consumption and can be viewed as a memory-efficient version of MTT.

Backdoor Attack. Backdoor attacks introduce malicious behavior into the model without degrading its performance on the benign sample. Gu et al. (2019) introduced the backdoor threat in DL with BadNets, which injects visible triggers into randomly selected training samples and mislabels them as a specified target class. To enhance attack stealthiness, Chen et al. (2017) proposed a blended strategy to make poisoned images indistinguishable from benign ones, improving their ability to evade human inspection. Furthermore, subsequent works explored stealthier attacks: WaNet (Nguyen and Tran 2020) used image warping; ISSBA (Li et al. 2021) employed deep steganography; Feng et al. (2022) and Wang et al. (2022) embedded triggers in the frequency domain; and Color Backdoor (Jiang et al. 2023) utilized uniform color space shifts as triggers.

Although existing works have demonstrated the vulnerability of deep networks to backdoor attacks, the exploration of such vulnerabilities in the context of DD remains limited. Only a few studies have evaluated the security risks associated with DD (Liu et al. 2023; Chung et al. 2024). This highlights the urgent need for a deeper investigation into the potential threats and vulnerabilities specific to DD.

Threat Model

In previous works (Liu et al. 2023; Chung et al. 2024), the threat model assumes all users are benign, the data owner is malicious, and the attack has access to the raw data and knowledge of the DD method used. These are highly restrictive and unrealistic assumptions, as raw data and DD methods are typically strictly protected by the owner in practice. In contrast, our threat model adopts a more practical and relaxed assumption, not requiring all users to be benign and permitting the attacker to operate without access to the raw data. A detailed introduction and comparison of the threat models can be found in the ***Appendix***.

Attack Scenario. In our threat model, the attacker intercepts the distribution process and injects backdoor information into the benign distilled dataset. The compromised dataset is then redistributed to users, allowing the attacker to manipulate the behavior of downstream models trained on the malicious dataset.

Attacker's Goal. The primary goal is to inject a backdoor into the distilled dataset, ensuring that downstream models trained on it exhibit malicious behavior when triggered, while maintaining high performance on benign inputs.

Attacker's Capability. In our threat model, attackers do not have access to the raw dataset and can only interact with the distilled dataset, with no prior knowledge of the specific DD method used to generate it.

Challenges. ***i) No Access to Raw Data:*** The attacker has no access to the raw dataset and must infer meaningful information solely from the significantly smaller distilled dataset, often less than one percent of the raw dataset's size. ***ii) Bridging the Gap Between Synthetic and Real Images:*** The distilled dataset is highly abstract and lacks the low-level visual details present in the raw data. The attacker must ensure that the injected backdoors are reliably triggered by real-world images in downstream tasks. ***iii) Maintaining Dataset Utility:*** The modified distilled dataset must remain effective for training models on legitimate tasks, ensuring the backdoor injection does not degrade overall performance.

Proposed Method

Problem Statement

As mentioned earlier, DD aims to extract knowledge from a large-scale dataset and construct a much smaller synthetic dataset, where models trained on it perform similarly to those trained on the raw dataset. Let $\mathcal{T}$ denote the target dataset and $\mathcal{S}$ the synthetic (distilled) dataset, where $|\mathcal{T}| \gg |\mathcal{S}|$, indicating that the distilled dataset is much smaller than the original. The loss between the prediction and ground truth is defined as ℓ. The DD process can then be formulated as (Lei and Tao 2024):

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell\left(\mathcal{M}_{\mathcal{T}}(x), y\right)\right] \simeq \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell\left(\mathcal{M}_{\mathcal{S}}(x), y\right)\right], \quad (1)$$

where $\mathcal{M}_{\mathcal{T}}$ and $\mathcal{M}_{\mathcal{S}}$ denote the downstream model $\mathcal{M}$ trained on $\mathcal{T}$ and $\mathcal{S}$, respectively. $\mathcal{D}$ denotes the real data distribution.

In this paper, we aim to update $\mathcal{S}$ to obtain a malicious synthetic dataset $\hat{\mathcal{S}}$, which is injected with backdoor information. The goal is to ensure that malicious behavior is effectively triggered when a model is trained on $\hat{\mathcal{S}}$. The process can be formulated as:

$$\mathbb{E}_{x\sim\mathcal{D}}\left[\mathcal{M}_{\hat{\mathcal{S}}}(x+T)\right] \approx y_T, \quad (2)$$

where T is the trigger and y_T denotes the target label.

$$\alpha\mathcal{L}_{BA} + (1-\alpha)\mathcal{L}_{tr}. \quad (3)$$

Moreover, for benign samples, the performance gap between models trained on $\mathcal{S}$ and $\hat{\mathcal{S}}$ should remain minimal to conceal the malicious behavior, which can be formulated as:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell\left(\mathcal{M}_{\hat{\mathcal{S}}}(x), y\right)\right] \simeq \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell\left(\mathcal{M}_{\mathcal{S}}(x), y\right)\right]. \quad (4)$$

Overview

The overview of the proposed method is illustrated in Figure 2. As described earlier, our threat model involves three entities: the dataset owner, the attacker, and the benign user. The dataset owner generates a benign distilled dataset $\mathcal{S}$

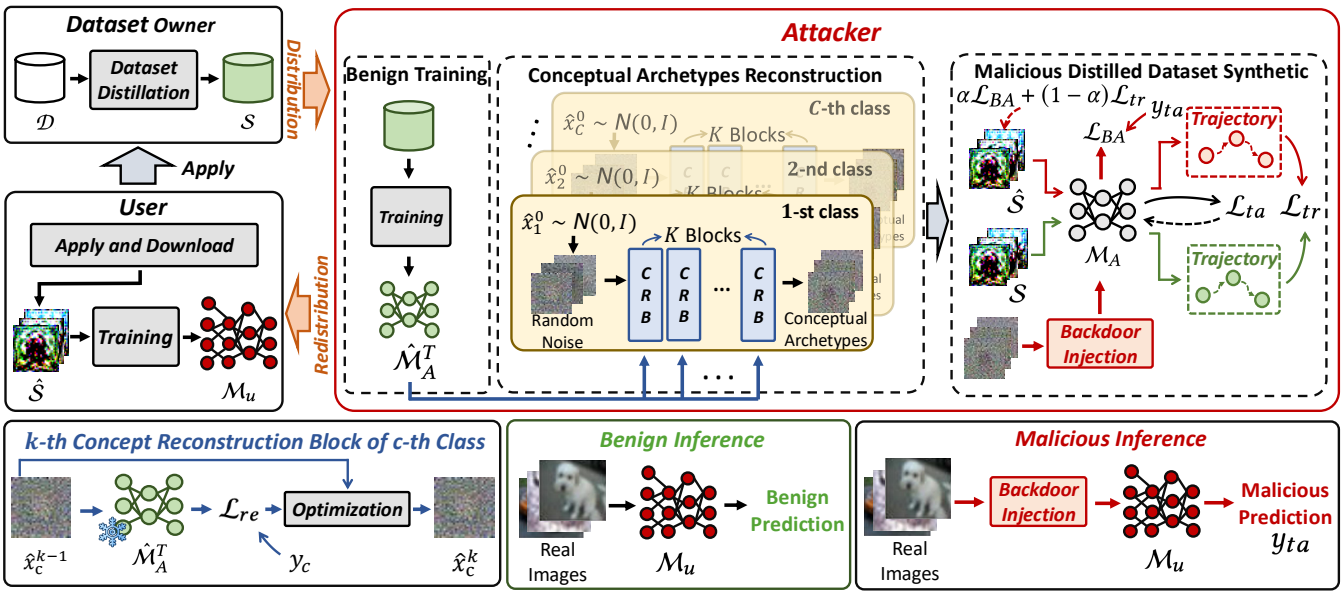

Figure 2: Overview of the proposed method.

from the raw dataset $\mathcal{D}$ and distributes it to users upon request. The attacker intercepts the distribution process and converts the benign distilled dataset into a malicious version.

Specifically, our attack method consists of three main phases. First, the attacker trains a downstream model using the benign distilled dataset $\mathcal{S}$. Next, leveraging the trained model, the attacker reconstructs conceptual archetypes for each class using the proposed Concept Reconstruction Blocks (CRBs). Finally, the attacker injects backdoor information into reconstructed conceptual archetypes and employs a hybrid loss to update the distilled dataset, ensuring that the backdoor is embedded while minimizing performance degradation. Once the malicious distilled dataset is created, it is redistributed to users.

The benign user then trains the local model $\mathcal{M}_u$. Finally, the attacker can target the user-side system by injecting triggers into real images, activating malicious behaviors in $\mathcal{M}_u$.

Proposed Attack Method

Our attack method consists of three main phases, which work together to effectively inject backdoor information while preserving the knowledge from the raw dataset. We detail each phase in the following sections:

Benign Training. After intercepting the distribution, the attacker first trains a benign downstream model using the distributed distilled dataset. The attacker-side trained downstream model is defined as $\hat{\mathcal{M}}_A^T$, which is the foundation of the subsequent phases.

Conceptual Archetypes Reconstruction. Under our strict assumption, the attacker has no access to real images and can only leverage the distilled dataset. However, during the inference phase, the system's input typically consists of real images. This raises a critical question: ***How can the backdoor be activated when injected into real images without relying on any raw data during backdoor training?***

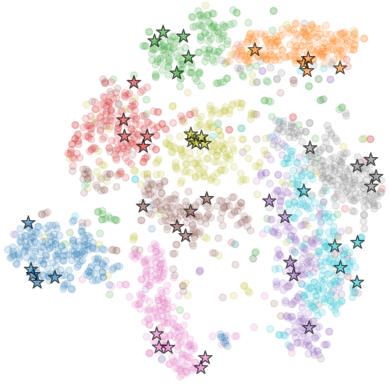

Figure 3: t-SNE visualization of the feature space. "Stars" and "Circles" represent the concept archetypes and real images, respectively.

To bridge the gap between distilled and real data, we propose reconstructing conceptual archetypes for each class. Although generating low-level, semantically similar images without access to raw data is infeasible, this limitation is not critical. In deep networks, accurate classification primarily relies on ensuring that latent feature representations of conceptual archetypes closely align with those of real images.

The reconstruction process aims to generate conceptual archetypes for each class by iteratively refining random noise to align with the high-level feature representations of the target class in $\hat{\mathcal{M}}_A^T$. Specifically, for the c-th class, the process consists of K **Concept Reconstruction Blocks (CRBs)**, each corresponding to an optimization step. The conceptual archetype initialization process for each class c can be formulated as:

$$\hat{x}_c^0 \sim \mathcal{N}(0, I), \tag{5}$$

where $\mathcal{N}(0, I)$ represents a Gaussian distribution with zero mean and identity covariance matrix. $\hat{x}_c^0 \in \mathbb{R}^{C \times H \times W}$ de-

notes the initialized conceptual archetype for the c-th class, where C, H, and W denote the channel, height, and width of the distilled data, respectively.

In the k-th CRB block, $\hat{x}_c^{k-1}$ is optimized to align the model's output with the c-th class representation. The optimization objective is defined as follows:

$$\mathcal{L}_{\text{re}}(\hat{x}_c^{k-1}, c) = -y_c \log\left(\hat{\mathcal{M}}_A^T(\hat{x}_c^{k-1})_c\right), \tag{6}$$

where $\mathcal{L}_{\text{re}}$ is the reconstruction loss, y_c represents the one-hot encoded label for class c.

The optimization process can be formulated as:

$$\hat{x}_c^k = \hat{x}_c^{k-1} - \eta \cdot \nabla_{\hat{x}_c^{k-1}} \mathcal{L}_{\text{re}}(\hat{x}_c^{k-1}, c), \tag{7}$$

where η is the learning rate, and $\nabla_{\hat{x}_c^{k-1}} \mathcal{L}_{\text{re}}(\hat{x}_c^{k-1}, c)$ represents the gradient of the reconstruction loss with respect to the input $\hat{x}_c^{k-1}$.

After K iterations, the reconstructed image $\hat{x}_c^K$ serves as the conceptual archetype for class c. This process is repeated m times for each class to generate m archetypes, with m set to 5 in this paper. Figure 3 illustrates a t-SNE visualization comparing the deep features of the archetypes with real MNIST (LeCun et al. 1998) images. The results show that the reconstructed archetypes closely align with the deep feature representations of real images, effectively bridging the gap between the distilled dataset and real images.

Malicious Distilled Dataset Synthesis. The goal of the attack is to synthesize a malicious distilled dataset such that the backdoor can be effectively activated by real images while maintaining the utility of the dataset for benign tasks. By reconstructing conceptual archetypes to bridge the gap between the distilled and real data, we can leverage them to embed malicious knowledge into the distilled dataset.

Specifically, for each conceptual archetype $\hat{x}$, we obtain the backdoored sample $\hat{x}'$ as follows:

$$\hat{x}'(h,w) = \begin{cases} v, & \text{if } h \geq H - t \text{ and } w \geq W - t, \\ \hat{x}(h,w), & \text{otherwise}, \end{cases} \tag{8}$$

where v and t represent the trigger value and size.

Then, a backdoor loss is designed to embed malicious information into the distilled dataset, ensuring that the backdoor behavior is learned by the model trained on the modified data. The backdoor loss is defined as:

$$\mathcal{L}_{BA} = -y_{ta} \log\left(\mathcal{M}_A(\hat{x}')_{ta}\right), \tag{9}$$

where y_{ta} represents the backdoor target label, and $\mathcal{M}_A$ is the attacker-side model, trained from scratch.

To conceal malicious behavior, it is crucial to minimize performance degradation. This is achieved by ensuring that the optimization trajectory of downstream models trained on the malicious distilled dataset closely matches those trained on the benign dataset. A trajectory consistency loss is introduced to enforce this alignment as follows:

$$\mathcal{L}_{tr} = \frac{1}{|\Theta|} \sum_{\theta \in \Theta} \|\nabla_\theta \mathcal{L}_{ta}(\mathcal{S}) - \nabla_\theta \mathcal{L}_{ta}(\hat{\mathcal{S}})\|^2, \tag{10}$$

where Θ denotes the set of model parameters of $\mathcal{M}_A$, $\mathcal{L}_{ta}$ represents the loss of the downstream task.

By constraining $\mathcal{L}_{tr}$, we can ensure that the malicious dataset maintains a similar optimization trajectory to the benign dataset, thereby concealing malicious behavior while minimizing the impact on the performance of downstream tasks. Finally, we combine both losses to form the overall objective for synthesizing the malicious distilled dataset. The hybrid loss function is defined as follows:

$$\mathcal{L}_{hybrid} = \alpha \mathcal{L}_{BA} + (1 - \alpha)\mathcal{L}_{tr}, \tag{11}$$

where α is the balancing parameter that controls the trade-off between embedding malicious information and maintaining trajectory consistency.

Then, $\hat{\mathcal{S}}$ is iteratively updated to minimize $\mathcal{L}_{hybrid}$ as:

$$\hat{\mathcal{S}} \leftarrow \hat{\mathcal{S}} - \eta \cdot \nabla \mathcal{L}_{\text{hybrid}}. \tag{12}$$

These steps are repeated for N iterations within a single epoch. To ensure that model $\mathcal{M}_A$ follows the next benign optimization trajectory, it is updated on $\mathcal{S}$ after each epoch. This entire process is repeated for E epochs.

Implementation. Once the attacker synthesizes the malicious distilled dataset $\hat{\mathcal{S}}$, it is redistributed to the users. Users then train their downstream models $\mathcal{M}_u$ on $\hat{\mathcal{S}}$ using their own training strategies. During the inference phase, the malicious behavior is activated when the trigger is injected into real images following Eq. (8) to produce malicious outputs aligned with the attacker's target, while maintaining normal performance on benign inputs.

Notably, our method remains effective even when $\mathcal{M}_u$ and $\mathcal{M}_A$ have different architectures. Moreover, it does not require fine-tuning any DD process on the dataset owner's side, nor does it require access to raw data. Therefore, our method is versatile and practical across various scenarios.

Experiments

Experimental Setting

Experiment Environment. Our method is implemented using the PyTorch framework and optimized with Stochastic Gradient Descent (SGD) (Kingma and Ba 2014) with a learning rate of 0.01. The malicious dataset is synthesized over 10 epochs. Experiments are conducted on a system equipped with an NVIDIA RTX 3090.

Experiment Setting. We use ConvNet (Krizhevsky, Sutskever, and Hinton 2012) as the default attacker-side downstream model. Additionally, AlexNet (Krizhevsky, Sutskever, and Hinton 2017), VGG11 and VGG16 (Simonyan and Zisserman 2014), ResNet18 and ResNet34 (He et al. 2016) are used as the user-side downstream networks.

Datasets. To assess the generalization ability of our method across different raw datasets, we evaluate it on CIFAR-10 (Krizhevsky, Hinton et al. 2009), CIFAR-100 (Krizhevsky, Hinton et al. 2009), MNIST (LeCun et al. 1998), FashionMNIST (Xiao, Rasul, and Vollgraf 2017), and SVHN (Netzer et al. 2011).

DD Methods. To evaluate the generalizability of our method, we test it across several different representative DD methods, including DC (Zhao, Mopuri, and Bilen 2021), DM (Zhao and Bilen 2023), DSA (Zhao and Bilen 2021), and MTT (Cazenavette et al. 2022).

	IPC	Epoch / Metric	10	20	30	40	50	60	70	80	90	100
CIFAR10	1	Baseline	26.88	27.25	27.56	27.85	27.78	27.85	27.83	27.45	27.52	27.77
		BA	25.07	25.47	25.42	25.50	25.43	25.82	25.31	25.71	25.32	25.79
		ASR	99.82	99.99	100.00	99.99	100.00	100.00	100.00	100.00	100.00	100.00
	10	Baseline	26.57	31.85	35.62	38.40	39.93	40.86	41.49	41.99	42.77	42.63
		BA	25.82	30.22	32.59	34.07	34.87	35.24	35.32	35.98	35.60	35.92
		ASR	99.99	100.00	100.00	100.00	100.00	100.00	99.99	100.00	100.00	100.00
CIFAR100	1	Baseline	3.36	6.13	7.99	9.05	9.46	10.22	10.52	10.69	11.11	10.77
		BA	3.05	5.95	7.24	8.36	8.59	9.19	9.39	9.75	9.81	9.97
		ASR	22.56	87.38	96.93	98.89	99.26	99.80	99.49	99.85	99.57	99.83
MNIST	1	Baseline	70.69	79.73	82.19	84.91	85.92	86.38	86.75	87.27	87.89	88.81
		BA	65.69	70.42	73.91	75.04	76.65	77.00	77.55	78.38	78.75	79.45
		ASR	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	10	Baseline	69.75	80.26	83.67	86.17	89.32	91.49	93.23	94.63	95.17	95.72
		BA	63.76	75.31	79.98	82.05	85.09	87.51	89.35	90.07	90.39	90.41
		ASR	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	50	Baseline	78.09	85.41	90.01	93.57	95.09	95.98	96.69	97.13	97.44	97.74
		BA	64.23	68.67	74.48	79.52	85.17	87.25	88.49	88.94	89.32	89.31
		ASR	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
FashionMNIST	1	Baseline	67.51	69.29	69.45	69.81	69.88	70.10	69.94	69.86	69.84	69.98
		BA	61.75	63.29	63.72	63.66	63.66	63.70	63.37	63.93	63.94	63.93
		ASR	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	10	Baseline	60.92	66.27	69.41	72.39	74.29	75.51	76.41	77.61	78.15	78.81
		BA	59.84	65.94	69.02	69.83	71.02	71.12	70.89	71.36	71.17	71.42
		ASR	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	50	Baseline	65.95	69.32	71.54	72.94	74.73	76.17	77.16	77.85	78.71	79.40
		BA	63.04	67.18	69.22	69.95	69.96	70.01	69.32	69.58	69.37	69.49
		ASR	100.00	100.00	100.00	100.00	100.00	99.94	99.93	99.64	99.58	99.51
SVHN	1	Baseline	29.10	31.58	31.37	30.63	29.91	29.90	30.51	29.47	30.20	30.64
		BA	29.30	30.59	29.70	29.05	28.69	29.57	28.88	28.99	29.56	29.47
		ASR	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

Table 1: Experimental results of different strategies and different raw datasets with DC (Avg, %).

Metrics. We use benign accuracy (BA) to measure performance on benign samples and attack success rate (ASR) to assess the attack's effectiveness. Neural networks exhibit inherent randomness due to variations in random seeds. To ensure result reliability, we conduct experiments with 10 different seeds and report the average (Avg) and standard deviation (STD) of the performance metrics.

Experiments about Different Training Strategies

In this subsection, we evaluate the impact of different training strategies on the effectiveness of our attack based on DC. Specifically, we analyze performance across different numbers of training epochs in user-side training, and we treat the performance of models trained directly on the benign distilled dataset as the baseline. As shown in Table 1, our attack remains effective across different training epochs, maintaining a high ASR with minimal BA degradation. Additionally, our method demonstrates strong generalizability across various raw datasets and IPC settings, ensuring robustness in diverse scenarios. Experiments on training strategies and different user-side models can be found in the ***Appendix***.

Experiment with Different DD Methods

To further validate our effectiveness, we extend our experiments to different DD methods, with the results summarized in Table 2. We conduct evaluations using user-side training strategies of 50 and 100 epochs. As shown in Table 2, our attack consistently demonstrates strong performance across various DD methods. In most cases, the attack achieves nearly 100% ASR, effectively embedding the backdoor into the distilled dataset, regardless of the specific DD approach employed. Additionally, the BA degradation remains within an acceptable range, which indicates that the overall utility of the dataset is well preserved. These results confirm the generalizability and robustness of our proposed attack method, demonstrating its effectiveness across different distillation strategies while maintaining the performance of downstream tasks.

Visualization

Figure 4 compares benign and malicious distilled datasets. The first row shows benign distilled images, while the second row illustrates the malicious versions after backdoor injection. Due to the inherent abstraction of distilled datasets, these images inherently lack fine-grained details, making it challenging for users to discern their authenticity based

Method	Dataset	IPC		ConvNet		AlexNet		VGG11		VGG16		ResNet18		ResNet34	
				50	100	50	100	50	100	50	100	50	100	50	100
DC	CIFAR10	1	Baseline	27.78	27.77	18.77	17.56	24.47	25.66	14.16	14.42	16.96	17.93	21.01	21.58
			BA	25.43	25.79	17.68	16.57	23.24	23.17	12.90	13.37	14.62	16.49	19.45	19.17
			ASR	100.00	100.00	68.35	65.03	100.00	100.00	99.40	98.90	51.77	51.24	89.41	99.21
		10	Baseline	39.93	42.63	12.79	21.48	34.67	35.27	23.71	26.16	17.20	18.52	22.08	22.88
			BA	34.87	35.92	22.26	16.38	28.48	29.07	20.76	22.15	14.02	14.25	17.73	18.47
			ASR	100.00	100.00	96.84	41.41	100.00	100.00	100.00	100.00	75.51	88.00	99.95	99.97
	CIFAR100	1	Baseline	9.46	10.77	1.21	1.68	8.41	9.01	3.15	4.61	1.53	1.67	2.26	3.16
			BA	8.59	9.97	2.11	1.33	7.06	7.85	2.85	4.09	1.34	1.39	1.71	2.30
			ASR	99.26	99.83	33.27	0.00	97.67	99.17	91.98	87.85	0.40	0.03	22.94	15.05
	FashionMNIST	1	Baseline	69.88	69.98	52.15	30.74	59.75	62.72	23.80	31.16	57.98	57.29	61.80	61.92
			BA	63.66	63.93	29.17	20.30	55.05	55.46	21.60	27.60	50.18	51.59	54.87	53.47
			ASR	100.00	100.00	74.46	68.27	100.00	100.00	100.00	100.00	99.66	98.43	100.00	100.00
		10	Baseline	74.29	78.81	25.84	53.17	77.46	78.00	56.81	63.27	56.49	57.77	59.97	62.40
			BA	71.02	71.42	19.12	34.29	70.53	70.55	46.80	55.02	42.64	46.31	46.77	47.26
			ASR	100.00	100.00	74.49	64.85	100.00	100.00	100.00	100.00	99.60	100.00	100.00	100.00
DM	CIFAR10	50	Baseline	48.50	54.19	20.23	35.50	42.50	42.97	28.18	29.88	25.77	26.11	26.36	27.79
			BA	33.42	35.26	20.39	23.99	29.00	29.45	18.35	20.23	15.92	16.04	17.92	18.49
			ASR	99.91	99.25	84.05	97.89	100.00	100.00	100.00	100.00	99.88	98.93	100.00	100.00
DSA	CIFAR10	1	Baseline	26.24	26.69	19.25	17.04	21.99	22.23	13.53	15.07	23.49	25.00	21.18	22.06
			BA	24.09	24.70	18.65	16.50	19.01	21.60	12.24	13.12	20.89	22.48	18.24	19.38
			ASR	100.00	99.99	68.90	67.72	97.38	99.76	89.64	90.52	96.27	99.83	89.81	97.34
		10	Baseline	38.17	44.23	14.48	27.51	33.25	37.30	19.60	24.15	24.53	29.56	21.10	25.50
			BA	32.60	34.82	17.92	22.32	26.54	31.21	17.80	19.15	19.62	23.82	18.70	20.40
			ASR	100.00	100.00	46.18	47.10	100.00	100.00	99.99	100.00	98.22	100.00	99.34	100.00
	CIFAR100	1	Baseline	8.78	9.99	1.24	2.55	6.15	8.27	2.07	3.11	2.94	5.63	3.31	5.15
			BA	7.66	9.10	2.41	1.97	5.79	7.35	2.23	3.05	2.20	4.07	3.14	4.27
			ASR	95.31	98.43	19.75	1.21	90.56	99.69	82.52	98.61	16.68	79.71	84.15	82.86
		10	Baseline	16.28	23.28	5.84	14.69	11.79	17.81	4.43	7.17	6.15	7.83	5.58	7.61
			BA	7.31	8.35	7.14	6.89	7.23	8.23	3.18	4.87	3.85	4.96	3.53	4.83
			ASR	42.76	41.90	35.68	25.76	96.23	60.21	97.17	75.47	95.90	75.79	98.16	67.59
	FashionMNIST	1	Baseline	67.89	69.22	42.81	45.75	53.03	57.14	21.53	27.98	62.66	66.32	57.96	58.88
			BA	61.32	62.55	32.27	32.55	47.77	49.78	19.57	28.21	56.31	57.36	49.29	50.07
			ASR	100.00	100.00	83.07	67.64	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
		10	Baseline	73.23	77.73	19.56	61.82	67.66	79.22	52.64	58.65	64.18	70.63	65.91	76.19
			BA	71.69	73.11	28.53	41.16	66.85	69.86	47.12	55.45	51.91	60.03	48.86	62.15
			ASR	100.00	100.00	79.51	80.63	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
MTT	CIFAR10	1	Baseline	38.76	38.80	10.87	14.16	17.97	21.49	11.37	11.04	14.02	14.79	16.76	16.99
			BA	31.87	32.58	11.27	12.27	18.11	19.73	10.62	11.52	12.29	13.82	16.28	16.45
			ASR	100.00	100.00	70.29	72.69	100.00	100.00	100.00	98.89	59.24	92.44	99.94	99.64
		10	Baseline	43.81	51.65	16.82	26.73	33.44	34.35	23.33	25.66	15.58	16.17	19.53	21.22
			BA	34.81	37.81	24.13	11.66	25.00	25.92	17.01	20.44	12.33	12.74	14.73	16.06
			ASR	100.00	100.00	94.08	54.31	100.00	100.00	100.00	100.00	79.19	71.56	100.00	100.00

Table 2: Experimental results based on different DD methods and different user-side models (Avg, %).

on individual distilled samples. This abstraction aids the attack, as the malicious changes are subtle and hard to detect. Although these perturbations are minor, the backdoor triggers remain effective, ensuring the model responds to the attacker's intended inputs.

Ablation Study

In previous experiments, we used ConvNet as the attacker's downstream model. In this experiment, we evaluate the impact of different architectures on the effectiveness of our attack. To demonstrate our generalizability, we conduct experiments on CIFAR-10 distilled using different DD methods, with IPC set to 10. The results in Table 3 indicate that our attack remains highly effective across various model architectures. It can be seen that our threat model operates under

Model		DC	DSA	MTT
AlexNet	Baseline	21.48±0.729	27.51±0.830	26.73±1.160
	BA	16.56±6.264	21.65±1.919	19.86±3.226
	ASR	23.04±38.896	21.36±16.998	18.33±18.708
VGG11	Baseline	25.66±0.952	35.27±0.497	34.35±1.110
	BA	23.58±0.991	29.01±1.107	20.54±1.054
	ASR	68.89±20.244	49.77±26.597	77.21±20.196
VGG16	Baseline	26.16±1.630	24.15±1.801	25.66±1.848
	BA	21.01±2.285	21.69±1.418	21.57±1.493
	ASR	56.70±37.613	17.99±24.918	73.78±35.649

Table 3: Attack performance with different attacker-side downstream models (Avg ± STD, %).

	Performance (%)		Time (s)		
	BA	ASR	Per Image	Per Epoch	All
$m = 5$	25.79	100.00	0.53	1.50	$0.53 \times 5 \times 10 + 1.50 \times 10 = 41.5$
$m = 10$	25.59	100.00	0.53	1.65	$0.53 \times 10 \times 10 + 1.65 \times 10 = 69.50$
$m = 20$	25.36	100.00	0.53	1.96	$0.53 \times 20 \times 10 + 1.96 \times 10 = 125.60$

Table 4: Computational complexity and attack performance with varying numbers of conceptual archetypes.

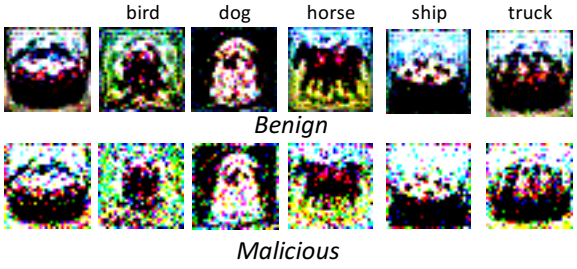

Figure 4: Visualization of benign and malicious distilled data: Without comparison, users may struggle to detect subtle differences due to the abstraction of distilled datasets.

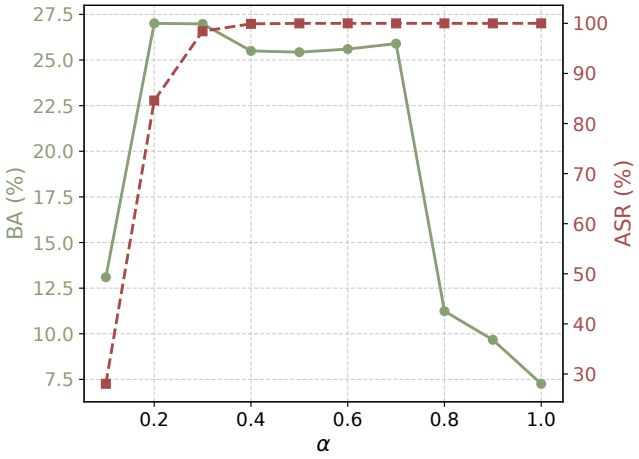

Figure 5: The performance under different α in Eq. (11).

relatively weak assumptions, making it highly practical in real-world scenarios. Despite these relaxed constraints, our attack maintains strong performance across various settings.

To analyze the impact of different components in our method, we conduct an ablation study on the effect of α in Eq. (11), which balances the tradeoff between attack effectiveness and benign performance. In this experiment, we use the distilled CIFAR-10 dataset by DC, with IPC set to 1. The results in Figure 5 show that as α increases, ASR remains high, but performance degradation on benign tasks becomes more pronounced. This occurs because a larger α emphasizes backdoor retention, potentially sacrificing the distilled dataset's utility. To achieve an optimal balance, we set α to 0.5 as the empirical recommendation. Besides, further experiments can be found in our ***Appendix***.

Computational Complexity

Our attack method is efficient and lightweight. We evaluate its computational cost on a distilled CIFAR-10 dataset by the DC method with IPC set to 1. The complexity depends on the number of reconstructed archetypes, with results summarized in Table 4. Different archetype numbers achieve effective attacks with minimal benign impact. We recommend using $m = 5$ for a balance between efficiency and attack effectiveness.

Our attack method consists of two phases: conceptual archetype reconstruction and malicious distilled dataset synthesis. Reconstructing each archetype takes 0.53s, this phase takes about 26.5s for CIFAR-10 under the default setting of five archetypes per class. In the second phase, each epoch of malicious dataset synthesis takes 1.5s on an NVIDIA GTX 3090, allowing the entire attack to complete in under one minute. Consequently, the entire attack can be completed in less than one minute. This minimal time overhead and the delay makes the attack virtually imperceptible to users.

Conclusion

In this paper, we propose a novel backdoor attack method for distilled datasets, enabling backdoor injection without access to raw data, knowledge of the DD process, or changes to the data owner's pipeline. Our approach leverages intrinsic properties of DD by reconstructing conceptual archetypes that align with the latent representations of real images, thereby bridging the gap between distilled and real data. We embed backdoor information into the distilled dataset, ensuring a consistent optimization trajectory with benign training to conceal malicious behavior. Extensive experiments across various DD methods, raw datasets, training strategies, and downstream architectures demonstrate the effectiveness, generalizability, and stealthiness of our method. Our findings highlight a critical security vulnerability in dataset distillation, challenging the belief that distilled datasets are resistant to backdoor attacks. We hope this work raises awareness of potential threats and encourages research into defense mechanisms for distilled dataset security.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62271335 and 62401381, in part by Sichuan Science and Technology Program under Grant 2025ZNSFSC0470, and in part by Sichuan University "From 0 to 1" Innovative Research Program under Grant 2022SCUH0016, and in part by the National Research Foundation, Singapore under its Digital Trust Centre Innovation Grant (DTC Award No: DTC-IGC-02), and is part by the Japan Science and Technology Agency (JST) and the A*STAR under the Japan-Singapore Joint Call (Project No. R24I6IR133).

References

Brown, T.; Mann, B.; Ryder, N.; et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, 1877–1901.

Cazenavette, G.; Wang, T.; Torralba, A.; Efros, A. A.; and Zhu, J.-Y. 2022. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4750–4759.

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

Chung, M.-Y.; Chou, S.-Y.; Yu, C.-M.; et al. 2024. Rethinking Backdoor Attacks on Dataset Distillation: A Kernel Method Perspective. In *International Conference on Learning Representations*.

Cui, J.; Wang, R.; Si, S.; and Hsieh, C.-J. 2023. Scaling up dataset distillation to imagenet-1k with constant memory. In *International Conference on Machine Learning*, 6565–6590. PMLR.

Du, J.; Shi, Q.; and Zhou, J. T. 2024. Sequential subset matching for dataset distillation. *Advances in Neural Information Processing Systems*, 36.

Du, J.; Zhang, X.; Hu, J.; Huang, W.; and Zhou, J. T. 2024. Diversity-Driven Synthesis: Enhancing Dataset Distillation through Directed Weight Adjustment. In *Advances in Neural Information Processing Systems*.

Feng, Y.; Ma, B.; Zhang, J.; et al. 2022. Fiba: Frequency-injection based backdoor attack in medical image analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20876–20885.

Gu, T.; Liu, K.; Dolan-Gavitt, B.; and Garg, S. 2019. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7: 47230–47244.

He, K.; Zhang, X.; Ren, S.; et al. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Jiang, W.; Li, H.; Xu, G.; and Zhang, T. 2023. Color backdoor: A robust poisoning attack in color space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8133–8142.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature*, 521(7553): 436–444.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lei, S.; and Tao, D. 2023. A comprehensive survey of dataset distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Lei, S.; and Tao, D. 2024. A Comprehensive Survey of Dataset Distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1): 17–32.

Li, Y.; Li, Y.; Wu, B.; et al. 2021. Invisible backdoor attack with sample-specific triggers. In *IEEE/CVF International Conference on Computer Vision*, 16463–16472.

Liu, Y.; Li, Z.; Backes, M.; Shen, Y.; and Zhang, Y. 2023. Backdoor attacks against dataset distillation. In *Network and Distributed System Security (NDSS) Symposium*.

Loo, N.; Hasani, R.; Amini, A.; and Rus, D. 2022. Efficient dataset distillation using random feature approximation. *Advances in Neural Information Processing Systems*, 35: 13877–13891.

Netzer, Y.; Wang, T.; Coates, A.; et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop*. Granada.

Nguyen, T. A.; and Tran, A. T. 2020. WaNet-Imperceptible Warping-based Backdoor Attack. In *International Conference on Learning Representations*.

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

Sun, P.; Shi, B.; Yu, D.; et al. 2024. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9390–9399.

Wang, T.; Yao, Y.; Xu, F.; et al. 2022. An invisible black-box backdoor attack through frequency domain. In *European Conference on Computer Vision*, 396–413. Springer.

Wang, T.; Zhu, J.-Y.; Torralba, A.; and Efros, A. A. 2018. Dataset distillation. *arXiv preprint arXiv:1811.10959*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xu, Z.; Chen, Y.; Pan, M.; et al. 2023. Kernel ridge regression-based graph dataset distillation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2850–2861.

Yu, R.; Liu, S.; and Wang, X. 2023. Dataset distillation: A comprehensive review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yu, R.; Liu, S.; Ye, J.; et al. 2024. Teddy: Efficient large-scale dataset distillation via taylor-approximated matching. In *European Conference on Computer Vision*, 1–17.

Zhao, B.; and Bilen, H. 2021. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, 12674–12685. PMLR.

Zhao, B.; and Bilen, H. 2023. Dataset Condensation with Distribution Matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

Zhao, B.; Mopuri, K. R.; and Bilen, H. 2021. Dataset Condensation with Gradient Matching. In *International Conference on Learning Representations*.