Journal of Machine Learning and Information Security
https://www.sciltp.com/journals/jmlis

Article

DeadMap: Open Yet Locked—Key-Controlled Dataset Protection with Feature-Level Collisions

Ting Yang [1], Mahabubur Rahman Miraj [1], Xinyu Lei [2], Nankun Mu [1] and Hongyu Huang [1,*]

[1] College of Computer Science, Chongqing University, Chongqing 400044, China
[2] Department of Computer Science, Michigan Technological University, Houghton, MI 49931, USA
* Correspondence: hyhuang@cqu.edu.cn

How To Cite: Yang, T.; Miraj, M.R.; Lei, X.; et al. DeadMap: Open Yet Locked—Key-Controlled Dataset Protection with Feature-Level Collisions. *Journal of Machine Learning and Information Security* **2026**, *2*(1), 5. https://doi.org/10.53941/jmlis.2026.100005

Received: 28 December 2025
Revised: 10 Feburary 2026
Accepted: 6 March 2026
Published: 19 March 2026

Abstract: The performance of deep learning models relies heavily on high-quality datasets. However, under the prevailing paradigm of pretrained-model–based applications, once a dataset is publicly released, data owners largely lose control over how it is used for model training. Existing data protection approaches either focus on post hoc accountability or irreversibly destroy the training utility of the data through perturbations, making it difficult to simultaneously satisfy the dual requirements of offline public sharing, authorized and controllable usage. In this paper, we propose DeadMap, a reversible training usability control framework designed for model fine-tuning scenarios. DeadMap introduces a secret label permutation combined with synchronized multi-layer feature-alignment perturbations. While preserving visual imperceptibility and keeping the surface labels unchanged, this design causes fine-tuning without the correct key to suffer severe performance degradation. In contrast, authorized users only need to possess a lightweight label-mapping key and apply a simple label remapping during training to recover performance close to that achieved with the original datasets. Experimental results show that DeadMap can effectively establish a significant performance difference between authorized and unauthorized settings, thus providing a lightweight and practical solution for balancing open data sharing and the controlled use of high-value datasets.

Keywords: dataset protection; model fine-tuning; feature collision; label permutation

1. Introduction

The success of deep learning models in visual, language, and multimodal tasks largely depends on the availability of large-scale, high-quality datasets [1–3]. Medical imaging, autonomous driving, and financial risk assessment all rely on carefully sorted high-quality data; in these key areas of security, the quality of the dataset determines the performance ceiling that any model can achieve and has a direct impact on the reliability of the deployed system [4,5]. The public sharing of such data sets helps to lower the research threshold, improve the comparability of model evaluation, and accelerate innovation at the algorithm and system levels [6,7]. Therefore, offline data sharing has long been a common goal pursued by academia and industry.

However, under the current research and application paradigm centered on the pre-training model, once the datasets are fully released to the public, its subsequent use will quickly get out of the control of the data owner [8,9]. In practice, publicly available datasets are more used to fine-tune general pre-training models or parameter efficient adaptation than to train models from the beginning [10–12]. This "download-fine-tuning-deployment" workflow is low cost, low threshold, and usually carried out in an offline environment, making it difficult for data owners to intervene in time or implement effective post-effect accountability [13,14]. As previously pointed out by previous research on multi-layer data security in sensitive areas such as IoT medical care [15], traditional defense measures are often surpassed by evolving cyber threats, which highlights the need for fundamental data-level protection beyond simple access control. Organizations that rely on proprietary data to maintain a competitive advantage or must comply with privacy regulations face a specific dilemma: how to share datasets for fine-tuning while still controlling their actual use.

The current protection technologies cannot solve this problem very well. Watermarks, fingerprints and traditional access control are largely passive-they track the distribution process or support post-traceability, but can do almost nothing about what happens after the data reaches the user's local storage [9,16,17]. Once the data is fully disclosed, these methods can either provide accountability evidence after the model training has been completed, or rely on strict access restrictions that fundamentally prevent open access and offline use [18]. Encryption and strong access control mechanisms can provide stricter protection, but they usually require online services, trusted hardware or complex key management infrastructure, bringing in a lot of system complexity and ongoing maintenance overhead [13,19]. Therefore, it is difficult for them to deploy in scenarios involving publicly shared and offline use of datasets.

In recent years, defensive methods based on active poisoning have tried to directly intervene in the training process by injecting adversarial perturbations into the data to disrupt unauthorized model training [20, 21]. Representative examples include Nightshade [22] and Glaze [23], which have shown a strong effect in suppressing unauthorized use in generative and fine-tuning scenarios. However, such perturbations are usually irreversible: once the data is contaminated, even if the data owner fully understands the perturbation process, it is impossible to restore the original training utility while retaining the perturbation [24,25]. Therefore, these methods are more suitable for scenarios aimed at permanently destroying the value of data training, rather than for open data release that must continue to support authorized fine-tuning and collaborative research.

In order to solve the above contradiction between "offline public sharing" and "controlled use", we propose DeadMap, a dataset training usability control framework specially designed for model fine-tuning scenarios. DeadMap combines secret label permutation with synchronous feature-level perturbation to build a key-based training availability control mechanism. The protected dataset can be fully published and downloaded offline, and the perturbation remains visually unperceptible. Without knowing the key, using the original label for standard fine-tuning will lead to a serious performance degradation, close to the accuracy of random guessing. In contrast, authorized users only need to hold a lightweight label mapping key (specifically, a permutation table with a length equal to the number of classes) and apply simple label remapping during the training process to restore the model performance close to the original data set. This does not require any other model, decryption program, or online service. Unlike previous work that either tracks abuse or permanently destroys training utility after the fact, DeadMap takes a different path: it makes unauthorized training fail in a controllable way while allowing authorized users to restore full utility at the lowest cost. This redefinition from "after-the-spot accountability" to "key-gated availability" has created a streamlined and scalable mechanism suitable for the release of high-value datasets. The main contributions of this work are summarized as follows:

1. We have formally defined the problem of reversible training usability control, clearly distinguished between data accessibility and data trainability, and introduced a new problem expression for data set protection research.
2. We proposed the DeadMap framework, which combines secret label permutation with multi-layer feature alignment to achieve strong protection while maintaining the recoverability of authorized user training utility.
3. We have verified the effectiveness of DeadMap on multiple benchmark data sets and network architectures. Experimental results show that under the typical fine-tuning settings, the accuracy of unauthorized training is significantly suppressed, and authorized users can restore most of the training utility. Even if unauthorized users try deeper fine-tuning, its performance is still far lower than authorized training.

2. Related Work

2.1. Passive Dataset Protection

The traditional dataset intellectual property protection method mainly focused on two aspects: privacy leakage prevention and post-event accountability [9, 18]. Differential privacy limited the risk of individual information leakage by injecting noise into the training process [26,27]; however, its design goal is not to prevent the dataset from being used for model training as a whole, and the injected noise often leads to an inevitable decrease in model accuracy. The cryptographic method-homomorphic encryption, trusted execution environment-provides stricter access level protection [13,19], but required services or dedicated hardware that are always online, which conflicted with the way the open research community actually shares data: offline sharing, not relying on infrastructure.

Watermark and fingerprint technologies adopted different strategies to support legal recourse after the discovery of infringement by embedding verifiable identifiers [16,17]. However, these methods do not work in the model training process and cannot directly prevent the use of unauthorized data during the training process [9].

2.2. Active Protection via Data Poisoning

In recent years, a series of works have explored the use of data poisoning technology as an active defense mechanism to reduce the effectiveness of unauthorized model training [20,21]. Glaze introduced imperceptible perturbations into artistic images, interfering with the learning of specific styles from text to image models [23]. Nightshade further proposed concept-level poisoning, in which contaminated samples induce incorrect semantic associations in the generation model, thus inhibiting large-scale data capture and training [22]. Similarly, methods such as Mist and Anti-DreamBooth used irreversible perturbations to prevent personal images from being used for LoRA or DreamBooth fine-tuning [25].

These methods prove that carefully designed data perturbations can significantly affect the results of model training. However, the perturbations they introduced are usually irreversible: once the data is released and used, even if the data owner fully understands the perturbation process, it cannot restore the original training utility while retaining the perturbation [24,28]. Therefore, such methods are more suitable for scenarios aimed at permanently destroying the value of data training, rather than for data-sharing environments that must continue to support authorized use.

Recent clean label backdoor attack methods, such as Sleeper Agent and Narcissus, also used feature collisions to achieve hidden poisoning [29–31]. These methods usually only disturb a small part of the target class and align their characteristics with the characteristics of the source class sample containing the trigger. As a result, the model learned the hidden association between the trigger and the target class while maintaining normal classification performance. The success of this kind of attack is characterized by the coexistence of a high attack success rate [32,33] and a high benign accuracy rate [28,34], so that the backdoor behavior can coexist with the normal model behavior.

Unlike the above methods, the goal of this work is not to implant the backdoor, but to focus on the reversible control of data set training usability. DeadMap applies feature-level perturbations to all training samples and combines secret label permutation with multi-layer feature alignment. While maintaining the clean label attribute, this design creates a systematic feature-label mismatch, so that users without the correct key cannot obtain a valid model through standard fine-tuning, while allowing authorized users to restore training utility through simple label remapping. We noticed that methods such as Glaze and Nightshade are specially designed to generative model scenarios (for example, to prevent style imitation or conceptual poisoning from text to image models), while DeadMap is aimed at classification fine-tuning, focusing on reversible training usability control. Since Glaze/Nightshade is aimed at generative models and DeadMap is for classification fine-tuning, the two operate under completely different threat assumptions, and it is meaningless to make a direct numerical comparison of protection intensity.

In terms of practicality, DeadMap follows the model of “one-time generation, free use”: the data owner takes perturbation generation as a one-time offline preprocessing step; after that, there is no need to recalculate for different users or different scenarios. After generation, the protected dataset can be distributed to any number of authorized parties without additional calculation. For each authorized user, the recovery process comes down to a permutation table containing C integers (10 for CIFAR-10 and 43 for GTSRB) and a one-time table search for each sample when the data is loaded, without increasing any training calculation, no model modification, no decryption steps, and no Internet connection is required. In contrast, the encryption-based schemes [13,19] will generate decryption overhead for each data access and require continuous key management infrastructure. The irreversible poisoning method [22,23,25] completely avoids runtime costs, but the cost is to permanently destroy the training value of everyone (including the data owner). DeadMap’s one-time generation and near-zero cost of use make it a natural choice when the same dataset needs to be distributed repeatedly to different authorized groups.

Table 1 provides a qualitative comparison of representative data set protection methods in key dimensions related to offline data release and fine-tuning control settings.

Table 1. Comparison of dataset protection methods for model fine-tuning.

Method	Offline Release	Usability Control	Recovery (Auth.)	Visual Impercept.	Key Based
Differential Privacy [26,27]	✓	×	×	✓	×
Watermarking / Fingerprinting [16,17]	✓	×	×	✓	×
Dataset Licensing / ToS [6,7]	✓	×	×	✓	×
Access Control / Encryption [13,19]	×	×	✓	×	✓
Fawkes [35]	✓	✓	×	✓	×
Mist / Anti-DreamBooth [25]	✓	✓	×	✓	×
Glaze / Nightshade [22,23]	✓	✓	×	✓	×
DeadMap (Ours)	✓	✓	✓	✓	✓

3. Method

3.1. Problem Definition

Given a classification dataset with C classes,

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}, \tag{1}$$

where

$$y_i \in \{0, 1, \ldots, C-1\}, \tag{2}$$

and

$$x_i \in [0,1]^{K\times H\times W}. \tag{3}$$

The data owner aims to publicly release a protected dataset

$$\mathcal{D}' = \{(\tilde{x}_i, y_i)\}_{i=1}^{N}, \tag{4}$$

where the protected sample is defined as

$$\tilde{x}_i = x_i + \delta_i. \tag{5}$$

The protected dataset is required to satisfy the following properties:

- Visual Indistinguishability: The perturbation is constrained under the ℓ_∞ norm,

$$\|\delta_i\|_\infty = \|\tilde{x}_i - x_i\|_\infty \le \epsilon, \tag{6}$$

  where $\epsilon = 16/255$ by default.
- Clean-label Property: The labels of the protected dataset remain identical to the original labels y_i, with no explicit label manipulation.
- Unauthorized Unusability: For any third-party user who does not possess the secret key, training a model on $\mathcal{D}'$ under standard fine-tuning settings yields an accuracy close to or lower than random guessing.
- Authorized Recoverability: Authorized users who possess the secret key can recover training performance close to that of the original dataset through a simple label remapping operation.

3.2. Threat Model

We assume the following conditions to define the security boundary of DeadMap.

- Data Owner Capabilities: The data owner has white-box access to widely used pretrained feature extractors f_θ (e.g., ImageNet-pretrained ResNet-18, VGG-16, MobileNetV2). The data owner can generate perturbations offline and release the protected dataset $\mathcal{D}'$, and can securely store and distribute the secret key π.
- Unauthorized User Capabilities: An unauthorized user can fully access the publicly released protected dataset $\mathcal{D}'$. The user may choose arbitrary pretrained models, training strategies, and fine-tuning depths. However, the user does not possess the secret key π and has no knowledge of the specific label permutation structure.
- Authorized User Capabilities: An authorized user obtains the secret key π from the data owner through a secure channel. During training, the authorized user applies label remapping

$$\tilde{y}_i = \pi(y_i), \tag{7}$$

  and performs standard fine-tuning on the remapped protected dataset $(\tilde{x}_i, \tilde{y}_i)$, thereby recovering model performance close to that obtained using the original dataset.

3.3. Overall Framework of DeadMap

The protection process of DeadMap consists of two stages: an offline generation stage and an authorized recovery stage, as illustrated in Figure 1.

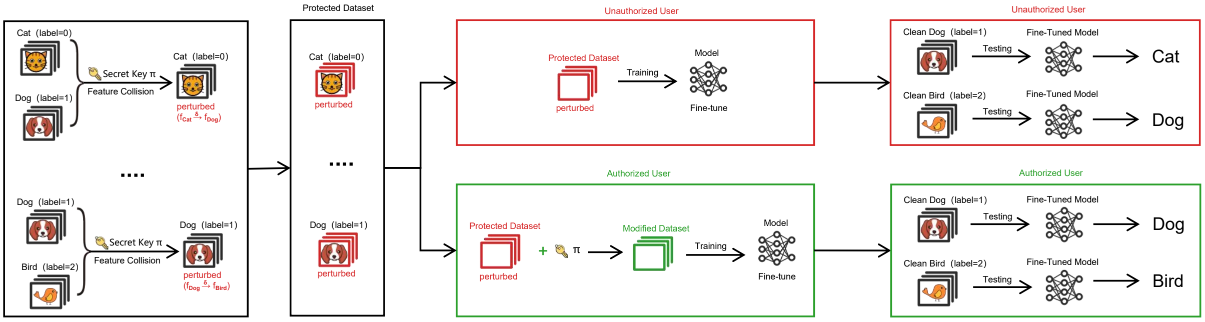

Figure 1. Overview of DeadMap.

3.3.1. Offline Generation Stage (Executed Once by the Data Owner)

- Label Permutation Key Selection: A permutation $\pi \in \mathcal{S}_C$ is selected from the symmetric group $\mathcal{S}_C$, which contains all bijective permutations defined over the label set $\{0, \ldots, C-1\}$. Although DeadMap is theoretically applicable to any non-identity permutation, different permutations yield different protection strengths. To avoid leaving any class semantics unaffected, we adopt a cyclic permutation that covers all classes in our experiments.
- Paired Sample Construction: For each class c, all samples belonging to class c are randomly shuffled and paired one-to-one with randomly shuffled samples from the target class $\pi(c)$. The number of pairs is determined by the minimum sample count of the two classes. This forms the paired set

$$\mathcal{P} = \{(x_o, x_t) \mid y_o = c,\ y_t = \pi(c),\ c \in \{0, \ldots, C-1\}\}, \tag{8}$$

  where each original sample x_o is uniquely matched with a target sample x_t.
- Feature-Collision Perturbation Generation: For each paired sample (x_o, x_t), a small perturbation δ is optimized such that the perturbed sample

$$\tilde{x}_o = x_o + \delta \tag{9}$$

  has multi-layer features highly aligned with those of x_t, while satisfying the ℓ_∞ perturbation budget and valid pixel range constraints (see Section 3.4).
- Protected Dataset Release: The data owner releases the protected dataset

$$\mathcal{D}' = \{(\tilde{x}_i, y_i)\}_{i=1}^{N}. \tag{10}$$

3.3.2. Authorized Recovery Stage (Executed by Authorized Users)

After obtaining the secret key π, an authorized user applies label remapping

$$\tilde{y}_i = \pi(y_i), \tag{11}$$

and performs standard fine-tuning on the remapped protected dataset $(\tilde{x}_i, \tilde{y}_i)$ to recover training utility. We emphasize that the authorized recovery process is computationally trivial: it requires only a label-mapping table of C entries and a single index lookup per sample during training. No additional model, decryption algorithm, or online service is needed, making DeadMap's recovery overhead negligible compared to encryption-based alternatives.

3.4. Synchronized Feature-Collision Perturbation Generation

3.4.1. Multi-layer Feature Encoders

Given a pretrained backbone feature extractor f_θ, we construct a sequence of L truncated encoders with increasing depth:

$$e_l(\cdot) \triangleq f_\theta^{(l)}(\cdot), \quad l = 0, \ldots, L-1, \tag{12}$$

where $f_\theta^{(l)}$ denotes the composition of the first l modules of the backbone network.

In implementation, an input image is first normalized using a preprocessing function $\mathcal{N}(\cdot)$ to match the pretrained model's input interface. The flattened feature representation at layer l is defined as

$$z_l(x) = \mathrm{vec}\big(e_l(\mathcal{N}(x))\big). \tag{13}$$

The truncation depth L depends on the network architecture. For ResNet-18 and VGG-16, the backbone is divided into five semantic stages ($L = 5$), corresponding to representations from low-level textures to high-level semantics. For MobileNetV2, due to its fine-grained inverted residual design, alignment constraints are applied to all 19 feature extraction modules ($L = 19$).

3.4.2. Optimization Objective

We minimize the weighted sum of multi-layer feature distances between an original sample and its paired target sample:

$$\min_{\delta} \mathcal{L}(x_o, x_t; \delta) = \sum_{l=0}^{L-1} w_l \cdot d\big(z_l(x_o + \delta), z_l(x_t)\big), \tag{14}$$

where w_l denotes the layer weight (set to $w_l = 1$ by default), and $d(\cdot,\cdot)$ is a feature distance metric. We adopt cosine distance as the default choice:

$$d_{\cos}(u, v) = 1 - \frac{u^\top v}{\|u\|_2 \, \|v\|_2}. \tag{15}$$

We precompute the target features $z_l(x_t)$ before optimization begins and freeze them throughout each batch; they act as fixed reference points for the alignment objective.

3.4.3. Constraints and Optimization

The optimization process uses Adam optimizer to update the symbol gradient. After each iteration, ℓ_∞ projection is carried out (to ensure that the perturbation is kept within the budget ε range) and pixel value cropping (to keep the image within the $[0, 1]$ interval). If the maximum loss of each sample in a batch falls below the threshold τ, it will be terminated early. The final perturbation is quantified to an 8-bit value to make the protected image conform to the standard storage format. Algorithm 1 gives the complete process.

Algorithm 1 DeadMap perturbation generation (single batch)

```
Require: Original batch $X_o$, target batch $X_t$, encoders $\{e_\ell\}_{\ell=0}^{L-1}$, normalization $\mathcal{N}(\cdot)$, budget $\epsilon$, max steps $T$,
    threshold $\tau$, layer weights $\{w_\ell\}$, learning rate $\eta$
Ensure: Protected batch $\tilde{X}$
Initialize $\delta \leftarrow \mathbf{0}$ with the same shape as $X_o$
Precompute target features $z_\ell^t \leftarrow \mathrm{vec}(e_\ell(\mathcal{N}(X_t)))$ for all $\ell$
for $t = 1$ to $T$ do
    Compute perturbed features $z_\ell^p \leftarrow \mathrm{vec}(e_\ell(\mathcal{N}(X_o + \delta)))$ for all $\ell$
    Compute loss $\mathcal{L} \leftarrow \sum_{\ell=0}^{L-1} w_\ell \cdot d(z_\ell^p, z_\ell^t)$
    if $\max(\mathcal{L}) \le \tau$ then
        break                                                    ▷ Early convergence
    end if
    Compute gradient $g \leftarrow \nabla_\delta \mathcal{L}$
    Sign gradient $g \leftarrow \mathrm{sign}(g)$
    Adam update $\delta \leftarrow \mathrm{Adam}(\delta, g, \eta)$
    $\ell_\infty$ projection $\delta \leftarrow \mathrm{clamp}(\delta, -\epsilon, \epsilon)$
    Pixel clipping $\delta \leftarrow \mathrm{clamp}(X_o + \delta, 0, 1) - X_o$
end for
Quantize to 8-bit $\tilde{X} \leftarrow \lfloor (X_o + \delta) \times 255 \rfloor / 255$
return $\tilde{X}$
```

3.5. Security and Key Space Analysis

The secret of DeadMap lies in the label permutation $\pi \in \mathcal{S}_C$. The complete symmetry group has $|\mathcal{S}_C| = C!$ elements—for CIFAR-10 ($C = 10$) has reached 3,628,800, and for GTSRB ($C = 43$), it is about 6×10^{52}. Violently exhausting a space of factorial size is obviously beyond the reach of any actual attacker.

However, not every permutation can provide the same level of protection. A permutation that only disrupts a few classes and keeps the rest of the classes unchanged will allow unauthorized users to maintain a good accuracy rate on unchanged classes. In order to eliminate this semantic immobility, we use cyclic permutation throughout the experiment.

$$\pi(i) = (i + 1) \bmod C, \tag{16}$$

The permutation remaps each category to produce consistent and uniform training degradation when the key is missing.

Strictly speaking, any derangement-that is, double projection permutation without fixed points - meets the requirements. In $D(10) = 1{,}334{,}961$ misaligned arrangements when $C = 10$, the cyclic shift is only one of them; we choose it because of its simplicity, reproducibility, and its uniform displacement pattern, which will not be biased to any specific category subset.

It may be expected that semantically similar category pairs (such as cars and trucks in CIFAR-10) are more likely to collide in the feature space, while class pairs with longer semantic distances require greater perturbation. However, our results show that the multi-layer alignment loss can reliably converge on all class pairs under cyclic permutation under the condition of $\varepsilon = 16/255$. Table 2 confirms this: on all data sets, the unauthorized accuracy rate is maintained below 5%, which has nothing to do with the semantic distance between category pairs.

4. Experiments

4.1. Experimental Setup

Datasets. We tested on four image classification benchmarks: CIFAR-10, MNIST, Fashion-MNIST (each with 10 classes), and GTSRB (43 classes). These four datasets differ in semantic difficulty, image type (grayscale and color), and the number of classes, which helps us to evaluate whether DeadMap can be generalized beyond a single data paradigm.

Model Architectures. Three convolutional architectures serve as backbone networks: ResNet-18, VGG-16, and MobileNetV2, all of which start from ImageNet pre-training weights. They differ significantly in depth, the use of residual connections, and the number of parameters, so the combination of the three provides a reasonable network design cross-section for robustness evaluation.

Fine-tuning Strategy. Following common practices in transfer learning, we control the number of unfrozen backbone modules via the parameter `tune_layers`, while always training the classification head. For datasets with relatively small domain gaps from ImageNet (CIFAR-10, MNIST, and Fashion-MNIST), fine-tuning only the classification head (`tune_layers` = 1) is sufficient to achieve strong baseline performance. For GTSRB, which exhibits a larger domain shift, we select the minimum fine-tuning depth that yields a baseline accuracy above 90% (ResNet-18: `tune_layers` = 2; VGG-16: `tune_layers` = 8; MobileNetV2: `tune_layers` = 4).

Training Configuration. All models are trained using the SGD optimizer with a learning rate of 0.05 for 20 epochs and a batch size of 256. The perturbation budget in DeadMap is set to $\epsilon = 16/255$, and cosine distance is used as the feature alignment loss. Label permutation is implemented using a cyclic mapping that covers all classes, defined as $\pi(i) = (i + 1) \bmod C$.

Evaluation Metrics. We report the following metrics. Baseline denotes the training and test accuracy achieved on the clean dataset. Unauthorized refers to the test accuracy obtained by training on the protected dataset using the original labels. Authorized denotes the test accuracy achieved by training on the protected dataset with correctly remapped labels. Recovery is defined as the ratio Authorized/Baseline and measures the extent to which training utility is restored for authorized users.

5. Main Results

Table 2 presents the protection effectiveness of DeadMap across different datasets and model architectures, where `tune` denotes the number of feature modules involved in fine-tuning. It should be noted that, due to the substantial distribution gap between GTSRB and ImageNet, deeper fine-tuning is required on this dataset to achieve a reasonable Baseline performance.

Key Findings.

1. Strong Protection Effectiveness. Under typical fine-tuning settings, the accuracy of unauthorized models is consistently suppressed to well below random-guessing levels. The lowest unauthorized accuracy we observe is 0.15% (MNIST + VGG-16)-well below the 10% random-guess baseline-confirming that standard fine-tuning without the key yields essentially unusable models.
2. Recoverable Training Utility. Of the 12 configurations, 11 reach recovery rates above 95% (average: 97.7%). Interestingly, for CIFAR-10 + VGG-16 the authorized accuracy slightly surpasses the clean-data baseline-a possible side-effect of the feature-level perturbations acting as an implicit regularizer during training.
3. Clear Bi-modal Behavior. On average, authorized users outperform unauthorized ones by 88.8% points-a gap large enough to confirm that the feature-label misalignment built into DeadMap does suppress unauthorized training in a reliable, across-the-board fashion.

4. Cross-setting Generalization. Protection trends hold steady across all four datasets and all three architectures, suggesting that the mechanism is not brittle to changes in image modality, number of classes, or backbone design.

Table 2. Protection effectiveness of DeadMap ($\epsilon = 16/255$).

Dataset	Model	`tune`	Baseline	Unauthorized	Authorized	Recovery
CIFAR-10	ResNet-18	1	87.45%	1.16%	86.78%	99.2%
CIFAR-10	VGG-16	1	86.96%	1.27%	87.01%	100.1%
CIFAR-10	MobileNetV2	1	88.21%	1.19%	87.44%	99.1%
MNIST	ResNet-18	1	96.54%	0.69%	96.46%	99.9%
MNIST	VGG-16	1	98.89%	0.15%	98.86%	100.0%
MNIST	MobileNetV2	1	97.12%	2.65%	92.79%	95.5%
F-MNIST	ResNet-18	1	88.58%	0.67%	84.83%	95.8%
F-MNIST	VGG-16	1	91.79%	0.58%	90.14%	98.2%
F-MNIST	MobileNetV2	1	89.20%	1.44%	80.46%	90.2%
GTSRB	ResNet-18	2	95.44%	2.22%	94.98%	99.5%
GTSRB	VGG-16	8	95.11%	2.37%	93.34%	98.1%
GTSRB	MobileNetV2	4	93.65%	2.79%	90.21%	96.3%

5.1. Effect of Fine-Tuning Depth

The choice of fine-tuning depth by downstream users depends on the specific task, so it is necessary to examine whether the protection effect of DeadMap is degraded or maintained as more backbone network layers are thawed. We emphasize that this analysis aims to understand the mechanism itself, not to simulate a real attacker who routinely thaws the entire backbone network.

Figure 2 shows the curve of the change of unauthorized accuracy on CIFAR-10 with the fine-tuning depth. When only a few modules can be trained (`tune` ≤ 3), the unauthorized accuracy rate of the three models remains below 5%. After exceeding this point, different structures begin to diverge. Even under full depth (32 layers), the unauthorized accuracy rate of VGG-16 is still below 3%. MobileNetV2 degenerates more slowly and is still within the acceptable range when `tune` ≤ 10. However, ResNet-18 has a significant rebound when `tune` ≥ 4, which may be because the skip connection provides a more direct path for the optimizer to reconstruct useful features in the presence of perturbations.

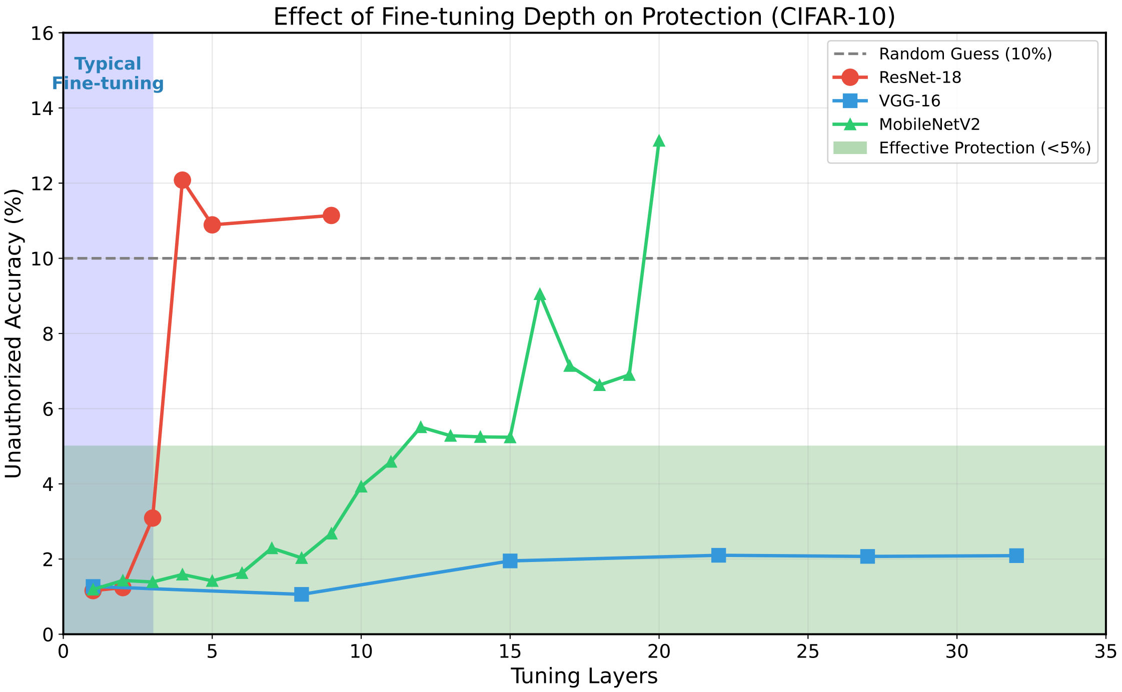

Figure 2. Effect of fine-tuning depth on protection (CIFAR-10).

Table 3 summarizes the depth range that meets two conditions at the same time-unauthorized accuracy rate of less than 5% and recovery rate of more than 90%. The results show that in typical shallow fine-tuning scenarios, DeadMap provides reliable protection on all evaluation architectures.

Table 3. Fine-tuning depth ranges with unauthorized accuracy $< 5\%$ and recovery $> 90\%$.

Dataset	ResNet-18	VGG-16	MobileNetV2
CIFAR-10	$t \leq 3$	$t \leq 32$	$t \leq 10$
MNIST	$t \leq 2$	$t \leq 32$	$t \leq 4$
F-MNIST	$t = 1$	$t \leq 8$	$t \leq 2$
GTSRB	$t = 2$–3	$t = 8$–32	$t = 4$–15

It should be noted that there is a fundamental asymmetry between authorized users and unauthorized users in fine-tuning strategies. Authorize users to have the correct label mapping, so follow the best practices of transfer learning, and usually use shallow fine-tuning to achieve optimal performance. In contrast, unauthorized users-even if they try to bypass protection through deeper fine-tuning-must bear additional computing costs and higher over-fitting risks, and still cannot obtain available model performance under most configurations.

5.2. Cross-Architecture Transferability

In order to evaluate the robustness of DeadMap when unauthorized users use a different architecture from the architecture used for perturbation generation, we conducted a cross-architecture experiment on CIFAR-10. Table 4 shows the experimental results.

Table 4. Cross-architecture transferability (CIFAR-10, `tune` = 1).

Perturbation Arch	Fine-Tuning Arch	Unauthorized Acc.	Baseline Acc.
ResNet-18	ResNet-18	1.16%	87.45%
ResNet-18	VGG-16	35.0%	86.96%
ResNet-18	MobileNetV2	57.36%	88.21%
VGG-16	VGG-16	1.27%	86.96%
VGG-16	ResNet-18	34.46%	87.45%
VGG-16	MobileNetV2	51.89%	88.21%

When the perturbation generation architecture matches the fine-tuning architecture, DeadMap provides strong protection, and the unauthorized accuracy rate is about 1%. Under the cross-architecture setting, the protection effect has decreased, and the unauthorized accuracy rate is between 35% and 57%. This is within the expectation, because different architectures present structural differences in feature spaces, and the feature collision optimized for one architecture cannot be completely migrated to another architecture. Nevertheless, even under cross-architecture settings, the unauthorized accuracy rate is still reduced by 30% to 53% points compared with the baseline, which represents a performance degradation of practical significance.

It is expected that the data owners who are expected to try multiple architectures can alleviate this gap by jointly optimizing perturbations for multiple backbone networks at the same time-we leave this strategy to future work (Section 6).

5.3. Ablation Study

In order to isolate the influence of various design choices, we use ResNet-18 in CIFAR-10 for ablation experiments.

5.3.1. Perturbation Strength

Table 5 breaks down the unauthorized accuracy rate according to the perturbation budget. Under shallow depth (`tune` $\leq$ 2), all three budgets control the unauthorized accuracy rate below 3%. The difference only appears when more layers are thawed: the larger ϵ value maintains a lower unauthorized accuracy at greater depth. Specifically, when `tune` = 5, the unauthorized accuracy rate under $\epsilon = 32/255$ drops to 4.18%, which is significantly lower than the level reached by $\epsilon = 8/255$ (10.89%) and $\epsilon = 16/255$ (10.89%).

Table 5. Effect of perturbation budget (CIFAR-10 + ResNet-18, cosine) on unauthorized accuracy (%).

`tune`	$\epsilon = 8/255$	$\epsilon = 16/255$	$\epsilon = 32/255$
1	1.45	1.16	1.29
2	1.77	1.24	1.15
3	2.94	3.09	1.63
4	8.07	12.08	6.08
5	10.89	10.89	4.18
9	21.55	11.14	10.01

A larger budget comes with a price: the perturbation becomes more obvious. We choose $\epsilon = 16/255$ as a practical compromise scheme-it maintains complete protection under the typical fine-tuning depth (`tune` ≤ 3) without producing visible artifacts.

5.3.2. Loss Function

Table 6 and Figure 3 compare three feature alignment loss functions: cosine distance, Smooth L1, and Smooth L2. At shallow depth (`tune` ≤ 2), the performance of the three is similar, and the unauthorized accuracy rate is less than 3%. The gap widens under a deeper setting: when `tune` = 5, the unauthorized accuracy of the cosine distance is 10.89%, while Smooth L1 is 17.21% and Smooth L2 is 21.84%; when `tune` = 9, the unauthorized accuracies are 11.14%, 23.45%, and 24.79%, respectively.

The advantage of the cosine distance may come from the fact that it constrains the characteristic direction rather than the amplitude. Direction alignment is more difficult to undo by deep fine-tuning, because even larger weight updates tend to retain the angle structure expressed in the middle rather than its scale. Smooth L1/L2 punishes absolute or square characteristic differences, and the constraint on inconsistent direction is weak, so it is easier to overcome. Based on these results, cosine distance is our default choice.

Table 6. Effect of loss function (CIFAR-10 + ResNet-18, $\epsilon = 16/255$) on unauthorized accuracy (%).

`tune`	Cosine	Smooth L1	Smooth L2
1	1.16	1.11	1.21
2	1.24	2.40	1.72
3	3.09	4.04	6.29
4	12.08	13.37	16.63
5	10.89	17.21	21.84
9	11.14	23.45	24.79

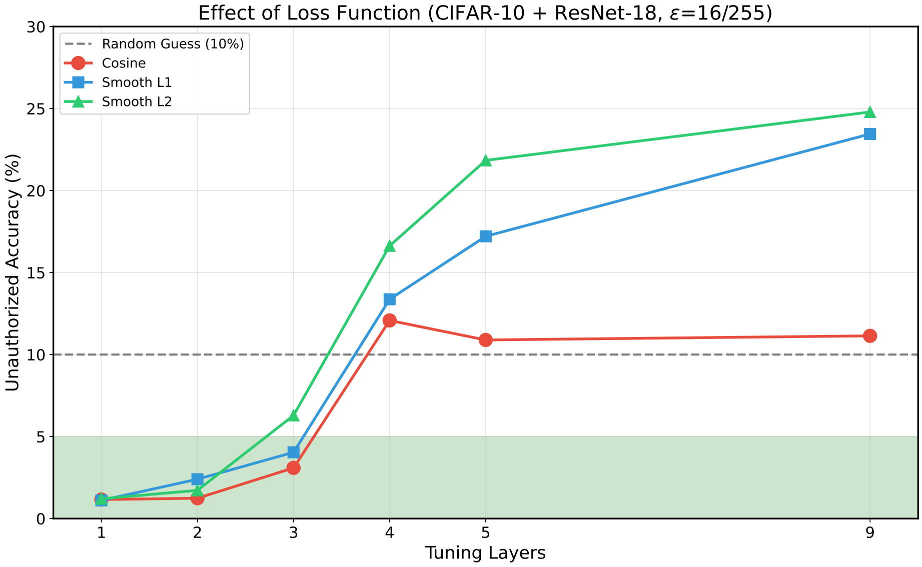

Figure 3. Effect of loss function (CIFAR-10 + ResNet-18, $\epsilon = 16/255$).

5.3.3. Layer Weight Ablation

Next, let's explore which feature layers are the most important for protection. The method is to melt the layer weight w_l. Three configurations were compared: all layers (default, a_0-$a_4 = 1$), only deep ($a_3, a_4 = 1$, the rest is 0) and only the shallow layer($a_0, a_1 = 1$, the rest is 0). The result of `tune` $= 1$ is shown in Table 7.

Table 7. Layer weight ablation (CIFAR-10 + ResNet-18, `tune` = 1).

Configuration	Layer Weights	Unauth. Acc.	Auth. Acc.
Full-layer (default)	a_0–$a_4 = 1$	1.16%	86.78%
Deep Only	$a_3, a_4 = 1$	1.01%	87.47%
Shallow Only	$a_0, a_1 = 1$	43.53%	29.40%

When only aligning the deep layer, the protection effect under the shallow layer fine-tuning is comparable to the full-layer setting, indicating that the semantic-level collision in the back stage of the network drives most of the effects of DeadMap. In contrast, the alignment of the shallow layer alone failed completely: the unauthorized accuracy rate jumped to 43.53%, and the authorization accuracy rate collapsed to 29.40%, indicating that the semantic confusion required to block learning cannot be created by relying on low-level texture perturbation alone.

Table 8. Full-layer vs. deep-only alignment at varying fine-tuning depths.

`tune`	Full-Layer (Unauth.)	Deep-Only (Unauth.)	Full-Layer (Auth.)	Deep-Only (Auth.)
1	1.16%	1.01%	86.78%	87.47%
2	1.24%	0.91%	90.61%	91.34%
3	3.09%	3.38%	87.63%	82.91%
4	12.08%	14.73%	63.79%	56.06%
5	10.89%	12.70%	69.05%	39.61%
9	11.14%	17.06%	59.49%	31.42%

However, once the `tune` $\geqslant 3$, the whole layer setting begins to lead in both protection strength and recovery ability. The gap is very obvious when `tune` $= 9$: only the deep configuration gives an unauthorized accuracy rate of 17.06%/31.42%, while the full configuration gives 11.14%/59.49%. The emerging picture is a complementary role-the deep layer bears the main protective load in the shallow fine-tuning, but when more layers are thawed, the shallow alignment becomes an indispensable stable support. Therefore, we keep the whole layer as the default configuration.

5.4. Robustness Against Purification Attacks

Can the opponent strip the perturbation through image space preprocessing? We tested three types of purification methods-JPEG compression, Gaussian blur, and added Gaussian noise-and applied each purification operation to clean data as a comparison, so as to attribute any decrease in accuracy to the purification itself rather than the change in data distribution.

Table 9. Robustness against purification (CIFAR-10 + ResNet-18, `tune` = 1).

Purification	Param	Protected Acc.	Clean Acc.	Gap
None	—	1.16%	87.45%	86.29 pp
JPEG	$q = 75$	11.88%	87.22%	75.34 pp
JPEG	$q = 50$	32.37%	86.60%	54.23 pp
Gaussian Blur	$k = 3$	20.71%	87.07%	66.36 pp
Gaussian Blur	$k = 5$	51.18%	85.88%	34.70 pp
Gaussian Noise	$\sigma = 0.03$	8.52%	70.14%	61.62 pp
Gaussian Noise	$\sigma = 0.05$	18.13%	62.73%	44.60 pp

Table 9 shows that under each setting, the accuracy of the protected data is much lower than the corresponding value of its clean data; the minimum gap is 34.70% (Gaussian blur, $k = 5$). Mild purification almost does not affect the protection effect: JPEG of $q = 75$ only increases the unauthorized accuracy rate from 1.16% to 11.88%. Radical purification causes greater damage to perturbations, but also undermines the training utility of everyone, for example, Gaussian noise of $\sigma = 0.05$ even reduces the accuracy of clean data to 62.73%.

Therefore, the opponent faces a trade-off: any preprocessing that is enough to weaken the DeadMap feature collision also destroys the training signal itself that the opponent wants to use.

5.5. Visual Imperceptibility Analysis

Figure 4 shows original, protected, and target images side by side for CIFAR-10 samples. Visually, the protected and original images are nearly indistinguishable; the perturbation shows up only as faint high-frequency noise that is hard to spot without pixel-level comparison.

Note that DeadMap's security does not hinge on hiding the perturbation in pixel space. Even with all three images in hand, an observer sees that the protected sample looks far more like the original than like the target-there is no visual cue from which the label mapping or the secret key could be reverse-engineered.

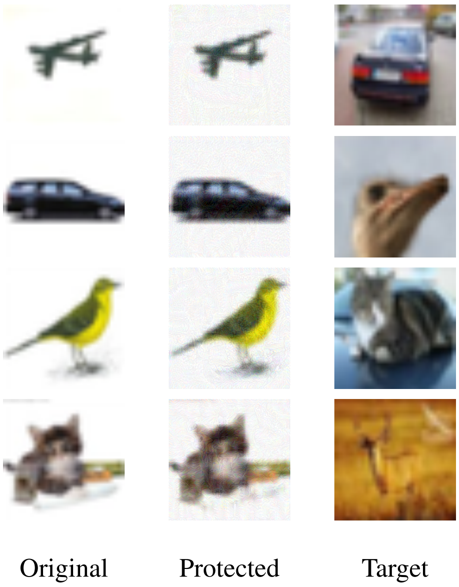

Figure 4. Visual imperceptibility of DeadMap.

5.6. Security Analysis

It is important to emphasize that the analysis in this subsection adopts a relatively conservative security assumption regarding the adversary's knowledge. Specifically, we assume that unauthorized users not only lack access to the exact label permutation π, but are also aware that a cyclic permutation structure is used in the implementation. Even under this assumption, an adversary would still need to enumerate over C possible cyclic shifts to recover the correct mapping.

In our experiments, we fix a specific cyclic permutation instance to generate the protected dataset and simulate unauthorized usage by training models on the protected data with the original labels. The experimental results show that, under the typical fine-tuning settings used in Table 2, the accuracy of unauthorized training is consistently and significantly suppressed across all datasets and model architectures, remaining below 5%. It should be noted that this conclusion relies on the use of permutations that cover all classes. For weaker permutations that involve only a small subset of classes, the degree of performance degradation under unauthorized training may be correspondingly reduced.

We noticed that the security analysis of DeadMap is based on the premise that the protected dataset is the only channel to obtain the data, such as proprietary medical image datasets or commercial annotation corpora that are not made public elsewhere. Public benchmark datasets such as CIFAR-10 are purely used as standardized evaluation and test platforms; in actual deployment, if clean data can be obtained from other sources for free, adversaries do not need to use the protected version. Under this threat model, we consider two potentially stronger attack vectors.

First, an adversary who knows the DeadMap mechanism may try to "un-align" the disturbed features by applying reverse perturbations. However, targeted feature alignment needs to know the collision direction of each sample, which is determined by the secret key π and the specific one-to-one target pairing. In the absence of π, the opponent cannot identify which characteristics have been redirected and does not know the direction of redirection, which makes systematic dealignment not feasible. Even if the opponent knows that the cyclic permutation structure is used, they still need to enumerate the possible displacement of C, and for each candidate shift, try to reverse the multi-layer feature alignment on all samples. This task is still computationally unbearable without the original clean feature as a reference point.

Second, adversary may use self-supervised pre-training (such as SimCLR, MAE) in protected data to learn representations that do not depend on labels. The representation learned by the self-supervised method comes from the disturbed images, and the deep features of these images have been systematically redirected to the target class $\pi(y)$. The contrastive learning goal encourages the enhancement of immutability, but does not distinguish between the original semantic content and the characteristic direction of confrontational injection; therefore, the representation space learned inherits the semantic distortion introduced by DeadMap.

However, we acknowledge a more nuanced scenario: with the advent of few-shot and zero-shot learning paradigms (e.g., CLIP zero-shot classification), an adversary could potentially combine self-supervised pre-training on the protected data with a small number of correctly labeled samples obtained from external sources to train a downstream linear classifier. For common-category datasets, obtaining such few-shot clean data is relatively straightforward. We note two mitigating factors: (1) DeadMap is primarily designed for proprietary, domain-specific datasets (e.g., medical imaging with specialized classes, industrial defect detection) where external few-shot labeled data for the specific domain classes is substantially harder to obtain; and (2) even when few-shot data is available, the self-supervised representations learned from perturbed images are themselves distorted-class c samples cluster in the feature region of class $\pi(c)$, which may reduce the effectiveness of a linear probe that must generalize from clean feature distributions to these shifted distributions. Nevertheless, we identify the self-supervised + few-shot probing attack as an important direction for future empirical investigation.

6. Conclusions

In this paper, we proposed DeadMap, a reversible dataset training usability control framework designed for model fine-tuning scenarios. By combining a secret label permutation with multi-layer feature-alignment perturbations, DeadMap enabled a practical balance between open data sharing and controlled usage: datasets can be fully released and downloaded offline, training becomes ineffective without the key, and training utility is recoverable for authorized users.

Experiments using three architectures (ResNet-18, VGG-16 and MobileNetV2) in four data sets - CIFAR-10, MNIST, Fashion-MNIST and GTSRB-show that: (1) Unauthorized training accuracy is consistently suppressed to a level far lower than the random guess, and the lowest accuracy rate observed reached 0.15%; (2) In 12 groups of experimental configurations, the recovery rate of 11 groups of authorized users exceeded 95%; (3) The average accuracy rate gap between authorized and unauthorized scenarios reached 88.8%.

The ablation study further verified the two design options of feature alignment based on cosine distance and $\epsilon = 16/255$ perturbation budget. By shifting the focus of dataset protection from whether data is accessible to whether data can be effectively trained, DeadMap has introduced a new design paradigm for publishing high-value datasets in academic sharing, commercial licensing, and privacy-sensitive applications. The future work will explore several promising directions. First, multi-architecture joint perturbation optimization, that is, the data owner aligns the characteristics of multiple architectures at the same time in the process of perturbation generation, which can improve the robustness of cross-architecture protection. Second, we plan to extend DeadMap to more complex visual tasks, such as target detection and semantic segmentation, in which label permutation can be applied to category-level annotation. Third, adapting the feature collision framework to non-image fields, including natural language processing and multimodal data, is an important direction that requires the redesign of perturbation strategies for discrete and sequential data modalities. Fourth, we plan to systematically study the robustness of DeadMap against self-supervised pre-training combined with external detection of a few samples, especially the performance under different numbers of available clean samples and different self-supervised targets. Finally, scaling up to larger-scale data sets and emerging architectures (such as Vision Transformer) remains a key priority.

Author Contributions

T.Y.: conceptualization, methodology, investigation, writing-original draft preparation; M.R.M.: writing—original draft preparation; X.L.: writing—review and editing, supervision; N.M.: writing—review and editing, supervision; H.H.: writing—review and editing, supervision. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Not applicable.

Conflicts of Interest

The authors declare no conflict of interest. Given the role as Editorial Board Member, Xinyu Lei had no involvement in the peer review of this paper and had no access to information regarding its peer-review process. Full responsibility for the editorial process of this paper was delegated to another editor of the journal

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used ChatGPT to polish the language of the manuscript. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the content of the published article.

References

1. Bayoudh, K.; Knani, R.; Hamdaoui, F.; et al. A Survey on Deep Multimodal Learning for Computer Vision: Advances, Trends, Applications, and Datasets. *Vis. Comput.* **2022**, *38*, 2939–2970.
2. Jabeen, S.; Li, X.; Amin, M.S.; et al. A Review on Methods and Applications in Multimodal Deep Learning. *ACM Trans. Multimed. Comput. Commun. Appl.* **2023**, *19*, 1–41.
3. Hatcher, W.G.; Yu, W. A Survey of Deep Learning Platforms and Applications. *IEEE Access* **2018**, *6*, 24411–24432.
4. Amiri, Z.; Heidari, A.; Navimipour, N.J.; et al. Deep Learning Techniques for Pattern Recognition in Cyber-Physical-Social Systems. *Multimed. Tools Appl.* **2024**, *83*, 22909–22973.
5. Rather, I.H.; Kumar, S.; Gandomi, A.H. Breaking the Data Barrier: Democratizing AI with Small Datasets. *Artif. Intell. Rev.* **2024**, *57*, 226.
6. Susha, I.; Zuiderwijk, A.; Janssen, M.; et al. Benchmarks for Evaluating Open Data Adoption. *Soc. Sci. Comput. Rev.* **2015**, *33*, 613–630.
7. Conrado, D.J.; Karlsson, M.O.; Romero, K.; et al. Open Innovation: Sharing Data, Models, and Workflows. *Eur. J. Pharm. Sci.* **2017**, *109*, S65–S71.
8. Li, Y.; Shao, S.; He, Y.; et al. Rethinking Data Protection in the Generative AI Era. *arXiv* **2025**, arXiv:2507.03034.
9. Xue, M.; Zhang, Y.; Wang, J.; et al. Intellectual Property Protection for Deep Learning Models. *IEEE Trans. Artif. Intell.* **2021**, *3*, 908–923.
10. Han, Z.; Gao, C.; Liu, J.; et al. Parameter-Efficient Fine-Tuning for Large Models. *arXiv* **2024**, arXiv:2403.14608.
11. Ding, N.; Qin, Y.; Yang, G.; et al. Parameter-Efficient Fine-Tuning of Large-Scale Language Models. *Nat. Mach. Intell.* **2023**, *5*, 220–235.
12. Xu, L.; Xie, H.; Qin, S.Z.J.; et al. Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models. *arXiv* **2023**, arXiv:2312.12148.
13. Kum, H.-C.; Ahalt, S. Privacy-by-Design: Understanding Data Access Models for Secondary Data. *AMIA Jt. Summits Transl. Sci. Proc.* **2013**, 2013, 126–130.
14. Upreti, R.; Lind, P.G.; Elmokashfi, A.; et al. Trustworthy Machine Learning for Security and Privacy. *Int. J. Inf. Secur.* **2024**, *23*, 2287–2314.
15. Almotairi, S.; Addula, S.R.; Alharbi, O.; et al. Personal Data Protection Model in IOMT-Blockchain on Secured Bit-Count Transmutation Data Encryption Approach. *Fusion: Pract. Appl.* **2024**, *16*, 152–170.
16. Mahajan, A.; Powell, D. Digital Watermarking for Authenticity and Provenance. *npj Digit. Med.* **2025**, *8*, 31.
17. Ye, P.; Ren, H.; Li, Z.; et al. Securing Large Language Models: A Survey of Watermarking and Fingerprinting Techniques. *ACM Comput. Surv.* **2026**, *58*, 1–35.
18. Wang, Z.; Ma, J.; Wang, X.; et al. Threats to Training: Poisoning Attacks and Defenses. *ACM Comput. Surv.* **2022**, *55*, 1–36.

19. Mousavi, S.K.; Ghaffari, A.; Besharat, S.; et al. Security of IoT Based on Cryptographic Algorithms. *Wirel. Netw.* **2021**, *27*, 1515–1555.
20. Zhao, P.; Zhu, W.; Jiao, P.; et al. Data Poisoning in Deep Learning: A Survey. *arXiv* **2025**, arXiv:2503.22759.
21. Goldblum, M.; Tsipras, D.; Xie, C.; et al. Dataset Security for Machine Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 1563–1580.
22. Shan, S.; Ding, W.; Passananti, J.; et al. Nightshade: Prompt-Specific Poisoning Attacks on Text-to-Image Generative Models. In Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2024; pp. 807–825.
23. Shan, S.; Cryan, J.; Wenger, E.; et al. Glaze: Protecting Artists from Style Mimicry by Text-to-Image Models. In Proceedings of the 32nd USENIX Conference on Security Symposium, Anaheim, CA, USA, 9–11 August 2023; pp. 2187–2204.
24. Hönig, R.; Rando, J.; Carlini, N.; et al. Adversarial Perturbations Cannot Reliably Protect Artists. *arXiv* **2024**, arXiv:2406.12027.
25. Li, Z.; Xie, L.; Zhou, J.; et al. Anti-Diffusion: Preventing Abuse of Diffusion Models. *arXiv* **2025**, arXiv:2503.05595.
26. Li, X.; Chen, Y.; Wang, C.; et al. When Deep Learning Meets Differential Privacy. *IEEE Netw.* **2022**, *35*, 148–155.
27. Chen, H.-L.; Chen, J.-Y.; Tsou, Y.-T.; et al. Evaluating the Risk of Data Disclosure Using Noise Estimation for Differential Privacy. In Proceedings of the 2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC), Christchurch, New Zealand, 22–25 January 2017; pp. 339–347.
28. Miraj, M.R.; Huang, H.; Yang, T.; et al. GK-SMOTE: A Hyperparameter-Free Noise-Resilient Gaussian KDE-Based Oversampling Approach. *arXiv* **2025**, arXiv:2509.11163.
29. Zeng, Y.; Pan, M.; Just, H.A.; et al. Narcissus: A Practical Clean-Label Backdoor Attack with Limited Information. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, Copenhagen, Denmark, 26–30 November 2023; pp. 771–785.
30. Zhang, C.; Sun, S.; Tu, J.; et al. Clean-Label Backdoor Attack via Sample-Customized Feature Alignment. *Expert Syst. Appl.* **2026**, *297*, 129481.
31. Zhao, S.; Tuan, L.A.; Fu, J.; et al. Exploring Clean-Label Backdoor Attacks and Defense in Language Models. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2024**, *32*, 3014–3024.
32. Zhao, J.; Huang, H.; Miraj, M.R. Covert Backdoor Attacks to Pre-Trained Encoders. In Proceedings of the 2025 International Joint Conference on Neural Networks (IJCNN), Rome, Italy, 30 June–5 July 2025; pp. 1–8.
33. Sommestad, T.; Holm, H.; Ekstedt, M. Estimates of Success Rates of Denial-of-Service Attacks. In Proceedings of the 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, Changsha, China, 16–18 November 2011; pp. 21–28.
34. Meng, Z.; Chen, C.; Zhu, Y.; et al. Diagnostic Performance of the Automated Breast Volume Scanner: A Systematic Review and Meta-Analysis. *Eur. Radiol.* **2015**, *25*, 3638–3647.
35. Shan, S.; Wenger, E.; Zhang, J.; et al. Fawkes: Protecting Privacy Against Unauthorized Deep Learning Models. In Proceedings of the 29th USENIX Conference on Security Symposium, Boston, MA, USA, 12–14 August 2020; pp. 1589–1604.